Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two; Ordered Sets, Cardinals, Integers

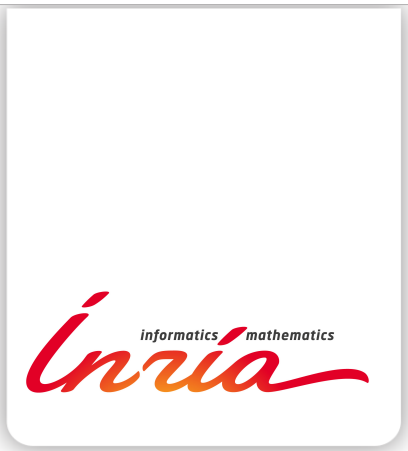
José Grimm

Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two Ordered Sets, Cardinals, Integers

José Grimm

RESEARCH REPORT
N° 7150
December 2009
Project-Team Marelle

ISSN 0249-6399 ISRN INRIA/RR--7150--FR+ENG



Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two Ordered Sets, Cardinals, Integers

José Grimm*

Project-Team Marelle

Research Report n° 7150 — version 4 — initial version December 2009 — revised version December 2011 — 446 pages

Abstract: We believe that it is possible to put the whole work of Bourbaki into a computer. One of the objectives of the Gaia project concerns homological algebra (theory as well as algorithms); in a first step we want to implement all nine chapters of the book Algebra. But this requires a theory of sets (with axiom of choice, etc.) more powerful than what is provided by Ensembles; we have chosen the work of Carlos Simpson as basis. This reports lists and comments all definitions and theorems of the Chapter "Ordered Sets, Cardinals, Integers".
Version 3 is based on the Coq ssreflect library. It implements some properties on ordinal numbers. The code (including some exercises) is available on the Web, under http://www-sop.inria.fr/apics/gaia.

Key-words: Gaia, Coq, Bourbaki, orders, cardinals, ordinals, integers

Work done in collaboration with Alban Quadrat, started when the author was in the Apics Team.

* Email: Jose.Grimm@sophia.inria.fr

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Implémentation de la théorie des ensembles de Bourbaki dans Coq partie 2
Ensembles Ordonnés, cardinaux, nombres entiers

Résumé : Nous pensons qu'il est possible de mettre dans un ordinateur l'ensemble de l'œuvre de Bourbaki. L'un des objectifs du projet Gaia concerne l'algèbre homologique (théeorie et algorithmes); dans une première étape nous voulons implémenter les neuf chapitres du livre Algèbre. Au préalable, il faut implémenter la théorie des ensembles. Nous utilisons l'Assistant de Preuve Coq; les choix fondamentaux et axiomes sont ceux proposés par Carlos Simpson. Ce rapport liste et commente toutes les définitions et théorèmes du Chapitre "Ensembles ordonnés, cardinaux, nombres entiers". Une petite partie des exercises a été résolue. Le code est disponible sur le site Web http://www-sop.inria.fr/apics/gaia.

Mots-clés : Gaia, Coq, Bourbaki, ordre, cardinaux, ordinaux, entiers

Chapter 1

Introduction

1.1 Objectives

Our objective (it will be called the *Bourbaki Project* in what follows) is to show that it is possible to implement the work of N. Bourbaki, "Éléments de Mathématiques"[4], into a computer, and we have chosen the Coq Proof Assistant, see [11, 2]. All references are given to the English version "Elements of Mathematics"[3], which is a translation of the French version (the only major difference is that Bourbaki uses an axiom for the ordered pair in the English version and a theorem in the French one). We start with the first book: theory of sets. It is divided into four chapters, the first one describes formal mathematics (logical connectors, quantifiers, axioms, theorems). Chapters II describes sets, unions, intersections, functions, products, equivalences; Chapter III defines orders, integers, cardinals, limits. The last chapter describes structures. The first part of this report[7] describes Chapter I and Chapter II, we consider here Chapter III.

1.2 Content of this document

This document describes the code found in the files *set5.v*, *set6.v*, *set7.v*, *set8.v*, *set9.v*, and *set10.v*, corresponding to sections 1 to 6 of Chapter III. The first section describes order relations and associated properties (like upper bounds, greatest elements, increasing functions, order isomorphisms). The second section studies well-ordered sets, and introduces the notion of transfinite induction. We show Zermelo's theorem (which is equivalent to the axiom of choice). Section 3 defines cardinals, addition, multiplication and order on cardinals (a cardinal is a representative of a class of equipotent sets; this class is not a set, and the axiom of choice is required). Section 4 defines natural integers as cardinals x such that $x \neq x+1$. It introduces induction on natural integers, so that a natural integer is any cardinal obtained by applying a "finite" number of times $x \mapsto x+1$ to the empty set. More formally, if E is a set containing zero and stable by $x \mapsto x+1$, it contains all natural integers. If E is any set with cardinal x, the set $\mathrm{E} \cup \{\mathrm{E}\}$ has cardinal $x+1$. As a consequence, one can define a mapping $n \mapsto \mathrm{E}_n$ that associates to each natural number n a set E_n of cardinal n (by transfinite induction, one can do this for any cardinal n). This set E_n is called an ordinal in [9] (For Bourbaki, an ordinal is a representative of well-ordered sets). It allows one to define finite cardinals without the use of the axiom of choice. There is no set containing all cardinals (thus no set containing all ordinals) but given a cardinal (or ordinal) a, there is a set containing all cardinals (or ordinals) less than a, and it is well-ordered. Every finite ordinal

is a natural integer; infinite ordinals possess strange properties (addition and multiplication are non-commutative) studied in the Exercises. Section 5 studies some properties of integers (for instance division, expansion to base b) and computes the number of elements of various sets (for instance the number of subsets of p elements of a set of n elements, the number of permutations, etc). Section 6 studies infinite sets. If there exists an infinite set, then there exists a set **N** containing all natural integers. An axiom is required in Bourbaki; in Coq, there is an infinite set, namely nat, and it is canonically isomorphic to the set of natural integers. We use this isomorphism in section 5 (for instance, the factorial function and binomial coefficient are defined by induction on the Coq type nat, then shown to satisfy the Bourbaki definitions). There are few infinite cardinals (i.e., for any cardinal x there is a cardinal y such that $y > x$, for instance 2^x, but one could add an axiom saying that $y > x$ implies $y \geq 2^x$), so that one gets result like: the number of permutations of E is the number of mapping $\mathbf{N} \mapsto$ E, it is also the number of orderings on E (see Exercices).

Section 6 defines direct limits and inverse limits. It is not yet implemented. There are many exercises, two third of them are solved.

In the current version of this document, we use von Neumann ordinals to define cardinals. This means that file *set7.v* contains the definition and basic properties of ordinal numbers (including comparison). An ordinal equipotent to its successor is called infinite; the least infinite ordinal is called ω (it exists since nat is infinite). The cardinal of a set is defined to be the least ordinal equipotent to it; a finite ordinal is a finite cardinal, so that $\mathbf{N} = \omega$ is the set of all integers. We moved the definition and basic properties of addition; multiplication and exponentiation from file *set7.v* into file *set8.v*.

Four additional files *sset11.v*, *sset12.v*, *sset13.v* and *sset14.v* study properties of ordinal numbers (addition, multiplication, exponentation, Cantor Normal Form) and also of infinite cardinals (cofinality, regular cardinals, inaccessible cardinals, the Generalized Continuum Hypothesis).

The reader is invited to read the introduction of the first part. It explains some implementation details (for instance, what is a set? what formulation of the axiom of choice is used?).

1.3 Terminology

Chapter III is much less formal that Chapter II. Let's for instance quote Definition 9 [3, p. 146]: "Two elements of a preordered set E are said comparable if the relation "$x \leq y$ or $y \leq x$" is true. A set E is said to be totally ordered if it is ordered and if any two elements of E are comparable. The ordering in E is then said to be a total ordering and the corresponding order relation a total order relation."

One has to understand this as follows. A preordered set is a pair (E, G) which satisfies the preorder condition. The notation $x \leq y$ stands for $(x, y) \in$ G. An ordered set is a preordered set where additional conditions are required for G. The ordering is G, the corresponding order relation is "$(x, y) \in$ G". By definition, E is uniquely determined by G. Instead of saying that E is totally ordered, one can say: G is a total ordering if it is an ordering and for every x and y in the substrate of G one has "$(x, y) \in$ G or $(y, x) \in$ G". This non ambiguous, and will be our definition. The following sentence is ambiguous "The set **R** of real numbers is totally ordered", since the order is not specified. Note that a sentence like "$(\mathbf{R}, \leq)$ is totally ordered" is ambiguous since $\leq$ can denote any ordering. One must say "The set **R** of real numbers is totally ordered by the usual order on real numbers." We shall use the notation $x \leq_{\mathbf{R}} y$; an

alternative would be $x \leq y \pmod{\mathbf{R}}$ as in [9].

Bourbaki says: "a well-ordered set is totally ordered". This is a short-hand for: for every (E, Γ), if Γ is a well-ordering on E, then Γ is a total ordering on E. It is impossible to say "for every equivalence relation R we have..." since relations cannot be quantified; there is only one theorem in E.II.6, the chapter on equivalence relations. It is of the form: a correspondence Γ between X and X is an equivalence if and only if... This might explain why Bourbaki defines an order as a correspondence, rather than a graph. As a consequence, there are few criteria (C59 to C63 define normal and transfinite induction).

1.4 Notations

The set of natural numbers is denoted by **N** is Bourbaki. In the code it will be Bnat, in order to distinguish it from the set nat of Coq integers (these two sets are naturally isomorphic).

A lemma whose name starts with OS_ (respectively, CS_ and BS_) says that some quantity is an ordinal number, a cardinal number, or a natural integer.

A lemma whose name ends with R, S, A or T says that some relation is reflexive, symmetric, antisymmetric, or transitive.

A lemma whose name ends with C, A D, or I says that an operation is commutative, associative, distributive or involutive.

The cardinal sum is denoted by card_sum, and properties of the sum are given in theorems starting with csum. Similarly, the ordinal product is denoted by ord_prod, and properties of the product are given in theorems starting with oprod. A suffix 2 is added in the case of a binary operation. Note that csum_Cn states commutativity of the cardinal sum in the case of arbitrary number of arguments.

A suffix M may indicate monotony; for instance sum_Mlele says how $a+b$ and $a'+b'$ compare when $a \leq a'$ and $b \leq b'$.

The following notations introduce some alternate names.

```
Notation "\osup" := union (only parsing).
Notation "\csup" := union (only parsing).
Notation "\opred" := union (only parsing).
Notation "\aleph" := omega_fct (only parsing).
```

We introduce some constants.

```
Notation "\0o" := ord_zero.
Notation "\1o" := ord_one.
Notation "\2o" := ord_two.
Notation "\0c" := card_zero.
Notation "\1c" := card_one.
Notation "\2c" := card_two.
Notation "\3c" := card_three.
Notation "\4c" := card_four.
Notation "\10c" := card_ten.
Notation "\omega" := ord_omega.
```

These are binary operators.

```
Notation "x \cg y" := (compose_graph x y) (at level 50).
Notation "f1 \co f2" := (compose f1 f2) (at level 50).
Notation "f1 \coP f2" := (composable f1 f2) (at level 50).
Notation "x \Eq y" := (equipotent x y) (at level 50).
Notation "x \Is y" := (order_isomorphic x y) (at level 50).
Notation "x <=o y" := (ordinal_le x y) (at level 60).
Notation "x <o y" := (ordinal_lt x y) (at level 60).
Notation "x <=c y" := (cardinal_le x y) (at level 60).
Notation "x <c y" := (cardinal_lt x y) (at level 60).
Notation "x +c y" := (card_sum2 x y) (at level 50).
Notation "x *c y" := (card_prod2 x y) (at level 40).
Notation "x ^c y" := (card_pow x y) (at level 30).
Notation "x -c y" := (card_diff x y) (at level 50).
Notation "x <=N y" := (Bnat_le x y) (at level 60).
Notation "x <N y" := (Bnat_lt x y) (at level 60).
Notation "x %/c y" := (card_quo x y) (at level 40).
Notation "x %%c y" := (card_rem x y) (at level 40).
Notation "x %|c y" := (BNdivides x y) (at level 40).
Notation "x +o y" := (ord_sum2 x y) (at level 50).
Notation "x *o y" := (ord_prod2 x y) (at level 40).
Notation "x -o y" := (ord_sub x y) (at level 50).
Notation "x ^o y" := (ord_pow x y) (at level 30).
Notation "x <<o y" := (ord_negl x y) (at level 60).
Notation "x ^<c y" := (cpow_less x y) (at level 30).
Notation "x +#o y" := (ord_natural_sum x y) (at level 50).
```

1.5 Tactics

We give here the list of tactics that are defined in the files associated to this document.

This is now the tactic that exploits properties of order relations.

```
Ltac order_tac:=
  match goal with
    | H1: gle ?r ?x _ |- inc ?x (substrate ?r)
      => exact (inc_arg1_substrate H1)
    | H1: glt ?r ?x _ |- inc ?x (substrate ?r)
      => move: H1 => [H1 _] ; order_tac
    | H1:gle ?r _ ?x |- inc ?x (substrate ?r)
      => exact (inc_arg2_substrate H1)
    | H1:glt ?r _ ?x |- inc ?x (substrate ?r)
      => move: H1 => [H1 _]; order_tac
    | H: order ?r, H1: inc ?u (substrate ?r) |- related ?r ?u ?u
      => rewrite -/gle - order_reflexivity //
    | H: order ?r |- inc (J ?u ?u) ?r
       => change (gle r u u); rewrite - order_reflexivity //
    | H: order ?r |- gle ?r ?u ?u
       =>  rewrite  -order_reflexivity //
    | H1: gle ?r ?x ?y, H2: gle ?r ?y ?x, H:order ?r |-
        ?x = ?y => exact (order_antisymmetry H H1 H2)
    | H:order ?r, H1:related ?r ?x ?y, H2: related ?r ?y ?x |- ?x = ?y
      => apply (order_antisymmetry H H1 H2)
    | H:order ?r, H1: inc (J ?x ?y) ?r , H2: inc (J ?y ?x) ?r |- ?x = ?y
      => apply (order_antisymmetry H H1 H2)
    | H:order ?r, H1:related ?r ?u ?v, H2: related ?r ?v ?w
```

```
      |- related ?r ?u ?w
      => apply (order_transitivity H H1 H2)
    | H:order ?r, H1:gle ?r ?u ?v, H2: gle ?r ?v ?w |- gle ?r ?u ?w
      => apply (order_transitivity H H1 H2)
    | H: order ?r, H1: inc (J ?u ?v) ?r, H2: inc (J ?v ?w) ?r |-
      inc (J ?u ?w) ?r
      => change (related r u w); apply (order_transitivity H H1 H2)
    |  H1: gle ?r ?x ?y, H2: glt ?r ?y ?x, H: order ?r |- _
      => elim (not_le_gt H H1 H2)
    | H1:glt ?r ?x ?y, H2: glt ?r ?y ?x, H:order ?r |- _
      => move: H1 => [H1 _] ; elim (not_le_gt H H1 H2)
    | H1:order ?r, H2:glt ?r ?x ?y, H3: gle ?r ?y ?z |- glt ?r ?x ?z
       => exact (lt_leq_trans H1 H2 H3)
    | H1:order ?r, H2:gle ?r ?x ?y, H3: glt ?r ?y ?z |- glt ?r ?x ?z
       => exact (leq_lt_trans H1 H2 H3)
    | H1:order ?r, H2:glt ?r ?x ?y, H3: glt ?r ?y ?z |- glt ?r ?x ?z
       => exact (lt_lt_trans H1 H2 H3)
    | H1:order ?r, H2:gle ?r ?x ?y |- glt ?r ?x ?y
       => split =>//
    | H:glt ?r ?x ?y |- gle ?r ?x ?y
       => by move: H=> []
end.
```

If a is an ordinal and $b \in a$, the following tactic says that b is an ordinal, and any element of b is an element of a.

```
Ltac ord_tac0 :=
 match goal with
  |  h1: is_ordinal ?a, h2: inc ?b ?a |- is_ordinal ?b =>
    apply: (elt_of_ordinal h1 h2)
  | h1: is_ordinal ?x, h2: inc ?y ?x, h3: inc ?z ?y |- inc ?z ?x =>
    apply: (((ordinal_transitive h1) _ h2) _ h3)
end.

Ltac eqtrans u:= apply equipotent_transitive with u.
Ltac eqsym:= apply equipotent_symmetric.
```

This tactic uses transitivity and antisymmetry of $\leq_{\mathrm{ord}}$.

```
Ltac ord_tac :=
  match goal with
    | h: ordinal_le _ ?x |- is_ordinal ?x
      => move: h => [_ [h _]]; exact h
    | h: ordinal_le ?x _ |- is_ordinal ?x
      => move: h => [h _]; exact h
    | h: ordinal_lt _ ?x |- is_ordinal ?x
      => move: h => [[_ [h _]] _]; exact h
    | h: ordinal_lt ?x _ |- is_ordinal ?x
      => move: h => [[h _] _]; exact h
    | h1: ordinal_le ?x ?y, h2:  ?y <=o ?x  |-  ?x = ?y
      => apply: (ord_leA h1 h2)
    | h1: ?x  <=o ?y, h2: ?y <=o ?z  |-  ?x  <=o ?z
      => apply: (ord_leT h1 h2)
    | h1: ?x <o ?y, h2: ?y <=o ?z  |-  ?x <o ?z
      => apply: (ord_lt_leT h1 h2)
    | h1: ?x <=o ?y, h2: ?y <o ?z  |-  ?x <o ?z
```

```
      => apply: (ord_le_ltT h1 h2)
    | h1: ?x <o ?y, h2: ?y <o ?z  |-  ?x <o ?z
      => apply: (ord_lt_ltT h1 h2)
    | h1: ordinal_le ?x ?y, h2: ordinal_lt ?y ?x  |-  _
      => elim (ord_leA1 h1 h2)
    | h1: is_ordinal ?x, h2: inc ?y ?x |- is_ordinal ?y
      => apply: (elt_of_ordinal h1 h2)
    | h1: is_ordinal ?x |- ordinal_le ?x ?x
      => apply: (ord_leR h1)
    | h1:  ordinal_lt ?x ?y |- ordinal_le ?x ?y
      => by move: h1 => []
end.
```

This tactic uses transitivity and antisymmetry of $\leq_{\text{card}}$.

```
Ltac co_tac := match goal with
  | Ha:cardinal_le ?a ?b, Hb: cardinal_le ?b ?c |- cardinal_le ?a ?c
     =>  apply: (cardinal_leT Ha Hb)
  | Ha:cardinal_lt ?a ?b, Hb: cardinal_le ?b ?c |- cardinal_lt ?a ?c
     =>  apply: (cardinal_lt_leT Ha Hb)
  | Ha:cardinal_le ?a ?b, Hb: cardinal_lt ?b ?c |- cardinal_lt ?a ?c
     =>  apply: (cardinal_le_ltT Ha Hb)
  | Ha:cardinal_lt ?a ?b, Hb: cardinal_lt ?b ?c |- cardinal_lt ?a ?c
     => induction Ha; co_tac
  | Ha: cardinal_le ?a ?b, Hb: cardinal_lt ?b ?a |- _
    => elim (not_card_le_lt Ha Hb)
  | Ha:cardinal_le ?x ?y, Hb:  cardinal_le ?y ?x |- _
   => solve [ rewrite (cardinal_leA Ha Hb) ; fprops ]
  | Ha: cardinal_le ?a _ |- is_cardinal ?a => induction Ha; assumption
  | Ha: cardinal_le  _ ?a |- is_cardinal ?a
   => destruct Ha as [_ [Ha _ ]]; exact Ha
  | Ha: cardinal_lt ?a _ |- is_cardinal ?a => induction Ha; co_tac
  | Ha: cardinal_lt  _ ?a |- is_cardinal ?a => induction Ha; co_tac
end.
```

This is used for intervals.

```
Ltac uf_interval :=
  rewrite /interval_cc/interval_oo/interval_co/interval_oc
  /interval_uu/interval_uo/interval_ou/interval_uc/interval_cu.
Ltac zztac2 v  := uf_interval; set_extens v ; aw; rewrite ? Z_rw;  aw.
```

Chapter 2

Order relations. Ordered sets

This chapter defines order relations and studies some properties of sets and subsets ordered by a relation. We define the notion of maximal element, greatest element, upper bound, least upper bound. Some ordered sets may be qualified as directed, lattice or totally ordered, or intervals. Functions weakly compatible with the ordering are called "increasing", and functions strongly compatible are called "order isomorphisms".

2.1 Definition of an order relation

In the last chapter of the first part of this document, we studied equivalence relations, that were reflexive, symmetric and transitive. If we replace symmetric by antisymmetric, we get an *order relation*. Remember that transitive means that if $x \sim y$ and $y \sim z$ then $x \sim z$, and symmetric means that if $x \sim y$ then $y \sim x$. A *reflexive* relation is such that $x \sim y$ implies $x \sim x$ and $y \sim y$. A symmetric and transitive relation is reflexive. A relation is *reflexive on* a set E if $x \in \mathrm{E}$ is equivalent to $x \sim x$. The support (or substrate) of a relation is the set of all x and y related by the relation.

We say that a relation (denoted here $\prec$) is *antisymmetric* if $x \prec y$ and $y \prec x$ imply $x = y$. An order relation on E is an order relation whose support is E (or equivalently, that is reflexive on E). A *preorder relation* is reflexive and transitive (we could also define preorder relations on E). The opposite relation of $\prec$, denoted by $\succ$, is such that $x \prec y$ and $y \succ x$ are equivalent.

```
Definition antisymmetric_r (r:Set -> Set -> Prop) :=
  forall x y, r x y -> r y x -> x = y.
Definition is_antisymmetric r :=
  is_graph r & forall x y, related r x y -> related r y x -> x = y.
Definition reflexive_rr (r:Set -> Set -> Prop) :=
  forall x y, r x y -> (r x x &  r y y).
Definition order_r(r:Set -> Set -> Prop) :=
  transitive_r r &  antisymmetric_r r & reflexive_rr r.
Definition order_re  (r:Set -> Set -> Prop) (x: Set) :=
  order_r r & reflexive_r r x.
Definition preorder_r (r:Set -> Set -> Prop) :=
  transitive_r r & reflexive_rr r.
Definition opposite_relation (r:Set -> Set -> Prop) :=
  fun x y => r y x.
```

Equality and inclusion are order relations. The opposite of an order relation is an order

relation.

```
Lemma equality_order_r: order_r (fun x y => x = y).
Lemma sub_order_r: order_r sub.
Lemma opposite_preorder_r r:
  preorder_r r -> preorder_r (opposite_relation r).
Lemma opposite_order_r r:
  order_r r -> order_r (opposite_relation r).
```

An *order* on a set E is a graph G such that the relation $(x, y) \in G$ is an order relation between x and y with substrate E. A *preorder* is similarly defined. The opposite order relation corresponds to the inverse graph, which is thus called the opposite order.

```
Definition order r :=
  is_reflexive r & is_transitive r & is_antisymmetric r.
Definition preorder r :=
  is_reflexive r & is_transitive r.
Definition opposite_order := inverse_graph.
Lemma order_graph r: order r -> is_graph r.
Lemma preorder_graph r: preorder r -> is_graph r.
Lemma order_preorder r: order r -> preorder r.
```

If we have an order relation on E, we can take its graph, and this gives an order (this is the same construction as for equivalence relations).

```
Lemma graph_on_rw3 r x u v:
  order_re r x ->
  (related (graph_on r x) u v <-> r u v).
Lemma order_has_graph0 r x:
  order_re r x -> is_graph_of (graph_on r x) r.
Lemma order_has_graph r x:
  order_re r x -> exists g, is_graph_of g r.
Lemma order_if_has_graph r g:
  is_graph g ->  is_graph_of g r ->
  order_r r -> order_re r (domain g).
Lemma order_if_has_graph2 r g:
  is_graph g ->  is_graph_of g r ->
  order_r r -> order g.
Lemma order_has_graph2 r x:
  order_re r x -> exists g,
    g = graph_on r x &
    order g & (forall u v, r u v <-> related g u v).
```

An order can be obtained from a relation by taking its graph on a set, provided that the relation "$x \in X$ and $y \in X$ and $x \prec y$" is an order relation. Its support is X provided that $x \prec x$ for all $x \in X$.

```
Lemma preorder_from_rel r x:
  preorder_r r -> preorder (graph_on r x).
Lemma order_from_rel r x:
  order_r r -> order (graph_on r x).
Lemma order_from_re11 r x:
  transitive_r r ->
  (forall u v, inc u x -> inc v x -> r u v -> (r u u  & r v v)) ->
  (forall u v, inc u x -> inc v x -> r u v -> r v u -> u = v) ->
```

```
  (forall u, inc u x -> r u u) ->
  order (graph_on r x).
Lemma graph_on_sr (r:Set -> Set -> Prop) x:
  (forall u, inc u x -> r u u) ->
  substrate (graph_on r x) = x.
```

The traditional notation for an order relation is $x \leq y$, or $y \geq x$, we shall use gle in our code; if the elements are distinct, we shall use the notations $x < y$, or $y < x$. This last relation is not reflexive. It satisfies some transitivity properties. We give here some simple properties of orders.[1]

```
Definition gle r x y := related r x y.
Definition glt r x y := gle r x y &  x <> y.

Lemma order_reflexivity_pr r x u v:
  order_re r x -> r u v -> (inc u x & inc v x).
Lemma order_symmetricity_pr  r x u v:
  order_re r x -> ((r u v & r v u) <-> (inc u x & inc v x & u = v)).
Lemma order_reflexivity r a:
  order r -> (inc a (substrate r) <-> gle r a a).
Lemma order_antisymmetry  r a b:
  order r -> gle r a b -> gle r b a -> a = b.
Lemma order_transitivity  r a b c:
  order r -> gle r a b -> gle r b c -> gle r a c.
Lemma lt_leq_trans r x y z:
  order r -> glt r x y -> gle r y z -> glt r x z.
Lemma leq_lt_trans r x y z:
  order r -> gle r x y -> glt r y z -> glt r x z.
Lemma lt_lt_trans r a b c:
  order r -> glt r a b -> glt r b c -> glt r a c.
Lemma not_le_gt r x y:
   order r -> gle r x y -> glt r y x -> False.
```

By reflexivity, the substrate of an order is the domain (the set of all x such that there exists y with $x \leq y$).

```
Lemma order_is_order r:
  order r -> order_r (related r).
Lemma substrate_domain_order r:
  order r -> substrate r = domain r.
Lemma opposite_order_sr r:
  order r -> substrate(opposite_order r) = substrate r.
```

Some properties of opposite order.

```
Lemma opposite_gle r x y:
  gle (opposite_order r) x y <-> gle r y x.
```

Examples of orders: the opposite of an order is an order; the intersection of orders is an order; equality is an order (it is also an equivalence).

```
Definition order_set z := (forall r, inc r z -> order r).
```

[1] Definition of gge changed in V3: related has been replaced by gle. Functions gge and ggt removed later on

```
Lemma inter_oa_or z: nonempty z -> order_set z ->
  order (intersection z).
Lemma opposite_or r:
  order r -> order (opposite_order r).
Lemma diagonal_or x: order (identity_g x).
```

¶ In many cases, we get an order on a set E by taking for $x \prec y$ a relation of the form $f(x) \subset f(y)$. This is an ordering if f is injective; moreover the ordering will be total. In the definition that follows, we consider the case where f is the identity function, and E is any set, or is the powerset of a set A.

```
Definition inclusion_suborder b :=
  graph_on sub b.
Definition inclusion_order a := inclusion_suborder (powerset a).

Lemma subinclusion_or a:
  order (inclusion_suborder a).
Lemma inclusion_or a:
  order (inclusion_order a).
Lemma subinclusion_sr a:
  substrate (inclusion_suborder a) = a.
Lemma inclusion_sr a:
  substrate (inclusion_order a) = powerset a.
Lemma subinclusion_gle_rw a u v:
  gle (inclusion_suborder a) u v <-> (inc u a & inc v a & sub u v).
Lemma inclusion_gle_rw a u v:
  gle (inclusion_order a) u v <-> (sub u a & sub v a & sub u v).
```

¶ Second example. We consider the set $\Phi(\mathrm{E},\mathrm{F})$ of functions from a subset of E to F and the extension relation. Denote by $\mathrm{S}(f)$, $\mathrm{T}(f)$ and $\mathrm{G}(f)$ the source, target and graph of a function f. We say that g extends f if $\mathrm{G}(f) \subset \mathrm{G}(g)$ and $\mathrm{T}(f) \subset \mathrm{T}(g)$. This implies $\mathrm{S}(f) \subset \mathrm{S}(g)$ and for any $x \in \mathrm{S}(f)$ we have $f(x) = g(x)$. The relation we consider is $g \leq f$ short for "$g \in \Phi(\mathrm{E},\mathrm{F})$ and $f \in \Phi(\mathrm{E},\mathrm{F})$ and g extends f". This is the same as "$g \in \Phi(\mathrm{E},\mathrm{F})$ and $f \in \Phi(\mathrm{E},\mathrm{F})$ and $\mathrm{G}(f) \subset \mathrm{G}(g)$". thus is an order on $\Phi(\mathrm{E},\mathrm{F})$.

```
Definition extension_order x y :=
  graph_on extends (set_of_sub_functions x y).
Lemma extends_in_prop:
  order_r extends &
  (forall x y u, inc u (set_of_sub_functions x y) -> extends u u).
Lemma extension_or x y:  order (extension_order x y).
Lemma extension_sr x y:
  substrate (extension_order x y) = (set_of_sub_functions x y).

Lemma extension_order_rw E F f g:
  gle (extension_order E F) g f <->
  (inc g (set_of_sub_functions E F) & inc f (set_of_sub_functions E F)
    & extends g f).
Lemma extension_order_pr1 E F f g:
  gle (extension_order E F) g f <->
  (inc g (set_of_sub_functions E F) & inc f (set_of_sub_functions E F)
    & sub (graph f) (graph g)).
Lemma extension_order_pr2 E F f g:
  gle (opposite_order (extension_order E F)) g f <->
  (inc g (set_of_sub_functions E F) & inc f (set_of_sub_functions E F)
```

```
    & sub (graph g) (graph f)).
Lemma extension_order_pr E F f g x:
  related (opposite_order (extension_order E F)) f g ->
  inc x (source f) ->  W x f =  W x g.
```

¶ Third example. Let E be a set, and W the subset of $\mathfrak{P}(\mathfrak{P}(E))$ formed of all partitions of E. Recall that a partition ϖ of E is a set of sets whose union is E, so that $\varpi \subset \mathfrak{P}(E)$. Additional conditions are: the empty set is not in ϖ, the elements of ϖ are mutually disjoint. We say that ϖ is *coarser* than ϖ' if for every $Y \in \varpi'$ there exists $X \in \varpi$ such that $Y \subset X$. We shall show that it is an order. But we start with another property: assume that X is a partition of E; if $x \in X$ then x is a subset of E hence $x \in \mathfrak{P}(E)$. We can consider the canonical injection $X \to \mathfrak{P}(E)$; this is the function defined on X that maps x to itself; the graph of this function is a family of sets; this family is a partition (in the other sense of this word) of E.

```
Definition set_of_partition_set x :=
  Zo(powerset(powerset x)) (fun z => partition z x).

Definition partition_fun_of_set y x :=
  canonical_injection y (powerset x).

Lemma partition_set_in_double_powerset y x:
  partition y x -> inc y (powerset (powerset x)).
Lemma set_of_partition_rw x y:
  inc y (set_of_partition_set x) <-> partition y x.
Lemma pfs_function y x:
  partition y x -> is_function (partition_fun_of_set y x).
Lemma pfs_W  y x a:
  partition y x -> inc a y ->  W a (partition_fun_of_set y x) = a.
Lemma pfs_partition y x:
  partition y x ->  partition_fam (graph (partition_fun_of_set y x)) x.
```

We now define coarser and show that it is an order on W.

```
Definition coarser x
    := graph_on coarser_c (set_of_partition_set x).
Lemma coarser_gle x y y':
  gle (coarser x) y y' <->
  (partition y x &  partition y' x & coarser_c y y').
Lemma coarser_gle_bis x y y':
  gle (coarser x) y y' <->
  (partition y x &  partition y' x &
    forall a, inc a y' -> exists b, inc b y & sub a b).
Lemma coarser_sr x:
  substrate (coarser x) = set_of_partition_set x.
Lemma coarser_or x: order (coarser x).
Lemma smallest_partition_is_smallest x y:
  nonempty x ->
  partition y x -> gle (coarser x) (smallest_partition x) y.
Lemma largest_partition_is_largest x y:
  partition y x -> gle (coarser x) y (largest_partition x).
```

Let ϖ be a partition; consider the set formed of all $A \times A$ with $A \in \varpi$; let $\tilde{\varpi}$ be the union of all these sets. We pretend that this set is the graph of the equivalence associated to the partition. The relation "ϖ coarser than ϖ'" is equivalent to $\tilde{\varpi} \supset \tilde{\varpi}'$. Bourbaki says that this shows that coarser is an order. The nontrivial point is antisymmetry: we must show that $\tilde{\varpi} = \tilde{\varpi}'$ implies

$\omega = \omega'$. This is a consequence of the fact that the sets $A \times A$ are mutually disjoint. If a and b are in the same element of ω and in A and B for ω', the pair (a,b) is in $\tilde{\omega}$, hence in $\tilde{\omega}'$, hence in $A \times A$ and $B \times B$, so that the intersection of A and B is not empty, and $A = B$.

```
Definition partition_relation_set_aux y x :=
  Zo (powerset (coarse x)) (fun z => exists a, inc a y & z = coarse a).
Definition partition_relation_set y x :=
  partition_relation (partition_fun_of_set y x) x.

Lemma prs_is_equivalence y x:
  partition y x -> is_equivalence (partition_relation_set y x).
Lemma partition_relation_set_pr1 y x a:
  partition y x ->
  inc a y -> inc (coarse a) (partition_relation_set_aux y x).
Lemma partition_relation_set_pr y x:
  partition y x ->
  partition_relation_set y x =
  union (partition_relation_set_aux y x).   (* 16 *)
Lemma sub_partition_relation_set_coarse y x:
  partition y x -> sub (partition_relation_set y x) (coarse x).
Lemma nondisjoint a b c: inc a b -> inc a c -> disjoint b c -> False.
Lemma partition_relation_set_order x y y:
  partition_set y x -> partition_set y' x ->
  (sub (partition_relation_set y' x)(partition_relation_set y x) <->
    gle (coarser x) y y').  (* 28 *)
Lemma partition_relation_set_order_antisymmetric x y y':
  partition y x -> partition y' x ->
  (partition_relation_set y' x = partition_relation_set y x) ->
   y = y'. (* 41 *)
```

Let G be a graph, Δ be the diagonal of the substrate of G. If $\Delta \subset G$ and $G \circ G \subset G$ then $G \circ G = G$. The two assumptions say that G is reflexive and transitive, this a preorder. Antisymmetry is $G \cap G^{-1} \subset \Delta$. Since Δ is symmetric (i.e., it is its inverse), reflexivity is also $\Delta \subset G^{-1}$, so that G is an order if and only $G \circ G = G$, and $G \cap G^{-1} = \Delta$ (we know that G is an equivalence if and only $G \circ G = G$ and $G = G^{-1}$). Proposition 1 of Bourbaki [3, p. 132] considers the case of a correspondence (see section 11.7).

```
Lemma preorder_prop1 g:
  is_graph g ->
  sub (identity_g (substrate g)) g -> sub (g \cg g) g ->
  g \cg g = g.
Theorem order_pr r:
  order r <->
    (r \cg r = r &
    intersection2 r (opposite_order r) = identity_g (substrate r)).  (* 37 *)
```

2.2 Preorder relations

Consider the relation "R is coarser than R′" in the set of all coverings of E. This is a preorder relation, since it is reflexive and transitive; it is not antisymmetric, for if R is a covering, $R' = R \cup \{X\}$ with $X \subset E$, then R′ is a covering which is coarser then R. Now, R is coarser than R′ if there is a $Y \in R$ such that $X \subset Y$. It can happen that $X \notin R$; in such a case, R is coarser than

R', R' is coarser than R, but $R \neq R'$. In the case of a partition, $X \in R'$, $Y \in R'$ and $X \subset Y$ implies $X = Y$.

Here is the definition and the basic properties.

```
Lemma preorder_is_preorder r:
  preorder r -> preorder_r (related r).
Lemma opposite_is_preorder1 r:
  preorder r -> preorder (opposite_order r).
```

The relation "$x \prec y$ and $y \prec x$" is an equivalence if $\prec$ is a preorder. Denote it by $\sim$. Then $x \prec y$ is compatible with $x \sim x'$ and $y \sim y'$. This means that if these three relations hold, then we have also $x' \prec y'$. If $\prec$ has a graph G, then $\sim$ has a graph, which is $G \cap G^{-1}$. If G is a graph, it is a preorder if and only if $\Delta \subset G$ and $G \circ G \subset G$. We know that it implies $G \circ G = G$.

```
Definition equivalence_associated_o r
   := intersection2 r (opposite_order r).

Lemma equivalence_preorder r:
  preorder_r r ->
  equivalence_r (fun x y => r x y & r y x).
Lemma compatible_equivalence_preorder
  r (s := (fun x y => r x y & r y x) :
  preorder_r r -> forall x y x' y', r x y -> s x x' -> s y y' -> r x' y'.
Lemma equivalence_preorder1 r:
  preorder r ->
  is_equivalence (equivalence_associated_o r).
Lemma equivalence_associated_o_sr r:
  preorder r ->
  substrate (equivalence_associated_o r) = substrate r.
Lemma compatible_equivalence_preorder1 r u v x y:
  preorder r -> related r x y ->
  related (equivalence_associated_o r) x u ->
  related (equivalence_associated_o r) y v->
  related r u v.
Lemma preorder_prop g:
  is_graph g ->
  (preorder g <-> (sub (identity_g (substrate g)) g & sub (g \cg g) g)).
Lemma preorder_prop2 g:
  preorder g -> g \cg g = g.
Lemma preorder_reflexivity r a:
  preorder r -> (inc a (substrate r) <-> related r a a).
```

Let $\prec$ be a preorder on a set E, and $\sim$ the equivalence associated to it. Assume that R is the graph of $\prec$, and let π be the canonical projection $E \to E/\sim$. Given two objects of the form $u = \pi(x)$ and $v = \pi(y)$ in the quotient, we can compare u and v via $x \prec y$, this is independent of the representatives x and y. This relation has a graph, namely $S \circ G \circ S^{-1}$, where S is the graph of π. This set is also $(\pi \times \pi)\langle G \rangle$, where $\pi \times \pi$ maps $E \times E$ into $(E/\sim) \times (E/\sim)$. This relation is an order on the quotient; it is called the order relation *associated* with $\prec$.

```
Definition order_associated r :=
  let s := graph (canon_proj (equivalence_associated_o r)) in
  (s \cg r) \cg (opposite_order s).

Lemma order_associated_graph r:
```

```
  is_graph (order_associated r).
Lemma compose3_related s r u v:
  related ((s \cg r) \cg (opposite_order s)) u v <->
  exists x, exists y, related s x u  & related s y v & related r x y.
Lemma eao_related r a b:
  related (equivalence_associated_o r) a b -> related r a b.

Section OrderAssociated.
Variable (r:Set).
Hypothesis pr: preorder r.

Lemma order_associated_related1 u v:
  related (order_associated r) u v <->
  ( inc u (quotient (equivalence_associated_o r)) &
    inc v (quotient (equivalence_associated_o r)) &
    exists x, exists y, inc x u & inc y v & related r x y).   (* 28 *)
Lemma order_associated_related2 u v:
  related (order_associated r) u v <->
  ( inc u (quotient (equivalence_associated_o r)) &
    inc v (quotient (equivalence_associated_o r)) &
    forall x y, inc x u -> inc y v -> related r x y).
Lemma order_associated_sr:   (* 21 *)
  substrate (order_associated r) = quotient (equivalence_associated_o r).
Lemma order_associated_or:
  order (order_associated r). (* 32 *)
Lemma order_associated_pr:
  order_associated r = image_by_fun (
    ext_to_prod(canon_proj (equivalence_associated_o r))
    (canon_proj (equivalence_associated_o r))) r.   (* 35 *)
End OrderAssociated.
```

2.3 Notation and terminology

Bourbaki introduces another notion of order. It is a correspondence $\Gamma = (E, F, G)$, where $E = F$, and the relation $(x, y) \in G$ is an order relation on E. Notice that this property implies $G \subset E \times E$ and E is the domain of G, so that (E, E, G) is a correspondence. One can use indifferently Γ or G. For instance, the opposite order is the inverse graph G^{-1} or the inverse correspondence Γ^{-1}.

An ordered set is a set with an ordering. An ordered group is a set with an internal law and an order relation, that are compatible (see Exercises). One has properties of the form $\forall a, b, c \in E$, if $a \leq b$ then $a + c \leq b + c$ and conversely. We shall see in a future chapter that such a relation is true when E is the set of integers, + the addition, and $\leq$ the natural ordering. The relation is equally true for the product (but the converse holds only if c is non-zero). Most authors define an ordered group as a triple $(E, +, \leq)$ (or maybe 5-uple $(E, +, 0, -, \leq)$) and an ordered field as a quadruple $(E, +, \times, \leq)$ (or maybe a 8-uple $(E, +, 0, -, \times, 1, /, \leq)$). Thus an ordered set would be a couple $(E, \leq)$.

The important point is that the relation "$x \in E$ and $y \in E$ and $x \leq y$" is an order relation on E. For instance, one may consider the set of integers and the cardinal ordering, and say that $(\mathbf{N}, \leq_{\mathrm{Card}})$ is an ordered set. We shall prove that if b, c are cardinals, then $c \leq_{\mathrm{Card}} b + c$ (because $0 \leq_{\mathrm{Card}} b$). From this, we deduce that the same relation holds if b and c are integers (lemmas csum_M0le and Bsum_M0le). If $\leq_{\mathbf{N}}$ is the ordering on $\mathbf{N}$, then $c \leq_{\mathbf{N}} b + c$ holds. This theorem is not needed, thus not proved. One could say that $(\mathbf{N}, \leq_{\mathbf{N}})$ is an ordered set. But this is the

same ordered set as $(\mathbf{N}, \leq_{\mathrm{Card}})$. Thus, it is better to define an ordered set by (E, G) where G is an ordering on E. Note that E is uniquely defined by G. In what follows, whenever Bourbaki considers an ordered set $(E, \leq)$, we replace it by a graph G. The relation $x \in E$ becomes 'inc x (substrate G)' and $x \leq y$ becomes 'gle G x y'.

At this point in the document, Bourbaki denotes order relations by $x \leq y$ instead of $x \prec y$. He defines an order on E by

($\mathrm{RO_I}$) The relation "$x \leq y$ and $y \leq z$" implies $x \leq z$.
($\mathrm{RO_{II}}$) The relation "$x \leq y$ and $y \leq x$" implies $x = y$.
($\mathrm{RO_{III}}$) The relation $x \leq y$ implies "$x \leq x$ and $y \leq y$".
($\mathrm{RO_{IV}}$) The relation $x \leq x$ is equivalent to $x \in E$.

Note the absence of quantifiers; in Bourbaki, if x is a free variable, $P\{x\}$ is equivalent to $(\forall x)(P\{x\})$. Hence the definition is correct if x, y and z are three distinct letters that do not appear in E or $\leq$. The first part of C58 is: let $\leq$ be an order relation and let x, y be two distinct letters; the relation $x \leq y$ is equivalent to "$x < y$ or $x = y$". Here, Bourbaki explicitly says that x and y are distinct. The statement is true only if $x = x$ implies $x \leq x$, for instance if $\leq$ is an order on E and $x \in E$.

```
Definition order_axioms r s :=
  (forall x y z, gle r x y ->  gle r y z -> gle r x z) &
  (forall x y, gle r x y ->  gle r y x -> x = y) &
  (forall x y, gle r x y ->  (inc x s & inc y s)) &
  (forall x, gle r x x <-> inc x s) &
  is_graph r.

Lemma axioms_of_order r:
  order r <-> (order_axioms r (substrate r)).
Lemma le_pr r x y:
  inc x (substrate r) -> order r ->
  (gle r x y <-> (glt r x y \/ x = y)).
```

We say that f is a *morphism* or *isomorphism* for two orders denoted $\leq_E$ or $\leq_F$ on E and F if f is a function from E to F compatible with the orders (this means the obvious: if $x \leq_E y$ then $f(x) \leq_F f(y)$ and conversely). Such a function is injective. An isomorphism is required to be bijective. Properties of morphisms will be studied later. A morphism is an isomorphism on its range.

```
Definition order_isomorphism f r r' :=
  (order r) & (order r') &
  (bijection f) & (substrate r = source f) &  (substrate r' = target f) &
  (forall x y, inc x (source f) -> inc y (source f) ->
    (gle r x y <-> gle r' (W x f) (W y f))).

Definition order_morphism f r r' :=
  (order r) & (order r') &
  (is_function f) & (substrate r = source f) &  (substrate r' = target f)  &
  (forall x y, inc x (source f) -> inc y (source f) ->
    (gle r x y <-> gle r' (W x f) (W y f))).
Lemma order_morphism_injective f r r':  order_morphism f r r'->
  injection f.

Lemma identity_is r: order r ->
  order_isomorphism (identity (substrate r)) r r.
Lemma identity_morphism  r: order r ->
```

```
  order_morphism (identity (substrate r)) r r.
Lemma inverse_order_is r r' f:
   order_isomorphism f r r' -> order_isomorphism (inverse_fun f) r' r.
Lemma compose_order_is r r' r'' f f':
 order_isomorphism f r r' -> order_isomorphism f' r' r''
  -> order_isomorphism (f' \co f) r r''.
Lemma compose_order_morphism r r' r'' f f':
  f' \coP f -> order_morphism f r r' -> order_morphism f' r' r''
  -> order_morphism (f' \co f) r r''.
```

2.4 Ordered subsets. Product of ordered sets

Let G be a graph, A a set. Define $G_A = G \cap (A \times A)$. This is a graph, whose substrate is a subset of A. If G is reflexive, with substrate E and $A \subset E$, then the substrate of G_A is A. As a consequence, if G is an order (or preorder) on E, then G_A is an order (or preorder) on A. It is called the order *induced* by G. By abuse of notations, if the order relation associated to G is denoted $x \leq y$ then the order relation associated to G_A also denoted $x \leq y$ instead of $x \leq_A y$. The first three lemmas here say that $x \leq_A y$ implies $x \leq y$, and $x <_A y$ implies $x < y$; moreover if $x \in A$ and $y \in A$, then $x \leq_A y$ is equivalent to $x \leq y$.

```
Definition induced_order r a :=
  intersection2 r (coarse a).

Lemma iorder_gle r a x y: inc x a -> inc y a ->
  (gle (induced_order r a) x y <-> gle r x y).
Lemma iorder_gle1 r a x y:
  gle (induced_order r a) x y -> gle r x y.
Lemma iorder_gle2 r a x y:
  glt (induced_order r a) x y -> glt r x y.

Lemma iorder_gle3 r a x y:
  gle (induced_order r a) x y -> (inc x a & inc y a).
Lemma iorder_gle4 r a x y:
  glt (induced_order r a) x y -> (inc x a & inc y a).
Lemma iorder_gle5 r a x y:
  gle (induced_order r a) x y <-> (inc x a & inc y a & gle r x y).
Lemma iorder_gle6 r a x y:
  glt (induced_order r a) x y <-> (inc x a & inc y a & glt r x y).

Lemma iorder_graph r a:
  is_graph r -> is_graph (induced_order r a).

Lemma iorder_sr1 r a:
  is_reflexive r ->
  sub a (substrate r) ->  substrate (induced_order r a) = a.
Lemma iorder_sr r a:
  order r ->
  sub a (substrate r) ->  substrate (induced_order r a) = a.
Lemma reflexive_iorder r a:
  sub a (substrate r) ->
  is_reflexive r -> is_reflexive (induced_order r a).
Lemma transitive_iorder r a:
  sub a (substrate r) ->
  is_transitive r -> is_transitive (induced_order r a).
```

```
Lemma iorder_preorder r a:
  sub a (substrate r) ->
  preorder r -> preorder (induced_order r a).
Lemma iorder_or r a:
  sub a (substrate r) ->
  order r -> order (induced_order r a).
Lemma iorder_substrate r:
  order r ->
  induced_order r (substrate r) = r.
Lemma order_exten r r': order r -> order r' ->
  ((r = r') <-> (forall x y, gle r x y <-> gle r' x y)).
Lemma iorder_opposite r x: order r ->
  (induced_order (opposite_order r) x = opposite_order (induced_order r x)).
Lemma iorder_trans a b c: sub c b ->
  (induced_order (induced_order a b) c = induced_order a c).
```

Let $\Phi(E,F)$ be the set of all mappings of subsets of E into F (this is the union of all $\mathscr{F}(I;F)$, for $I \subset E$). Denote here by $\Psi(E,F)$ the set of functional graphs included in $E \times F$ (one could show that this is the union of all F^I for $I \subset E$). The bijection between $\mathscr{F}(I;F)$ and F^I, that associates to each function its graph extends to $\Phi(E,F) \to \Psi(E,F)$ and is an order isomorphism where the source is endowed with the relation "g extends f" between f and g (this is the opposite order of extension_order) and the target with the inclusion order.

```
Definition set_of_fgraphs x y :=
  (Zo(powerset (product x y)) (fun z => functional_graph z)).

Definition graph_of_function x y :=
  BL graph (set_of_sub_functions x y)
  (set_of_fgraphs x y).

Lemma graph_of_function_sub x y z:
  inc z (set_of_sub_functions x y) -> sub (graph z) (product x y).
Lemma set_of_graphs_pr x y z:
  inc z (set_of_sub_functions x y) -> inc (graph z) (set_of_fgraphs x y).
Lemma graph_of_fonction_function x y:
  is_function (graph_of_function x y).
Lemma graph_of_function_W x y f:
  inc f (set_of_sub_functions x y) -> W f (graph_of_function x y) = graph f.
Lemma graph_of_function_bijection x y:
  bijective (graph_of_function x y).
Lemma graph_of_function_is x y:
  order_isomorphism (graph_of_function x y)
  (opposite_order (extension_order x y))
  (inclusion_suborder (set_of_fgraphs x y)).
```

¶ If E is a set, we consider the mapping $\varpi \mapsto \tilde{\varpi}$, that maps a partition to the graph of the associated equivalence. We know that "ϖ coarser than ϖ'" is equivalent to $\tilde{\varpi} \supset \tilde{\varpi}'$. This mapping is an isomorphism on its image (when the source is endowed with the opposite of the coarse relation, and the target with $\subset$). The source is the set of partitions, the target is some subset of $\mathfrak{P}(E \times E)$.

```
Definition graph_of_partition x :=
  BL(fun y => partition_relation_set y x)
  (set_of_partition_set x)(powerset (coarse x)).
```

```
Lemma gop_axioms x:
  transf_axioms (fun y => partition_relation_set y x)
     (set_of_partition_set x) (powerset (coarse x)).
Lemma gop_W x y:
  partition y x ->
  W y (graph_of_partition x) = partition_relation_set y x.
Lemma gop_morphism x:
  order_morphism (graph_of_partition x) (coarser x)
  (opposite_order (inclusion_order (coarse x))).
```

¶ We consider the set of all preorders on E; more precisely the set of all graphs that are preorders, and order them by inclusion. A preorder s is finer than t if $s \subset t$; this is the same as to say that elements related by s are related by t.

```
Definition set_of_preorders x :=
  Zo (powerset (coarse x))(fun z => substrate z = x & preorder z).
Definition coarser_preorder x :=
  inclusion_suborder (set_of_preorders x).

Lemma set_of_preorders_rw x z:
  inc z (set_of_preorders x) <-> (substrate z = x & preorder z).

Lemma coarser_preorder_or x:
  order (coarser_preorder x).
Lemma coarser_preorder_sr x:
  substrate (coarser_preorder x) = set_of_preorders x.
Lemma coarser_preorder_gle x u v:
  gle (coarser_preorder x) u v <->
  (preorder u & preorder v & substrate u = x & substrate v = x & sub u v).
Lemma coarser_preorder_gle1 x u v:
  gle (coarser_preorder x) u v <->
  (preorder u & preorder v & substrate u = x & substrate v = x &
  forall a b, inc a x -> inc b x -> related u a b -> related v a b).
```

¶ Consider now a family of orders G_ι with substrate E_ι. Denote the relation associated to G_ι by $\leq_\iota$. On the product $\prod E_\iota$ we can consider the relation: $\forall \iota$, $x_\iota \leq_\iota x'_\iota$ between $(x_\iota)_\iota$ and $(x'_\iota)_\iota$. This is an order relation; it will be called the *product* of the order relations and its graph will be called the *product* of the orders. The graph is $\prod G_\iota$ transported from $\prod(E_\iota \times E_\iota)$ to $\prod E_\iota \times \prod E_\iota$ via the canonical bijection.

```
Definition fam_of_substrates g :=
  L (domain g) (fun i => substrate (V i g)).
Definition prod_of_substrates g := productb (fam_of_substrates g).
Definition order_fam g :=
  fgraph g &  (forall x, inc x (domain g) -> order (V x g)).
Definition order_product_r g x x' :=
   forall i, inc i (domain g) -> gle (V i g) (V i x) (V i x').
Definition order_product g :=
  graph_on (order_product_r g)(prod_of_substrates g).

Lemma prod_of_substrates_rw g x:
  order_fam g ->
  (inc x  (prod_of_substrates g) <-> (fgraph x & domain x = domain g &
    forall i, inc i (domain g) -> inc (V i x) (substrate (V i g)))).
```

```
Lemma order_product_or g:
  order_fam g -> order (order_product g).
Lemma order_product_gle g x x':
  order_fam g ->
  (gle (order_product g) x x' <->
   (inc x (prod_of_substrates g) & inc x' (prod_of_substrates g) &
   forall i, inc i (domain g) -> gle (V i g) (V i x)(V i x'))).
Lemma order_product_sr g:
  order_fam g -> substrate (order_product g) = prod_of_substrates g.
Lemma product_order_def g (f := fam_of_substrates g):   (* 48 *)
  order_fam g ->
  image_by_fun (prod_of_products_canon f f) (order_product g)= (productb g).
```

Special case of a product of two orders.

```
Definition order_product2 f g :=
  graph_on (fun x x' => gle f (P x) (P x') & gle g (Q x) (Q x'))
    (product (substrate f)(substrate g)).

Lemma order_product2_pr f g x x':
  gle (order_product2 f g) x x' <->
  (inc x (product (substrate f)(substrate g)) &
    inc x' (product (substrate f)(substrate g)) &
    gle f (P x) (P x') & gle g (Q x) (Q x')).
Lemma order_product2_sr1 f g:
  preorder f -> preorder g ->
  substrate (order_product2 f g) = product (substrate f) (substrate g).
Lemma order_product2_sr f g:
  order f -> order g ->
  substrate (order_product2 f g) = product (substrate f) (substrate g).
Lemma order_product2_preorder f g:
  preorder f -> preorder g -> preorder (order_product2 f g).
Lemma order_product2_or f g:
  order f -> order g -> order (order_product2 f g).
```

Since F^E is a product, an order on F gives an order on the set of graphs of functions. We can compare two functions f and g via $f(x) \leq g(x)$. This gives a natural order on $\mathscr{F}(E;F)$. The function that associates to a function $\mathscr{F}(E;F)$ its graph in F^E is an isomorphism.

```
Definition order_graph_r x g z z' :=
  forall i, inc i x -> gle g (V i z) (V i z').

Definition order_graph x y g :=
  graph_on (order_graph_r x g) (set_of_gfunctions x y).

Definition order_function_r x y r f g :=
  is_function f & is_function g &  source f = x & target f = y
  & source g =x & target g = y &
  forall i, inc i x -> gle r (W i f) (W i g).

Definition order_function x y r :=
  graph_on(fun u v => order_function_r x y r u v)
  (set_of_functions x y).
Definition order_isomorphic r r':=
  exists f, order_isomorphism f r r'.
Notation "x \Is y" := (order_isomorphic x y) (at level 50).
```

Now some properties of graph orderings.

```
Lemma order_fam_cst x g: order g ->
  order_fam (cst_graph x g).
Lemma order_graph_r_pr x g z z':
  order_graph_r x g z z' <->
  order_product_r  (cst_graph x g) z z'.
Lemma order_graph_pr1 x y g:
   substrate g = y ->
    (set_of_gfunctions x y) = prod_of_substrates (cst_graph x g).
Lemma order_graph_pr x y g: substrate g = y ->
  order_graph x y g = order_product (cst_graph x g).
Lemma order_graph_or x y g: order g -> substrate g = y ->
  order (order_graph x y g).
Lemma order_graph_sr x y g: order g -> substrate g = y ->
  substrate (order_graph x y g) = set_of_gfunctions x y.
```

Now some properties of function orderings.

```
Section OrderFunction.
Variables (x y r: Set).
Hypothesis (or: order r) (sr: substrate r = y).

Lemma order_function_reflexive u:
  inc u (set_of_functions x y) -> gle (order_function x y r) u u.
Lemma order_function_sr:
  substrate (order_function x y r) = set_of_functions x y.
Lemma order_function_pr f f':
  gle (order_function x y r) f f' <->
  (inc f (set_of_functions x y) &
    inc f' (set_of_functions x y) &
    forall i, inc i x ->   gle r (W i f) (W i f')).
Lemma order_function_or : order (order_function x y r).
Lemma function_order_is:
  order_isomorphism (BL graph (set_of_functions x y)(set_of_gfunctions x y))
  (order_function x y r)(order_graph x y r).
```

The last theorem says that there exists an order isomorphism $f : r \to r'$ where r is the function order and r' the graph order. Bourbaki writes this as $\mathrm{Is}(r, r')$. We prefer an infix notation r \Is r'.

```
Lemma order_function_is1:
  (order_function x y r) \Is (order_graph x y r).
End OrderFunction.
```

2.5 Increasing mappings

Consider two sets E and F ordered by $\leq_E$ and $\leq_F$ and a function $f : E \to F$. We say that f is *increasing* if $x \leq_E y$ implies $f(x) \leq_F f(y)$ and *decreasing* if $x \leq_E y$ implies $f(x) \geq_F f(y)$. A function is *strictly increasing* or *strictly decreasing* if $x < y$ implies $f(x) < f(y)$ or $f(x) > f(y)$. (Bourbaki considers the case of preorders in the non-strict case). A function that is increasing or decreasing is *monotone*.[2]

[2] The definition has been simplified in Version 3.

```
Definition increasing_fun f r r' :=
  is_function f & order r & order r' & substrate r = source f
  & substrate r' = target f  &
  forall x y,  gle r x y  -> gle r' (W x f) (W y f).
Definition decreasing_fun f r r':=
  is_function f & order r & order r' & substrate r = source f
  & substrate r' = target f  &
  forall x y,  gle r x y  -> gle r' (W y f) (W x f).
Definition monotone_fun f r r' :=
  increasing_fun f r r' \/  decreasing_fun f r r'.

Definition strict_increasing_fun f r r' :=
  is_function f & order r & order r' & substrate r = source f
  & substrate r' = target f  &
  forall x y,  glt r x y  -> glt r' (W x f) (W y f).
Definition strict_decreasing_fun f r r' :=
  is_function f & order r & order r' & substrate r = source f
  & substrate r' = target f  &
  forall x y,  glt r x y  -> glt r' (W y f) (W x f).
Definition strict_monotone_fun f r r' :=
  strict_increasing_fun f r r' \/  strict_decreasing_fun f r r'.
```

Some consequences when we replace one order by its opposite.

```
Lemma increasing_fun_reva f r r':
   increasing_fun f r r' -> decreasing_fun f r (opposite_order r').
Lemma increasing_fun_revb f r r':
   increasing_fun f r r' -> decreasing_fun f (opposite_order r) r'.
Lemma decreasing_fun_reva f r r':
   decreasing_fun f r r' -> increasing_fun f r (opposite_order r').
Lemma decreasing_fun_revb  f r r':
   decreasing_fun f r r' -> increasing_fun f (opposite_order r) r'.
Lemma monotone_fun_reva f r r':
   monotone_fun f r r' -> monotone_fun f r (opposite_order r').
Lemma monotone_fun_revb  f r r':
   monotone_fun f r r' -> monotone_fun f (opposite_order r) r'.
```

Same for strictly monotone.

```
Lemma strict_increasing_fun_reva  f r r':
  strict_increasing_fun f r r' -> strict_decreasing_fun f r (opposite_order r').
Lemma strict_increasing_fun_revb f r r',:
  strict_increasing_fun f r r' -> strict_decreasing_fun f (opposite_order r) r'.
Lemma strict_decreasing_fun_reva f r r':
  strict_decreasing_fun f r r' -> strict_increasing_fun f r (opposite_order r').
Lemma strict_decreasing_fun_revb f r r':
  strict_decreasing_fun f r r' -> strict_increasing_fun f (opposite_order r) r'.
Lemma strict_monotone_fun_reva f r r':
  strict_monotone_fun f r r' -> strict_monotone_fun f r (opposite_order r').
Lemma strict_monotone_fun_revb f r r':
   strict_monotone_fun f r r' -> strict_monotone_fun f (opposite_order r) r'.
```

If f is constant, then f is increasing and decreasing. Conversely, the identity function is increasing and decreasing on a set ordered by equality. This function is not constant when the set has more than one element. Let E be set, f the function that maps $X \subset E$ to its complementary. This is a strictly decreasing function when the powerset is ordered by inclusion.

```
Lemma constant_fun_increasing f r r':
  order r -> order r' -> substrate r = source f  ->
  substrate r' = target f -> is_constant_function f ->
  increasing_fun f r r'.
Lemma  constant_fun_decreasing f r r':
  order r -> order r' -> substrate r = source f  ->
  substrate r' = target f -> is_constant_function f ->
  decreasing_fun f r r'.
Lemma identity_increasing_decreasing x (r := identity_g x) :
    (increasing_fun (identity x) r r &decreasing_fun (identity x) r r).
Lemma complementary_decreasing E:
  strict_decreasing_fun(BL (fun X => complement E X)(powerset E)(powerset E))
  (inclusion_order E)  (inclusion_order E).
```

Let U_x be the set of upper bounds of $\{x\}$ (the case of a non-singleton will be studied later). We have $x \leq y$ if and only if $U_y \subset U_x$. The function $x \mapsto U_x$ is strictly decreasing.

```
Definition set_of_majorants1 r x :=
  Zo (substrate r)(fun y => gle r x y).
Lemma set_of_majorants1_pr x y r:
  order r -> inc x (substrate r) -> inc y (substrate r) ->
  (gle r x y <-> sub (set_of_majorants1 r y) (set_of_majorants1 r x)).
Lemma set_of_majorants1_decreasing r: (* 19 *)
 order r ->
 strict_decreasing_fun(BL (set_of_majorants1 r)
   (substrate r)(powerset (substrate r))) r  (inclusion_order (substrate r)).
```

If a function is injective, monotone implies strictly monotone. If a function is bijective, it is an isomorphism if and only if the function and its inverse are increasing. An isomorphism remains one if the ordering on the source and target are replaced by the opposite ones.

```
Lemma strict_increasing_from_injective f r r':
  injection f -> increasing_fun f r r' -> strict_increasing_fun f r r'.
Lemma strict_decreasing_from_injective f r r':
  injection f ->decreasing_fun f r r' -> strict_decreasing_fun f r r'.
Lemma strict_monotone_from_injective f r r':
  injection f -> monotone_fun f r r' -> strict_monotone_fun f r r'.
Lemma order_isomorphism_increasing f r r':
  order_isomorphism f r r' ->
  strict_increasing_fun f r r'.
Lemma order_morphism_increasing f r r':
  order_morphism f r r' ->
  strict_increasing_fun f r r'.
Lemma order_isomorphism_pr f r r':
  order r -> order r' ->
  bijection f -> substrate r = source f -> substrate r' = target f ->
  (order_isomorphism f r r' <->
    (increasing_fun f r r' & increasing_fun (inverse_fun f) r' r)).
Lemma order_isomorphism_opposite g r r':
  order_isomorphism g r r' ->
  order_isomorphism g (opposite_order r) (opposite_order r').
```

Assume that we have two ordered sets E and E′, decreasing functions u and v from E to E′ and E′ to E. Assume $u(v(x)) \geq x$ and $v(u(x')) \geq x'$ for all x and x'. Proposition 2 [3, p. 139] says $u \circ v \circ u = u$ and $v \circ u \circ v = v$.

```
Theorem decreasing_composition u v r r':
  decreasing_fun u r r' -> decreasing_fun v r' r ->
  (forall x, inc x (substrate r) -> gle r (W (W x u) v) x) ->
  (forall x', inc x' (substrate r') -> gle r' (W (W x' v) u) x') ->
  (u \co v \co u = u & v \co u \co v = v).  (* 20 *)
```

2.6 Maximal and minimal elements

Bourbaki says: if E is a set with a preorder, then $a \in E$ is *minimal* (resp. *maximal*) in E if $x \leq a$ (resp. $x \geq a$) implies $x = a$. Our definition applies to any graph[3]

```
Definition maximal_element r a :=
  inc a (substrate r) & forall x, gle r a x -> x = a.
Definition minimal_element r a :=
  inc a (substrate r) & forall x, gle r x a -> x = a.
```

Examples. In $\mathfrak{P}(E)$, the empty set is the least element for inclusion. If we remove it, minimal elements are singletons. On the set of partial functions ordered by extension, maximal elements are total functions (because non-total functions can be extended).

```
Lemma maximal_element_opp r a:
  order r -> maximal_element r a -> minimal_element (opposite_order r) a.
Lemma minimal_element_opp r a:
  order r -> minimal_element r a -> maximal_element (opposite_order r) a.
Lemma maximal_opposite r x:
  order r -> (maximal_element (opposite_order r) x <-> minimal_element r x).

Lemma minimal_inclusion E y:
  let F:= complement (powerset E) (singleton emptyset) in
  inc y F -> (minimal_element (inclusion_suborder F) y <-> is_singleton y).

Lemma maximal_prolongation E F x: (* 21 *)
  nonempty F -> inc x (set_of_sub_functions E F) ->
  (maximal_element (opposite_order (extension_order E F)) x <-> (source x = E)).
```

2.7 Greatest element and least element

We start with the definition of the *greatest* and *least* element of an order relation. Trivial properties follow. If $\leq$ is an order on E, then a is a greatest element for the order if $a \in E$ and if for all $x \in E$ we have $x \leq a$ (the condition $a \in E$ could be relaxed if E is not empty). A greatest element need not exist, but is unique. It is maximal. A set that has a greatest element has a unique maximal element. The greatest (resp. least) element is sometimes denoted max (resp. min). We give two properties: if E has a least element then $\min E$ is a least element, and if x is a least element of E then $\min E = x$. If $E' \subset E$ then $\min E = \min E'$ provided that the big set has a least element that belongs to the small set. If $\min E = \max E$, then E has a single element.

```
Definition greatest_element r a :=
  inc a (substrate r) & forall x, inc x (substrate r) -> gle r x a.
Definition least_element r a :=
```

[3] Definition simplified in V3; it was originally: $\forall x \in E,\ x \leq a \implies x = a$. We deduce $x \in E$ from $x \leq a$.

```
  inc a (substrate r) & forall x, inc x (substrate r) -> gle r a x.

Definition the_least_element r :=  (select (least_element r) (substrate r)).
Definition the_greatest_element r := (select (greatest_element r) (substrate r)).

Lemma unique_greatest a b r:
  order r -> greatest_element r a -> greatest_element r b  -> a = b.
Lemma unique_least a b r:
  order r -> least_element r a -> least_element r b -> a = b.

Lemma the_least_element_pr r:
  order r -> (exists u, least_element r u) ->
  least_element r (the_least_element r).

Lemma the_greatest_element_pr r:
  order r -> (exists u, greatest_element r u) ->
  greatest_element r (the_greatest_element r).

Lemma the_least_element_pr2 r x: order r ->
  least_element r x ->   the_least_element r = x.
Lemma the_greatest_element_pr2 r x: order r ->
  greatest_element r x ->   the_greatest_element r = x.
Lemma greatest_induced r s x:
  order r -> sub s (substrate r) -> inc x s ->
  greatest_element r x ->
  the_greatest_element r = the_greatest_element (induced_order r s).
Lemma least_induced r s x:
  order r -> sub s (substrate r) -> inc x s ->
  least_element r x ->
  the_least_element r = the_least_element (induced_order r s).

Lemma least_and_greatest r x: order r ->
  least_element r x -> greatest_element r x ->
  is_singleton (substrate r).
```

More simple properties.

```
Lemma greatest_opposite a r:
  order r -> greatest_element r a -> least_element (opposite_order r) a.
Lemma least_oposite a r:
  order r -> least_element r a -> greatest_element (opposite_order r) a.
Lemma the_least_opposite r: order r ->
  (exists a, greatest_element r a) ->
  the_least_element (opposite_order r) =  the_greatest_element r.
Lemma greatest_maximal a r:
  order r -> greatest_element r a -> maximal_element r a.
Lemma least_minimal a r:
  order r -> least_element r a -> minimal_element r a.
Lemma greatest_unique_maximal a b r:
  greatest_element r a -> maximal_element r b -> a = b.
Lemma least_unique_minimal a b r:
  least_element r a -> minimal_element r b -> a = b.
```

¶ Now some examples. If $\mathfrak{S}$ is a subset of $\mathfrak{P}(E)$, then the upper and lower bounds of $\mathfrak{S}$ are the intersection and union; we anticipate a bit: for the moment being, we just say that the least and greatest elements are the intersection and union, provided they are in $\mathfrak{S}$. Note that E need not be mentioned in the theorems.

```
Lemma least_is_intersection s a:
 least_element (inclusion_suborder s) a ->
  nonempty s -> a = intersection s.
Lemma greatest_is_union s a:
  greatest_element  (inclusion_suborder s) a -> a = union s.
Lemma intersection_is_least s:
  inc (intersection s) s ->
  least_element (inclusion_suborder s) (intersection s).
Lemma union_is_greatest s:
  inc (union s) s -> greatest_element (inclusion_suborder s) (union s).
```

Some applications. On $\mathfrak{P}(E)$, the least element is $\emptyset$, the greatest is E. On the set of partial functions from E to F, there is no greatest element if F has at least two elements (constant functions defined on the whole of E are maximal; if there is a greatest element, all constants are equal).

```
Definition empty_function_tg F := BL id emptyset F.

Lemma emptyset_is_least E:
  least_element (inclusion_order E) emptyset.
Lemma wholeset_is_greatest E:
  greatest_element (inclusion_order E) E.
Lemma empty_function_tg_function F:
  is_function (empty_function_tg F).
Lemma least_prolongation E F:
  least_element (opposite_order (extension_order E F))
  (empty_function_tg F).
Lemma greatest_prolongation E F x:
  greatest_element (opposite_order (extension_order E F)) x ->
  nonempty E -> small_set F. (* 21 *)
```

Bourbaki notices that if E is a set, Δ the diagonal of E, then Δ is the least equivalence or preorder on E (for the inclusion order induced on $\mathfrak{P}(E \times E)$). This a consequence of the next lemma. Note: We have defined finer_equivalence and shown that if S and R are equivalences on a same set, then S is finer than R if and only if $S \subset R$. Thus, on the set of equivalences on E, the inclusion order is finer_equivalence and we have already shown that the least element is the diagonal and the greatest one is $E \times E$.

```
Lemma least_equivalence r:
  is_reflexive r -> sub (diagonal (substrate r)) r.
```

Proposition 3 [3, p. 140] in Bourbaki says: Let E be an ordered set and let E' be the disjoint union of E and a set $\{a\}$ consisting of a single element. Then there exists a unique ordering on E' which induces the given ordering on E and for which a is the greatest element of E'.

We first discard the term "disjoint" and assume $a \notin E$. Let G be the ordering of E, $E' = E \cup \{a\}$, and $G' = G \cup (E' \times \{a\})$; this is an ordering on E' which induces the given ordering on E (i.e. $G'_E = G$) and for which a is the greatest element of E' and is in fact the unique ordering satisfying these conditions.

Assume $a \in E$ and that there is some $b \in E$ such that $a < b$. Whatever E' that has an ordering G' such that $G'_E = G$, the relation $b \leq a$ is false (because $a < b$ is true in E'). Thus Proposition 3 is false. What Bourbaki means is the following. Given two sets E and A. There exists $\bar{E}$ and $\bar{A}$, two disjoint sets, as well as two bijections $f : E \to \bar{E}$ and $g : A \to \bar{A}$. The union E' of $\bar{E}$ and $\bar{A}$ is called the disjoint union of E and A. It is of course not unique. There is a

unique ordering on $\bar{E}$ that makes f an order isomorphism. If A is ordered, there is a unique ordering on $\bar{A}$ that makes g an order isomorphism, and there a unique ordering on E' such that $\bar{x} \leq \bar{y}$ if and only if either (1) $\bar{x} = f(x)$ and $\bar{y} = f(y)$ and $x \leq y$ or (2) $\bar{x} = g(x)$ and $\bar{y} = g(y)$ and $x \leq y$ or (3) $\bar{x} = f(x)$ and $\bar{y} = g(y)$, for some x and y. This is called the ordinal sum of E and A (modulo f and g). If $A = \{a\}$, there is a unique ordering on A, $\bar{A}$ is a singleton, and the ordinal sum of E and $\{a\}$ is the answer to the Proposition. It is not unique (since f and g are not unique). One could however state: let E be an ordered set; all ordered sets (E', G') such that $E' = E \cup \{a\}$, and $G'_E = G$ and $a = \max E'$, these sets are are order-isomorphic. This equivalence class[4] is called the ordinal successor of the ordinal of E.

```
Lemma order_transportation f r (r':= image_by_fun (ext_to_prod f f) r) :
  bijection f -> order r -> substrate r = source f ->
  order_isomorphism f r r'. (* 70 *)

Definition order_with_greatest r a :=
  union2 r (product (tack_on (substrate r) a) (singleton a)).
Lemma order_with_greatest_pr   (* 51 *)
  r a (r':=order_with_greatest r a) :
    order r ->  ~ (inc a (substrate r)) ->
    (order r' & substrate r' = tack_on (substrate r) a &
     r = induced_order r' (substrate r) & greatest_element r' a).
Theorem adjoin_greatest r a E:
  order r -> substrate r = E -> ~ (inc a E) ->
  exists_unique (fun r' => order r' & substrate r' = tack_on E a &
     r = induced_order r' E & greatest_element r' a).  (* 35 *)
```

If r is an order (denoted by $\leq$) on E (Bourbaki considers the case of pre-orders) we say that a subset A of E is *cofinal* (or *coinitial*) if for all $x \in E$ there is a $y \in A$ such that $x \leq y$ (or $y \leq x$). Bourbaki says "To say that an ordered set E has a greatest element therefore means that E has a cofinal subset consisting of a single element". The lemmas given here talk about an object r and its substrate E. We do not assume that r is an order: we do not even assume that r is a graph (what happens is the following: if r' is the set of pairs (x, y) in r, then r' is the the graph of the relation associated to r, for which x and y are related if and only if $(x, y) \in r$).

```
Definition cofinal_set r a :=
  sub a (substrate r) &
  (forall x, inc x (substrate r) -> exists y, inc y a & gle r x y).
Definition coinitial_set r a :=
  sub a (substrate r) &
  (forall x, inc x (substrate r) -> exists y, inc y a & gle r y x).
Lemma exists_greatest_cofinal r:
  (exists x, greatest_element r x) <->
  (exists a, cofinal_set r a & is_singleton a).
Lemma exists_least_coinitial r:
  (exists x, least_element r x) <->
  (exists a, coinitial_set r a & is_singleton a).
```

2.8 Upper and lower bounds

Given an order r on a set E denoted by $\leq$ and a set X, an element x is said to be an *upper bound* for r and X if $y \in X$ implies $y \leq x$. A *lower bound* is an element x such that $y \in X$

[4] in fact, the order-type of any such set, because there is no set containing all orderings order-isomorphic to a given ordering.

implies $x \leq y$. If the set X is not empty, we deduce that $x \in$ E; in order to cover the case $X = \emptyset$, we add the condition $x \in$ E; the set of upper bounds of the empty set is thus E. Note that Bourbaki assumes that r is a preorder (but in most examples, and in the next section, it will be an order).

```
Definition upper_bound r X x  :=
  inc x (substrate r) & forall y, inc y X -> gle r y x.
Definition lower_bound r X x :=
  inc x (substrate r) & forall y, inc y X -> gle r x y.
```

The first properties given here are trivial. If we have an order on E and if X is a subset of E, we can consider the order induced on X; this may have a least element m or a greatest element M. If these quantities exist, they are in X and are an upper or lower bound of X for the relation on E. Converse holds: if X has an upper bound in X, it is M.

```
Lemma opposite_upper_bound x X r: order r ->
  (upper_bound r X x <-> lower_bound (opposite_order r) X x).
Lemma opposite_lower_bound x X r: order r ->
  (lower_bound r X x <-> upper_bound  (opposite_order r) X x).
Lemma smaller_lower_bound x y X r: preorder r ->
  lower_bound r X x -> gle r y x -> lower_bound r X y.
Lemma greater_upper_bound x y X r: preorder r ->
  upper_bound r X x -> gle r x y -> upper_bound r X y.
Lemma sub_lower_bound  x X Y r:
  lower_bound r X x -> sub Y X-> lower_bound r Y x.
Lemma sub_upper_bound x X Y r:
  upper_bound r X x -> sub Y X -> upper_bound r Y x.
Lemma least_element_pr X r: order r -> sub X (substrate r) ->
  ((exists a, least_element (induced_order r X) a) <->
   (exists x, lower_bound r X x & inc x X)).
Lemma greatest_element_pr X r: order r -> sub X (substrate r) ->
  ( (exists a, greatest_element (induced_order r X) a) <->
    (exists x, upper_bound r X x & inc x X)).
```

We consider now *bounded* sets, that are sets that have a bound.

```
Definition bounded_above r X := exists x, upper_bound r X x.
Definition bounded_below r X := exists x, lower_bound r X x.
Definition bounded_both r X := bounded_above r X & bounded_below r X.

Lemma bounded_above_sub X Y r:
  sub Y X ->bounded_above r X  -> bounded_above r Y.
Lemma bounded_below_sub X Y r:
  sub Y X ->bounded_below r X  -> bounded_below r Y.
Lemma bounded_both_sub X Y r:
  sub Y X ->bounded_both r X  -> bounded_both r Y.
Lemma singleton_bounded x r:
  is_singleton x -> order r -> sub x (substrate r) -> bounded_both r x.
```

2.9 Least upper bound and greatest lower bound

The Bourbaki definition is: *let* E *be an ordered set and let* X *be a subset of* E. *An element of* E *is said to be the* greatest lower bound *of* X *in* E *if it is the greatest element of the set of*

lower bounds of X *in* E. Let W_X be the set of lower bounds of X; it is a subset of E, hence can be ordered with the order induced from E. This may have a greatest element x. This is in W_X hence is a lower bound of X and if z is another lower bound we have $z \leq x$. The converse is true, hence we have a characterization of the greatest lower bound that does not involve the set W_X. Similarly the *least upper bound* of X is an upper bound x such that if z is another upper bound, then $x \leq z$.

```
Definition greatest_lower_bound r X x :=
  greatest_element (induced_order r (Zo (substrate r) (lower_bound r X))) x.
Definition least_upper_bound r X x :=
  least_element (induced_order r (Zo (substrate r) (upper_bound r X))) x.

Lemma greatest_lower_bound_pr r X x:
  order r -> sub X (substrate r) ->
  (greatest_lower_bound r X x <-> (lower_bound r X x
    & forall z, lower_bound r X z -> gle r z x)).
Lemma least_upper_bound_pr r X x:
  order r -> sub X (substrate r) ->
  (least_upper_bound r X x <-> (upper_bound r X x
    & forall z, upper_bound r X z -> gle r x z)).
```

The greatest lower bound and least upper bound are also called supremum and infimum and denoted by $\sup_E X$ and $\inf_E X$. As usual, the order is $\leq$ and the substrate is E; in some cases the set E is not mentioned. If $X = \{x, y\}$ we often write $\sup(x, y)$ and $\inf(x, y)$. The sup does not always exists, but is unique since there is a unique least element. We give a characterization of the sup and the inf when they exist. The case of two arguments is also provided.[5]

```
Lemma supremum_unique  x y X r: order r ->
  least_upper_bound r X x  -> least_upper_bound r X y  -> x = y.
Lemma infimum_unique x y X r: order r ->
  greatest_lower_bound r X x  -> greatest_lower_bound r X y  -> x = y.

Definition supremum r X  := select (least_upper_bound r X) (substrate r).
Definition infimum r X  := select (greatest_lower_bound r X) (substrate r).
Definition has_supremum r X :=(exists x, least_upper_bound r X x).
Definition has_infimum r X :=(exists x, greatest_lower_bound r X x).
Definition sup r x y := supremum r (doubleton x y).
Definition inf r x y := infimum r (doubleton x y).

Lemma supremum_pr1 X r:
  has_supremum r X ->
  least_upper_bound r X (supremum r X).
Lemma infimum_pr1 X r:
  has_infimum r X ->
  greatest_lower_bound r X (infimum r X).
Lemma supremum_pr2 r X a: order  r ->
  least_upper_bound r X a -> a = supremum r X.
Lemma infimum_pr2 r X a: order  r ->
  greatest_lower_bound r X a -> a = infimum r X.
Lemma inc_supremum_substrate X r:
  order r -> sub X (substrate r) -> has_supremum r X ->
  inc (supremum r X) (substrate r).
Lemma inc_infimum_substrate  X r:
```

[5] Some lemmas simplified in V3

```
  order r -> sub X (substrate r) -> has_infimum r X ->
  inc (infimum r X) (substrate r).
Lemma supremum_pr X r:
  order r ->  sub X (substrate r) -> has_supremum r X ->
  (upper_bound r X (supremum r X) &
    forall z, upper_bound r X z -> gle r (supremum r X) z).
Lemma infimum_pr X r:
  order r ->  sub X (substrate r) -> has_infimum r X ->
  (lower_bound r X (infimum r X) &
    forall z, lower_bound r X z -> gle r z (infimum r X)).
Lemma sup_pr a b r:
  order r -> inc a (substrate r) -> inc b (substrate r)
  -> has_supremum r (doubleton a b) ->
  (gle r a (sup r a b) &  gle r b (sup r a b) &
    forall z, gle r a z ->  gle r b z -> gle r (sup r a b) z).
Lemma inf_pr a b r:
  order r -> inc a (substrate r) -> inc b (substrate r)
  -> has_infimum r (doubleton a b) ->
  (gle r (inf r a b) a &  gle r (inf r a b) b&
    forall z, gle r z a ->  gle r z b -> gle r z (inf r a b)).
Lemma least_upper_bound_doubleton r x y z:
  order r -> gle r x z -> gle r y z ->
  (forall t, gle r x t -> gle r y t -> gle r z t) ->
  least_upper_bound r (doubleton x y) z.
Lemma greatest_lower_bound_doubleton r x y z:
  order r -> gle r z x  -> gle r z y ->
  (forall t, gle r t x -> gle r t y -> gle r t z) ->
  greatest_lower_bound r (doubleton x y) z.
```

We show here the following claim: if a subset X of E has a greatest element a, then a is the least upper bound of X in E.

```
Lemma greatest_is_sup r X a:
  order r -> sub X (substrate r) ->
  greatest_element (induced_order r X) a -> least_upper_bound r X a.
Lemma least_is_inf r X a:
  order r -> sub X (substrate r) ->
  least_element (induced_order r X) a -> greatest_lower_bound r X a.
```

The roles of inf and sup are exchanged if we replace the order by its opposite.

```
Lemma inf_sup_opp r X a:
  order r -> sub X (substrate r) ->
  (greatest_lower_bound r X a <-> least_upper_bound (opposite_order r) X a).
Lemma sup_inf_opp r X a:
  order r -> sub X (substrate r) ->
  (least_upper_bound r X a <-> greatest_lower_bound (opposite_order r) X a).
```

Examples. We study the sup and inf of the empty set.

```
Lemma set_of_lower_bounds_emptyset r :
  Zo (substrate r) (lower_bound r emptyset) = substrate r.
Lemma set_of_upper_bounds_emptyset r:
  Zo (substrate r) (upper_bound r emptyset) = substrate r.
Lemma least_upper_bound_emptyset r x: order r ->
```

```
  (least_upper_bound r emptyset x <-> least_element r x).
Lemma greatest_lower_bound_emptyset r x: order r ->
  greatest_lower_bound r emptyset x = greatest_element r x.
```

If $\mathfrak{S}$ is a subset of $\mathfrak{P}(E)$, then the upper and lower bounds of $\mathfrak{S}$ are the union and intersection, as claimed before. If $\mathfrak{S}$ is empty, the intersection is empty, and the greatest lower bound is the greatest element, namely E. Assume $\mathfrak{S} \subset \mathfrak{F}$ and $\mathfrak{F} \subset \mathfrak{P}(E)$; then the upper and lower bounds of $\mathfrak{S}$ in $\mathfrak{F}$ are the union and intersection, provided that these elements are in $\mathfrak{F}$.

```
Lemma intersection_is_inf s E: sub s (powerset E) ->
  greatest_lower_bound (inclusion_order E) s
  (Yo (nonempty s) (intersection s) E).
Lemma union_is_sup s E: sub s (powerset E) ->
  least_upper_bound (inclusion_order E) s (union s).
Lemma union_is_sup1 s F E:
  sub F (powerset E) ->
  sub s F -> inc (union s) F ->
  least_upper_bound (inclusion_suborder F) s (union s).
Lemma intersection_is_inf1 s F E:
   sub F (powerset E) ->
  sub s F -> inc (Yo (nonempty s) (intersection s) E) F ->
  greatest_lower_bound (inclusion_suborder F) s
  (Yo (nonempty s) (intersection s) E).
```

Third example. If u is in $\Phi(E,F)$, the set of partial functions from E to F, we denote its domain by $D(u)$. If Θ is subset of $\Phi(E,F)$, it has a least upper bound if and only if for all u and v in the family, for all $x \in D(u) \cap D(v)$ we have $u(x) = v(x)$. Denote this property by $P(u,v)$. We use an auxiliary result: if $u \leq w$, the condition $x \in D(u) \cap D(w)$ is equivalent to $x \in D(u)$, and $P(u,w)$ is true. Thus, if u and v are bounded by w, $P(u,v)$ is true. Conversely, we know that if P is true on Θ there is a f function defined on the union of the $D(u)$ such that $u \leq f$. This is the least upper bound.

```
Lemma sup_extension_order1 E F T f:
  sub T (set_of_sub_functions E F) ->
  least_upper_bound (opposite_order (extension_order E F)) T f ->
  forall u v x, inc u T -> inc v T -> inc x (source u) -> inc x (source v) ->
    W x u = W x v.
Lemma sup_extension_order2 E F T:
  sub T (set_of_sub_functions E F) ->
  (forall u v x, inc u T -> inc v T -> inc x (source u) -> inc x (source v) ->
    W x u = W x v) ->
  exists x, least_upper_bound (opposite_order (extension_order E F)) T x &
     (source x = unionf T source) &
     (range (graph x) = unionf T (fun u => (range (graph u))))&
     (graph x) = unionf T graph.   (* 29 *)
```

If f is a function with source A and if its target is an ordered set, the supremum of the image $f\langle A\rangle$ is denoted by $\sup_{x\in A} f(x)$. The infimum is denoted by $\inf_{x\in A} f(x)$. For typographical reasons, for in-text formulas, the notations $\sup_{x\in A} f(x)$, $\inf_{x\in A} f(x)$ are preferred. If f is the identity function, one can write $\sup_{x\in A} x$ or $\inf_{x\in A} x$. We give a characterization of the sup and inf of a function.

```
Definition is_sup_fun r f := least_upper_bound r (image_of_fun f).
```

```
Definition is_inf_fun r f := greatest_lower_bound r (image_of_fun f).

Lemma is_sup_fun_pr r f x: order r -> substrate r = target f ->
  is_function f ->
  (is_sup_fun r f x <-> (inc x (target f)
   & (forall a, inc a (source f) -> gle r (W a f) x)
   & forall z, inc z (target f) -> (forall a, inc a (source f)
     -> gle r (W a f) z) -> gle r x z)).
Lemma is_inf_fun_pr r f x: order r -> substrate r = target f ->
  is_function f ->
  (is_inf_fun r f x <->  (inc x (target f)
   & (forall a, inc a (source f) -> gle r x (W a f))
   & forall z, inc z (target f) -> (forall a, inc a (source f)
     -> gle r z (W a f)) -> gle r z x)).
```

In general, we consider a family rather than a function (i.e., a functional graph instead of a function).

```
Definition is_sup_graph r f := least_upper_bound r (range f).
Definition is_inf_graph r f := greatest_lower_bound r (range f).
Definition has_sup_graph r  f := has_supremum r (range f).
Definition has_inf_graph r  f := has_infimum r (range f).
Definition sup_graph r f := supremum r (range f).
Definition inf_graph r f := infimum r (range f).
```

Here are the characteristic properties.

```
Lemma is_sup_graph_pr1 r f:
  order r -> sub (range f) (substrate r) -> has_sup_graph r f ->
  least_upper_bound r (range f) (sup_graph r f).
Lemma is_inf_graph_pr1 r f:
  order r -> sub (range f) (substrate r) -> has_inf_graph r f ->
  greatest_lower_bound r (range f) (inf_graph r f).

Lemma is_sup_graph_pr r f x: order r -> sub (range f) (substrate r) ->
  fgraph f ->
  (is_sup_graph r f x <-> (inc x (substrate r)
   & (forall a, inc a (domain f) -> gle r (V a f) x)
   & forall z, inc z (substrate r) -> (forall a, inc a (domain f)
     -> gle r (V a f) z) -> gle r x z)).
Lemma is_inf_graph_pr r f x: order r -> sub (range f) (substrate r) ->
  fgraph f ->
  (is_inf_graph r f x <-> (inc x (substrate r)
   & (forall a, inc a (domain f) -> gle r x (V a f))
   & forall z, inc z (substrate r) -> (forall a, inc a (domain f)
     -> gle r z (V a f)) -> gle r z x)).
```

Assume that $A \subset E$ is a set that has an infimum and a supremum. If A is empty, we know that these elements are the least and greatest elements; otherwise, if $y \in A$ we have $\inf A \leq y \leq \sup A$, hence $\inf A \leq \sup A$. This is Proposition 4 [3, p. 142].

```
Theorem compare_inf_sup1 r A: order r -> sub A (substrate r) ->
  has_supremum r A -> has_infimum r A ->
  A = emptyset ->
  (greatest_element r (infimum r A) &  least_element r (supremum r A)).
```

```
Theorem compare_inf_sup2 r A: order r -> sub A (substrate r) ->
  has_supremum r A -> has_infimum r A ->
  nonempty A -> gle r (infimum r A) (supremum r A).
```

Proposition 5 [3, p. 142] says that sup is increasing and inf is decreasing (as a function from $\mathfrak{P}(E)$ into E, where $\mathfrak{P}$ is ordered by inclusion). Of course, these are only partial functions. As a corollary, consider a family $(x_\iota)_{\iota\in I}$ and $J \subset I$; we have $\sup_{\iota\in J} x_\iota \leq \sup_{\iota\in I} x_\iota$ if both quantities are defined. Note that the first term is the supremum of the restriction of the family to J.

```
Theorem sup_increasing r A B: order r -> sub A (substrate r) ->
  sub B (substrate r) -> sub A B ->
  has_supremum r A -> has_supremum r B ->
  gle r (supremum r A) (supremum r B).
Theorem inf_decreasing r A B: order r -> sub A (substrate r) ->
  sub B (substrate r) -> sub A B ->
  has_infimum r A -> has_infimum r B ->
  gle r (infimum r B) (infimum r A) .
Lemma sup_increasing1 r f j:
  order r -> fgraph f -> sub (range f) (substrate r) -> sub j (domain f) ->
  has_sup_graph r f  -> has_sup_graph r (restr f j) ->
  gle r (sup_graph r (restr f j)) (sup_graph r f).
Lemma inf_decreasing1 r f j:
  order r -> fgraph f -> sub (range f) (substrate r) -> sub j (domain f) ->
  has_inf_graph r f  -> has_inf_graph r (restr f j) ->
  gle r (inf_graph r f) (inf_graph r (restr f j)) .
```

Proposition 6 [3, p. 143] says that if f and g are two functions of type $F \to E$, if $f(x) \leq g(x))$ for all $x \in F$ then $\sup f \leq \sup g$, provided that both quantities are defined. In fact, it is stated as: if for all $\iota \in I$ we have $x_\iota \leq y_\iota$, then $\sup_{\iota\in I} x_\iota \leq \sup_{\iota\in I} y_\iota$, and $\inf_{\iota\in I} x_\iota \leq \inf_{\iota\in I} y_\iota$.

```
Lemma sup_increasing2 r f f':
  order r -> fgraph f ->  fgraph f' ->  domain f = domain f' ->
  sub (range f) (substrate r) -> sub (range f') (substrate r) ->
  has_sup_graph r f -> has_sup_graph r f' ->
  (forall x , inc x (domain f) -> gle r (V x f) (V x f')) ->
  gle r (sup_graph r f) (sup_graph r f').
Lemma inf_increasing2 r f f':
  order r -> fgraph f ->  fgraph f' ->  domain f = domain f' ->
  sub (range f) (substrate r) -> sub (range f') (substrate r) ->
  has_inf_graph r f  ->  has_inf_graph r f' ->
  (forall x , inc x (domain f) -> gle r (V x f) (V x f')) ->
  gle r (inf_graph r f) (inf_graph r f').
```

Proposition 7 [3, p. 143] is the following. Consider a family $(x_\iota)_{\iota\in I}$, and let $(J_\lambda)_{\lambda\in L}$ be a covering of I. The family $(x_\iota)_{\iota\in J_\lambda}$ is the restriction of (x_ι) to J_λ; we assume that it has a supremum $\sup_{\iota\in J_\lambda} x_\iota$, and we consider the family $(\sup_{\iota\in J_\lambda} x_\iota)_{\lambda\in L}$. This family has a supremum if and only if $(x_\iota)_{\iota\in I}$ has one, and the values are the same; the second equality in (1) is true under similar conditions.

$$\sup_{\iota\in I} x_\iota = \sup_{\lambda\in L}\Big(\sup_{\iota\in J_\lambda} x_\iota\Big), \qquad \inf_{\iota\in I} x_\iota = \inf_{\lambda\in L}\Big(\inf_{\iota\in J_\lambda} x_\iota\Big). \tag{1}$$

The first lemma here says that if x is a least upper bound for one family, it is also the least upper bound for the other one. Finally, since the supremum is a least upper bound, we get the result by uniqueness.

```
Lemma sup_distributive r f c x: (* 48 *)
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  covering c (domain f) ->
  (forall l, inc l (domain c) -> has_sup_graph r (restr f (V l c))) ->
  (is_sup_graph r f x  <->
   is_sup_graph r (L (domain c) (fun l => sup_graph r (restr f (V l c)))) x).
Lemma inf_distributive r f c x:  (* 48 *)
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  covering c (domain f) ->
  (forall l, inc l (domain c) -> has_inf_graph r (restr f (V l c))) ->
  (is_inf_graph r f x <->
   is_inf_graph r (L (domain c) (fun l => inf_graph r (restr f (V l c)))) x).
Lemma sup_distributive1 r f c:
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  covering c (domain f) ->
  (forall l, inc l (domain c) -> has_sup_graph r (restr f (V l c))) ->
  (has_sup_graph r f <->
   has_sup_graph r (L (domain c) (fun l => sup_graph r (restr f (V l c))))).
Lemma inf_distributive1 r f c:
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  covering c (domain f) ->
  (forall l, inc l (domain c) -> has_inf_graph r (restr f (V l c))) ->
  (has_inf_graph r f <->
   has_inf_graph r (L (domain c) (fun l => inf_graph r (restr f (V l c))))).

Theorem sup_distributive2 r f c: (* 19 *)
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  covering c (domain f) ->
  (forall l, inc l (domain c) -> has_sup_graph r (restr f (V l c))) ->
  ((has_sup_graph r f <->
    has_sup_graph r (L (domain c) (fun l => sup_graph r (restr f (V l c))))) &
  (has_sup_graph r f -> sup_graph r f =
    sup_graph r (L (domain c) (fun l => sup_graph r (restr f (V l c)))))).

Theorem inf_distributive2 r f c:  (* 19 *)
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  covering c (domain f) ->
  (forall l, inc l (domain c) -> has_inf_graph r (restr f (V l c))) ->
  ((has_inf_graph r f <->
    has_inf_graph r (L (domain c) (fun l => inf_graph r (restr f (V l c))))) &
  (has_inf_graph r f -> inf_graph r f =
    inf_graph r (L (domain c) (fun l => inf_graph r (restr f (V l c)))))).
```

¶ Corollary. Let $(x_{\lambda\mu})_{(\lambda,\mu)\in L\times M}$ be a double family of elements of an ordered set E such that for each $\mu \in M$ the family $(x_{\lambda\mu})_{\lambda\in L}$ has a least upper bound in E. This family is the restriction of the double family to $L \times \{\mu\}$. For the double family to have a least upper bound in E it is necessary and sufficient that $(\sup_{\lambda\in L} x_{\lambda\mu})_{\mu\in M}$ has a least upper bound, and the bounds are the same. The second equality in (2) holds under similar conditions.

$$\sup_{(\lambda,\mu)\in L\times M} x_{\lambda\mu} = \sup_{\mu\in M}\big(\sup_{\lambda\in L} x_{\lambda\mu}\big), \qquad \inf_{(\lambda,\mu)\in L\times M} x_{\lambda\mu} = \inf_{\mu\in M}\big(\inf_{\lambda\in L} x_{\lambda\mu}\big). \tag{2}$$

```
Definition partial_fun f x m := restr f (product x (singleton m)).

Lemma sup_distributive3 r f x y:
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  domain f = product x y ->
  (forall m, inc m y -> has_sup_graph r (partial_fun f x m)) ->
  ((has_sup_graph r f <->
    has_sup_graph r (L y (fun m => sup_graph r  (partial_fun f x m)))) &
  (has_sup_graph r f -> sup_graph r f =
    sup_graph r (L y (fun m => sup_graph r (partial_fun f x m))))).
Lemma inf_distributive3 r f x y r:
  order r -> fgraph f ->  sub (range f) (substrate r) ->
  domain f = product x y ->
  (forall m, inc m y -> has_inf_graph r (partial_fun f x m)) ->
  ((has_inf_graph r f <->
    has_inf_graph r (L y (fun m => inf_graph r  (partial_fun f x m)))) &
  (has_inf_graph r f -> inf_graph r f =
    inf_graph r (L y (fun m => inf_graph r (partial_fun f x m))))).
```

Proposition 8 [3, p. 144] says that if we have a family of ordered sets E_ι, a subset A of $\prod E_\iota$, and if $A_\iota = \mathrm{pr}_\iota A$, then A has a least upper bound of the form $(x_\iota)_\iota$ if and only if each A_ι has one, and there is equality; a similar property holds for the greatest lower bound.

$$\sup A = (\sup A_\iota)_{\iota\in I} = \Big(\sup_{x\in A} \mathrm{pr}_\iota x\Big)_{\iota\in I}, \qquad \inf A = (\inf A_\iota)_{\iota\in I} = \Big(\inf_{x\in A} \mathrm{pr}_\iota x\Big)_{\iota\in I}.$$

```
Theorem sup_in_product g A: (* 93 *)
  let f := fam_of_substrates g in
  let Ai:= fun i => (image_by_fun (pr_i f i) A) in
    let has_sup :=
      forall i, inc i (domain g) -> has_supremum (V i g) (Ai i) in
    order_fam g -> sub A (productb f) ->
    ((has_sup <-> has_supremum (order_product g) A) &
      (has_sup -> supremum (order_product g) A =
        L (domain g) (fun i => supremum (V i g) (Ai i)))).
Theorem inf_in_product g A:  (* 93 *)
  let f := fam_of_substrates g in
  let Ai:= fun i => (image_by_fun (pr_i f i) A) in
    let has_inf :=
      forall i, inc i (domain g) -> has_infimum (V i g) (Ai i) in
    order_fam g -> sub A (productb f) ->
    ((has_inf <-> has_infimum (order_product g) A) &
      (has_inf -> infimum (order_product g) A =
        L (domain g) (fun i => infimum (V i g) (Ai i)))).
```

Proposition 9 [3, p. 144] assumes that E is an ordered set, F is a subset of E and A is a subset of F. It can happen that one of $\sup_E A$ and $\sup_F A$ exists, but not the other; they may be unequal. If the objects exist we have

$$\sup_E A \le \sup_F A, \qquad \inf_E \ge \inf_F A \qquad (F \subset E).$$

If $\sup_E A$ exists and is in F, it is $\sup_F A$.

```
Theorem sup_induced1 r A F: order r -> sub F (substrate r) -> sub A F ->
  has_supremum r A -> has_supremum (induced_order r F) A ->
```

```
  gle r (supremum r A) (supremum (induced_order r F) A).
Theorem inf_induced1 r A F: order r -> sub F (substrate r) -> sub A F ->
  has_infimum r A -> has_infimum (induced_order r F) A ->
  gle r  (infimum (induced_order r F) A) (infimum r A).
Theorem sup_induced2 r A F: order r -> sub F (substrate r) -> sub A F ->
  has_supremum r A -> inc (supremum r A)  F ->
  (has_supremum (induced_order r F) A &
    supremum r A = supremum (induced_order r F) A).
Theorem inf_induced2 r A F: order r -> sub F (substrate r) -> sub A F ->
  has_infimum r A -> inc (infimum r A)  F ->
  (has_infimum (induced_order r F) A &
    infimum r A = infimum (induced_order r F) A).
```

2.10 Directed sets

An ordered set is said left or right *directed* if every doubleton is bounded (above or below).

```
Definition right_directed r :=
  order r & forall x, forall y, inc x (substrate r) -> inc y (substrate r) ->
    bounded_above r (doubleton x y).
Definition left_directed r :=
  order r & forall x, forall y, inc x (substrate r) -> inc y (substrate r) ->
    bounded_below r (doubleton x y).
```

We rewrite the definition as: for all x and y there is a z such that $x \leq z$ and $y \leq z$. A set that has a greatest element is right directed. A product of directed sets is directed[6]. A cofinal set of a directed set is directed for the induced order.

```
Lemma right_directed_pr r:
  right_directed r <-> (order r &
    forall x, forall y, inc x (substrate r) -> inc y (substrate r) -> exists z,
      inc z (substrate r) & gle r x z  & gle r y z).
Lemma left_directed_pr r:
  left_directed r <-> (order r &
    forall x, forall y, inc x (substrate r) -> inc y (substrate r) -> exists z,
      inc z (substrate r) & gle r z x  & gle r z y).

Lemma greatest_right_directed  r: order r ->
  (exists a, greatest_element r a )  -> right_directed r .
Lemma least_left_directed r: order r ->
  (exists a, least_element r a )  -> left_directed r.
Lemma opposite_right_directed  r: is_graph r ->
  (right_directed r <-> left_directed(opposite_order r)).
Lemma opposite_left_directed r: is_graph r ->
  (left_directed r <-> right_directed(opposite_order r)).
Lemma product_right_directed g: (* 20 *)
  order_fam g ->
  (forall i, inc i (domain g) -> right_directed (V i g))
  -> right_directed (order_product g).
Lemma product_left_directed g:  (* 20 *)
  order_fam g ->
  (forall i, inc i (domain g) -> left_directed (V i g))
  -> left_directed (order_product g).
```

[6] This requires the axiom of choice

```
Lemma cofinal_right_directed r A:
  right_directed r -> cofinal_set r A -> right_directed (induced_order r A).
Lemma coinitial_left_directed  r A:
  left_directed r -> coinitial_set r A -> left_directed (induced_order r A).
```

Proposition 10 [3, p. 145] says that in a right directed set, a maximal element is the greatest element.

```
Theorem right_directed_maximal  r x:
  right_directed r -> maximal_element r x -> greatest_element r x.
Theorem left_directed_minimal r x:
  left_directed r -> minimal_element r x -> least_element r x.
```

2.11 Lattices

A *lattice* is an ordered set on which each pair has a least upper bound and a greatest lower bound.

```
Definition lattice r :=  order  r &
  forall x, forall y, inc x (substrate r) -> inc y (substrate r) ->
    (has_supremum r (doubleton x y) & has_infimum r (doubleton x y)).

Lemma lattice_sup_pr r a b:
  lattice r -> inc a (substrate r) -> inc b (substrate r) ->
  (gle r a (sup r a b) &  gle r b (sup r a b) &
    forall z, gle r a -> gle r b z -> gle r (sup r a b) z).
Lemma lattice_inf_pr r a b:
  lattice r -> inc a (substrate r) -> inc b (substrate r) ->
  (gle r (inf r a b) a &  gle r (inf r a b) b&
    forall z, gle r z a ->  gle r z b -> gle r z (inf r a b)).
```

The powerset is a lattice. In fact each set has a supremum and an infimum.

```
Lemma inf_inclusion A x y: sub x A -> sub y A  ->
  greatest_lower_bound (inclusion_order A) (doubleton x y) (intersection2 x y).
Lemma sup_inclusion A x y: sub x A -> sub y A  ->
  least_upper_bound (inclusion_order A) (doubleton x y) (union2 x y).
Lemma powerset_lattice A: lattice (inclusion_order A).
```

The product of lattices is a lattice. This is an easy consequence of Proposition 8, and the fact that $\mathrm{pr}_\iota A$ is a doubleton if A is a doubleton. A lattice is a directed set.

```
Lemma product_lattice g:
  order_fam g ->
  (forall i, inc i (domain g) -> lattice (V i g))
  -> lattice (order_product g).  (* 37 *)
Lemma lattice_directed  r:
  lattice r -> (right_directed r & left_directed r).
```

Other examples. The set of integers, with the order "x divides y" is a lattice. The set of subgroups of a group (ordered by inclusion) is a lattice. The set of topologies on a set is a lattice. The set of real functions on an interval is a lattice. The opposite of a lattice is a lattice.

```
Lemma lattice_opposite: forall r, lattice r -> lattice (opposite_order r).
```

2.12 Totally ordered sets

Two elements of a preordered set E are said comparable if the relation "$x \leq y$ or $y \leq x$" is true. A set E is said to be *totally ordered* if is is ordered and if any two elements of E are comparable.

```
Definition total_order r :=
  order r & forall x y, inc x (substrate r) -> inc y (substrate r) ->
    (gle r x y \/ gle r y x).
```

An order satisfies $G \circ G = G$ and $G \cap G^{-1} = \Delta_E$. It is total if moreover $G \cup G^{-1} = E \times E$. We have $x < y$ or $x > y$ or $x = y$. We have $x < y$ or $y \leq x$. A subset of a totally ordered set is totally ordered. A small set is totally ordered. The opposite of a totally ordered set is totally ordered.

```
Lemma total_order_pr r:
  total_order r <->
    (r \cg r = r &
    intersection2 r (inverse_graph r) = identity_g (substrate r)&
    union2 r (inverse_graph r) =  coarse (substrate r)).  (* 22 *)
Lemma total_order_pr1 r x y:
  total_order r ->  inc x (substrate r) -> inc y (substrate r) ->
    (glt r x y \/ glt r y x \/ x = y).
Lemma total_order_pr2  r x y:
  total_order r ->  inc x (substrate r) -> inc y (substrate r) ->
    (glt r x y \/ gle r y x).

Lemma total_order_sub r x:
  total_order r ->  sub x (substrate r) -> total_order (induced_order r x).
Lemma total_order_counterexample:
  ~ (total_order (inclusion_order two_points)).
Lemma total_order_opposite r:
 total_order r -> total_order (opposite_order r).
```

If $x \leq y$, then $\sup(x, y) = y$ and $\inf(x, y) = x$, hence a totally ordered set is a lattice.

```
Lemma sup_comparable r x y: gle r x y ->
  order r -> least_upper_bound r (doubleton x y) y.
Lemma inf_comparable r x y: gle r x y ->
  order r -> greatest_lower_bound r (doubleton x y) x.
Lemma sup_comparable1 r x y: order r -> gle r x y -> sup r x y = y.
Lemma inf_comparable1 r x y: order r -> gle r x y -> inf r x y = x.
Lemma total_order_lattice  r:  total_order r -> lattice r.
Lemma total_order_directed r:
  total_order r -> (right_directed r & left_directed r).
```

Proposition 11 [3, p. 147] says that if f is strictly monotone and the ordering on the source is total, then f is injective. If f is strictly increasing, it is a morphism (an isomorphism onto the image). We first show that if f is strictly increasing, then it is increasing.

```
Lemma increasing_fun_from_strict f r r':
  strict_increasing_fun f r r' -> increasing_fun f r r'.
Lemma decreasing_fun_from_strict f r r':
  strict_decreasing_fun f r r' -> decreasing_fun f r r'.
Theorem total_order_monotone_injective f r r':
```

```
  total_order r -> strict_monotone_fun f r r' -> injection f.
Theorem total_order_increasing_morphism f r r':
 total_order r -> strict_increasing_fun f r r' -> order_morphism f r r'.
```

Proposition 12 [3, p. 147] says that in a totally ordered set E, an element x is the least upper bound of a subset X if and only if it is an upper bound and, for all $y < x$, there is a $z \in X$ such that $y < z$ and $z \leq x$.

```
Theorem sup_in_total_order r X x: total_order r -> sub X (substrate r)->
  (least_upper_bound r X x <-> (upper_bound r X x &
    (forall y, glt r y x -> exists z, inc z X & glt r y z &  gle r z x))).
Theorem inf_in_total_order r X x: total_order r -> sub X (substrate r)->
  (greatest_lower_bound r X x <-> (lower_bound r X x &
    (forall y, glt r x y -> exists z, inc z X & glt r z y &  gle r x z))).
```

2.13 Intervals

There are many definitions of an *interval*. The set of all x such that $a \leq x \leq b$ is called the closed interval and denoted $[a, b]$; the set of all x such that $a < x < b$ is called the open interval and denoted $]a, b[$; intervals can be semi open. One can drop one of the conditions, for instance the set of all x such that $x < b$ is denoted by $]\leftarrow, b[$, this is an unbounded interval.

The letters o, c, u stand for open, close, and unbounded.

```
Definition interval_oo r a b := Zo(substrate r)(fun z => glt r a z & glt r z b).
Definition interval_oc r a b := Zo(substrate r)(fun z => glt r a z & gle r z b).
Definition interval_ou r a  := Zo (substrate r) (fun z  => glt r a z).
Definition interval_co r a b := Zo(substrate r)(fun z => gle r a z & glt r z b).
Definition interval_cc r a b := Zo(substrate r)(fun z => gle r a z & gle r z b).
Definition interval_cu r a  := Zo (substrate r) (fun z  => gle r a z).
Definition interval_uo r b := Zo (substrate r) (fun z  =>  glt r z b).
Definition interval_uc r b := Zo (substrate r) (fun z => gle r  z b).
Definition interval_uu r := Zo (substrate r) (fun z => True).

Definition is_closed_interval r x := exists a, exists b,
  inc a (substrate r) & inc b (substrate r) & gle r a b & x = interval_cc r a b.
Definition is_open_interval r x := exists a, exists b,
  inc a (substrate r) & inc b (substrate r) & gle r a b & x = interval_oo r a b.
Definition is_semi_open_interval r x := exists a, exists b,
  inc a (substrate r) & inc b (substrate r)  & gle r a b &
  (x = interval_oc r a b \/ x = interval_co r a b).
Definition is_bounded_interval r x := is_closed_interval r x \/
  is_open_interval r x \/ is_semi_open_interval r x.
Definition is_left_unbounded_interval r x :=
  exists b, inc b (substrate r) & (x = interval_uc r b \/ x = interval_uo r b).
Definition is_right_unbounded_interval r x :=
  exists a, inc a (substrate r) & (x = interval_cu r a \/ x = interval_ou r a).
Definition is_unbounded_interval r x :=
  is_left_unbounded_interval r x \/ is_right_unbounded_interval r x  \/
  x = interval_uu r.
Definition is_interval r x :=
  is_bounded_interval r x  \/ is_unbounded_interval r x.
```

A non-empty interval $[a, b]$ has a least and greatest elements, which are a and b respectively.

```
Lemma the_least_interval r a b: order r ->
  gle r a b ->   the_least_element (induced_order r (interval_cc r a b)) = a.
Lemma the_greatest_interval r a b: order r ->
  gle r a b -> the_greatest_element (induced_order r (interval_cc r a b)) = b.
```

A closed interval is never empty; however $[a, a[$, $]a, a]$ and $]a, a[$ are empty.

```
Lemma nonempty_closed_interval r x:
  order r -> is_closed_interval r x -> nonempty x.
Lemma singleton_interval r a:
  order r -> inc a (substrate r) -> is_singleton (interval_cc r a a).
Lemma empty_interval r a:
  order r -> inc a (substrate r) ->
  (empty (interval_co r a a) &  empty (interval_oc r a a)&
    empty (interval_oo r a a)).
```

The only non trivial result here is Proposition 13 [3, p. 148] that says that, in a lattice, the intersection of two intervals is an interval. We start with a short proof.

Let's say that an interval is of type L if it is left unbounded, of type R if it is right unbounded (the interval U $=] \leftarrow, \rightarrow [$ is of both types, and $U \cap X = X$ for any interval X). Obviously, each interval is the intersection of two intervals of type L and R (if the interval is unbounded, consider intersection with U). The intersection of two intervals is thus of the form $(L_1 \cap R_1) \cap (L_2 \cap R_2) = (L_1 \cap L_2) \cap (R_1 \cap R_2)$. This is of the form $L_3 \cap R_3$.

```
Definition is_lu_interval r x :=
   x = interval_uu r \/ is_left_unbounded_interval r x.
Definition is_ru_interval r x :=
   x= interval_uu r \/ is_right_unbounded_interval r x.
Lemma intersection4  A B C D:
  intersection2 (intersection2 A B) (intersection2 C D)
  = intersection2 (intersection2 A C) (intersection2 B D).
Lemma intersection_i1  r x:
  is_interval r x -> intersection2 x (interval_uu r) = x. (* 18 *)
Lemma intersection_i2  r x:
  is_interval r x ->
  (exists u, exists v, is_lu_interval r u & is_ru_interval r v &
    intersection2 u v = x). (* 40 *)
Lemma intersection_i3  r x y: lattice r ->
  is_left_unbounded_interval r x -> is_left_unbounded_interval r y ->
  is_left_unbounded_interval r (intersection2 x y). (* 38 *)
Theorem intersection_interval r x y:
  lattice r -> is_interval r x -> is_interval r y ->
  is_interval r (intersection2 x y). (* 49 *)
```

The total size of the proof of the theorem is 150 lines, including the tactics designed for this theorem. Originally, its size was 841 lines, reduced to 615 by using adequate tactics. Details can be found in section 11.7.

Chapter 3

Well-ordered sets

This chapter defines the notion of a well-ordering, and the lexicographic ordering of a product of ordered sets. We show Zermelo's theorem (there exists a well-ordering) and Zorn's lemma (every inductive ordered set has a maximal element). These theorems are equivalent to the axiom of choice, thus non-constructive. We introduce the principle of transfinite induction: given a well-ordered set and a term T, there exists a unique function f such that $f(x)$ is $\mathrm{T}(f'_x)$, where f'_x is the restriction of f to the set of all y such that $y < x$.

3.1 Segments of a well-ordered set

A relation $\mathrm{R}\{x, y\}$ is said to be a *well-ordering relation* between x and y if R is an order relation between x and y and if for each non-empty set E on which $\mathrm{R}\{x, y\}$ induces an order relation, E, ordered by this relation, has a least element. A set E ordered by an ordering Γ is said to be *well-ordered* if the relation $y \in \Gamma\langle x\rangle$ is a well ordering between x and y; Γ is then said to be a well-ordering on E.

These definitions are a bit complicated. We first rewrite "E, ordered by this relation, has a least element" as "the ordering induced by this relation on E has a least element", or simply "the graph of R on E has a least element". Assume that R is an order relation; then $\mathrm{R}\{x, y\}$ induces an order relation on E provided that it is reflexive on E, thus, whenever $x \in \mathrm{E}$, we have $\mathrm{R}\{x, x\}$. Let's consider the second definition. Recall that for Bourbaki, Γ is a correspondence and $\Gamma\langle x\rangle$ is (by abuse of notations) the image of the singleton $\{x\}$ under the correspondence. Let G be the graph of Γ. Then $y \in \Gamma\langle x\rangle$ is the same as $(x, y) \in \mathrm{G}$. It implies that x and y belong to E (conversely, if $x \in \mathrm{E}$, then $(x, x) \in \mathrm{G}$). Denote this relation by $\mathrm{R}\{x, y\}$. The condition "A is a set on which $\mathrm{R}\{x, y\}$ induces an order relation" is equivalent to $\mathrm{A} \subset \mathrm{E}$. The condition "A, ordered by this relation, has a least element", written as "the graph of R on A has a least element" becomes "the order induced by G on A has a least element". Here are the two definitions

```
Definition worder_r (r: Set -> Set -> Prop) :=
  order_r r & forall x, (forall a, inc a x -> r a a) -> nonempty x ->
    exists y, least_element (graph_on r x) y.

Definition worder r :=
  order r & forall x, sub x (substrate x) -> nonempty x ->
    exists y, least_element (induced_order r x) y.
```

```
Lemma worder_or r: worder r -> order r.
Lemma wordering_pr r x: worder_r r ->
  (forall a, inc a x -> r a a)  ->
  (substrate (graph_on r x) = x & worder (graph_on r x)).
```

Bourbaki notes that a totally ordered set with two elements is well-ordered. In fact, it contains a and b such that $a < b$, so that the graph is the set with three elements (a,a), (a,b) and (b,b). All total orders on sets with two elements are isomorphic. We consider here the case where a and b are the elements of the doubleton two_points.

```
Definition canonical_doubleton_order :=
  union2 (doubleton (J TPa  TPa) (J TPb TPb)) (singleton (J TPa TPb)).
Lemma canonical_doubleton_order_gle x y:
  gle canonical_doubleton_order x y <->
  ( (x= TPa & y = TPa) \/  (x= TPb & y = TPb) \/  (x= TPa & y = TPb)).
Lemma canonical_doubleton_order_sr:
  substrate canonical_doubleton_order = two_points.
Lemma canonical_doubleton_order_wor:
  worder canonical_doubleton_order. (* 31 *)
```

By considering the least element of the doubleton $\{x, y\}$, one deduces that a well-ordering is a total ordering. Every subset that is bounded above has a least upper bound. A subset of a well-ordered set is well-ordered. Adjoining a greatest element to a well-ordered set gives a well-ordered set. A nonempty well-ordered set has a least element.

```
Lemma worder_total r: worder r -> total_order r.
Lemma worder_hassup r A: worder r -> sub A (substrate r) ->
  bounded_above r A -> has_supremum r A.
Lemma induced_trans r x y:
  order r -> sub x y-> sub y (substrate r) ->
  induced_order r x = induced_order (induced_order r y) x.
Lemma induced_wor r A: worder r -> sub A (substrate r) ->
  worder (induced_order r A).
Lemma worder_adjoin_greatest r a: worder r-> ~ (inc a (substrate r))
  -> worder (order_with_greatest r a). (* 22 *)
Lemma worder_least r: worder r -> nonempty (substrate  r) ->
  exists y, least_element r y.
```

Some useful small lemmas. If $\leq$ is a relation on E, then $x < y$ means $x \leq y$ and $x \neq y$. Thus $x < x$ is false and $x < y$ implies that x and y are in E. If we have a total order on E, if $x \in$ E and $y \in$ E then $x < y$ or $y \leq x$; in the case of an order, one of these relations is false.

```
Lemma inc_lt1_substrate r x y: glt r x y -> inc x (substrate r).
Lemma inc_lt2_substrate r x y: glt r x y -> inc y (substrate r).
Lemma not_le_gt r x y: order r -> gle r x y -> glt r y x -> False.
Lemma not_lt_self r x: glt r x x -> False.
```

In 1908, Zermelo presented an alternative, simpler proof of his theorem (see [12, pages 183-189]), this is Zermelo-bis below.

```
Lemma Zermelo_bis E: exists r, worder r & substrate r = E.
```

Consider two strictly increasing functions f and g, with the same source and range. These functions are equal provided that the source is well-ordered (on **Z**, the mapping $x \mapsto$

$x+1$ is a strictly increasing bijection, different from the identity, so that the condition on the order is necessary). Assume $f \neq g$, and let x be the smallest element such that $f(x) \neq g(x)$. Since $f(x)$ is in the range of g, there is z such that $f(x) = g(z)$. If $z < x$ then $f(z) < f(x) = g(z)$, contradicting the definition of x. Since the source is totally ordered, we get $x < z$ (since $x \neq z$), hence $g(x) < g(z) = f(x)$. Since $g(x)$ is in the range of f, the same argument gives $f(x) < g(x)$, absurd.

```
Lemma strict_increasing_extens f g r r':
  strict_increasing_fun f r r'-> strict_increasing_fun g r r' -> worder r ->
  range (graph f) = range (graph g) ->
  f = g. (* 42 *)
```

A *segment* S in an ordered set E is such that, if $x \in S$ and $y \in E$ and $y \leq x$, then $y \in S$. Note that if E is the substrate of $\leq$ then $y \leq x$ implies $y \in E$. An interval of the form $]\leftarrow, x[$ is a segment; it will be denoted by S_x, and is called the *segment with endpoint x*. The interval $]\leftarrow, x]$ will be called the closed segment.

```
Definition is_segment r s :=
  sub s (substrate r) & forall x y, inc x s -> gle r y x -> inc y s.
Definition segment r x := interval_uo r x.
Definition segment_c r x := interval_uc r x.
```

The 20-some following lemmas sometimes assume that we have an order $\leq$ on E, and that S is a segment. We assume $x' \in E$. They state that if $x \in S$ and $y < x$ then $y \in S$. If $y \in]\leftarrow, x[$ then $y < x$. We have $S \subset E$. We have $]\leftarrow, x[\subset E$, $]\leftarrow, x] \subset E$. We have $x \notin]\leftarrow, x[$ and $x' \in]\leftarrow, x']$. We have $y < x'$ if and only if $y \in]\leftarrow, x'[$ and $y \leq x'$ if and only if $y \in]\leftarrow, x']$. We have $]\leftarrow, x'] =]\leftarrow, x'[\cup \{x'\}$. We have $]\leftarrow, x[_F \subset F$, where the notation W_F means that we restrict $\leq$ to F. The empty set and E are segments. Intersections and unions of segments are segments. If S' is a segment for the order induced on S, then S' is a segment. Finally $]\leftarrow, x'[$ is a segment.

```
Lemma lt_in_segment r s x y:
  is_segment r s ->  inc x s -> glt r y x -> inc y s.
Lemma inc_segment r x y: inc y (segment r x) -> glt r y x.
Lemma not_in_segment  r x:  inc x (segment r x) -> False.
Lemma sub_segment r x: sub (segment r x) (substrate r).
Lemma sub_segment1 r s:  is_segment r s -> sub s (substrate r).
Lemma sub_segment2 r x y:
  sub (segment (induced_order r x) y) x.
Lemma segment_inc r x y:
  glt r y x -> inc y (segment r x).
Lemma segment_rw r x y:
  inc y (segment r x) <-> glt r y x.
Lemma segmentc_rw r x y:
  inc y (segment_c r x) <-> gle r y x.
Lemma inc_bound_segmentc r x: order r -> inc x (substrate r) ->
  inc x (segment_c r x).
Lemma sub_segmentc r x: sub (segment_c r x) (substrate r).
Lemma segment_c_pr r x: order r -> inc x (substrate r) ->
  tack_on (segment r x) x = segment_c r x.
Lemma empty_is_segment r: is_segment r emptyset.
Lemma substrate_is_segment r: order r -> is_segment r (substrate r).
Lemma intersection_is_segment r s: order r -> nonempty s ->
```

```
  (forall x, inc x s -> is_segment r x) -> is_segment r (intersection s).
Lemma union_is_segment r s: order r ->
  (forall x, inc x s -> is_segment r x) -> is_segment r (union s).
Lemma unionf_is_segment r j s: order r ->
  (forall x, inc x j -> is_segment r (s x)) -> is_segment r (unionf j s).
Lemma subsegment_is_segment r s s': order r ->
  is_segment r s -> is_segment (induced_order r s) s' -> is_segment r s'.
Lemma segment_is_segment r x: order r ->
  inc x (substrate r) -> is_segment r (segment r x).
```

Proposition 1 [3, p. 149] is the converse of the previous lemma. In a well-ordered set, a segment is either the whole set or of the form S_x. If an ordered set has a least element a, then $S_x = [a, x[$; hence in a well-ordered set E, a segment S is either E, or else E is not empty, has a least element a and $S = [a, x[$.

```
Theorem well_ordered_segment r s: worder r -> is_segment r s ->
  s = substrate r \/ (exists x,  inc x (substrate r) &  s = segment r x). (* 22 *)
Lemma segment_alt r x a: order r -> least_element r a ->
  segment r x = interval_co r a x.
Lemma segment_alt1 r s: worder r -> is_segment r s ->
  s = substrate r \/ (exists x, exists a, s = interval_co  r a x).
```

Some useful lemmas. We consider a well-ordered set. If S and S′ are segments, then $S \subset S'$ or $S' \subset S$. If $S \subset S'$, if $x \in S$, the segments with endpoint x in S or S′ coincide. If $x \leq y$, then $S_x \subset S_y$ and $S_x \times S_x \subset S_y \times S_y$. If $z \leq y$ and $y \in S_x$ then $z \in S_x$. The set $]\leftarrow, x]$ is a segment. If S is a segment and $x \in S$, then S_x is the segment with endpoint x for the order induced on S. It is also the segment with endpoint x for the order induced on $]\leftarrow, y]$ or $]\leftarrow, y[$ if $x < y$.

```
Lemma segment_monotone r x y: order r -> gle r x y ->
  sub (segment r x)  (segment r y).
Lemma segment_dichot_sub r x y:
  worder r -> is_segment r x -> is_segment r y ->
 (sub x y \/ sub y x).
Lemma le_in_segment r x y z: order r -> inc x (substrate r) ->
  inc y (segment r x) -> gle r z y -> inc z (segment r x).
Lemma coarse_segment_monotone r x y: order r -> gle r x y ->
  sub (coarse (segment r x)) (coarse (segment r y)).
Lemma tack_on_segment r x:
  order r -> inc x (substrate r) ->
  is_segment r (segment_c r x).
Lemma segment_induced_a r s x:
  order r -> is_segment r s -> inc x s ->
  segment (induced_order r s) x = segment r x.
Lemma segment_induced r x x0: order r ->  glt r x0 x ->
  segment (induced_order r (segment r x)) x0 = segment r x0.
Lemma segment_induced1 r x x0: order r ->  glt r x0 x ->
  segment (induced_order r (segment_c r x)) x0 = segment r x0.
```

In a totally ordered set E, the union of all segments is E minus its greatest elements (there is at most one such element). In fact the union is the set of all x for which there is a y such that $x < y$ (remember that $x \leq y$ implies $x \in E$ and $y \in E$).

```
Lemma union_segments r: total_order r ->  (* 22 *)
  let E := substrate r in
```

```
    let A := union (fun_image E (fun x => (segment r x))) in
      ( (forall x, ~ (greatest_element r x)) -> A = E)
      & (forall x, greatest_element r x -> A = complement E (singleton x)).
```

The function $x \mapsto]\leftarrow, x[$ is strictly increasing , hence injective in a totally ordered set.

```
Lemma segment_monotone1 r x y: total_order r ->
  inc x (substrate r) ->  inc y (substrate r) ->
  sub (segment r x)(segment r y) -> gle r x y.
Lemma segment_injective r x y: total_order r ->
  inc x (substrate r) ->  inc y (substrate r) -> segment r x = segment r y ->
  x = y.
Lemma segment_injective1 r x y: worder r ->
  inc x (substrate r) ->  inc y (substrate r) -> segment r x = segment r y ->
  x = y.
```

Assume that E is well-ordered. The mapping $x \mapsto S_x$ is a bijection on the set $E^* \setminus \{E\}$ where E^* is the set of all segments of E. This mapping is an order isomorphism. From this, we deduce that $E^* \setminus \{E\}$, hence E^*, is well-ordered. This is Proposition 2 in [3, p. 149].

```
Definition set_of_segments_strict r:=
  fun_image (substrate r) (fun x => (segment r x)).
Definition set_of_segments r:=
  tack_on (set_of_segments_strict r) (substrate r).
Definition  set_of_segments_iso r:=
  BL(segment r) (substrate r) (set_of_segments_strict r).

Lemma inc_set_of_segments r x: worder r ->
  (is_segment r x <-> inc x (set_of_segments r)).
Lemma segmentc_insetof r x: worder r -> inc x (substrate r) ->
  inc (segment_c r x) (set_of_segments r).
Lemma segment_insetof r x: worder r -> inc x (substrate r) ->
  inc (segment r x) (set_of_segments r).
Lemma sub_set_of_segments r x: worder r ->
  inc x (set_of_segments r) -> sub x (substrate r).
Lemma set_of_segments_axiom r: worder r ->
  bl_axioms (segment r) (substrate r) (set_of_segments_strict r).
Lemma set_of_segments_iso_bijective r: worder r ->
  bijection (set_of_segments_iso r).
Theorem set_of_segments_iso_is r: worder r ->
  order_isomorphism (set_of_segments_iso r) r
  (inclusion_suborder (set_of_segments_strict r)).
Theorem set_of_segments_worder r: worder r ->
  worder (inclusion_suborder (set_of_segments r)).  (* 43 *)
```

We state Lemma 1 [3, p. 150]. *Let* $(X_\alpha)_{\alpha\in A}$ *be a family of ordered sets, directed with respect to the relation* $\subset$. *Suppose that, for each pair of indices* (α,β) *such that* $X_\alpha \subset X_\beta$, *the ordering induced on* X_α *by that of* X_β *is identical with the given ordering on* X_α. *Under these conditions there exists a unique ordering on then set* $E = \bigcup_{\alpha\in A} X_\alpha$ *which induces the given ordering on each* X_α.

We start with a bunch of definitions. We introduce first the notion of a family of orderings or well-orderings. We say that the family is monotone if, whenever $X_\alpha \subset X_\beta$, the ordering induced on X_α by that of X_β is identical with the given ordering on X_α. We consider two special cases of monotone families: in the first case the family is directed, this means that for

any α and β, there is γ such that $X_\gamma \subset X_\alpha$ or $X_\gamma \subset X_\beta$. In the second case, we assume that X'_α is a segment of X_β or X_β is a segment of X_α. Finally, we define the extension of a family.

```
Definition order_fam g :=
  graph g &  (forall x, inc x (domain g) -> order (V x g)).
Definition worder_fam g :=
  graph g &  (forall x, inc x (domain g) -> worder (V x g)).
Definition monotone_order_fam g :=
  (forall a b, inc a (domain g) -> inc b (domain g) ->
    sub (substrate (V a g)) (substrate (V b g)) ->
    V a g = induced_order (V b g) (substrate (V a g))).
Definition common_extension_order_axiom g :=
  order_fam g &
  (forall a b, inc a (domain g) -> inc b (domain g )-> exists c,
    inc c (domain g) & sub (substrate (V a g)) (substrate (V c g))
    & sub (substrate (V b g)) (substrate (V c g))) &
  monotone_order_fam g.
Definition common_worder_axiom g:=
   worder_fam g &
  (forall a b, inc a (domain g) -> inc b (domain g) ->
    is_segment (V a g) (substrate (V b g))
    \/ is_segment (V b g) (substrate (V a g))) &
  monotone_order_fam g.
Definition common_extension_order g h:=
  order h & substrate h = unionf (domain g) (fun a =>  (substrate (V a g))) &
  (forall a, inc a (domain g) -> V a g = induced_order h (substrate (V a g))).
```

Here is the lemma.

```
Lemma order_merge1 g:
  let G :=  (unionb g) in
  common_extension_order_axiom g -> common_extension_order g G.
Lemma order_merge2 g h1 h2:  common_extension_order_axiom g ->
  common_extension_order g h1 ->
  common_extension_order g h2 -> (h1 = h2).
```

We consider now Proposition 3 [3, p. 149]. It says *Let* $(X_\iota)_{\iota\in I}$ *be a family of well-ordered sets such that for each pair of indices* (ι,κ) *one of the sets* X_ι, X_κ *is a segment of the other. Then there exists a unique ordering on the set* $E = \bigcup_{\iota\in I} X_\iota$ *which induces the given ordering on each of the* X_ι. *Endowed with this ordering,* E *is a well-ordered set. Every segment of* X_ι *is a segment of* E*; for each* $x \in X_\iota$*, the segment with endpoint* x *in* X *is equal to the segment with endpoint* x *in* E*; and each segment of* E *is either* E *itself or a segment of one of the* X_ι. Note that if $X_\alpha = X_\beta$, then each set is a segment of the other; the orderings G_α and G_β may differ, in which case (since orders are total) there is a pair (x, y) such that $x < y$ for one order and $y < x$ for the other one, and no compatible order exists on the union. Thus an additional condition is needed (the same one as in the previous lemma). With this definition, the next lemmas become trivial.

```
Lemma order_merge3 g:
  common_worder_axiom g -> common_extension_order_axiom g.
Lemma order_merge4 g:
  common_worder_axiom g -> common_extension_order g (unionb g).
Lemma order_merge5 g h1 h2:  common_worder_axiom g ->
```

```
  common_extension_order g h1 ->
  common_extension_order g h2 -> (h1 = h2).

Theorem worder_merge g: common_worder_axiom g ->  (* 58 *)
 let G := (unionb g) in
 ( (common_extension_order g G) &
   worder G &
   (forall a x, inc a (domain g) -> is_segment (V a g) x
     -> is_segment G x) &
   (forall a x, inc a (domain g) ->  inc x (substrate (V a g)) ->
     segment (V a g) x = segment G x) &
   (forall x, is_segment G x ->
     x = substrate G \/ exists a,  inc a (domain g) & is_segment (V a g) x)).
       exists a,  inc a (domain g) & is_segment (V a g) x)).
```

3.2 The principle of transfinite induction

The next result is Lemma 2 [3, p. 151]. It says that, given a well-ordered set E and a set $\mathfrak{S}$ of segments of E, all segments are in $\mathfrak{S}$ if $\mathfrak{S}$ is stable by union and by adjunction of a greatest element (i.e., if the segment S_x belongs to $\mathfrak{S}$ then $S_x \cup \{x\}$ belongs to $\mathfrak{S}$). As a consequence, the substrate of the order is in $\mathfrak{S}$. **Note:** The assumption that elements of $\mathfrak{S}$ are segments is superfluous.

```
Section TransfinitePrinciple.
Variables r s: Set.
Hypothesis wor: worder r.
Hypothesis u_stable: forall s', sub s' s -> inc (union s') s.
Hypothesis adj_stable:
 (forall x, inc x (substrate r) -> inc (segment r x) s
    -> inc (segment_c r x) s).

Lemma transfinite_principle1 x: (* 47 *)
  is_segment r x -> inc x s.
Lemma transfinite_principle2: inc (substrate r) s.
End TransfinitePrinciple.
```

Let H(x) be the assumption: "$x \in$ E, and $p(y)$ holds for all $y \in$ E such that $y < x$". Assume that $\leq$ is a well-ordering on E and for all x, H(x) implies $p(x)$; then p is true on E, by application of the previous lemma (a direct proof is given as an exercise). This is C59 ("Principle of transfinite induction"), [3, p. 151].

```
Theorem transfinite_principle r (p:Set -> Prop):
  worder r ->
  (forall x, inc x (substrate r) ->
       (forall y, inc y (substrate r) -> glt r y x -> p y)
     -> p x)
   ->  forall x, inc x (substrate r) -> p x.
```

We now define a mapping by transfinite induction. We consider a well-ordered set E, a function g, and an element $x \in$ E. Let $g^{(x)}$ denote the restriction of g to $]\leftarrow, x[$ as a surjective function.

```
Definition restriction_to_segment r x g :=
```

```
  restriction1 g (segment r x).
Definition restriction_to_segment_axiom r x g :=
  worder r & inc x (substrate r) & is_function g & sub (segment r x) (source g).

Lemma sub_image_target g x: is_function g ->
  sub (image_by_fun g x) (target g).
Lemma rts_function r x g:  restriction_to_segment_axiom r x g  ->
  is_function (restriction_to_segment r x g).
Lemma rts_W r x g a: restriction_to_segment_axiom r x g  ->
  glt r a x -> W a  (restriction_to_segment r x g) = W a g.
Lemma rts_surjective r x g:  restriction_to_segment_axiom r x g  ->
  surjection (restriction_to_segment r x g).

Lemma rts_extensionality r s x f g:
  worder r -> inc x (substrate r) -> worder s -> inc x (substrate s) ->
  segment r x = segment s x ->
  is_function f -> sub (segment r x) (source f) ->
  is_function g -> sub (segment s x) (source g) ->
  (forall a , inc a (segment r x) -> W a f = W a g) ->
  restriction_to_segment r x f = restriction_to_segment s x g.
```

We can now state Criterion C60 (Definition of a mapping by transfinite induction) [3, p. 151]: *Let u be a letter, $\mathrm{T}\{u\}$ a term in the theory $\mathcal{T}$ (in which* E *is a set well-ordered by a relation denoted $\leq$). There exists a set* U *and a mapping f of* E *onto* U *such that for all $x \in \mathrm{E}$ we have $f(x) = \mathrm{T}\{f^{(x)}\}$. Furthermore the set* U *and the mapping f are uniquely determined by these conditions.*

We shall denote by $f_{\mathrm{S}}^{(x)}$ the restriction of f to the segment x for the ordering S (more precisely, the ordering induced on S by the given ordering). Let's denote by $\mathscr{I}(\mathrm{S},\mathrm{T},f)$ the property of the criterion, namely, that f is a surjective function defined on S such that $f(x) = \mathrm{T}\{f_{\mathrm{S}}^{(x)}\}$ for every $x \in \mathrm{S}$.

Assume that $\mathscr{I}(\mathrm{E},\mathrm{T},f)$ and $\mathscr{I}(\mathrm{E},\mathrm{T},f')$ are true. Assume $x \in \mathrm{E}$, f and f' agree on S_x. Then $f_{\mathrm{E}}^{(x)} = f_{\mathrm{E}}^{\prime(x)}$ (the functions have the same target, because they are surjective, and take the same value). By assumption, $\mathrm{T}\{f_{\mathrm{E}}^{(x)}\} = \mathrm{T}\{f_{\mathrm{E}}^{\prime(x)}\}$, so that f and f' agree on $\mathrm{S}_x \cup \{x\}$. We consider the set of all segments $\mathfrak{S}$ on which f and f' agree. This set is clearly stable by union, and we have seen that, if it contains S_x, then it contains $\mathrm{S}_x \cup \{x\}$. Hence it contains E. Since f and f' agree on E, they are equal. We can state: there is is at most one function f satisfying $\mathscr{I}(\mathrm{E},\mathrm{T},f)$.

```
Definition transfinite_def r (p: Set -> Set) f:=
  surjection f & source f = substrate r &
  forall x, inc x (substrate r) -> W x f = p (restriction_to_segment r x f).
Definition transfinite_defined r p:= choose (fun f => transfinite_def r p f).

Lemma transfinite_unique1 r p  f f' z: worder r ->
  inc z (substrate r) ->
  transfinite_def r p f -> transfinite_def r p f' ->
  (forall x : Set, inc x (segment r z) -> W x f = W x f') ->
  restriction_to_segment r z f = restriction_to_segment r z f'.
Lemma transfinite_unique r p  f f': worder r ->
  transfinite_def r p f -> transfinite_def r p f' -> f = f'.  (* 37 *)
```

Assume now that $\mathscr{I}(\mathrm{S}',\mathrm{T},f')$ and $\mathscr{I}(\mathrm{S}'',\mathrm{T},f'')$ are true, and S' and S'' are segments such that $\mathrm{S}' \subset \mathrm{S}''$. Let f be the restriction of f'' on S'. This is a surjective function. We have $f_{\mathrm{S}'}^{(x)} = f_{\mathrm{S}'}^{\prime\prime(x)}$ (because the functions are surjective and take the same values on S'). We have

$f''^{(x)}_{S'} = f''^{(x)}_{S''}$ because these are two restrictions to identical segments. We have $f(x) = f''(x)$. Assumption $\mathscr{I}(S'', T, f'')$ gives thus $f(x) = T\{\!\{f^{(x)}_{S'}\}\!\}$, in other words, $\mathscr{I}(S', T, f)$, hence $f = f'$ by uniqueness.

```
Lemma transfinite_aux1 r p s' s'' f' f'':
  worder r -> is_segment r s' -> is_segment r s'' -> sub s' s'' ->
  transfinite_def (induced_order r s')  p f' ->
  transfinite_def (induced_order r s'') p f'' ->
  f' = restriction1 f'' s'.  (* 31 *)
```

Assume we have a set of segments $\mathfrak{S}$ and, for $s \in \mathfrak{S}$, we have a function f_s such that $\mathscr{I}(s, T, f_s)$. Let t be the union of the targets of the f_s, and let g_s be the function that has the same source and graph as f_s but target t. If s and s' are two segments, we have $s \subset s'$ or $s' \subset s$; thus g_s and $g_{s'}$ agree on the intersection and we can find a common extension g. The target of this function is t, so that the function is surjective. Assume $x \in S$ and $S \in \mathfrak{S}$. We have $g(x) = f_S(x)$ because $f_S(x) = g_S(x)$. Thus the quantities $g^{(x)}_t$ and $f^{(x)}_{St}$ are well defined and equal. We have $f^{(x)}_{St} = f^{(x)}_{SS}$ because these two functions are restriction of f_S to the segment with endpoint x. We have $f_S(x) = T\{\!\{f^{(x)}_{SS}\}\!\}$ by assumption. Thus $g(x) = T\{\!\{g^{(x)}_t\}\!\}$. This means that $\mathscr{I}(t, T, g)$/

Assume now that $\mathscr{I}(S_y, T, f)$. Let's extend f as f' to $S_y \cup \{y\}$ via the formula $f'(y) = T\{\!\{f^{(y)}\}\!\}$. It follows $\mathscr{I}(S_y \cup \{y\}, T, f')$.

```
Lemma transfinite_aux2 r p s td f: worder r ->  (* 81 *)
  (forall z, inc z s -> is_segment r z) ->
  (forall z, inc z s ->  transfinite_def (induced_order r z) p (td f z)) ->
  let t := unionf s (fun z => target (td f z)) in
  let new_target := fun t f =>  corresp (source f) t (graph f) in
  let f := (common_ext s id (fun z => new_target t (td f z)) t) in
    (transfinite_def (induced_order r (union s)) p f & target f = t).
Lemma transfinite_aux3 r p x g:
  worder r ->  inc x (substrate r)
  -> transfinite_def (induced_order r (segment r x)) p g
  -> transfinite_def (induced_order r (segment_c r x)) p
    (tack_on_f g x (p (restriction_to_segment r x g))). (* 44 *)
```

Let $\mathfrak{S}$ be the the set of segments for which there exists f such that $\mathscr{I}(S, T, f)$ holds. Applying the principle of transfinite induction shows that $E \in \mathfrak{S}$. Thus, there exists f such that $\mathscr{I}(E, T, f)$. The unique f satisfying this property will be called the *mapping defined by transfinite induction on* E *via* T. We shall denote it by transfinite_defined E T.

Assume that there is a set F such that, whenever f is a function defined on a segment of E whose target is a subset of F, then $T\{\!\{f\}\!\} \in F$. We can modify $\mathscr{I}$ and add the requirement that the target of f is a subset of F. The two previous lemmas remain true, so that the target of the mapping defined by transfinite induction is a subset of F.

In the general case, we use the axiom of choice (since we do not know in which set lies f); in the particular case, we know that f belongs to the set of subfunctions $E \to F$, and we could avoid using AC.

```
Theorem transfinite_definition r p:
  worder r -> exists_unique (transfinite_def r p).

Lemma transfinite_pr r x p:
```

```
  worder r -> transfinite_def r p x ->
  transfinite_defined r p = x.
Lemma transfinite_defined_pr r p:
  worder r ->  transfinite_def r p  (transfinite_defined r p).

Theorem transfinite_definition_stable r p F:
  worder r ->
  (forall f, is_function f -> is_segment r (source f) -> sub (target f) F ->
    inc (p f) F) ->
  sub (target (transfinite_defined r p)) F. (* 53 *)
```

3.3 Zermelo's theorem

We show here that every set is the substrate of a well-ordering. This requires quite a number of small results. Consider two well-orderings, Γ and Γ' on E and E$'$; denote by S_x and S'_x the segments with endpoints x in Γ and Γ' (these are subsets of E and E$'$). Let V be the set of all $x \in E \cap E'$ such that $S_x = S'_x$ and $(S_x \times S_x) \cap \Gamma = (S'_x \times S'_x) \cap \Gamma'$; the last condition says that Γ and Γ' induce the same ordering on S_x. This set is a segment for Γ and Γ', and both orderings coincide.

```
Definition common_ordering_set r r' :=
  Zo (intersection2 (substrate r) (substrate r'))
  (fun x => segment r x = segment r' x &
    induced_order r (segment r x) = induced_order r' (segment r' x)).

Lemma Zermelo_aux0 r r':
  common_ordering_set r r' = common_ordering_set r' r.
Lemma Zermelo_aux1 r r': worder r ->  worder r' ->
   is_segment r (common_ordering_set r r'). (* 33 *)
Lemma Zermelo_aux2 r r' v: worder r ->  worder r' ->
  v = common_ordering_set r r' ->  sub(induced_order r v)(induced_order r' v).
```

Let Q(Γ) denote the following property: Γ is a well-ordering of a subset of E, and if S_x denotes the segment of Γ with endpoint x, we have $S_x \in \mathfrak{S}$ and $p(S_x) = x$.

```
Definition Zermelo_axioms E p s r:=
  worder r &
  sub (substrate r) E &
  (forall x, inc x (substrate r) -> inc (segment r x) s) &
  (forall x, inc x (substrate r) -> p (segment r x)  = x).
```

Let $q(\Gamma, \Gamma')$ be the property that, if E and E$'$ are the substrates of Γ and Γ', then E $\subset$ E$'$, E is a segment of Γ', and Γ is the ordering induced by Γ' on E. We pretend that if Q(Γ) and Q(Γ') are true, then either $q(\Gamma, \Gamma')$ or $q(\Gamma', \Gamma)$ is true. The previous lemmas show that this is true if V is either E or E$'$. Otherwise, we may consider x and x' the least element of E or E$'$ not in V, so that V = S_x and V = $S_{x'}$ (for the orderings Γ and Γ'). By application of p we get $x = x'$. This implies $x \in V$, absurd.

```
Lemma Zermelo_aux3 E s p r r':
  let q := fun r r' => sub (substrate r) (substrate r')
    & r = induced_order r' (substrate r) & is_segment r' (substrate r) in
    Zermelo_axioms E p s r -> Zermelo_axioms E p s r' ->
    q r r' \/ q r' r.  (* 50 *)
```

Let Γ be a well-ordering and x an element not in the substrate. We can extend Γ as Γ' by adjoining x as greatest element. This is a well-ordering, a segment of this order is either the substrate of Γ', the substrate of Γ, or a segment S_x of Γ.

```
Lemma Zermelo_aux4 r a: worder r ->
  let owg := order_with_greatest r a in
    ~ (inc a (substrate r)) ->
    (worder owg &  segment owg a = (substrate r) &
      forall x, inc x (substrate owg) -> x = a \/
      segment owg x = segment r x).  (* 23 * )
```

Let E be a set, $\mathfrak{S}$ a part of $\mathfrak{P}(E)$ and p a function from $\mathfrak{S}$ into E such that $p(X) \notin X$. Consider $\mathfrak{M}$, the set of orderings Γ that satisfy property Q; by virtue of Zermelo_aux3, there is a well-ordering Γ (the union of the elements of $\mathfrak{M}$) that extends the orderings of $\mathfrak{M}$. It satisfies Q. Its substrate M is not in $\mathfrak{S}$. Proof: if $M \in \mathfrak{S}$, and $a = p(M)$, we know $a \notin M$, so that we can extend Γ to Γ' by adjoining a as greatest element. This new order satisfies Q (for all y, $S_y \in \mathfrak{S}$ and $p(S_y) = y$; this is true if y is in the support of Γ, otherwise $y = a$ and $S_a = M \in \mathfrak{S}$, $p(S_a) = p(M) = a$). This means that the new order is in $\mathfrak{M}$, its support is a subset of M, and $a \in M$, absurd.

```
Lemma Zermelo_aux E s p: sub s (powerset E) ->
  (forall x, inc x s -> inc (p x) E) & ~ (inc (p x) x))->
  exists r, Zermelo_axioms E p s r & (~  (inc (substrate r) s)). (* 68 *)
```

Let now $\mathfrak{S}$ be the set of all subsets of E but E itself. If p is the representative of the complement of x in E (this exists, by the axiom of choice), the order defined by the lemma Zermelo_aux has its substrate in $\mathfrak{P}(E) - \mathfrak{S}$. Thus, it is a well-ordering on E. This is Theorem 1 [3, p. 153].

```
Theorem Zermelo E: exists r, worder r & substrate r = E. (* 17 *)
```

3.4 Inductive sets

An ordered set is said to be *inductive* if every totally ordered subset of E has an upper bound in E. More precisely, let r be an order and E its substrate, then every subset X of E, for which the order induced by r is total, has an upper bound for r. The set $\Phi(A, B)$ of partial functions is inductive, another example is given page 261.

```
Definition inductive_set r :=
  forall X, sub X (substrate r) -> total_order (induced_order r X) ->
    exists x, upper_bound r X x.

Lemma inductive_graphs a b:
  inductive_set (opposite_order (extension_order a b)). (* 35 *)
```

Consider an ordered set E; assume that each well-ordered subset of E is bounded above. Let $p(S)$ be an upper bound of S that is not in S; it exists by the axiom of choice. Let $\mathfrak{S}$ be the set of sets $S \subset E$ for which such a $p(S)$ exists. By Zermelo_aux, there is Γ that satisfies Q, i.e. Γ is a well-ordering of a subset M of E, and if S_x denotes the segment of Γ with endpoint x, we have $S_x \in \mathfrak{S}$ and $p(S_x) = x$. This last condition says that if $y < x$ for Γ, it is true for the ordering

on E; hence Γ is the restriction to M of the ordering of E. By assumption, M is bounded, say by m. This element is maximal (this is Proposition 4 [3, p. 154]).

Theorem 2 [3, p. 154] says that every inductive ordered set has a maximal element. This is a trivial consequence of the previous result.

```
Theorem Zorn_aux r: order  r ->
  (forall s, sub s (substrate r) -> worder (restriction_order r s) ->
    (bounded_above r s)) ->
  exists a, maximal_element r a.  (* 48 *)
Theorem Zorn_lemma r: order  r -> inductive_set  r ->
  exists a, maximal_element r a.
```

Corollary. If E is inductive, $a \in$ E, F is the set of all $x \geq a$, then F is inductive (if X is a totally ordered set in F, then $X \cup \{a\}$ is totally ordered; an upper bound m is in F since it satisfies $a \leq m$). Hence there is a maximal element m such that $a \leq m$. Second corollary: if $\mathfrak{F}$ is a subset of the powerset of E such that for every subset $\mathfrak{G}$ of $\mathfrak{F}$ which is totally ordered by inclusion, the union (resp. intersection) of the sets of $\mathfrak{G}$ belongs to $\mathfrak{F}$, then $\mathfrak{F}$ has a maximal or minimal element. The trick with the intersection is that the intersection is not defined if $\mathfrak{G}$ is empty. The set is nevertheless inductive, provided that $\mathfrak{F}$ is not empty (in the case of the union, it contains the empty set).

```
Lemma inductive_max_greater r a: order r ->  inductive_set r ->
  inc a (substrate r) ->
  exists m, maximal_element r m & gle r a m. (* 32 *)
Lemma inductive_powerset  A F: sub A (powerset F) ->
  (forall S, (forall x y, inc x S -> inc y S -> sub x y \/ sub y x) ->
    sub S A ->inc (union S) A) ->
  inductive_set (inclusion_suborder A).
Lemma maximal_in_powerset A F: sub A (powerset F) ->
  (forall So, (forall x y, inc x So -> inc y So -> sub x y \/ sub y x) ->
    sub So A -> inc (union So) A) ->
  exists a, maximal_element (inclusion_suborder A) a.
Lemma minimal_in_powerset A F: sub A (powerset F) -> nonempty A ->
  (forall So, (forall x y, inc x So -> inc y So -> sub x y \/ sub y x) ->
    sub So A -> inc (intersection So) A) ->
  exists a, minimal_element (inclusion_suborder A) a.
```

3.5 Isomorphisms of well-ordered sets

Assume that E and F are two well-ordered sets. We show Theorem 3 [3, p. 155]: Let $I(u, v, f)$ be the property that f is an order isomorphism from u onto a segment w of v. We claim that there exists a unique f such that I(E, F, f), or there exists a unique f such that I(F, E, f). Note: The two cases are not excluded; in that case, f is bijection between E and F.

In order to show uniqueness we start with a lemma: if f is increasing and g is strictly increasing, if the image of f is a segment of F, then $f(x) \leq g(x)$ for all x. The proof is by contradiction. If a is the least element such that $g(a) < f(a)$, since the image of f is a segment there is a z such that $g(a) = f(z)$. Since f is increasing this gives $z < a$, hence $f(z) \leq g(z) < g(a)$, absurd.

```
Lemma increasing_function_segments r r' f g:
  worder r -> worder r' ->
```

```
  increasing_fun f r r' -> strict_increasing_fun g r r'->
  is_segment r' (range (graph f)) ->
  forall x, inc x (source f) -> gle r' (W x f) (W x g).  (* 31 *)
Lemma isomorphism_worder_unique r r' x y:
  worder r -> worder r' ->  is_segment r' (range (graph x)) ->
  is_segment r' (range (graph y)) ->
  order_morphism x r r' -> order_morphism y r r'
  ->  x = y.
```

Given a totally ordered subset X of $\mathfrak{F}$, we can apply lemma sup_extension_order2, that says that there exists a function f that extends all elements in X; we know that the source and range of f are the union of the sources and ranges of the elements of X, hence are segments. Given a and b in the source of f, there is a function g that is defined for both a and b (because X is totally ordered); since $a \leq b$ is equivalent to $g(a) \leq g(b)$ and $f(a) = g(a)$ and $f(b) = g(b)$ we deduce that $a \leq b$ is equivalent to $f(a) \leq f(b)$. As a consequence, f is increasing and hence is a morphism. Consider now a maximal element f. If the source of f is E, then I(E, F, f) is true. If the range of f is F, then f^{-1} is a bijection from F onto a subset of E, hence I(F, E, f^{-1}). Otherwise, if a is the least element of E not in the source of f and b the least element not in the range of b, we can extend f to a function g by saying $g(a) = b$. This function is in $\mathfrak{F}$. This contradicts the maximality of f.

```
Theorem isomorphism_worder r r': (* 158 *)
 worder r -> worder r' ->
  let iso:= (fun u v f =>
    is_segment v (range (graph f)) & order_morphism f u v) in
  exists_unique (fun f => iso r r' f) \/ exists_unique (fun f => iso r' r f).
```

Corollary 1. The only isomorphism from a well-ordered set into a segment of itself is the identity.

```
Lemma unique_isomorphism_onto_segment r f: worder r ->
  is_segment r (range (graph f)) ->  order_morphism f r r ->
  f = identity (substrate r).
```

Corollary 2. If E and F are two well-ordered sets, f an isomorphism of E onto a segment of F, g an isomorphism of F onto a segment of E, then f and g are inverse bijections.

```
Lemma bij_pair_isomorphism_onto_segment r r' f f':
  worder r -> worder r' ->
  is_segment r' (range (graph f)) ->  order_morphism f r r' ->
  is_segment r (range (graph f')) ->  order_morphism f' r' r ->
  (order_isomorphism f r r' & order_isomorphism f' r' r &
    f = inverse_fun f'). (* 52 *)
```

Finally, we show that every subset of a well-ordered set is isomorphic to a segment of E.

```
Lemma isomorphic_subset_segment r a:
  worder r -> sub a (substrate r) ->
  exists w, exists f, is_segment r w &
    order_isomorphism f (induced_order r a) (induced_order r w). (* 46 *)
```

3.6 Lexicographic products

Consider the follow definition. We assume that $(X_\iota)_{\iota\in I}$ is a family of sets, with domain I, g is a family of orderings (for each ι, g_ι is an ordering on f_ι) and r is a well-ordering on I. Denote the order relation associated to r by $\leq$, and the order relation associated to g_ι by $\leq_\iota$. The *lexicographic product* is the order associated to the relation: x and y are elements of the product $\prod X_\iota$, and either $x = y$, or, if ι is the least index (for the relation $\leq$) such that $x_\iota \neq y_\iota$ then $x_\iota \leq_\iota y_\iota$.

```
Definition order_prod_r r g :=
  fun x x' =>
    forall j, least_element (induced_order r (Zo (domain g)
      (fun i =>  V i x <> V i x'))) j -> glt (V j g) (V j x)(V j x').

Definition orprod_ax r g:=
  worder r & substrate r = domain g & order_fam g.
Definition order_prod r g :=
  graph_on (order_prod_r r g)(prod_of_substrates g).
```

It is obvious that the lexicographic product is well-defined; it is a bit more longish to prove that it is an order on the product.

```
Lemma orprod_substrate_aux r g x:
  orprod_ax r g ->  order_prod_r r g x x.
Lemma orprod_sr r g:
  orprod_ax r g ->
  substrate(order_prod r g) = prod_of_substrates g.
Lemma orprod_or r g:
   orprod_ax r g -> order (order_prod r g). (* 86 *)
Lemma orprod_gle1 r g x x':
  orprod_ax r g ->
  (related (order_prod r g) x x' <->
  (inc x (prod_of_substrates g)&  inc x' (prod_of_substrates g) &
    forall j, least_element (induced_order r (Zo (domain g)
      (fun i =>  V i x <> V i x'))) j -> glt (V j g) (V j x)(V j x'))).
Lemma orprod_gle r g x x' :  orprod_ax r g ->
  (gle (order_prod r g) x x' <->
  (inc x (prod_of_substrates g) & inc x' (prod_of_substrates g) &
    (x= x' \/ exists j, inc j (substrate r)  &
      glt (V j g) (V j x) (V j x') &
      forall i, glt r i j -> V i x = V i x'))).
```

If all orders are total so is the lexicographic product.

```
Lemma orprod_total r g:
  orprod_ax r g ->
  (forall i, inc i (domain g) -> total_order (V i g)) ->
  total_order (order_prod r g).  (* 23 *)
```

Chapter 4

Equipotent Sets. Cardinals

Bourbaki denotes by Eq(X, Y) the property that there is a bijection between X and Y and denotes by Card(X) the set $\tau_Z(\mathrm{Eq}(X,Z))$. He calls this *the cardinal of* X. He does not define "a cardinal". The only possible interpretation of "$\mathfrak{r}$ is a cardinal" is "the object $\mathfrak{r}$ is of the form Card(E) for some set E". Using a specific font for cardinals suggests that a cardinal is some special object, and that we should perhaps introduce a type for these cardinals. If $A = \{\emptyset\}$ and $\mathfrak{a}$ denotes the cardinal of A, it is impossible to prove $\mathfrak{a} = A$ or $\mathfrak{a} \neq A$ (non-definiteness of τ).

The cardinal of a set is sometimes called the *the power of* X. It is interesting to notice that no name is given to the notation $\mathfrak{a}^{\mathfrak{b}}$ when this means the cardinal of the set of mappings from one set into another (the operation is nevertheless called "exponentiation of cardinals"). The term "power" is used only in the phrase "power of the continuum", where it means the cardinal of the set of real numbers (to be defined elsewhere), or, equivalently, the cardinal of $\mathfrak{P}(\mathbf{N})$ (where $\mathbf{N}$ is the set of natural integers, defined in Chapter 6). For us, the term "power" will only be used to denote $\mathfrak{a}^{\mathfrak{b}}$.

One can define addition and multiplication of cardinals. For instance, $A \times B$ is equipotent to $A' \times B'$ when A is equipotent to A' and B is equipotent to B'. Thus $\mathrm{Card}(A \times B) = \mathrm{Card}(A' \times B')$ if $\mathrm{Card}(A) = \mathrm{Card}(A')$ and $\mathrm{Card}(B) = \mathrm{Card}(B')$. For instance, if $\mathfrak{a}$ is as above then $\mathfrak{a}.\mathfrak{a} = \mathfrak{a}$. In order to prove properties of these operations (like associativity, commutativity), one needs the notion of a family of cardinals, which is some f that associates to each element i of a given set I a cardinal f_i. Technically, f is a functional graph, and its range is a set. This means that we can consider a set of cardinals, so that a cardinal is a set. One could define the cardinal product $\mathfrak{a}.\mathfrak{b}$ as the cardinal of $A \times B$, whenever $\mathrm{Card}(A) = \mathfrak{a}$ and $\mathrm{Card}(B) = \mathfrak{b}$. Since $\mathfrak{a}$ and $\mathfrak{b}$ are sets that satisfy these conditions, it is simpler to define it as the cardinal of $\mathfrak{a} \times \mathfrak{b}$. This definition then makes sense for any two sets; moreover A.B = B.A, even when A and B are not cardinals.

In the Exercises, Bourbaki denotes by $\mathrm{Is}(\Gamma, \Gamma')$ the property that Γ and Γ' are ordered sets, and there is an order isomorphism between Γ and Γ', he denotes by $\mathrm{Ord}(\Gamma)$ the ordered set $\tau_\Delta(\mathrm{Is}(\Gamma, \Delta))$, and calls it the *order-type* of Γ. He defines an *ordinal* as the order type of a well-ordered set (the cardinal of any set is a cardinal, the order-type of an ordered set is not always an ordinal). Ordinals are generally denoted by lower-case Greek letters such as ω. The ordinal sum and lexicographic product of orderings induce two operations (sum and product) on the family of order-types, denoted by $\lambda + \mu$ and $\lambda\mu$. These operations are non-commutative: for instance $\lambda + 1$ and $1 + \lambda$ correspond to the orderings obtained by adjoining a greatest and a least element, respectively. The relation "there is an order isomorphism between Γ and a sub-ordering of Γ'", denoted by $\Gamma \prec \Gamma'$ is a preorder. The sum and product of ordinals are ordinals, and the relation $\lambda \prec \mu$ is a well-ordering, compatible with the two operations. Let

A = {∅} be as above; there is a unique ordering on this set, which is a well-ordering. Let λ be its ordinal. The support of λ is a singleton, but it is undecidable whether or not the support is A.

In 1923 von Neumann noted that the axiom of choice (i.e. the use of τ) is not needed for defining ordinals. To each well-ordered set (E, ≤) is uniquely associated a numeration, and hence a set F, with a natural ordering o(F), and (F, o(F)) is isomorphic to (E, ≤). The von Neumann ordinal of the ordered set (E, ≤) is the set F. It is equipotent to E. Zermelo's theorem asserts that any set E has a well-ordering, so that one can consider the least ordinal equipotent to E. This is called the von Neumann cardinal of E. Let A = {∅} be as above; the von Neumann ordinal of the unique ordering of A is A itself, and the von Neumann cardinal of A is A.

In this chapter, we shall use the von Neumann point of view. We shall introduce the notion of finite set, and see that any finite ordinal is a cardinal. In a future chapter, we shall see that if X is an infinite set, then X and X × X are equipotent. This can be restated as: if A is an infinite cardinal, then A = A.A. This result is equivalent to the axiom of choice: if there is no choice function on X, then X has no cardinal, according to von Neumann.

The first section of this chapter studies some properties of ordinals. Sections 2 to 7 correspond to §3.1 to §3.6 of Chapter III.

4.1 Ordinals

4.1.1 Auxiliary results

We start this section with some properties of bijections, order isomorphisms, and order morphisms. We say that two ordered sets are order-isomorphic if there exists an order-isomorphism. This is an equivalence relation.

```
Lemma orderIR r: order r -> r \Is r.
Lemma orderIS r r': r \Is r' -> r' \Is  r.
Lemma orderIT r r' r'': r \Is r' -> r' \Is r'' -> r \Is r''.
```

Let f be an order morphism (or isomorphism) E → F. By definition, $x \le y$ if and only if $f(x) \le f(y)$ for x and y in E. We can replace ≤ by <. We give a variant where E is replaced by the substrate of <. Finally, if $x < y$, then x and y are in E, and this implies $f(x) < f(y)$.

```
Lemma order_isomorphism_morphism r r' f:
   order_isomorphism f r r' -> order_morphism f r r'.
Lemma order_morphism_pr1 r r' f a b:
  order_morphism f r r' -> inc a (source f) -> inc b (source f) ->
  (glt r a b <-> glt r' (W a f) (W b f)).
Lemma order_morphism_pr0 r r' f a b:
  order_morphism f r r' -> inc a (substrate r) -> inc b (substrate r) ->
  (glt r a b <-> glt r' (W a f) (W b f)).
Lemma order_morphism_pr2 r r' f a b:
  order_morphism f r r' -> glt r a b ->
  glt r' (W a f) (W b f).

Lemma order_isomorphism_pr1 r r' f a b:
  order_isomorphism f r r' -> inc a (source f) -> inc b (source f) ->
  (glt r a b <-> glt r' (W a f) (W b f)).
```

```
Lemma order_isomorphism_pr0 r r' f a b:
  order_isomorphism f r r' -> inc a (substrate r) -> inc b (substrate r) ->
  (glt r a b <-> glt r' (W a f) (W b f)).
Lemma order_isomorphism_pr2 r r' f a b:
  order_isomorphism f r r' -> glt r a b ->
  glt r' (W a f) (W b f).
```

If f is injective (resp. bijective) and satisfies $f(x) \leq f(y)$ then f is an order morphism (resp. isomorphism) if the source is totally ordered. We refine the theorem about well-ordered sets isomorphisms: given two well-ordered sets E and E′, either E is isomorphic to E′, or E isomorphic to a strict segment of E′ or E′ isomorphic to a strict segment of E. Two isomorphic segments of E are equal.

```
Lemma total_order_morphism f r r':
  total_order r -> order r' ->
  injection f ->  substrate r = source f -> substrate r' = target f ->
  (forall x y, inc x (source f) -> inc y (source f) ->
    gle r x y -> gle r' (W x f) (W y f)) ->
  order_morphism f r r'.
Lemma total_order_isomorphism f r r':
  total_order r -> order r' ->
  bijection f ->  substrate r = source f -> substrate r' = target f ->
  (forall x y, inc x (source f) -> inc y (source f) ->
    gle r x y -> gle r' (W x f) (W y f)) ->
  order_isomorphism f r r'.

Lemma segments_iso1 a A B: worder a -> sub A B ->
  is_segment a A -> is_segment a B ->
  (induced_order a B) \Is (induced_order a A) -> A = B. (* 31 *)
Lemma segments_iso2 a A B: worder a ->
   inc A (set_of_segments a) -> inc B (set_of_segments a)  ->
   (induced_order a A)  \Is (induced_order a B) -> A = B.
Lemma isomorphism_worder2 r r': (* 38 *)
   worder r -> worder r' ->
  r \Is r'
  \/ (exists x, inc x (substrate r) &
    (induced_order r (segment r x)) \Is r')
  \/ (exists x', inc x' (substrate r') &
   (induced_order r' (segment r' x')) \Is r).
```

Let's write $A \prec B$ if there is an order morphism $f : A \to B$. The first lemma says that f is an order isomorphism onto its target. Conversely, if we have an order isomorphism $A \to C$, where C is the ordering induced by B on some subset X of its domain, it can be extended to an order morphism $A \to B$ with target X.

```
Lemma order_le_alt r r':
   order r -> order r' ->
   (exists f, order_morphism f r r')
  ->  (exists f, exists x,  sub x (substrate r') &
       order_isomorphism f r (induced_order r' x)).
Lemma order_le_alt2 f a b x: order a -> order b ->
  sub x (substrate b) ->
  let F := (BL (fun z => W z f) (substrate a) (substrate b)) in
    order_isomorphism f a (induced_order b x)
    -> (order_morphism F a b  & range (graph F) = x).
```

We show here that if X ∪ {a} is equipotent to Y ∪ {b}, then X and Y are equipotent, and conversely, provided $a \notin$ X and $b \notin$ Y. This will be interpreted later on as: the successor function is injective. Moreover, if X ∪ {a} is equipotent to X, and X is a subset of Y, then Y ∪ {b} is equipotent to Y. This will be interpreted later on as: a subset of a finite set is finite.

```
Lemma tack_on_injective_card1 x y a b:
  ~ (inc a x)  -> ~ (inc b y) ->
   (tack_on x a) \Eq (tack_on y b) -> x \Eq y.   (* 35 *)
Lemma tack_on_injective_card2 x y a b:
  ~ (inc a x)  -> ~ (inc b y) ->
   ((tack_on x a) \Eq (tack_on y b) <-> x \Eq y).
Lemma tack_on_injective_card3 x y a b:
  ~ (inc a x)  -> ~ (inc b y) ->
  sub x y -> x \Eq (tack_on x a) ->
  y \Eq (tack_on y b).   (* 42 *)
```

Let's show the Cantor Bernstein theorem: if F is a subset of E, and f an injection E → F, then E is equipotent to F. (Let D be the smallest set invariant by f that contains E − F. The bijection is the function that is f on D, the identity elsewhere). It follows that if there is an injection f : X → Y and an injection g : Y → X, then there is a bijection X → Y (let Z be the image of g, it is equipotent to Y since g is injective; it is equipotent to X, being a subset of X, and $g \circ f$ is an injection X → Z).

```
Lemma equipotent_restriction1  f x:
  sub x (source f) -> injection f ->
  x \Eq (image_by_fun f x).
Lemma equipotent_range f: injection f ->
  (source f) \Eq (range (graph f)).

Lemma Cantor_Bernstein1 E F f:
  sub F E ->
  (forall x, inc x E -> inc (f x) F) ->
  (forall x y, inc x E -> inc y E -> f x = f y  -> x = y) ->
  E \Eq F.    (* 46 *)
Theorem CantorBernstein X Y:
  (exists f, injection f & source f = X & target f = Y) ->
  (exists f, injection f & source f = Y & target f = X) ->
  (exists f, bijection f & source f = X & target f = Y).  (* 21 *)
```

4.1.2 Definition of ordinals

A set E is said to be *transitive* if $a \in b$ and $b \in$ E implies $a \in$ E, it is *irreflexive* if E $\notin$ E, it is *decent* if all elements of E are irreflexive, it is *asymmetric* if one of $x \in y$ and $y \in x$ is false.

We say that < is *irreflexive* if $x < x$ is false. We say that < is *asymmetric* if at least one of $x < y$, $y < x$ is false. An asymmetric relation is obviously antisymmetric and irreflexive. A strict well-ordering is an asymmetric relation $a < b$ such that "$a < b$ or $a = b$" is a well-ordering. If E is any set, we denote by o(E) the relation "$x \in$ E and $y \in$ E and $x \subset y$". We know that this is an ordering. We denote by o'(E) the relation $x \in$ E and $y \in$ E and $x \in y$ or $x = y$. This is an ordering if $\in$ is transitive and antisymmetric (in particular if it is asymmetric, i.e., if E is asymmetric). The set $x \cup \{x\}$, denoted by x^+, will be called the ordinal successor of x. One of the previous theorems can then be restated as: if x and y are irreflexive, then x and y are equipotent if and only if x^+ and y^+ are equipotent.

```
Definition transitive_set X:= forall x, inc x X -> sub x X.
Definition decent_set x := forall y, inc y x -> ~ (inc y y).
Definition trans_dec_set X := transitive_set X & decent_set X.
Definition asymmetric_set E :=
  forall  x y, inc x E -> inc y E -> inc x y -> inc y x -> False.
Definition ordinal_oa E :=  graph_on (fun a b => inc a b \/ a = b) E.
Definition ordinal_o x := inclusion_suborder x.
Definition succ_o x := tack_on x x.
```

There are different ways of defining an ordinal E.

- The definition of Krivine [9] is very basic. It is: E is transitive, $\in$ is transitive, E is asymmetric, and $\in$ satisfies the properties of a well-ordering.
- An equivalent form of above: E is asymmetric and transitive, and $o'(\mathrm{E})$ is a well-ordering.
- If E is an ordinal, then $o(\mathrm{E})$ and $o'(\mathrm{E})$ are the same orderings.
- The definition of von Neumann (see [12]) is: $o(\mathrm{E})$ is a well-ordering, and $\mathrm{S}_x = x$ for $x \in \mathrm{E}$, where S_x is the segment of x. An easy consequence is that $o(\mathrm{E}) = o'(\mathrm{E})$.
- The Bourbaki definition (that we shall adopt) is: any transitive subset of E is either E or an element of E.

We prove here that if x_i is a family of transitive and decent sets, so is the union and intersection of the family, as well as the successor of each x_i.

```
Lemma transitive_union x: (forall y, inc y x -> transitive_set y)
  -> transitive_set (union x).
Lemma trans_dec_union x: (forall y, inc y x -> trans_dec_set y)
  -> trans_dec_set (union x).
Lemma trans_dec_intersection x:
  (forall y, inc y x -> trans_dec_set y)
  -> nonempty x -> trans_dec_set (intersection x).
Lemma trans_dec_succ y:
  trans_dec_set y -> trans_dec_set (succ_o y).
```

Let $p(\mathrm{X})$ be the property that X is transitive and decent; let E be an ordinal, and Y be the set of all subsets of E that satisfy p. Let $t = \bigcup \mathrm{Y}$. We have shown that t and t^+ satisfy p. Since t is a transitive subset of E, we have $t \in \mathrm{E}$ or $t = \mathrm{E}$. The first case is excluded, for otherwise we would have $t^+ \subset \mathrm{E}$ and $t^+ \in \mathrm{Y}$; since $t \in t^+$, we get $t \in \bigcup \mathrm{Y}$, which is $t \in t$, contradicting the fact that t^+ is decent. Thus $t = \mathrm{E}$, so that any ordinal is transitive and decent.

```
Definition is_ordinal X:=
  forall Y, sub Y X -> transitive_set Y -> Y <> X -> inc Y X.
Lemma OS_succ x: is_ordinal x ->  is_ordinal (succ_o x).
Lemma ordinal_trans_dec x:  is_ordinal x -> trans_dec_set x.
Lemma ordinal_transitive x: is_ordinal x -> transitive_set x.
Lemma ordinal_decent x:  is_ordinal x -> decent_set x.
Lemma ordinal_irreflexive x: is_ordinal x -> ~ (inc x x).
```

As a consequence, if y is an ordinal, then $x \in y$ implies that x is a strict subset of y. The converse holds if x is transitive (in particular if x is an ordinal). Assume $y = z^+$. We deduce: if z is an ordinal $x \in z^+ \implies x \subset z$, and if x is an ordinal $x \in z^+ \iff x \subset z$.

```
Lemma ordinal_sub x y:
  is_ordinal x -> is_ordinal y -> sub x y ->
  x = y \/ inc x y.
Lemma ordinal_sub2 x y: is_ordinal y ->
  inc x y -> strict_sub x y.
Lemma ordinal_sub3 x y: is_ordinal y ->
  inc x (succ_o y) -> sub x y.
Lemma ordinal_sub4 x y: is_ordinal x ->  is_ordinal y ->
  (sub x y <-> inc x (succ_o y)).
```

Consider a non-empty set X of ordinals, and its intersection Y. This is a transitive and decent set. The relation $Y \notin Y$ says that for some $A \in X$ we have $Y \notin A$. Since Y is a transitive subset of A we get Y = A. We restate this as $Y \in X$. In the case where X has two elements, x and y, this says that $x \cap y$ is either x or y, or equivalently that $x \subset y$ or $y \subset x$. It follows one of $x \in y$, $y \in x$, $x = y$.

```
Definition ordinal_set E :=  (forall i, inc i E -> is_ordinal i).
Lemma ordinal_intersection x: nonempty x -> ordinal_set x ->
  inc (intersection x) x.
Lemma ordinal_trichotomy x y:
  is_ordinal x -> is_ordinal y ->
  (inc x y \/ inc y x \/ x = y).
```

A transitive set X whose elements are ordinals is an ordinal (hint: let Y be a transitive strict subset of X, a an element of X not in Y, and $t \in Y$. Since a and t are in X, hence ordinals, we have $t = a$, $a \in t$, or $t \in a$. We deduce $t \in a$, thus $Y \subset a$, etc). The elements of an ordinal are ordinals (hint: Let Z the union of all subsets Y of X such that Y is transitive and contains only ordinals. This set is transitive and contains only ordinals, thus is an ordinal, and we have one of Z = X, $Z \in X$ or $X \in Z$, etc). We deduce that if x^+ is an ordinal, then x is an ordinal.

```
Lemma ordinal_pr x: transitive_set x ->  ordinal_set x ->
  is_ordinal x.
Lemma elt_of_ordinal x y: is_ordinal x -> inc y x ->
  is_ordinal y. (* 23 *)
Lemma OS_succr x: is_ordinal (succ_o x) -> is_ordinal x.
```

Consequences. There is not set containing all ordinals, for if X is the set of all ordinals, it is transitive, satisfies the previous criterion, hence is an ordinal, but $X \notin X$ is true. An ordinal is asymmetric. The two orderings $o(x)$ and $o'(x)$ are the same. Any subset of an ordinal set has a least element (for inclusion) which is the intersection. Thus $o(X)$ is a well-ordering. Any property p satisfied by some ordinal (say x) is satisfied by a least ordinal (the intersection of the set all y satisfying p that are in x^+).

```
Definition least_ordinal (p: Set -> Prop) x:= intersection (Zo (succ_o x) p).

Lemma non_collectivizing_ordinal:
  ~(exists x, forall a, is_ordinal a -> inc a x).
Lemma ordinal_asymmetric E: is_ordinal E -> asymmetric_set E.
Lemma ordinal_o_lt a b c: is_ordinal c ->
   inc a c -> inc b c ->
   (glt (ordinal_o c) a b <-> inc a b).
Lemma ordinal_same_wo x: is_ordinal x ->
  ordinal_oa x = ordinal_o x.
```

```
Lemma ordinal_worder1 x: (p: Set -> Prop),
  is_ordinal x -> (p x) ->
  let y:= least_ordinal p x in
    (is_ordinal y & p y & forall z, is_ordinal z -> p z -> sub y z).
Lemma ordinal_worder2 x: is_ordinal x ->
  worder (ordinal_o x).
Lemma ordinal_worder x: is_ordinal x ->
  worder (ordinal_oa x).
```

Let E be a transitive and decent set. Let S_x be the segment for $o'(E)$ with endpoint x. This is the set of all $y \in E$ such that $y \in x$ and $x \neq y$. Since E is decent, the condition $x \neq y$ is unnecessary. Thus $S_x = x \cap E$. Since E is transitive, $x \in E$ implies $x \subset E$ and $x \cap E = x$. Thus $S_x = x$. This relation is true when E is an ordinal, and with $o(E)$ instead of $o'(E)$.

Assume moreover E well-ordered by $o'(E)$ and F transitive. Then $E \cap F$ is a segment of $o'(E)$, thus is either E or an initial segment S_x, with $x \in E$. But $S_x = x$, thus either $E \subset F$ or $E \cap F \in E$. Assume $F \subset E$. We get $F = E$ or $F \in E$. Thus E is an ordinal. We restate this as: an ordinal is a set E which is transitive, asymmetric, and well-ordered by $o'(E)$.

```
Lemma ordinal_segment1 E x: trans_dec_set E ->
  inc x E -> segment (ordinal_oa E) x = x.
Lemma ordinal_segment E x: is_ordinal E ->
  inc x E -> segment (ordinal_o E) x = x.
Lemma ordinal_pr1 E:
  is_ordinal E <-> (transitive_set E & worder (ordinal_oa E) & asymmetric_set E).
```

4.1.3 Comparing ordinals

Any well-ordered set is order-isomorphic to unique ordinal. More precisely, given a well-ordering r, there is a unique set E, denoted ord(r) such that $o(E) = r$.

Existence. Consider a set X well-ordered by $\leq$. Let f be the function defined by transfinite induction via the property p (where $p(x)$ is the target of x). Then f is the unique surjective function defined on X such that $f(x) = p(f_x)$, where f_x is the surjective function, defined on the segment S_x, that agrees with f. Thus $f(x)$ is the set of all $f(y)$ for $y < x$. In particular $y < x$ implies $f(y) \in f(x)$. Obviously, $y \leq x$ implies $f(y) \subset f(x)$. If $x \in f(a)$, then $x = f(b)$ for some $b < a$, thus $f(b) \subset f(a)$ and $f(a)$ is transitive. It is an ordinal (consider the least a such that $f(a)$ is not an ordinal and apply ordinal_pr). Since f is surjective, its target is transitive, the same argument says that the target is an ordinal. If $f(a) \in f(a)$, then $f(a) = f(b)$ for some $b < a$, and $f(b) \in f(b)$; thus there no least such a, and since X is well-ordered, there is no such a. Thus, the image of f is decent; but since the source is totally ordered, it says that f is injective; in particular f is an order isomorphism.

Uniqueness. Consider two ordinals X and Y and an order-isomorphism $f : o(X) \to o(Y)$. Then f is the identity function. For otherwise, there would be a least element $x \in X$ such that $f(x) \neq x$. Then $f(y) = y$ for $y \in x$. This implies that x is a subset of Y. It is transitive and cannot be Y (by injectivity of f), thus is an element z of y. Since f is increasing we have $f(z) \subset z$; thus, if $x \in z$ we have $f(x) \in z$, and otherwise $z \in f(x)$. Thus z is not in the image of f, contradicting surjectivity.

```
Lemma ordinal_isomorphism_unique  x y f:
  is_ordinal x -> is_ordinal y ->
  order_isomorphism f (ordinal_o x)(ordinal_o y) ->
```

```
  (x = y & f = identity x).  (* 63 *)
Lemma ordinal_isomorphism_exists r: worder r ->
  let f := transfinite_defined r target in
  is_ordinal (target f) &
  order_isomorphism f r (inclusion_suborder (target f)). (* 60 *)
```

We shall rewrite the existence theorem as: if E is a well-ordering then ord(E) is an ordinal, and E is isomorphic to o(ord(E)). Note that for Bourbaki, the ordinal of E, denoted by Ord(E), is a quantity that has the same properties as o(ord(E)). If x is an ordinal, then ord($o(x)$) = x. The uniqueness theorem says: if x and y are two ordinals, if $o(x)$ and $o(y)$ are isomorphic (or isomorphic to a common ordering z), then $x = y$.

```
Definition ordinal r := target (transfinite_defined r target).

Lemma ordinal_p1 r: worder r -> is_ordinal (ordinal r).
Lemma ordinal_p2 r: worder r ->
  r \Is  (ordinal_o (ordinal r)).
Lemma ordinal_p3 E: is_ordinal E->
  ordinal (ordinal_o E) = E.
Lemma ordinal_p4 x y:  is_ordinal x -> is_ordinal y ->
  (ordinal_o x) \Is (ordinal_o y) -> x = y.
Lemma ordinal_p5  x y z: is_ordinal x -> is_ordinal y ->
  (ordinal_o x) \Is z -> (ordinal_o y) \Is  z ->
  x = y.
Lemma ordinal_p7 E: is_ordinal E -> substrate (ordinal_o E) = E.
```

Let $r \leq_{\mathrm{ord}} r'$ be the relation "r and r' are orderings and there is an order isomorphism from r onto a subset of r'". This is the same as: there is an order morphism $r \to r'$. Let $x \leq_{\mathrm{ord}} y$ be the relation "x and y are ordinals, and $o(x) \leq_{\mathrm{ord}} o(y)$". The relation $r \leq_{\mathrm{ord}} r'$ is compatible with order-isomorphism (one can replace each argument by an isomorphic one). In particular, if r and r' are well-orders, $r \leq_{\mathrm{ord}} r'$ is equivalent to $\mathrm{ord}(r) \leq_{\mathrm{ord}} \mathrm{ord}(r')$. Since any subset of a well-ordered set is isomorphic to a segment we get: $x \leq_{\mathrm{ord}} y$ if and only if x and y are ordinals, and there is an order isomorphism f with source x, whose target is a segment S of $o(y)$. Using von Neumann ordinals (rather than the axiom of choice) has the following advantage: S is an ordinal, and f is the identity, hence $x \subset y$. As a consequence, $\leq_{\mathrm{ord}}$ is a well-ordering (given a family of ordinals, the intersection is the least element).

```
Definition order_le r r' :=
  order r & order r' &
  exists f, exists x,
    sub x (substrate r') & order_isomorphism f r (induced_order r' x).
Definition ordinal_leD1 r r' :=
  is_ordinal r & is_ordinal r' & order_le (ordinal_o r)(ordinal_o r').

Lemma order_le_compatible r r' r1 r1':
  r \Is r1 -> r' \Is r1' ->
  (order_le r r' <-> order_le r1 r1').  (* 52 *)
Lemma order_le_compatible1 r r':
  worder r -> worder r' ->
  (order_le r r' <-> ordinal_leD1 (ordinal r) (ordinal r')).

Lemma ordinal_le_pr x x':
  ordinal_leD1 x x' <-> (
    is_ordinal x & is_ordinal x' &
```

```
     exists f, exists S,
      is_segment (ordinal_o x') S &
      order_isomorphism f (ordinal_o x) (induced_order (ordinal_o x') S)).

Lemma ordinal_le_pr1 x x':
   ordinal_leD1 x x' <-> (
    is_ordinal x & is_ordinal x' &
     exists f, is_segment (ordinal_o x') (range (graph f)) &
         order_morphism f (ordinal_o x)(ordinal_o x')).
Lemma ordinal_le_pr0 x y:
  ordinal_leD1 x y <-> (is_ordinal x & is_ordinal y & sub x y). (* 27 *)
Lemma ordinal_le_pr2 x y: is_ordinal x -> is_ordinal y -> sub x y ->
   induced_order (ordinal_o y) x = (ordinal_o x).
Lemma ordinal_le_pr3 x y: is_ordinal y -> inc x y ->
   induced_order (ordinal_o y) x = (ordinal_o x).
```

Consider two well-orderings r and r'. Then "$r \preceq_{\mathrm{ord}} r'$ and r and r' are non-isomorphic" if and only if there exists an order morphism $r \to r'$ whose range is a segment S_z. If r and r' are ordinals, they are are non-isomorphic whenever they are different, and the condition can be rewritten as $r <_{\mathrm{ord}} r'$.

```
Lemma ordinal_lt_pr1 x x':
    (ordinal_leD1 x x' & x <> x') <-> (
    is_ordinal x & is_ordinal x' &
     exists f,  exists y,
        inc y x' &
        (range (graph f)) = segment  (ordinal_o x') y
        & order_morphism f (ordinal_o x) (ordinal_o x')).  (* 25 *)

Lemma ordinal_lt_pr2 a b: (* 26 *)
  worder b ->  (ordinal_leD1 a (ordinal b) & a <> (ordinal b)) ->
     exists f,  exists x,
        inc x (substrate b) &
        (range (graph f)) = segment b x & order_morphism f (ordinal_o a) b.
```

We know that $x \leq_{\mathrm{ord}} y$ is equivalent to "x is an ordinal and y is an ordinal and $x \subset y$". We use this as the definition. We deduce: if y is an ordinal, then $x <_{\mathrm{ord}} y$ is equivalent to $x \in y$ and $x \leq_{\mathrm{ord}} y$ is equivalent to $x \in y^+$.

```
Definition ordinal_le x y :=
   is_ordinal x & is_ordinal y & sub x y.
Definition ordinal_lt x y := ordinal_le x y & x <> y.

Notation "x <=o y" := (ordinal_le x y) (at level 60).
Notation "x <o y" := (ordinal_lt x y) (at level 60).

Lemma ordinal_lt_pr0 x y:
  x <o y <-> (is_ordinal x & is_ordinal y & inc x y).
Lemma set_ord_lt_rw a x: is_ordinal a ->  (x <o a <-> inc x a).
Lemma set_ord_le_rw a x: is_ordinal a ->
  (x <=o a <-> inc x (succ_o a)).
Lemma ordinal_le_pr0 x y:
  (x <=o y <-> (is_ordinal x & is_ordinal y & sub x y)).
Theorem wordering_ordinal_le: worder_r ordinal_le.
```

Let's state some properties of $\leq_{\mathrm{ord}}$.

```
Lemma ord_leR x: is_ordinal x ->
  x <=o x.
Lemma ord_leT x y z:
  x <=o y -> y <=o z -> x <=o z.
Lemma orl_leA x y:
  x <=o y -> y <=o x ->  x = y.
Lemma ord_leA1 a b:
  a <=o b -> b <o a -> False.
Lemma ord_lt_leT a b c:
  a <o b -> b <=o c -> a <o c.
Lemma ord_le_ltT a b c:
  a <=o b -> b <o c -> a <o c.
Lemma ord_lt_ltT a b c:
  a <= b -> b <o c -> a <o c.
Lemma ord_le_to_el a b:
  is_ordinal a -> is_ordinal b ->  a <=o b \/ b <oa.
Lemma ord_le_to_ell a b:
  is_ordinal a -> is_ordinal b -> (a = b \/ a <o b \/ b <o a).
Lemma ord_le_to_ee a b:
  is_ordinal a -> is_ordinal b -> a <=o b \/ b <=o a.
Lemma ord_le_to_si a b:
  is_ordinal a -> is_ordinal b -> (sub a b \/ inc b a).
```

Let p be a property, satisfied by at least some x. Then there is y that satisfies p such that $p(z)$ implies $y \leq z$ (we have already shown this result with $y \subset z$ instead of $y \leq z$). Assume that p is false for some ordinal; then there exists y, that does not satisfy p, such that if $z < y$, then $p(z)$ holds.

```
Lemma ordinal_worder4 x (p: Set -> Prop):
 is_ordinal x -> p x ->
 let y := least_ordinal p x in
   is_ordinal y & p y & (forall z, is_ordinal z -> p z -> y <=o z).
Lemma ordinal_worder3 x (p: Set -> Prop):
 is_ordinal x -> ~ (p x) ->
 let y := least_ordinal (fun z => (~ p z)) x in
   is_ordinal y & ~(p y) & (forall z, z <o y -> p z).
```

The union U of a family of ordinals is an ordinal, it is the least upper bound of the family (in the sense that $U = \sup_E X$, where E is any set of ordinals containing U and the elements of X). We shall sometimes use the notation \osup for it[1].

```
Notation "\osup" := union (only parsing).
Notation "\csup" := union (only parsing).
Notation "\opred" := union (only parsing).

Lemma OS_sup E: ordinal_set E ->
  is_ordinal (\osup E).
Lemma ord_sup_pr3 E: ordinal_set E ->
  forall i, inc i E -> i <=o (\osup E).
Lemma ord_sup_pr4 E y: ordinal_set E ->
  is_ordinal y -> (forall i, inc i E -> i <=o y) ->
  (\osup E) <=o y.
Lemma ord_sup_prop E: ordinal_set E ->
```

[1] We shall see below that this is also the cardinal supremum, and the ordinal predecessor, so that we give three names to this object.

```
  exists_unique (fun x => is_ordinal x &
   (forall y,  x <=o y <-> (is_ordinal y & forall i, inc i E -> i <=o y))).
```

The empty set is the least ordinal; we shall write 0_o instead of $\emptyset$ (later on, we shall see that the empty set is also the least cardinal, and write 0_c for it). If $0 < x$ then $0 \in x$; conversely, if $0 \in x$ and x is an ordinal, then $0 < x$. Note that, if x is an non-empty ordinal, then $0 < x$.

```
Definition ord_zero := emptyset.
Notation "\0o" := ord_zero.

Lemma OS0: is_ordinal \0o.
Lemma ozero_least x: is_ordinal x -> \0o <=o x.
Lemma ozero_least_1 x: \0o <o x -> inc \0o x.
Lemma ozero_least_2 x: is_ordinal x-> nonempty x -> \0o <o x.
Lemma ozero_least_3 x: is_ordinal x ->  inc \0o x -> \0o <o x.
Lemma ozero_least_4 x: is_ordinal x -> x <> \0o -> inc \0o x.
Lemma ozero_least_5 x: x <=o \0o -> x = \0o.
Lemma ozero_least_8 x: is_ordinal x -> x <> \0o -> \0o <o x.
```

4.1.4 Limit ordinals

Consider a well-ordering $\leq$. Let's assume that the relation "$x < a$ and $a \leq y$" is collectivizing in a; if $x < y$, then the set $\{a, x < a \text{ and } a \leq y\}$ has a least element, that depends only on x, and will be called the successor of x. Such an element exists if $\leq$ is an ordering with a graph, i.e., with a support E, provided that x is not the greatest element of E; it exists if $\leq$ is a relation without graph and if $a \leq y$ is collectivizing and x not maximal; for instance $\leq_{\mathrm{Card}}$ and $\leq_{\mathrm{ord}}$ are such relations. We show here that the successor for $\leq_{\mathrm{ord}}$ is the ordinal successor as defined above. The successor of a finite cardinal x is $x+1$; no explicit formula exists for infinite cardinals. For instance, let $\aleph_0$ denote the cardinal of $\mathbf{N}$ and $\aleph_1$ the cardinal of $\mathfrak{P}(\mathbf{N})$. The Cantor Theorem says $\aleph_0 < \aleph_1$, the "continuum hypothesis" is the assertion that $\aleph_1$ is the successor of $\aleph_0$. This is an unprovable statement.

According to Cantor, the limit x of a strictly increasing sequence of ordinals x_i is the least upper bound of the family $(x_i)_i$, and is called a *limit ordinal*. Set $\mathrm{X} = \{y, y <_{\mathrm{ord}} x\}$. We have $x \leq \sup \mathrm{X}$, since the family x_i is a subset of X, and we also have $\sup \mathrm{X} \leq x$, so that $x = \sup \mathrm{X}$. Note that, if x is the successor of y, then $\sup \mathrm{X} = y$, so that a limit ordinal is not a successor. In the case of von Neumann ordinals, $y <_{\mathrm{ord}} x$ is the same as $y \in x$, thus $\mathrm{X} = x$ and $\sup \mathrm{X} = \bigcup x$, and the condition becomes $x = \bigcup x$. Thus, either x is a successor, or $x = \bigcup x$; one case excludes the other. Note that zero is not a successor, but not a limit ordinal either. In the definition below, a limit ordinal contains zero and is stable by the successor function.

The union of x is called its *predecessor*. Any ordinal is the predecessor of its successor. It follows that any ordinal that is a successor is the successor of its predecessor. The lemma limit_ordinal_pr1 can be viewed as: if x is a limit ordinal, then x is the predecessor of itself, or x is the supremum of all y with $y \in x$, or x is the union of all y with $y \in x$.

```
Definition is_a_successor x := exists y, x = succ_o y.
Definition limit_ordinal x:=
  is_ordinal x & inc \0c x & (forall y, inc y x -> inc (succ_o y) x).
Lemma ordinal_succ_pr x: is_ordinal x ->
  let z := succ_o x in
  x <o z & (forall w, x <o w -> z <=o w).
Lemma limit_ordinal_pr0 x: is_ordinal x -> (* 27 *)
```

```
  let p := is_a_successor x in
  let q := x = \opred x in
  (p \/ q) & (p-> q-> False).
Lemma limit_ordinal_pr1 x: is_ordinal x -> (* 24 *)
  ((limit_ordinal x) <-> (nonempty x & x = union x)).
Lemma limit_ordinal_pr2 x: is_ordinal x ->
  x = \0o \/ is_a_successor x \/ limit_ordinal x.
Lemma ordinal_predecessor y: is_ordinal y -> y = \opred (succ_o y).
Lemma ordinal_predecessor1 x: is_ordinal x ->
  is_a_successor x -> x = succ_o (\opred x).
```

Let's say that an ordinal is *infinite* if it is equipotent to its successor, and *finite* otherwise. In the definition that follow, the qualification "ordinal" is missing, so that any non-irreflexive set is infinite (for instance, if $x = \{x\}$, then x is infinite). We restate one of the previous lemmas as: two ordinals are equipotent if and only if their successors are equipotent (this will imply that the successor function on cardinals is injective). This implies that x is infinite if and only if its successor is infinite. It implies that x is finite if and only if its successor is finite (note that x is an ordinal if and only if its successor is an ordinal). Another lemma says that if $x \in y$, both elements are ordinals and x is infinite, then y is infinite. Thus, any element of a finite ordinal is a finite ordinal.

```
Definition infinite_o u := u \Eq (succ_o u).
Definition finite_o u := is_ordinal u & ~ (infinite_o u).

Lemma succ_injective_o x y: is_ordinal x -> is_ordinal y ->
  ((succ_o x) \Eq (succ_o y) <->  x \Eq y).
Lemma infinite_o_increasing x y: is_ordinal x -> is_ordinal y ->
  inc x y -> infinite_o x -> infinite_o y.
Lemma finite_o_increasing x y:
  inc x y -> finite_o y -> finite_o x.
Lemma infinite_o_pr x: is_ordinal x ->
  (infinite_o (succ_o x) <-> infinite_o x).
Lemma finite_o_pr x: finite_o x <-> finite_o (succ_o x).
```

A limit ordinal x is infinite. Proof: let x be a limit ordinal. There is then a least limit ordinal ω. Since $\omega \leq x$, it suffices to prove that ω is infinite. Define $f(x)$ on ω^+ by $f(x) = x^+$ if $x \in \omega$ and $f(\omega) = \emptyset$. Since ω is limit, we have $f(x) \in \omega$ for all x. This an injective function. Since ω is the least limit ordinal, the function is surjective, thus bijective.

```
Lemma limit_infinite x: ordinal_limit x -> infinite_o x. (* 48 *)
```

4.1.5 Cardinals

For any set x, there is a well-ordering $(\mathsf{worder\_of}\ x)$ on x (The explicit form of this ordering is given page 255); and its ordinal is equipotent to x. The least ordinal equipotent to x is called the *von Neumann cardinal* of x.

```
Definition cardinalVp x y :=
  is_ordinal y & x \Eq y &
  (forall z, is_ordinal z ->  x \Eq z -> sub y z).
Definition worder_of (E:Set): Set ...
Definition cardinal_of x :=
```

```
  (least_ordinal (equipotent x) (ordinal (worder_of x))).
Lemma Zermelo_ter E:
  worder (worder_of E) & substrate (worder_of E) = E. (* 167 *)
Lemma cardinalV_unique x y z:
  cardinalVp x y -> cardinalVp x z -> y = z.
Lemma cardinalV_exists x: cardinalVp x (cardinal_of x).
```

The following piece of code make cardinal provably equal to cardinal_of, but Coq will never replace one definition by the other; this makes some proofs faster.

```
Module Type CardinalSig.
Parameter cardinal : Set -> Set.
Axiom cardinalE: cardinal = cardinal_of.
End CardinalSig.

Module Cardinal: CardinalSig.
Definition cardinal := cardinal_of.
Lemma cardinalE: cardinal = cardinal_of. Proof. by []. Qed.
End Cardinal.
```

We say that y is a cardinal if y is the cardinal of some set x. This definition can be simplified a bit and we get: For any set x, there is a set y, called its *cardinal*, denoted card(x), such that y is an ordinal equipotent to x and any ordinal z equipotent to x satisfies $y \subset z$. We say that y is a *cardinal* if it is an ordinal and any ordinal z equipotent to y satisfies $y \subset z$. Bourbaki uses uses the notation Card(x) to mean some set equipotent to x; it satisfies the property that two sets have the same cardinal if and only if they are equipotent. We use the notation card(x) only when we need to disambiguate with Card.

```
Notation cardinal := Cardinal.cardinal.
Definition is_cardinal x:=
 is_ordinal x & (forall z, is_ordinal z -> x \Eq z -> sub x z).
```

Some trivial properties. Proposition 1 [3, p. 158] states that X and Y are equipotent if and only if they have the same cardinal. One implication is trivial since any set is equipotent to its cardinal. The converse depends on the definition of an ordinal. In the von Neumann case, if Z is any ordinal, X equipotent to Z implies card(X) $\subset$ Z. Take Z = card(Y), this gives card(X) $\subset$ card(Y), thus equality.

```
Lemma cardinalV_pr x:
  is_ordinal (cardinal x) & x \Eq (cardinal x) &
  (forall z, is_ordinal z -> x \Eq z -> sub (cardinal x) z).
Lemma cardinal_pr x: (cardinal x) \Eq x.
Lemma cardinal_pr0 x: x \Eq (cardinal x).
Lemma cardinal_of_cardinal x: is_cardinal x -> cardinal x = x.
Lemma CS_cardinal x: is_cardinal (cardinal x).
Lemma OS_cardinal x: is_cardinal x ->  is_ordinal x.
Lemma cardinal_ordinal_le x: is_ordinal x -> cardinal x <=o x.
Lemma double_cardinal  x: cardinal (cardinal x) = cardinal x.
Lemma is_cardinal_pr x:
  is_cardinal x <-> (is_ordinal x & forall z, inc z x -> ~ (x \Eq z)).
Theorem cardinal_equipotent x y:
   (cardinal x = cardinal y) <-> (x \Eq y).
```

We define here zero, one and two as $\emptyset$, $\{\emptyset\}$ and $\{\emptyset,\{\emptyset\}\}$. These quantities will be denoted by 0_c, 1_c and 2_c, and have 0_o, 1_o and 2_o, as alternate names. Unfolding definitions shows that $1_o = 0_o^+$ and $2_o = 1_o^+$ so that these quantities are finite ordinals.

The Bourbaki definition of one is $1_b = \mathrm{Card}(\{\emptyset\}) = \mathrm{Card}(1_c)$. This means that 1_b is some set with one element, but could be any singleton. With our definition, 1_c is a cardinal, so that $1_b = \mathrm{card}(1_c)$ holds. On the other hand, $0_b = \mathrm{Card}(\emptyset)$, thus is $\emptyset$, since this is the only set with zero elements.

```
Definition card_zero := emptyset.
Definition card_one := singleton emptyset.
Definition card_two := doubleton emptyset (singleton emptyset).
Definition ord_one := card_one.
Definition ord_two := card_two.

Notation "\0c" := card_zero.
Notation "\1c" := card_one.
Notation "\2c" := card_two.
Notation "\1o" := ord_one.
Notation "\2o" := ord_two.

Lemma succ_o_zero: succ_o \0o = \1o.
Lemma succ_o_one: succ_o \1o = \2o.

Lemma OS1:  is_ordinal \1o.
Lemma OS2:  is_ordinal \2o.

Lemma equipotent_to_emptyset x:
  x \Eq  emptyset -> x = emptyset.
Lemma cardinal_emptyset: cardinal emptyset = \0c.
Lemma cardinal_nonemptyset x:
  cardinal x = \0c -> x = emptyset.
Lemma cardinal_nonemptyset1 x:
  nonempty x -> cardinal x <> \0c.
Lemma card1_nz: \1c <> \0c.
Lemma card2_nz: \2c <> \0c.
Lemma card_one_not_two: \1c <> \2c.

Lemma finite_zero: finite_o \0o.
Lemma finite_one: finite_o \1o.
Lemma finite_two: finite_o \2o.
```

Assume that x is a finite ordinal. Then it is not a limit ordinal, thus is either 0 or a successor y^+ and y is finite. Conversely, 0 and x^+ are finite ordinals. We pretend that moreover x is a cardinal. Since 0 is a cardinal, it suffices to show that y^+ is a cardinal if y^+ is finite. Assume that there is $z \in y^+$ is equipotent to y^+. We get a bijection $y^+ \to z$, and by restriction, an injection $y \to z$. The Cantor Bernstein theorem then says that y and z are equipotent, thus y equipotent to y^+, absurd as y is finite. We deduce that 0, 1 and 2 are cardinals. Note that an infinite cardinal cannot be a successor.

```
Lemma finite_succ  x:
  finite_o x <-> (x = emptyset \/ (exists y, finite_o y & x = succ_o y)).

Lemma CS_succ_o x: finite_o x -> is_cardinal (succ_o x).
Lemma CS0: is_cardinal \0c.
Lemma CS1: is_cardinal \1c.
Lemma CS2: is_cardinal \2c.

Lemma CS_finite_o x: finite_o x -> is_cardinal x.
```

```
Lemma infinite_pr1 y z: is_cardinal z ->  infinite_o z ->
  z = succ_o y  -> False.
```

4.1.6 Finite sets

We say that a cardinal is *finite* or *infinite* if it is finite or infinite as an ordinal, and we say that a set is finite or infinite if its cardinal is finite or infinite. As noted above, a finite ordinal is a cardinal. A finite cardinal is called a *natural integer*. The *cardinal successor* of x is the cardinal of the ordinal successor of x. Note that if x is an infinite cardinal, it is equal to its successor[2]. A cardinal cannot be both finite and infinite. In particular, an infinite cardinal is non-empty.

```
Definition finite_c  :=  finite_o.
Definition infinite_c a := is_cardinal a & infinite_o a.
Definition finite_set E := finite_c (cardinal E).
Definition infinite_set E := infinite_o (cardinal E).
Definition succ x := cardinal (succ_o x).

Lemma infinite_dichot1 x: finite_c x -> infinite_c x -> False.
Lemma infinite_dichot2 x:
  finite_c (cardinal x) -> infinite_set x -> False.
Lemma CS_finite2 x: finite_c x -> is_cardinal x.
Lemma infinite_nz y:  infinite_c y -> y <> \0c.
```

We know that A is equipotent to B if and only if $A \cup \{a\}$ is equipotent to $B \cup \{b\}$, whenever $a \notin A$ and $b \notin B$. We deduce Proposition 8 [3, p. 162], two cardinals that have the same successor are equal. If B is the cardinal of A we get: if $a \notin A$, the cardinal of $A \cup \{a\}$ is the successor of the cardinal of A. This is the same as: if $c \in C$, then the cardinal of C is the successor of the cardinal of $C - \{c\}$. Thus, if C is equipotent to $C - \{c\}$, it is an infinite set.

```
Lemma cardinal_irreflexive x: is_cardinal x -> ~ (inc x x).
Theorem succ_injective1: forall a b, is_cardinal a -> is_cardinal b ->
   succ a = succ b -> a = b.
Lemma card_succ_pr a b: ~ (inc b a) ->
  cardinal (tack_on a b) = succ (cardinal a).
Lemma card_succ_pr1 a b:
 cardinal (tack_on (compl_singl a b) b) =
  succ (cardinal (compl_singl a b)).
Lemma card_succ_pr2 a b: inc b a ->
  cardinal a = succ (cardinal (compl_singl a b)).
Lemma infinite_set_pr a b: inc b a ->
  a \Eq (compl_singl a b) ->
  infinite_set a.
Lemma infinite_set_pr1 a b: inc b a ->
  a \Eq (compl_singl a b) ->
  infinite_set (compl_singl a b).
Lemma infinite_set_pr2 x:
   infinite_o x -> ~(inc x x) -> infinite_set x.
```

Bourbaki has an axiom that says that there is an infinite set. This axiom is not needed in Coq, because $\mathbb{N}$ (the type nat) is infinite, since the successor function is a bijection $\mathbb{N} \to$

[2] Later on, we shall define the cardinal successor of an infinite cardinal as the least cardinal greater than x, and denote it succ_c x

$\mathbb{N} - \{0\}$. We can then consider the least infinite ordinal, and call it ω. If x is an ordinal, then x is infinite if and only if $\omega \subset x$. This can be restated as: x is a finite ordinal if and only if $x \in \omega$.

Since the successor of a finite ordinal is finite, ω is a limit ordinal. Since a limit ordinal is infinite, ω is the least limit ordinal. Since any infinite cardinal is a limit ordinal, then ω is the least infinite cardinal. More properties of ordinal numbers will be given in Chapter 8.

```
Definition ord_omega := least_ordinal infinite_o (cardinal nat).
Notation "\omega" := ord_omega.

Lemma nat_infinite_set: infinite_set nat.
Lemma omega0_pr:
   is_ordinal \omega & infinite_o \omega &
   (forall z, is_ordinal z -> infinite_o z -> sub \omega z).

Lemma OS_omega: is_ordinal \omega.
Lemma omega_infinite: infinite_o \omega.
Lemma omega_pr1 x: is_ordinal x ->
  (infinite_o x <-> sub \omega x).
Lemma omega_pr2 x: inc \omega <-> finite_o x.
Lemma CS_omega: is_cardinal \omega.
Lemma omega_infinitec: infinite_c \omega.
Lemma omega_limit: \omega = \opred omega.
Lemma omega_limit1: limit_ordinal \omega.
Lemma omega_limit2 x:  limit_ordinal x -> \omega <=o x.
Lemma infinite_card_limit1 x: infinite_c x -> \opred x = x.
Lemma infinite_card_limit2 x: infinite_c x -> limit_ordinal x.
Lemma omega_limit3 x: infinite_c x -> sub \omega x.
Lemma infinite_set_rw x: infinite_set x <-> infinite_c (cardinal x).
Lemma infinite_set_pr3 x: \omega <=o x ->  infinite_c (cardinal x).
```

4.1.7 Properties of equipotent sets

We state here some lemmas that will be useful when defining operations on cardinals. This section is independent of the previous ones.

We denote by Eq(X, Y) or equipotent X Y the property that there is a bijection between X and Y. We know that this relation is reflexive, symmetric and transitive (but is not an equivalence, because it has no graph). If a set E contains the n distinct elements $x_1, x_2 \ldots, x_n$, and if a set F contains the n distinct elements $y_1, y_2 \ldots, y_n$, then the set $G = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is the graph of a bijection between E and F. We may use this method to prove equipotency of doubletons.

```
Lemma singletons_equipotent x y:
  singleton x \Eq singleton y.
Lemma doubleton_equipotent1 x y x' y':  (* 25 *)
  x <> x' -> y <> y' -> doubleton x x' \Eq doubleton y y'.
```

Products of equipotent sets are equipotent. We first consider the case of productf since ext_map_prod is a bijection from $\prod E_\iota$ to $\prod F_\iota$ (See Part I, section 6.7). We also consider the case of the product of two sets (the bijection is ext_to_prodC).

```
Definition fgraphs_equipotent x y :=
  fgraph x & fgraph y & domain x = domain y
```

```
  & (forall i, inc i (domain x) ->  (V i x) \Eq (V i y)).
Lemma equipotent_ex_pr E F (z:=  equipotent_ex E F):
  equipotent E F ->
   (bijection z & source z = E & target z = F).

Lemma equipotent_productf I p1 p2:
  (forall i, inc i I ->  (p1 i) \Eq (p2 i)) ->
  productf I p1 \Eq productf I p2.
Lemma equipotent_productb x y:
  fgraphs_equipotent  x y ->  (productb x) \Eq (productb y).
Lemma equipotent_product a b a' b':
  a \Eq a' ->  b \Eq b' ->
  (product a b) \Eq (product a' b').
```

If A, B and C are sets, then $A \times B$ is equipotent to $B \times A$, (bijection is inv_graph_canon) and $A \times (B \times C)$ is equipotent to $(A \times B) \times C$ (the bijection maps $(a,(b,c))$ to $((a,b),c)$). We have $(A \cup B) \times C = (A \times C) \cup (B \times C)$. Finally, if B is a singleton, A and $A \times B$ are equipotent.

```
Lemma equipotent_product_sym a b:
  (product a b) \Eq (product b a).
Lemma product2A a b c:
  (product a (product b c)) \Eq (product (product a b) c).
Lemma distrib_inter_prod2 a b c:
  product (union2 a b) c = union2 (product a c) (product b c).
Lemma distrib_inter_prod3 a b c:
  product c (union2 a b) = union2 (product c a) (product c b).
Lemma equipotent_a_times_singl a b:
  a \Eq (product a (singleton b)).
Lemma equipotent_singl_times_a a b:
  a \Eq (product (singleton b) a).
```

Two sets $A \times B$ and $A' \times B'$ are disjoint if B and B' are disjoint; this is the case when B and B' are distinct singletons.

```
Lemma disjoint_pr a b:
  (forall u, inc u a -> inc u b -> False) -> disjoint a b.
Lemma disjoint_union2_pr0 a b x y:
  disjoint x y -> disjoint (product a x) (product b y).
Lemma disjoint_union2_pr1 x y:
  x <> y -> disjoint (singleton x) (singleton y).
Lemma disjoint_union2_pr a b x y:
  x <> y -> disjoint (product a (singleton x)) (product b (singleton y)).
```

Two unions $\bigcup X_\iota$ and $\bigcup Y_\iota$ with the same index set are equipotent if the sets are equipotent and if each family is mutually disjoint. For if f_ι is a function from X_ι into Y_ι, we can find a function f such that $f(x) = f_\iota(x)$ whenever $x \in X_\iota$, provided that x is in a unique X_ι (i.e., the family is mutually disjoint). If the family $\bigcup Y_\iota$ is mutually disjoint, then $f(x) = f(y)$ implies that there is ι such that $x \in X_\iota$ and $y \in X_\iota$, hence $f_\iota(x) = f_\iota(y)$. Thus f is injective if each f_ι is injective. As a particular case, we get conditions for $A \cup B$ and $A' \cup B'$ to be equipotent.

Two disjoint unions of equipotent sets are equipotent. Remember that the disjoint union of a family of sets X_ι is the union of the sets $X'_\iota = X_\iota \times \{\iota\}$. Two lemmas mentioned above say that each X_ι is equipotent to X'_ι and the family is disjoint.

```
Lemma equipotent_disjoint_union X Y:
```

```
  fgraphs_equipotent X Y ->
  mutually_disjoint X ->  mutually_disjoint Y ->
  (unionb X) \EQ (unionb Y).  (* 39 *)
Lemma equipotent_disjoint_union1 X Y:
  fgraphs_equipotent X Y ->
  (disjoint_union X) \Eq (disjoint_union  Y).
Lemma union2Lv a b: union2 a b = unionb (variantLc a b).
Lemma disjointLv a b: disjoint a b ->
   mutually_disjoint (variantLc a b).
Lemma equipotent_disjoint_union2 a b a' b':
  disjoint a b -> disjoint a' b' -> a \Eq a' -> b \Eq b' ->
  (union2 a b) \Eq (union2 a' b').
```

Given two sets A and B, we can consider a family f defined on a doubleton $\{x, y\}$ such that $f(x) = \mathrm{A}$ and $f(y) = \mathrm{B}$. An example is the canonical family variantLc, denoted here by F. We pretend that there is a bijection g such that $f = \mathrm{F} \circ g$ (since f and F are graphs, we compose with the graph of g). The lemma asserts that F is a functional graph whose domain is the target of g, since these are the conditions of the associativity theorems of the sum and product.

```
Definition doubleton_fam f a b :=
  exists x, exists y, x<>y & fgraph f &  domain f = doubleton x y &
    V x f = a &   V y f = b.
Lemma two_terms_bij a b f: doubleton_fam f a b ->
  let F := (variantLc a b) in
  exists g, (bijection g & target g = domain F & fgraph F &
    f = gcompose F (graph g)).    (* 40 *)
```

4.2 The cardinal of a set

The cardinal of a set, and some basic properties have been introduced earlier. We show here that x has cardinal one if and only if it is a singleton, that x has cardinal two if and only if it is a doubleton.

```
Lemma set_of_card_one x: cardinal x = \1c -> is_singleton x.
Lemma set_of_card_two x: cardinal x = \2c  ->
  exists u, exists v, u<>v & x = doubleton u v.
Lemma cardinal_singleton x: cardinal(singleton x) = \1c.
Lemma cardinal_doubleton x x':
  x <> x' ->  cardinal (doubleton  x x') = \2c.
```

4.3 Order relation between cardinals

Bourbaki defines $\mathfrak{r} \leq_{\mathrm{Card}} \mathfrak{n}$ as $\mathfrak{r}$ and $\mathfrak{n}$ are cardinals, and $\mathfrak{r}$ is equipotent to a subset of $\mathfrak{n}$. If $f : \mathrm{A} \to \mathrm{C}$ is bijective, and $\mathrm{C} \subset \mathrm{B}$, we can extend $f : \mathrm{A} \to \mathrm{B}$; the result will be injective. Conversely, if $f : \mathrm{A} \to \mathrm{B}$ is a function with range C, it induces a surjective function $\mathrm{A} \to \mathrm{C}$, which is bijective when f is injective.

```
Definition restriction_to_image f :=
  restriction2  f (source f) (image_of_fun f).
Lemma restriction_to_image_axioms f: is_function f ->
```

```
  restriction2_axioms f (source f) (image_of_fun f).
Lemma restriction_to_image_surjective f: is_function f ->
  surjection (restriction_to_image f).
Lemma restriction_to_image_bijective f: injection f ->
  bijection (restriction_to_image f).
```

We study here some properties of the relation: x is equipotent to a subset of y.

```
Definition equipotent_to_subset x y:= exists z, sub z y & x \Eq z.
Lemma inj_compose1 f f':
  injection f -> injection f' -> source f' = target f ->
  injection (f'\co f).
Lemma eq_subset_ex_inj x y:
  equipotent_to_subset x y <->
  (exists f, injection f & source f = x & target f = y).
Lemma eq_subset_pr2 a b a' b':
   a \Eq a' -> b \Eq b' ->
   equipotent_to_subset a b -> equipotent_to_subset a' b'.
Lemma eq_subset_card x y:
  equipotent_to_subset x y <-> equipotent_to_subset (cardinal x) (cardinal y).
```

We define $a \leq_{\mathrm{card}} b$ to be: a and b are (von Neumann) cardinals and $a \subset b$. Since a cardinal is an ordinal, it implies $a \leq_{\mathrm{ord}} b$ (so that $\leq_{\mathrm{card}}$ is a well-ordering). We claim that $a \leq_{\mathrm{card}} b$ is equivalent to $a \leq_{\mathrm{Card}} b$. Obviously, if a is a subset of b, it is equipotent to a subset of b. Assume that a is equipotent to a subset c of b, and that a is not a subset of b; since a and b are ordinal numbers, we have $b \subset a$. The Cantor-Bernstein theorem says that b is equipotent to c, hence to a, and thus $a = b$.

```
Definition cardinal_le x y :=
  is_cardinal x & is_cardinal y & sub x y.
Definition cardinal_lt a b := cardinal_le a b & a <> b.
Notation "x <=c y" := (cardinal_le x y) (at level 60).
Notation "x <c y" := (cardinal_lt x y) (at level 60).

Lemma ordinal_cardinal_le x y:
  x <=c y -> x <=o y.
Lemma cardinal_le_aux1 x y:
  x <=c y -> equipotent_to_subset x y
Lemma cardinal_le_aux2 x y: is_cardinal x -> is_cardinal y ->
  (equipotent_to_subset x y <-> x <=c y).
```

The relation $\leq_{\mathrm{Card}}$ is an ordering: the non-trivial point to to show antisymmetry. It is a consequence of the Cantor-Bernstein theorem. We consider the $\leq_{\mathrm{card}}$ which is trivially a well-ordering (given a non family of cardinals, the least ordinal is obviously the least cardinal).

```
Lemma cardinal_leR x: is_cardinal x -> x <=c x.
Lemma cardinal_leT a b c:
  a <=c b -> b <=c c -> a <=c c.
Lemma cardinal_leA x y:
  x <=c y -> y <=c x -> x = y.
Theorem cardinal_le_wor: worder_r cardinal_le.
```

We list some useful properties.

```
Definition cardinal_set X := forall x, inc x X -> is_cardinal x.
Lemma eq_subset_card1 x y:
  equipotent_to_subset x y <-> (cardinal x) <=c (cardinal y).
Lemma incr_fun_morph f: injection f ->
  (cardinal (source f)) <=c (cardinal (target f)).
Lemma sub_smaller a b:
  sub a b ->  (cardinal a) <=c (cardinal b).
```

Some consequences. Note that $\leq_{\mathrm{Card}}$ is a total ordering since $\leq_{\mathrm{ord}}$ is total.

```
Lemma not_card_le_lt a b: a <=c b -> b <c a -> False.
Lemma cardinal_lt_leT a b c:
  a <c b -> b <=c c -> a <c c.
Lemma cardinal_le_ltT a b c:
  a <=c b -> b <c c -> a <c c.
Lemma wordering_cardinal_le_pr x:
  cardinal_set x ->
  (substrate (graph_on cardinal_le x) = x &
  worder (graph_on cardinal_le x)).
Lemma card_le_to_ell a b:
  is_cardinal a -> is_cardinal b ->
  a = b \/ a <c b  \/ b <c a.
Lemma card_le_to_el a b:
  is_cardinal a -> is_cardinal b ->
  a <=c b \/ b <c a.
Lemma card_le_to_ee a b:
  is_cardinal a -> is_cardinal b ->
  a <=c b \/ b <=c a.
```

These lemmas show how $\leq_{\mathrm{ord}}$ and $\leq_{\mathrm{card}}$ are related.

```
Lemma ordinal_cardinal_le1 y x:
  x <=o y -> (cardinal x) <=c (cardinal y).
Lemma ordinal_cardinal_lt  x y: x <c y -> x <o y.
Lemma ordinal_cardinal_le2 x y:  is_cardinal x ->  is_ordinal y ->
   ((y <o x) <-> (cardinal y <c x)).
```

We show here that x is a finite (resp. infinite) cardinal if and only if it is a cardinal and $x \neq x+1$ (resp. $x = x+1$). A cardinal is either finite or infinite, a set set is either finite or infinite.

```
Lemma finite_c_pr x: finite_c x <->  (is_cardinal x & x <> succ x).
Lemma infinite_c_pr x: infinite_c x <-> (is_cardinal x & x = succ x).
Lemma finite_dichot x: is_cardinal x ->
  finite_c x \/ infinite_c x.
Lemma finite_dichot1 x:  finite_set x \/ infinite_set x.
```

If a is finite and b is infinite, then $a < b$. If $b \leq c$ then c is infinite. If $d \leq a$ then d is finite.

```
Lemma finite_le_infinite a b:
  finite_c a -> infinite_c b -> a <=c b.
Lemma finite_lt_infinite a b:
  finite_c a -> infinite_c b -> a <c b.
Lemma ge_infinite_infinite a b:
  infinite_c a -> a <=c b -> infinite_c b.
Theorem le_finite_finite a b:  finite_c b ->
  a <=c b -> finite_c a.
```

A subset of a finite set is finite. A cardinal x is infinite if and only if $\omega \leq x$.

```
Lemma infinite_c_pr2 x: infinite_c x <-> (\omega <=c x).
Lemma sub_finite_set x y: sub x y -> finite_set y -> finite_set x.
```

We show here that if x is an ordinal, it is a finite or infinite set if and only if it is a finite or infinite ordinal, and this implies $x < \omega$ or $\omega \leq x$.

```
Section OrdinalFinite.
Variable x:Set.
Hypothesis ox: is_ordinal x.

Lemma ordinal_finite1: finite_set x -> finite_o x.
Lemma ordinal_finite2: infinite_set x -> infinite_o x.
Lemma ordinal_finite3: finite_set x -> x <o \omega.
Lemma ordinal_finite4: infinite_set x -> \omega <=o x.
End OrdinalFinite.
```

We have $0 \leq \mathfrak{a}$ for every cardinal $\mathfrak{a}$, and $1 \leq \mathfrak{a}$ if moreover $\mathfrak{a} \neq 0$. We have Card(E) ≥ 1 if and only if E is non-empty.

```
Lemma ordinal_cardinal_le3 x y:
   is_cardinal x -> is_cardinal y -> x <=o y -> x <=c y.
Lemma ordinal_cardinal_lt3 x y:
   is_cardinal x -> is_cardinal y -> x <o y -> x <c y.

Lemma czero_least x: is_cardinal x -> \0c <=c x.
Lemma czero_least_7 x: x <c \0c -> False.
Lemma czero_least_5 a: a <=c \0c -> a = \0c.
Lemma oone_small_1 x: is_ordinal x -> x <> \0o -> \1o <=o x.
Lemma oone_small_2 x: x <o \1o -> x = \0o.
Lemma oone_small_3 x: \1o <=o x -> x <> \0o.
Lemma oone_small_4 x:  (\1o <=o x) <-> (\0o <o x).
Lemma oone_small_5 x: is_ordinal x -> x <> \0o ->  (\1o <=o x).
Lemma oone_small_6 x: is_ordinal x -> x <> \0o ->  x <> \1o -> (\1o <o x).

Lemma cone_small_1 x: is_cardinal x -> x <> \0c -> \1c <=c x.
Lemma cone_small_2 x:  x <c \1o -> x = \0c.
Lemma cone_small_3 x:  \1c <=c x -> x <> \0o.
Lemma cone_small_4 x:  (\1c <=c x) = (\0c <c x).
Lemma cone_small_5 x: is_cardinal x ->  ((\1c <=c x) <-> (x <> \0c)).
Lemma ctwo_small x: is_cardinal x ->  x <> \0c ->  x <> \1c  -> \2c <=c x.
Lemma card_le_one_prop E: \1c <=c (cardinal E) <-> nonempty E.
```

We have Card(E) ≥ 2 if and only if E has at least two elements.

```
Lemma card_le_two_prop E:
  \2c <=c  (cardinal E)
  <-> exists a, exists b, inc a E & inc b E & a <> b.
```

One can consider the set of cardinals $\leq_{\mathrm{Card}} a$, when a is a cardinal (because it is a subset of the set of ordinals $\leq_{\mathrm{ord}} a$). One can also consider the set of cardinals that are $< a$.

```
Definition set_of_cardinals_le a:=
```

```
  Zo (succ_o a) is_cardinal.
Definition set_of_cardinals_lt x :=
  Zo x (fun z => z <c x).

Lemma set_of_cardinals_le_rw a : is_cardinal a ->
  forall b, (inc b (set_of_cardinals_le a) <-> (b <=c a)).
Lemma set_of_cardinals_lt_rw x: is_cardinal x ->
  forall y, inc y (set_of_cardinals_lt x) = (y <c x).
```

Given a family of cardinals there is a least upper bound (an upper bound a such that, for all other upper bounds b we have $a \leq b$). It is called the supremum of the family. This is Proposition 2 in [3, p. 160]. We first show the same result for a set of cardinals. We can use Bourbaki's argument or notice that the union a of the family is the least upper bound (considered as an ordinal). Let b an ordinal equipotent to a. Assume $b <_{\mathrm{ord}} a$. It is not an upper bound of the family, thus for some element c of the family we have $b <_{\mathrm{ord}} c \leq_{\mathrm{ord}} a$. Let A, B and C be the cardinals of these quantities. We get $\mathrm{B} \leq_{\mathrm{Card}} \mathrm{C} \leq_{\mathrm{Card}} \mathrm{A}$. Since A = B, these three cardinals are equal. Thus b and c are equipotent. Since c is a cardinal, we get $c \leq_{\mathrm{ord}} b$, absurd.

```
Definition cardinal_fam x :=
  fgraph x & (forall a, inc a (domain x) -> is_cardinal (V a x)) ->

Lemma CS_sup E: cardinal_set E -> is_cardinal (\csup E). (* 23 *)
Lemma card_sup_pr3 E: cardinal_set E ->
  forall i, inc i E -> i <=c (\csup E).
Lemma card_sup_pr4 E: cardinal_set E ->
  forall y, is_cardinal y -> (forall i, inc i E -> i <=o y) ->
  (\csup E) <=c y.
Lemma card_sup_pr5 E x: cardinal_set E -> \osup E <c x ->
   (forall t, inc t E -> t <c x).
Lemma card_sup_pr6 E f g:
  (forall x, inc x E -> f x <=c g x) ->
  \csup (fun_image E f) <=c \csup (fun_image E g).

Lemma cardinal_supremum1 x:
  cardinal_set x ->
  exists_unique (fun b => is_cardinal b &
    (forall a, inc a x -> a <=c b) &
    (forall c, is_cardinal c ->  (forall a, inc a x -> a <=c c) ->
      b <=c c)).
Theorem cardinal_supremum2 x:
  cardinal_fam x ->
  exists_unique (fun b => is_cardinal b &
    (forall a, inc a (domain x) -> (V a x) <= c b) &
    (forall c, is_cardinal c ->
      (forall a, inc a (domain x) -> (V a x) <=c c) ->
      b <=c c)).
```

Proposition 3 in [3, p. 160] says that Card(Y) ≤ Card(X) if there is a surjection of X onto Y. As a consequence, the range of a function is not bigger than the source.

```
Theorem surjective_cardinal_le x y:
  (exists z, surjection z & source z = x & target z = y) ->
  cardinal y <=c cardinal x.
Lemma image_smaller_cardinal f: is_function f ->
  cardinal (image_of_fun f) <=c cardinal (source f).
```

4.4 Operations on cardinals

The remainder of the chapter is independent of the definition of the cardinals.

Given a family of cardinals $(\mathfrak{a}_\iota)_{\iota\in I}$, the cardinal of the sum of these sets is called the *cardinal sum* and denoted by $\sum_{\iota\in I}\mathfrak{a}_\iota$; the cardinal of the product is called the *cardinal product* and denoted $\mathsf{P}_{\iota\in I}\mathfrak{a}_\iota$. The qualificative "cardinal" will be omitted if there is no risk of confusion. The associated operations are called *addition* and *multiplication.* The notation $\prod_{\iota\in I}\mathfrak{a}_\iota$ will later be used for both the normal product and the cardinal product.

```
Definition card_sum x := cardinal (disjoint_union x).
Definition card_prod x := cardinal (productb x).
```

Proposition 4 of [3, p. 160] says that the cardinal sum or cardinal product of the family $\mathrm{Card}(E_\iota)$ is the cardinal of the sum or the product of the sets E_ι. In other terms, if $\mathfrak{a}_\iota = \mathrm{Card}(E_\iota)$ then $\mathrm{Card}(\prod E_\iota) = \mathsf{P}\mathfrak{a}_\iota$ and $\mathrm{Card}(\mathsf{S}E_\iota) = \sum\mathfrak{a}_\iota$. One can notice that the cardinal of a union is at most the cardinal of the disjoint union[3].

```
Theorem cprod_pr x: fgraph x ->
  cardinal (productb x) =
  cardinal_prod (L (domain x) (fun a => cardinal (V a x))).
Theorem csum_pr x: fgraph x ->
  cardinal (disjoint_union x) =
  card_sum (L (domain x) (fun a => cardinal (V a x))).
Lemma csum_pr3: forall X Y, fgraph X -> fgraph Y ->
  domain X = domain Y ->
  (forall i, inc i (domain X) -> cardinal (V i X) = cardinal (V i Y)) ->
  card_sum X = card_sum Y.
Lemma csum_pr1 x: fgraph x ->
  (cardinal (unionb x))
  <=c (card_sum (L (domain x) (fun a => cardinal (V a x)))).
```

Proposition 5 [3, p. 161] says that if f is a bijection from K to I and if $\mathfrak{a}_\iota$ is a cardinal then ("commutativity" of the sum and product):

$$\sum_{\kappa\in K}\mathfrak{a}_{f(\kappa)} = \sum_{\iota\in I}\mathfrak{a}_\iota, \qquad \mathsf{P}_{\kappa\in K}\mathfrak{a}_{f(\kappa)} = \mathsf{P}_{\iota\in I}\mathfrak{a}_\iota. \tag{4}$$

If the family $(J_\lambda)_{\lambda\in L}$ is a partition of I, then ("associativity" of the sum and product):

$$\sum_{\iota\in I}\mathfrak{a}_\iota = \sum_{\lambda\in L}\Big(\sum_{\iota\in J_\lambda}\mathfrak{a}_\iota\Big), \qquad \mathsf{P}_{\iota\in I}\mathfrak{a}_\iota = \mathsf{P}_{\lambda\in L}\Big(\mathsf{P}_{\iota\in J_\lambda}\mathfrak{a}_\iota\Big). \tag{5}$$

Let $((\mathfrak{a}_{\lambda,\iota})_{\iota\in J_\lambda})_{\lambda\in L}$ be a family of families of cardinals. Let $I = \prod J_\lambda$. Distributivity of product over sum is

$$\mathsf{P}_{\lambda\in L}\Big(\sum_{\iota\in J_\lambda}\mathfrak{a}_{\lambda,\iota}\Big) = \sum_{f\in I}\Big(\mathsf{P}_{\lambda\in L}\mathfrak{a}_{\lambda,f(\lambda)}\Big). \tag{6}$$

Note that we do not need $\mathfrak{a}_\iota$ be a cardinal in any of these theorems. The relations are trivial for the product, and in the case of the union, we have to check that the families are disjoint. These formulas are numbered (4), (5) and (6), in order to respect the original Bourbaki numbering.

[3] Bourbaki has no notation for the disjoint union; we use a big S by analogy with the big P

```
Theorem csum_Cn X f:
  fgraph X ->  target f = domain X -> bijection f ->
  card_sum X = card_sum (gcompose X (graph f)). (* 27 *)
Theorem cprod_Cn X f:
  fgraph X ->  target f = domain X ->  bijection f ->
  card_prod X = card_prod (gcompose X (graph f)).
Theorem csum_An f g:
  fgraph f  -> partition_fam g (domain f) ->
  card_sum f  = card_sum (L (domain g) (fun l =>
    card_sum (restr f (V l g)))).   (* 42 *)
Theorem cprod_An f g:
  fgraph f  -> partition_fam g (domain f) ->
  card_prod f  = card_prod (L (domain g) (fun l =>
    card_prod (restr f (V l g)))).
Theorem cprod_sumDn f:  (* 57 *)
  fgraph f ->
  (forall l, inc l (domain f) -> fgraph (V l f)) ->
  card_prod (L (domain f) (fun l => card_sum (V l f))) =
  card_sum (L (productf (domain f) (fun l => (domain (V l f))))
    (fun g => (card_prod (L (domain f) (fun l => V (V l g) (V l f)))))).
```

Given two sets $\mathfrak{a}$ and $\mathfrak{b}$, we can consider a family F defined on a doubleton $\{x, y\}$ such that $F(x) = \mathfrak{a}$ and $F(y) = \mathfrak{b}$. By commutativity, the cardinal sum and cardinal product of the family depends only on $\mathfrak{a}$ and $\mathfrak{b}$. It is denoted by $\mathfrak{a} + \mathfrak{b}$ and $\mathfrak{a}.\mathfrak{b}$ respectively. Commutativity is obvious.

```
Definition card_sum2 a b :=  card_sum (variantLc a b).
Definition card_prod2 a b := card_prod (variantLc a b).
Notation "x +c y" := (card_sum2 x y) (at level 50).
Notation "x *c y" := (card_prod2 x y) (at level 40).


Lemma CS_sum2 a b: is_cardinal (a +c b).
Lemma CS_prod2 a b: is_cardinal (a *c b).

Lemma csum2_pr a b f:
  doubleton_fam f a b -> a +c b = card_sum f.
Lemma cprod2_pr a b f:
  doubleton_fam f a b -> a *c b = card_prod f.

Lemma card_commutative_aux a b:
  doubleton_fam (Lvariantc b a) a b.
Lemma csumC a b: a +c b = b +c a.
Lemma cprodC a b: a *c b = b *c a.
```

If α and β are the two elements of the canonical doubleton I, if f is any function, the sum (resp. product) of the graph of f on I is $f(\alpha) + f(\beta)$ and $f(\alpha) \cdot f(\beta)$ respectively.

```
Definition TPas := singleton TPa.
Definition TPbs := singleton TPb.

Lemma doubleton_fam_canon f:
  doubleton_fam (L two_points f) (f TPa) (f TPb).

Lemma csum2_pr0 f:
```

```
  card_sum (L two_points f) = (f TPa) +c (f TPb).
Lemma cprod2_pr0 f:
  card_prod (L two_points f) = (f TPa) *c (f TPb).

Lemma disjoint_union2_pr3 a b x y: y <> x ->
  (a +c b) \Eq
    (union2 (product a (singleton x)) (product b (singleton y))).
Lemma disjoint_union2_pr4 a b:
  (a +c b) \Eq (union2 (product a TPas) (product b TPbs)).
Lemma csum2_pr4 a b a' b': a \Eq a' -> b \Eq b' ->
  a +c b = a' +c b'.
Lemma csum2_pr3 a b a' b':
  disjoint a b -> a \Eq a' -> b \Eq b' ->
  cardinal (union2 a b) = a' +c b'.
Lemma cprod2_pr1 a b:
  a *c b = cardinal (product a b).

Lemma csum2_pr2 a b a' b':
  cardinal a = cardinal a' -> cardinal b = cardinal b' ->
  a +c b = a' +c b'.
Lemma cprod2_pr2 a b a' b':
  cardinal a = cardinal a' -> cardinal b = cardinal b' ->
  a *c b = a' *c b'.
Lemma csum2_pr2b x y: x +c (cardinal y) = x +c y.
Lemma cprod2_pr2b x y: x *c (cardinal y) = x *c y.
```

We have $\sum_{i\in \mathrm{I}} f_i + \sum_{i\in \mathrm{I}} g_i = \sum_{i\in \mathrm{I}} (f_i + g_i)$. The proof is a bit tricky. Let $\mathrm{K} = \mathrm{I} \times \{\alpha, \beta\}$, where α and β are two distinct elements. We define a function h on K that associates f_i to (i, α) and g_i to (i, β). We have $\sum f_i = \sum_\alpha h_i$ where the index α denotes the restriction of h to the first part of K (this is the commutativity theorem, we use some auxiliary lemmas in order to ease the proof). We have $\sum f_i + \sum g_i = \sum h_k$ by associativity. If we consider K as the disjoint union where the first component is fixed, we can reapply the associativity theorem.

```
Lemma cardinal_commutativity_aux X f I:
  (forall x, inc x I -> inc (f x) (domain X)) ->
  (forall x y, inc x I -> inc y I -> f x = f y -> x = y) ->
  (forall y, inc y (domain X)-> exists x, inc x I & f x = y) ->
  fgraph X ->
  let F := (BL f I (domain X)) in
    (target F = domain X & bijection F &
       gcompose X (graph F) = L I (fun z : Set => V (f z<) X)).

Lemma csum_cn2 X f I:
  (forall x, inc x I -> inc (f x) (domain X)) ->
  (forall x y, inc x I -> inc y I -> f x = f y -> x = y) ->
  (forall y, inc y (domain X)-> exists x, inc x I & f x = y) ->
  fgraph X ->
  card_sum X = card_sum (L I (fun z : Set => V (f z) X)).
Lemma cprod_Cn2 X f I:
  (forall x, inc x I -> inc (f x) (domain X)) ->
  (forall x y, inc x I -> inc y I -> f x = f y -> x = y) ->
  (forall y, inc y (domain X)-> exists x, inc x I & f x = y) ->
  fgraph X ->
  card_prod X = card_prod (L I (fun z : Set => V (f z) X)).

Lemma sum_of_sums f g I:
```

```
 (card_sum (L I f)) +c (card_sum (L I g)) =
  card_sum (L I (fun i =>  (f i) +c (g i))). (* 86 *)
Lemma prod_of_prods f g I:
  (cardprod (L I f)) *c (card_prod (L I g)) =
  card_prod (L I (fun i =>  (f i) *c (g i))).  (* 86 *)
```

As a corollary, if $\mathfrak{a}$, $\mathfrak{b}$ and $\mathfrak{c}$ are cardinals we have

(1) $$\mathfrak{a}+\mathfrak{b}=\mathfrak{b}+\mathfrak{a} \qquad \text{and} \qquad \mathfrak{a}\mathfrak{b}=\mathfrak{b}\mathfrak{a},$$

(2) $$\mathfrak{a}+(\mathfrak{b}+\mathfrak{c})=(\mathfrak{a}+\mathfrak{b})+\mathfrak{c} \qquad \text{and} \qquad \mathfrak{a}(\mathfrak{b}\mathfrak{c})=(\mathfrak{a}\mathfrak{b})\mathfrak{c},$$

(3) $$\mathfrak{a}(\mathfrak{b}+\mathfrak{c})=\mathfrak{a}\mathfrak{b}+\mathfrak{a}\mathfrak{c}.$$

Note that the formulas are true even if $\mathfrak{a}$, $\mathfrak{b}$ and $\mathfrak{c}$ are not cardinals. Associativity of the product is a consequence of equipotency of $A \times (B \times C)$ and $(A \times B) \times C$. Associativity of the sum is a consequence of associativity of $\cup$, commutativity of $+$ and $\cup$, and the property that $\mathfrak{a}+(\mathfrak{b}+\mathfrak{c})$ is equipotent $\mathfrak{a}_i \cup (\mathfrak{b}_j \cup \mathfrak{c}_k)$, if the indices are distinct; the current proof is four times shorter than the original one. The same idea can be used for (3). The quantity $\mathfrak{a}(\mathfrak{b}+\mathfrak{c})$ is equipotent to $\mathfrak{a} \times (\mathfrak{b}_i \cup \mathfrak{c}_j)$ hence to $(\mathfrak{a} \times \mathfrak{b}_i) \cup (\mathfrak{a} \times \mathfrak{c}_j)$, and $(\mathfrak{a} \times \mathfrak{b})_i \cup (\mathfrak{a} \times \mathfrak{c})_j$, by associativity of the product. We show in fact $(\mathfrak{b}+\mathfrak{c})\mathfrak{a}=\mathfrak{b}\mathfrak{a}+\mathfrak{c}\mathfrak{a}$, because, basically we use equipotency of $(A \cup B) \times C$ and $(A \times C) \cup (B \times C)$. The same techniques as above show

(7) $$\mathfrak{a}\sum_{\iota\in I}\mathfrak{b}_\iota=\sum_{\iota\in I}\mathfrak{a}\mathfrak{b}_\iota.$$

```
Lemma cprodA a b c:
  a *c (b *c c) = (a *c b) *c c.
Lemma csumA a b c: (* 21 *)
   a +c (b +c c) = (a +c b) +c c.
Lemma equipotent_product1 a b c:
  equipotent a b -> equipotent (product a c) (product b c).
Lemma cprod_sumDl  a b c:
  a *c (b +c c) = (a *c b) +c (a *c c).  (* 20 *)
Lemma cprod_sumDr  a b c:
  (b +c c) *c a = (b *c a) +c (c *c a).
Lemma distrib_prod2_sum A f:
  product A (unionb f) = unionb (L (domain f) (fun x => product A (V x f))).
Lemma cprod2_sumDn a f: is_cardinal a -> fgraph f ->
  a *c (card_sum f) =
  card_sum (L (domain f) (fun i => a *c (V i f))).
```

4.5 Properties of the cardinals 0 and 1

If a family is empty, the sum is zero and the product is one. If a family has a single element that is a cardinal, this element is the sum or the product.

```
Lemma csum_trivial f: domain f = emptyset ->
  card_sum f = \0c.
Lemma cprod_trivial f: domain f = emptyset ->
  card_prod f = \1c.
Lemma csum_trivial2 x f: domain f = singleton x ->
```

```
  card_sum f =  cardinal (V x f).
Lemma cprod_trivial2 x f: fgraph f -> domain f = singleton x ->
  card_prod f =  cardinal (V x f).
Lemma csum_trivia1 x f: domain f = singleton x ->
  is_cardinal (V x f) -> card_sum f =  V x f.
Lemma cprod_trivial1 x f: fgraph f -> domain f = singleton x ->
  is_cardinal (V x f) -> card_prod f =  V x f.
Lemma csum_trivial3 x a: is_cardinal a ->
  card_sum (cst_graph (singleton x) a) = a.
```

One can remove 0 in a sum and 1 in a product. This is Proposition 6 [3, p. 162]. The result is clear for the sum, because $0_i = \emptyset$ (where 0_i means $0 \times \{i\}$). In the case of a product, it is a trivial consequence of bijective_prj. If the family has two elements, this gives nice results. If a factor of a product is zero, so is the product itself.

```
Theorem csum_zero_unit f j:
  fgraph f -> sub j (domain f) ->
  (forall i, inc i (complement (domain f) j) -> (V i f) = \0c) ->
  card_sum f = card_sum (restr f j).
Theorem cprod_one_unit f j:
  fgraph f -> sub j (domain f) ->
  (forall i, inc i (complement (domain f) j) -> (V i f) = \1c) ->
  card_prod f = card_prod (restr f j).

Lemma csum0r a: is_cardinal a ->  a +c \0c = a.
Lemma csum0l a: is_cardinal a -> \0c +c a = a.
Lemma cprod1r a: is_cardinal a -> a *c \1c = a.
Lemma cprod1l a: is_cardinal a ->\1c *c a = a.
Lemma cprod0r a:  a *c \0c = \0c.
Lemma cprod_zero_absorbing f:
  fgraph f ->  (exists i, inc i (domain f) & cardinal (V i f) = \0c)
  -> card_prod f = \0c.
```

Let $\mathfrak{a}$ and $\mathfrak{b}$ be two cardinals; consider a set I equipotent to $\mathfrak{b}$ and the two families $\mathfrak{a}_\iota = \mathfrak{a}$ and $\mathfrak{c}_\iota = 1$. Then

$$\mathfrak{a}\mathfrak{b} = \sum_{\iota\in I} \mathfrak{a}_\iota; \qquad \mathfrak{b} = \sum_{\iota\in I} \mathfrak{c}_\iota. \tag{8}$$

The first formula is obtained from the second after multiplication by $\mathfrak{a}$, and using distributivity. We prove a similar result for $I = \mathfrak{b}$, and when $\mathfrak{a}$ and $\mathfrak{b}$ are any sets. The cardinal successor of x is $x+1$ (since this is $\mathrm{Card}(x^+)$ and x^+ is the disjoint union of x and singleton). Since 2 is the successor of 1, it follows $1+1=2$ and $x+x=2x$.

```
Lemma csum_of_ones b:   card_sum (cst_graph b \1c) = cardinal b.
Lemma csum_of_same a b: card_sum (cst_graph b a) = a *c b.
Lemma csum_of_ones1 b j: is_cardinal b -> b \Eq j ->
  card_sum (L j (fun _ => \1c)) = b.
Lemma csum_of_same1 a b j:
  is_cardinal a -> is_cardinal b -> b \Eq j ->
  card_sum (cst_graph j a) = a *c b.
Lemma card_succ_pr3 x:
   succ (cardinal x) = x +c \1c.
Lemma card_succ_pr4 x: is_cardinal x ->  succ x = x +c \1c.
Lemma card_two_pr: \1c +c \1c = \2c.
Lemma two_times_n n: \2c *c n = n +c n.
```

Proposition 7 [3, p. 162] says that a cardinal product is non-zero if and only if each factor is non-zero (because a product is non-empty if and only if no factor is empty). Proposition 8 [3, p. 162] asserts injectivity of the successor function, namely that if $\mathfrak{a}$ and $\mathfrak{b}$ are two cardinals such that $\mathfrak{a}+1=\mathfrak{b}+1$ then $\mathfrak{a}=\mathfrak{b}$. In effect, there exists X equipotent to $\mathfrak{a}$, Y equipotent to $\mathfrak{b}$, and $u\notin X$, $v\notin Y$ such that $X\cup\{u\}=Y\cup\{v\}$. If $u=v$, then $X=Y$; otherwise, if $Z=Y\cap Y$, we have $X=Z\cup\{v\}$ and $Y=Z\cup\{u\}$, so that if $\mathfrak{c}=\mathrm{Card}(Z)$ we have $\mathfrak{a}=\mathfrak{b}=\mathfrak{c}+1$.

```
Theorem cprod_nz f: fgraph f ->
 ((forall i, inc i (domain f) -> V i f <> \0c) <-> (card_prod f <> \0c)).
Lemma cprod2_nz a b: a <> \0c -> b <> \0c ->  a *c b <> \0c.
Theorem succ_injective a b: is_cardinal a ->  is_cardinal b ->
  a +c \1c = b +c \1c -> a = b.  (* 91 *)
```

4.6 Exponentiation of cardinals

If $\mathfrak{a}$ and $\mathfrak{b}$ are two cardinals, the cardinal of the set of functions from $\mathfrak{b}$ to $\mathfrak{a}$ is denoted $\mathfrak{a}^{\mathfrak{b}}$, by abuse of notations[4]. Proposition 9 [3, p. 163] says that we can replace $\mathfrak{a}$ and $\mathfrak{b}$ by equipotent sets.

```
Definition card_pow a b := cardinal (set_of_functions b a).
Notation "x ^c  y" := (card_pow x y) (at level 30).

Lemma cpow_pr a b a' b':
  a \Eq a' -> b \Eq b' -> a ^c b = a' ^c b'.
Theorem cpow_pr1 x y:
  cardinal (set_of_functions y x) = (cardinal x) ^c (cardinal y).
```

Proposition 10 [3, p. 163] says that if $\mathfrak{a}$ and $\mathfrak{b}$ are two cardinals, I is a set with cardinal $\mathfrak{b}$ and $\mathfrak{a}_\iota$ is the constant family $\mathfrak{a}$, then $\mathfrak{a}^{\mathfrak{b}}=\prod_{\iota\in I}\mathfrak{a}_\iota$. This is a trivial consequence of the fact that the set of functions and the set of graphs of functions are equipotent. Note: we do not need $\mathfrak{a}$ and $\mathfrak{b}$ to be cardinals. Note also that $\mathrm{Card}(I)=\mathfrak{b}$ implies that $\mathfrak{b}$ is a cardinal, hence, we give also another version of the theorem.

A consequence is that, if $\mathfrak{a}$ and $\mathfrak{b}$ are cardinals, $(\mathfrak{a}_\iota)_{\iota\in I}$ and $(\mathfrak{b}_\iota)_{\iota\in I}$ are families of cardinals we have

$$\mathfrak{a}^{\sum_{\iota\in I}\mathfrak{b}_\iota}=\prod_{\iota\in I}\mathfrak{a}^{\mathfrak{b}_\iota},\qquad \Big(\prod_{\iota\in I}\mathfrak{a}_\iota\Big)^{\mathfrak{b}}=\prod_{\iota\in I}\mathfrak{a}_\iota^{\mathfrak{b}}. \tag{9}$$

The proof does not make use of the fact that the sets are cardinals, so we dropped the assumption. The proof of the first formula is as follows. Let $\mathfrak{a}_\iota=\mathfrak{a}$. We have $\mathfrak{a}^{\sum_{\iota\in I}\mathfrak{b}_\iota}=\prod_{J}\mathfrak{a}_\iota$, where J is any set whose cardinal is $\sum_{\iota\in I}\mathfrak{b}_\iota$; we chose the disjoint union of the sets $\mathfrak{b}_\iota$. We have a natural partition of J and we can apply the associativity of the product. For the second formula, let $\mathfrak{a}_{\iota\beta}=\mathfrak{a}_\iota$ for $\iota\in I$ and $\beta\in\mathfrak{b}$. This is a family defined on the product $I\times\mathfrak{b}$, for which we have have two natural partitions (all elements with the same ι, or all elements with the same β); we can apply associativity of the product twice. We also have

$$\mathfrak{a}^{\mathfrak{b}+\mathfrak{c}}=\mathfrak{a}^{\mathfrak{b}}\mathfrak{a}^{\mathfrak{c}},\qquad (\mathfrak{a}\mathfrak{b})^{\mathfrak{c}}=\mathfrak{a}^{\mathfrak{c}}\mathfrak{b}^{\mathfrak{c}},\qquad \mathfrak{a}^{\mathfrak{b}\mathfrak{c}}=(\mathfrak{a}^{\mathfrak{b}})^{\mathfrak{c}}. \tag{10}$$

[4]The trouble seems to be that 4^2 and 2^4 denote the set of graphs of mappings from 2 to 4 or from 4 to 2; these sets are obviously distinct, but have the same number of elements; hence $4^2=2^4$ is true with these new notations, see page 92. Some authors write ${}^{E}F$ for the set of functions $E\to F$.

```
Lemma cpow_pr2 a b:
  card_prod (cst_graph b a) = a ^c b.
Theorem cpow_pr3 a b j: cardinal j = b ->
  card_prod (cst_graph j a) = a ^c b.
Lemma cpow_sum a f: fgraph f ->
  a ^c (card_sum f) =
  card_prod (L (domain f) (fun i => a ^c (V i f))).  (* 27 *)
Lemma cpow_prod b f: fgraph f ->
  (card_prod f) ^c b =
  card_prod (L (domain f) (fun i => (V i f) ^c b)). (* 72 *)
Lemma cpow_sum2 a b c:   a ^c (b +c c) = (a ^c b) *c (a ^c c).
Lemma cpow_prod2 a b c:  (a *c b) ^c c =  (a ^c c) *c (b ^c c).
Lemma cpow_pow a b c:  a ^c (b *c c) = (a ^c b) ^c c.
```

Proposition 11 [3, p. 164] states that

$$\mathfrak{a}^0 = 1, \qquad \mathfrak{a}^1 = \mathfrak{a}, \qquad 1^{\mathfrak{a}} = 1, \qquad 0^{\mathfrak{b}} = 0 \quad (\mathfrak{b} \neq 0). \tag{11}$$

The Bourbaki proof is the following. We want to compute the number of functions from F to E is some cases. If F is empty, there is only the empty function; if F is a singleton then E^F are E equipotent (the bijection is product1_canon), if E has a single element, there is only one function, a constant; finally if the source is non-empty and the target is empty, there is no function. We use different properties. In the first two cases, we replace the power by a product whose index set has 0 or 1 element, and simplify the result. In the third case we rewrite 1 as a product whose index set is empty, and use distributivity (9b).

Note that $\mathfrak{a}^2 = \mathfrak{a}.\mathfrak{a}$.

```
Lemma cpowx0 a: a ^c \0c =  \1c.
Lemma cpow00: \0c ^c \0c =  \1c.
Lemma cpowx1c a: a ^c \1c = cardinal a..
Lemma cpowx1 a: is_cardinal a -> a ^c \1c = a.
Lemma cpow1x a: \1c ^c  a = \1c.
Lemma cpow0x a: a <> \0c ->\0c ^c a = \0c.
Lemma cpowx2 a: a ^c \2c = a *c a.
```

The final result in this section is Proposition 12 [3, p. 164], it states that the cardinal of the power set of X is 2^X (for each subset Y of X, we can consider the characteristic function, whose value is 0 on Y and 1 elsewhere).

```
Lemma card_powerset X:
  cardinal (powerset X) = \2c ^c  X. (* 73 *)
```

4.7 Order relation and operations on cardinals

We shall write $a - b$ for the cardinal of the complement of b in a. This operation will be studied later on.

```
Definition card_diff a b := cardinal (complement a b).
Notation "x -c y" := (card_diff x y) (at level 50).
```

Proposition 13 [3, p. 164] states that $\mathfrak{a} \geq \mathfrak{b}$ if and only if there exists $\mathfrak{c}$ such that $\mathfrak{a} = \mathfrak{b} + \mathfrak{c}$ (for simplicity, we assume all three quantities to be cardinals, although $\mathfrak{c}$ could be any set, and both relations $\mathfrak{a} \geq \mathfrak{b}$ and $\mathfrak{a} = \mathfrak{b} + \mathfrak{c}$ imply that $\mathfrak{a}$ is a cardinal).

Proof. Assume B equipotent to a subset X of A and let C be the complement. Then B + C is equipotent to $B_1 \cup C_2$ (where $B_1 = B \times \{1\}$). We can replace B by X, then omit the indices, so that B + C is equipotent to A. Conversely, if A is equipotent $B_1 \cup C_2$, there is a bijection $B_1 \cup C_2 \to A$, and by restriction, an injection $B_1 \to A$, and by composition, an injection $B \to A$.

Using von Neumann cardinals simplifies the proof: let c be the cardinal of the complement of b in a. If a and b are cardinals and $b \leq a$ then $b \subset a$ and $a = b + c$. This is also $a = b + (a - b)$.

```
Lemma cardinal_complement A E: sub A E ->
  (cardinal A) +c (E -c A)  = cardinal E.
Lemma cardinal_complement1 a b: b <=c a ->
  b +c (a -c b) = a.
Theorem cardinal_le_when_complement a b:
  is_cardinal a -> is_cardinal b ->
  ((b <=c a) <-> (exists c, is_cardinal c & b +c c = a)).
```

Proposition 14 [3, p. 165] says that if $\mathfrak{a}_\iota \geq \mathfrak{b}_\iota$ (for two families of cardinals) we have

$$\sum_{\iota\in I} \mathfrak{a}_\iota \geq \sum_{\iota\in I} \mathfrak{b}_\iota, \qquad \prod_{\iota\in I} \mathfrak{a}_\iota \geq \prod_{\iota\in I} \mathfrak{b}_\iota. \tag{12}$$

The first formula is shown as follows. We have a bijection from $\mathfrak{b}_\iota$ into a subset E_ι of $\mathfrak{a}_\iota$, hence a bijection from $\mathfrak{b}_\iota \times \{\iota\}$ into a subset $E_\iota \times \{\iota\}$ of $\mathfrak{a}_\iota \times \{\iota\}$. This gives a bijection from the disjoint union $\bigcup \mathfrak{b}_\iota \times \{\iota\}$ into a subset $\bigcup E_\iota \times \{\iota\}$ of $\bigcup \mathfrak{a}_\iota \times \{\iota\}$. The proof of the second formula is similar: we get a bijection from $\prod \mathfrak{b}_\iota$ into a subset $\prod E_\iota$ of $\prod \mathfrak{a}_\iota$.

As a corollary, we obtain a smaller result if we restrict the domain of the sum or the product; in the case of a product, we assume all factors nonzero (proof: missing terms are replaced by zero, or one). The power is increasing with respect to both arguments. [Note: using von Neumann cardinals simplifies the proof, since in the first three cases, we must show Card(A) ≤ Card(B) and A ⊂ B is trivial].

```
Theorem csum_increasing f g:
  fgraph f -> fgraph g -> domain f = domain g ->
  (forall x, inc x (domain f) -> (V x f) <=c (V x g)) ->
  (card_sum f) <=c (card_sum g).
Theorem cprod_increasing f g:
  fgraph f -> fgraph g -> domain f = domain g ->
  (forall x, inc x (domain f) -> (V x f) <=c (V x g)) ->
  (card_prod f) <=c (card_prod g).

Lemma csum_increasing1 f j: cardinal_fam f ->
  sub j (domain f) -> (card_sum (restr f j)) <=c (card_sum f).
Lemma cprod_increasing1 f j: cardinal_fam f ->
  (forall x, inc x (domain f) -> V x f <> \0c) ->
  sub j (domain f) -> (card_prod (restr f j)) <=c (card_prod f).
Lemma csum_Mlele a b a' b':
  a <=c a' -> b <=c b' -> (a +c b) <=c (a' +c b').
Lemma cprod_Mlele a b a' b':
  a <=c a' -> b <=c b' -> (a *c b) <=c (a' *c b').
Lemma csum_M0le a b: is_cardinal a ->is_cardinal b ->
  a <=c (a +c b).
Lemma cprod_M1le a b: is_cardinal a ->is_cardinal b ->
  b <> \0c -> a <=c (a *c b).
```

```
Lemma cpow_Mlele a b a' b':
  a <> \0c -> a <=c a' -> b <=c b' ->  (a ^c b) <=c (a' ^c b').
Lemma cpow_Mleeq x y z: x <=c y  ->  x <> \0c -> x ^c  z <=c y ^c z.
Lemma cpow_Meqle x a b:
  x <> \0c -> (cardinal a) <=c (cardinal b) ->
  x ^c a <=c x ^c b.
Lemma cpow_Meqle0 x a b: x <> \0c -> a <=c b -> x ^c a <=c x ^c b.
Lemma cpow_Mle1 a b:
  is_cardinal a -> a <>  \0c -> b <> \0c  -> a <=c (a ^c b).
```

To conclude this chapter, we prove Cantor's theorem (Theorem 2, [3, p. 165]) stating that $2^{\mathfrak{a}} > \mathfrak{a}$ for every cardinal $\mathfrak{a}$, so that there is no set containing all cardinals.

```
Theorem cantor a: is_cardinal a ->
  a <c (\2c ^c  a).
Lemma cantor_bis: ~ (exists a, forall x, is_cardinal x -> inc x a).
Lemma cpow_M2le x y: x <=c y -> \2c ^c x <=c \2c ^c y.
Lemma infinite_powerset x: infinite_c x -> infinite_c (\2c ^c x).
```

Chapter 5

Natural integers. Finite sets

Bourbaki makes a distinction between finite and infinite cardinals. Finite cardinals are identified with *natural integers*, which are entities satisfying some arithmetic properties (addition, multiplication, subtraction and division are studied in the next chapter) derived from an induction principle and a successor function. There is a set **N** containing all finite cardinals, so that we have statements of the form: if $n \in \mathbf{N}$ then $n \neq n+1$, instead of: if $\mathfrak{a}$ is an infinite cardinal then $\mathfrak{a} = \mathfrak{a} + 1$. In the Bourbaki theory, a cardinal is a set: one could try to prove $1+1=2$ by showing that $x \in 1+1$ is equivalent to $x \in 2$. However since 2 is constructed via the axiom of choice, the statement $1 \in 2$ is unprovable (all we we know is that 2 is a set with two distinct elements). For this reason, natural integers are sometimes considered as urelements (objects that may appear at the right-hand side of $\in$, but never the left-hand-side). In Version 4 of this document, a natural integer will be a finite von Neumann ordinal. This gives an explicit form for integers (for instance 2 is $\{\emptyset, \{\emptyset\}\}$). This form is however not adapted to computations (there is no explicit form of the sum of two ordinals, and the ordinal sum of two cardinals is not always a cardinal).

Integers are presented in [8] as follows. There is a symbol O and a symbol S, and two operations $a+b$ and $a \cdot b$ (sum and product), defined on integers, which are a finite (maybe empty) sequences of letters S followed by a single O. The five axioms are

Axiom 1 $\forall a, \mathrm{S}a \neq \mathrm{O}$.
Axiom 2 $\forall a, a + \mathrm{O} = a$.
Axiom 3 $\forall a \forall b, a + \mathrm{S}b = \mathrm{S}(a+b)$.
Axiom 4 $\forall a, a \cdot \mathrm{O} = \mathrm{O}$.
Axiom 5 $\forall a \forall b, a \cdot \mathrm{S}b = (a \cdot b) + a$.

The first axiom has an unusual form, since most axioms are of the form $a \implies b = c$. This axiom is built-in in Coq: an object of a type with n constructors is defined by a single constructor: an integer is either O or Sa, but nor both. This means that if c an integer, one and only one of axioms 2 and 3 apply to $a+c$.

The axiom implies injectivity of S (by induction on the size of the arguments). Note that Coq defines addition an multiplication by induction on the first argument.

In the system presented above, it is impossible to prove $\forall a, a = \mathrm{O} + a$, although the result is obvious for any a. Thus a new principle is needed. It says something like: "If all the strings in a pyramidal family are theorems, then so is the universally quantified string which summarizes them". (We get a pyramid if we center the statements $a = \mathrm{O} + a$, for consecutive values of a). The whole pyramid has an infinite number of statements, and proving it requires an infinite proof. Assume that each line can be shown from the previous one, using exactly the same argument. Then the proof has the form P and Q and Q and Q, etc. It is infinite, but not

too much, hence is accepted. The induction principle is: "Suppose u is a variable, and $X\{u\}$ is a well-formed formula in which u occurs free. If both $\forall u : \langle X\{u\} \supset X\{Su/u\}\rangle$ and $X\{0/u\}$ are theorems, then $\forall u : X\{u\}$ is also a theorem." This is built-in in Coq, under the form

```
nat_ind =
fun P : nat -> Prop => nat_rect P
     : forall P : nat -> Prop,
       P 0 -> (forall n : nat, P n -> P (S n)) -> forall n : nat, P n.
```

In this chapter we shall prove that the Bourbaki integers satisfy the induction principle, under the form

```
Lemma cardinal_c_induction: forall r:Set -> Prop,
  (r card_zero) -> (forall n, inc n  Bnat-> r n -> r (succ n))
  ->  (forall n, inc n Bnat -> r n).
```

and as a consequence, that all these definitions are essentially the same. The proof of the principle is as follows: the least element of the set (assumed non-empty) of elements not satisfying a property is either O or Sa. This is a consequence of the fact that **N** is well-ordered. Note that the property shown by induction (X, P, r, in the examples) is quantified in Coq, but neither in [8] nor in Bourbaki.

An important property of integers is the possibility of defining a function by induction. This is a Coq example

```
Fixpoint add (n m:nat) {struct n} : nat :=
   match n with
   | O => m
   | S p => S (add p m)
   end.
```

This definition says that the source is $\mathbb{N}\times\mathbb{N}$, induction is on n, and the result is of type $\mathbb{N}$. By induction, there is at most one function satisfying Axioms 2 and 3. In Bourbaki, one could define, for each m, a function $f_m : \mathbf{N} \to E_m$, which is the unique surjective function satisfying the two axioms, show that $E_m \subset \mathbf{N}$, extend the function $f'_m : \mathbf{N} \to \mathbf{N}$, then merge all these functions to get $f : \mathbf{N}\times\mathbf{N} \to \mathbf{N}$. This function cannot be defined without first showing the existence of the set of integers. It is however possible to prove by induction that addition satisfies the two axioms (if one replaces "$n \in \mathbf{N}$" by "n is a finite cardinal" in the induction principle). Obviously, definition by induction is a particular case of definition by transfinite induction. Exercise 2.17 defines sum and product, exercise 2.18 defines exponentiation of ordinals. The situation is simpler: since there is no set containing all ordinals, what has to be defined is not a function; it is called an "ordinal functional symbol", this is something that associates an ordinal to a pair of ordinals.

If we define 'succ x' as $x+1$ for every cardinal x, we can define by induction the following function

```
Fixpoint nat_to_B (n:nat) :=
  match n with 0 => card_zero | S m => succ (nat_to_B m) end.
```

then define the set **N** as the image of this function. One can then show that **N** is the set of all finite integers. In a version 2 this document, this isomorphism between $\mathbb{N}$ and **N** was used to convert theorems proved in the standard library of Coq into theorems about finite cardinals. This identification has been removed in Version 3.

The function S on $\mathbb{N}$ has the following properties: it is injective; 0 is not in the range, and every non-zero element is in the range. Thus S is a bijection between $\mathbb{N}$ and $\mathbb{N}-\{0\}$. We use here two features of the axiom system of Carlos Simpson. The first is that there is a mapping $\mathscr{R}$ such that $\mathscr{R}x$ is a set, for any natural number x. The second feature (General Scheme of Replacement) is that there is a set containing all $(\mathscr{R}x, \mathscr{R}\mathrm{S}x)$ for $x:\mathbb{N}$. This set is easily seen to be the graph of a bijection (in the Bourbaki sense) between $\mathbb{N}$ and $\mathbb{N}-\{0\}$. The function $\mathscr{R}$ and the General Scheme of Replacement have been used a lot in the first part of this report in order to prove existence of some sets (union, product, powerset, etc). We do not use it here except for a single purpose: to show that there exists an infinite set. This avoids introducing an axiom asserting the existence of an infinite set. Note also that applying the replacement scheme to the function nat_to_B yields a set containing zero and stable by succ; this set has to be infinite, by injectivity of succ. This is another way of proving existence of infinite sets, without using $\mathscr{R}$, but it presupposes that defining functions by induction in Coq is compatible with the Bourbaki theory. In Version 4, the definition has been moved to the previous chapter.

5.1 Definition of integers

Recall that x is *finite* if it is an ordinal not equipotent to its successor x^+. A finite ordinal is a finite cardinal. We know that x is finite if and only if $x <_{\mathrm{ord}} \omega$, and this is the same as $x \in \omega$. We shall sometimes write **N**, or Bnat in the Coq code, instead of ω. A finite cardinal will be called an *integer*,

We say that x is an infinite ordinal (resp. infinite cardinal) if it is an ordinal (resp. cardinal) that is not finite. Let $\bar{x}$ be the cardinal successor of x; this is $\mathrm{Card}(x^+)$. Assume x finite; then $\bar{x} = x^+$, and x^+ is finite. Assume that x is an infinite cardinal; then $\bar{x} = x$ so that $\bar{x}$ is infinite. We state: (Proposition 1 [3, p. 166]): a cardinal x is finite if and only if its successor is finite.

```
Definition Bnat := \omega.
Lemma inc_Bnat a: inc a Bnat <-> finite_c a.

Lemma CS_succ a: is_cardinal (succ a).
Lemma card_succ_pr5 a: cardinal (succ a) = succ a.
Theorem is_finite_succ x:  is_cardinal x ->
  (finite_c x <-> finite_c (succ x)).
```

The following lemmas are trivial.

```
Lemma BS_succ x: inc x Bnat -> inc (succ x) Bnat.
Lemma CS_Bnat x: inc x Bnat -> is_cardinal x.
Lemma Bnat_cardinal x: inc x Bnat -> cardinal x = x.
Lemma bsum0r x: inc x Bnat -> x +c \0c = x.
Lemma bsum0l x: inc x Bnat -> \0c +c x = x.
Lemma BS_le_int a b: a <=c b -> inc b Bnat ->  inc a Bnat.
Lemma Bsucc_rw x: inc x Bnat ->  succ x = x +c \1c
Lemma Bnat_dichot x: is_cardinal x -> inc x Bnat \/ infinite_c x.
Lemma Bnat_le_infinite a b: inc a Bnat -> infinite_c b -> a <=c b.
```

5.2 Inequalities between integers

The successors of zero are one, two, three and four. We list some trivial properties. We show here that $2n = n + n$, hence $2 + 2 = 2.2 = 4$ and $2^4 = 4^2$.

```
Definition card_three := succ card_two.
Definition card_four := succ card_three.
Notation "\3c" := card_three.
Notation "\4c" := card_four.


Lemma succ_zero: succ \0c = \1c.
Lemma succ_one: succ \1c  = \2c.
Lemma BS0: inc \0c Bnat.
Lemma BS1: inc \1c Bnat.
Lemma BS2: inc \2c Bnat.
Lemma BS3: inc \3c Bnat.
Lemma BS4: inc \4c Bnat.

Lemma two_plus_two:  \2c +c \2c = \4c.
Lemma two_times_two:  \2c *c \2c = \4c.
Lemma power_2_4:  \2c ^c \4c = \4c ^c \2c.
```

Proposition 2 [3, p. 166] says that if $\mathfrak{a}$ is a cardinal and n an integer, if $\mathfrak{a} \leq n$ then $\mathfrak{a}$ is an integer. If n is an integer and $n \neq 0$, then there is a unique integer m such that $n = m + 1$. In this case $a < n$ is equivalent to $a \leq m$.

We consider here the first statement. If $\mathfrak{a} = \mathfrak{a} + 1$ then $(\mathfrak{a} + \mathfrak{b}) + 1 = \mathfrak{a} + \mathfrak{b}$ (by associativity and commutativity). Thus, if $\mathfrak{a} + \mathfrak{b}$ is finite so is $\mathfrak{a}$. Now, if $\mathfrak{a} \leq n$ there is $\mathfrak{b}$ such that $\mathfrak{a} + \mathfrak{b} = n$. Bourbaki concludes that if n is finite, then $\mathfrak{a}$ is finite also. Later one, Bourbaki shows that the quantity $\mathfrak{b}$ introduced above is an integer, is unique (and is called the difference).

We have already proved this statement in the form: if $a \leq n$ and $n < \omega$ then $a < \omega$. We just state: if $a + b \in \mathbf{N}$; then a and b are integers. If a is an integer and $b \leq a$, then $a = b + (a - b)$ so that $a - b$ is an integer.

Assume $\mathfrak{b}$ non-zero, so that is $\mathrm{Card}(\mathfrak{b}) = \mathfrak{c}+1$ for some $\mathfrak{c}$. Then $\mathfrak{ab} = \mathfrak{ac}+\mathfrak{a}$. If this expression is finite, so is $\mathfrak{a}$.

```
Lemma card_le_succ0 a: is_cardinal a ->
  a <=c (succ a).
Lemma Bnat_in_sum a b: is_cardinal b ->
  inc (a +c b) Bnat ->inc b Bnat.
Lemma Bnat_in_sum2 a b: is_cardinal a ->
  inc (a +c b) Bnat ->inc a Bnat.
Lemma Bnat_in_product a b: is_cardinal a ->
  b <> \0c -> inc (a *c b) Bnat -> inc a Bnat.
Lemma BS_diff0 a b: inc a Bnat ->
  b <=c a -> inc (a -c b) Bnat.
Lemma BS_nsucc x: is_cardinal x -> inc (succ x) Bnat -> inc x Bnat.
```

Consider now the second statement of the proposition: if n is a non-zero integer, there is a unique m such that $n = m+1$. Uniqueness follows from injectivity of the successor function. If a cardinal is a von Neumann ordinal, we can consider the ordinal predecessor. This gives an explicit form of the predecessor.

```
Definition cpred := union.
Lemma cpred_pr n: inc n Bnat -> n <> \0c ->
  (inc (cpred n) Bnat &  n = succ (cpred n)).
```

We have $a < b$ iff $a+1 \leq b$ and similar relations. Note that for any cardinal a, the successor of the predecessor of a is a (if a is infinite, the successor and the predecessor of a are a).

```
Lemma card_le_succ a: inc a Bnat -> a <=c (succ a).
Lemma card_lt_succ n: inc n Bnat -> n <c (succ n).
Lemma cdiff_nz a b: inc a Bnat -> b <c a -> (a -c b ) <> \0c.
Theorem card_lt_succ_le a n: inc n Bnat ->
  (a <c (succ n) <-> a <=c n).
Lemma card_le_succ_succ a b: is_cardinal a -> inc b Bnat ->
  (a <=c b <-> succ a <=c succ b).
Lemma succ_nz  n: succ n <> \0c.
Lemma card_comp_zero_one i: i <c \1c <-> (i = \0c).
Lemma cpred_pr1 n: is_cardinal n -> cpred (succ n) = n.
Lemma cpred_pr2 n: inc n Bnat -> cpred (succ n) = n.
Lemma cpred_pr3 n: inc n Bnat ->
  n = card_zero \/ exists m, inc m Bnat & n = succ m.
Lemma succ_positive a: \0c <c  (succ a).
Lemma cardinal_lt01: \0c <c \1c.
Lemma cardinal_le01: \0c <=c \1c.
Lemma cardinal_lt12: \1c <c \2c.
Lemma cardinal_lt02: \0c <c \2c.
Lemma less_than_two a: a <c \2c ->  a = \0c \/ a = \1c.
Lemma card_le_succ_lt0 a b: is_cardinal a -> finite_c b ->
  (succ a <=c b <-> a <c b).
Lemma card_le_succ_lt a b: inc a Bnat -> inc b Bnat ->
  (succ a <=c b <-> a <c b).
Lemma ctwo_small x: is_cardinal x ->  x <> \0c ->  x <> \1c  -> \2c <=c x.
```

As a corollary, every subset of a finite set is finite. If $X \subset Y$, $X \neq Y$ and Y is finite, then $\mathrm{Card}(X) < \mathrm{Card}(Y)$. Bourbaki says that the converse is true by definition. In fact, if Y is empty, it is finite, otherwise there is $x \in Y$, and if X is the complement of $\{x\}$ in Y, then $\mathrm{Card}(Y) = \mathrm{Card}(X) + 1$. The relation $\mathrm{Card}(X) < \mathrm{Card}(Y)$ implies that Card(X) is finite, hence $\mathrm{Card}(X) + 1$ is also finite.

```
Lemma card_finite_set x: finite_set x <-> inc (cardinal x) Bnat.

Lemma sub_finite_set x y: sub x y -> finite_set y ->
  finite_set x.
Lemma strict_sub_smaller x y: stricy_sub x y -> finite_set y ->
   (cardinal x) <c (cardinal y).
Lemma emptyset_finite: finite_set (emptyset).
Lemma strict_sub_smaller1 y:
  (forall x, strict_sub x y -> (cardinal x) <c (cardinal y)) ->
  finite_set y.
```

The image of a finite set by a function is finite. We give some variants of this property. Consider two sets E and F with the same cardinal, and a function $f : E \to F$. Assume E finite. If f is injective and not surjective, then $f(E)$ is a strict subset of F equipotent to F, thus cannot be finite. Hence we claim: if f injective, then f is bijective. Assume f surjective. It has a right inverse, which is injective, so that f is also bijective.

```
Lemma finite_image f: is_function f -> finite_set (source f) ->
  finite_set (image_of_fun f).
Lemma finite_image_by f A: is_function f ->  sub A (source f) ->
  finite_set A -> finite_set (image_by_fun f A).
Lemma finite_fun_image a f: finite_set a ->
  finite_set  (fun_image a f).
Lemma finite_range f: fgraph f -> finite_set(domain f) ->
  finite_set(range f).
Lemma finite_graph_domain f: fgraph f ->
  finite_set f -> finite_set (domain f).
Lemma finite_graph_range f: fgraph f ->
  finite_set f -> finite_set (range f).
Lemma finite_domain_graph f: fgraph f ->
  finite_set (domain f) -> finite_set f.
Lemma bijective_if_same_finite_c_inj f:
  cardinal (source f) = cardinal (target f) -> finite_set (source f) ->
  injection f -> bijection f.
Lemma sub_image_of_fun f x: is_function f -> sub x (source f) ->
  sub (image_by_fun f x) (image_of_fun f).
Lemma bijective_if_same_finite_c_surj f:
  cardinal (source f) = cardinal (target f) -> finite_set (source f) ->
  surjection f -> bijection f.
```

5.3 The set of natural integers

As explained as the start of the chapter, we introduced the set of integers as early as possible (Bourbaki introduced it very lately). This short section contain only one simple result, namely that **N** can be well-ordered by the relation "$x \in \mathbf{N}$ and $y \in \mathbf{N}$ and $x \leq_{\mathrm{Card}} y$" denoted $x \leq_{\mathbf{N}} y$. Thus every non-empty subset of **N** has a least element.

```
Definition Bnat_order := graph_on cardinal_le Bnat.
Definition Bnat_le x y := inc x Bnat & inc y Bnat & x <=c y.
Definition Bnat_lt x y := Bnat_le x y & x<>y.
Notation "x <=N y" := (Bnat_le x y) (at level 60).
Notation "x <N y" := (Bnat_lt x y) (at level 60).

Lemma Bnat_order_sr: substrate Bnat_order = Bnat.
Lemma Bnat_order_wor: worder Bnat_order.
Lemma Bnat_order_le x y:
  gle Bnat_order x y <-> x <=N y.

Lemma Bnat_wordered X: sub X Bnat -> nonempty X ->
  inc \0c X \/
  (exists a, inc a Bnat & inc (succ a) X & ~ (inc a X)).
```

We consider some properties of intervals (there will be more of them in the next chapter).

```
Lemma Bnat_interval_cc_pr a b x: inc a Bnat -> inc b Bnat ->
  (inc x (interval_cc Bnat_order a b) <-> (a <=N x & x <=N b)).
Lemma Bnat_interval_co_pr a b x: inc a Bnat -> inc b Bnat ->
  (inc x (interval_co Bnat_order a b) <-> (a <=N x & x <N b)).
Lemma Bnat_interval_cc_pr1 a b x: inc a Bnat -> inc b Bnat ->
  (inc x (interval_cc Bnat_order a b) <-> (a <=c x & x <=c b)).
Lemma Bnat_interval_co_pr1 a b x: inc a Bnat -> inc b Bnat ->
  (inc x (interval_co Bnat_order a b) <-> (a <=c x & x <c b)).
```

5.4 The principle of induction

Bourbaki states the principle of induction, the Criterion C61, in the following form: *Let* R$\{n\}$ *be a relation in a theory* $\mathcal{T}$ *(where* n *is not a constant of* $\mathcal{T}$*). Suppose that the relation*

$$\mathrm{R}\{0\} \text{ and } (\forall n)((n \text{ is an integer and } \mathrm{R}\{n\}) \implies \mathrm{R}\{n+1\})$$

is a theorem in $\mathcal{T}$*. Under these conditions, the relation*

$$(\forall n)((n \text{ is an integer }) \implies \mathrm{R}\{n\})$$

is a theorem in $\mathcal{T}$*.*

The proof is by contradiction. Assume the result false for some n, and consider the least element m of the set of all integers $\leq n$ that do not satisfy p, (it exists, since the set is non-empty and is well-ordered). Our proof is similar (with "the set of all integers $\leq n$ that do not satisfy p" replaced by "the set of integers that do not satisfy p").

We give four variants of the principle. Let S(n) be the relation: n is an integer and R is true for all integers $p < n$. If S implies R, then R is true for all integers.

If R(a) is true, and if R(n) together with $a \leq n$ (respectively $a \leq n < b$) implies R($n+1$), then for all n such that $a \leq n$ (respectively $a \leq n \leq b$), the relation R is true. In both cases n must be an integer (in the second case, we assume b integer, so that $n < b$ or $n \leq b$ implies that n is an integer). Bourbaki uses induction on $a \leq n < b \implies \mathrm{R}(n)$, but it is simpler to use $a \leq n \leq b \implies \mathrm{R}(n)$.

The last variant is: if $a \leq n < b$ and R($n+1$) implies R(n), if moreover R(b) is true, then $a \leq n \leq b$ implies R(n). The proof is by induction on P = ¬R. If R is false for some n with $a \leq n \leq b$ then P is true for some c with $a \leq c < b$. On the other hand, if $a \leq n < b$ then P(n) implies P($n+1$).

```
Lemma cardinal_c_induction (r:Set -> Prop):
  (r \0c) -> (forall n, inc n  Bnat-> r n -> r (succ n))
  ->  (forall n, inc n Bnat -> r n).

Lemma cardinal_c_induction1 (r:Set -> Prop):
  let s:= fun n => forall p, inc n Bnat -> inc p Bnat ->
    p <c n -> r p in
  (forall n, inc n Bnat -> s n -> r n) ->
  (forall n, inc n Bnat -> r n).
Lemma cardinal_c_induction2 (r:Set -> Prop) k:
  inc k Bnat -> r k ->
  (forall n, inc n Bnat -> k <=c n -> r n -> r (succ n))
  ->  (forall n, inc  n Bnat  -> k <=c n -> r n).
Lemma cardinal_c_induction3 (r:Set -> Prop) a b:
  inc a  Bnat -> inc b Bnat ->  r a ->
  (forall n, a <=c n -> n <c b -> r n -> r (succ n))
  ->  (forall n, a <=c n  -> n <=c b -> r n).
Lemma cardinal_c_induction4: (r:Set -> Prop) a b:
  inc a Bnat -> inc b Bnat ->  r b ->
  (forall n, a <=c n -> n <c b -> r (succ n) -> r n)
  ->  (forall n, a <=c n  -> n <=c b -> r n).
```

We rewrite our induction principle variants 3 and 4, replacing $a \leq n \leq b$ by $n \in [a, b]$.

```
Lemma cardinal_c_induction3_v (r:Set -> Prop) a b:
  inc a Bnat -> inc b Bnat ->  r a ->
  (forall n, inc n (interval_co Bnat_order a b) -> r n -> r (succ n))
  ->  (forall n, inc n (interval_cc Bnat_order a b) -> r n).
Lemma cardinal_c_induction4_v (r:Set -> Prop) a b:
  inc a Bnat -> inc b Bnat ->  r b ->
  (forall n, inc n (interval_co Bnat_order a b) -> r (succ n) -> r n)
  ->  (forall n, inc n (interval_cc Bnat_order a b) -> r n).
```

The empty set is finite, and if X is finite then $X \cup \{x\}$ is finite. We then show a partial converse, that will be useful for induction on finite sets. If X has cardinal zero, it is the empty set, and if X has cardinal $n+1$, it is of the form $X' \cup \{x\}$, where X' has cardinal n.

```
Lemma tack_on_finite X x:
  finite_set X -> finite_set(tack_on X x).
Lemma singleton_finite x: finite_set(singleton x).
Lemma doubleton_finite x y: finite_set(doubleton x y).
Lemma tack_if_succ_card x n: is_cardinal n -> cardinal x = succ n ->
  exists u, exists v, x = tack_on u v & ~(inc v u) & cardinal u = n.
```

The induction principle on finite sets is now: If a property P is true for the empty set, if P(a) implies P($a \cup \{b\}$), then P is true for every finite set. In general P has the form: if A then B. Note: if $b \in a$, then $a \cup \{b\} = a$, and we have a version where we add the condition $b \notin a$.

```
Lemma finite_set_induction0 (s:Set -> Prop):
  s emptyset -> (forall a b, s (a) ->  ~(inc b a) -> s (tack_on a b)) ->
  forall x, finite_set x -> s x.
Lemma finite_set_induction (s:Set -> Prop):
  s emptyset -> (forall a b, s (a) -> s (tack_on a b)) ->
  forall x, finite_set x -> s x.
Lemma finite_set_induction1 (A B:Set -> Prop) x:
  (A emptyset -> B emptyset)
  -> (forall a b, (A a -> B a) -> A(tack_on a b) -> B(tack_on a b))
  -> finite_set x -> A x -> B x.
```

In some cases P is false for the empty set. If P is true for all singletons, then P is true for every non-empty finite set.

```
Lemma finite_set_induction2 (A B:Set -> Prop) x:
  (forall a, A (singleton a) -> B (singleton a))
  -> (forall a b, (A a -> nonempty a -> B a) ->
    nonempty a -> A(tack_on a b) -> B(tack_on a b))
  -> finite_set x -> A x -> nonempty x -> B x.
```

If $s(x)$ is the successor of x, we have $a + s(b) = s(a+b)$ and $a \cdot s(b) = a \cdot b + a$, $a^{s(b)} = a^b.a$ We deduce by induction that **N** is stable by addition, multiplication and power.

```
Lemma csum_via_succ a b: inc b Bnat ->
  a +c (succ b) = succ (a +c b).
Lemma csum_via_succ1 a b: inc a Bnat ->
   (succ a) +c b = succ (a +c b).
Lemma cprod_via_sum a b: inc a Bnat -> inc b Bnat ->
  a *c (succ b) = (a *c b) +c a.
```

```
Lemma pow_succ a b: inc b Bnat ->
  a ^c (succ b) = (a ^c b) *c a.

Lemma BS_sum a b: inc a Bnat -> inc b Bnat ->
  inc (a +c b) Bnat.
Lemma BS_prod a b: inc a Bnat -> inc b Bnat ->
  inc (a *c b) Bnat.
Lemma BS_pow a b: inc a Bnat -> inc b Bnat ->
  inc (a ^c b) Bnat.
Lemma BS_pow2 n: inc n Bnat -> inc (\2c ^c n) Bnat.
```

5.5 Finite subsets of ordered sets

Let $\leq$ be an order relation on a set E that makes it a directed set, a lattice, or a totally ordered set; and let X be a finite non-empty subset of E. Then X has an upper bound, or has a least upper bound and a greatest lower bound, or has a least and greatest element respectively (Proposition 3, [3, p. 170]). We have to show that there is an x such that $P(x,X)$. By assumption, this is true if X is a doubleton (therefore, if X is a singleton). If $X = Y \cup \{b\}$ and $P(a,X)$ we have to show the property for the doubleton $\{a,b\}$.

```
Lemma finite_set_induction3 (p:Set -> Set -> Prop) E X:
  (forall a b, inc a E -> inc b E -> exists y, p (doubleton a b) y) ->
  (forall a b x y, sub a E -> inc b E -> p a x -> p (doubleton x b) y->
    p (tack_on a b) y) ->
  (forall X x, sub X E -> nonempty X -> p X x -> inc x E) ->
  nonempty X -> finite_set X -> sub X E -> exists x, p X x.

Lemma finite_subset_directed_bounded r X:
  right_directed r ->finite_set X -> sub X (substrate r) -> nonempty X
  -> bounded_above r X.
Lemma finite_subset_lattice_inf r X:
  lattice r ->finite_set X -> sub X (substrate r) -> nonempty X
  -> exists x, greatest_lower_bound r X x.  (* 20 *)
Lemma finite_subset_lattice_sup r X:
  lattice r ->finite_set X -> sub X (substrate r) -> nonempty X
  -> exists x, least_upper_bound r X x.  (* 20 *)
Lemma finite_subset_torder_greatest r X:
  total_order r ->finite_set X -> sub X (substrate r) -> nonempty X
  -> exists x, greatest_element (induced_order r X) x. (* 25 *)
Lemma finite_subset_torder_least r X:
  total_order r ->finite_set X -> sub X (substrate r) -> nonempty X
  -> exists x, least_element (induced_order r X) x.
```

Some consequences. A nonempty finite set[1] has a maximal element, and if totally ordered, has a greatest element. A finite totally ordered set is well-ordered.

```
Lemma finite_set_torder_greatest r:
  total_order r ->finite_set (substrate r) -> nonempty (substrate r)
  -> exists x, greatest_element r x.
Lemma finite_set_torder_wor r:
  total_order r ->finite_set (substrate r) -> worder r.
```

[1] The word "nonempty" is missing in Bourbaki

```
Lemma finite_set_maximal r:
  order r ->finite_set (substrate r) -> nonempty (substrate r) ->
  exists x, maximal_element r x.
```

5.6 Properties of finite character

If E is a set, a property P{X} (where X is a subset of E) is said to be of *finite character* if the set $\mathfrak{S}$ of all X satisfying P is of finite character; this means $X \in \mathfrak{S}$ if and only if every finite subset Y of X satisfies $Y \in \mathfrak{S}$. Example: the set of totally ordered subsets of an ordered set. Theorem 1 [3, p. 171] states: Every nonempty[2] set $\mathfrak{S}$ of subsets of a set E which is of finite character has a maximal element (when ordered by inclusion).

```
Definition of_finite_character s:=
  forall x, (inc x s) <-> (forall y, (sub y x & finite_set y) -> inc y s).

Lemma of_finite_character_example r: order r ->
  of_finite_character(Zo (powerset (substrate r)) (fun z =>
    total_order (induced_order r z))).
Lemma maximal_inclusion s: of_finite_character s -> nonempty s ->
  exists x, maximal_element (inclusion_suborder s) x. (* 39 *)
Lemma maximal_inclusion_aux: let s := emptyset in
  of_finite_character s &
  ~ (exists x, maximal_element (inclusion_suborder s) x).
```

¶ Define by induction on the type nat a function $n \to \bar{n}$ by $\bar{0} = 0$ and $\overline{n+1} = \bar{n}+1$ (the first $n+1$ is the successor on nat, the second is the cardinal successor. Then $\bar{n}$ is a natural number, and each natural number can be uniquely written in the form $\bar{n}$. This function respects operations (addition, multiplication, exponentiation).

```
Fixpoint nat_to_B (n:nat) :=
  if n is m.+1 then  succ (nat_to_B m) else  \0c.

Lemma nat_to_B_succ n:
  succ (nat_to_B n) = (nat_to_B n.+1).
Lemma nat_to_B_Bnat n: inc (nat_to_B n) Bnat.
Lemma nat_to_B_injective n m:
   nat_to_B n =  nat_to_B m -> n = m.
Lemma nat_to_B_surjective x: inc x Bnat -> exists n, x = nat_to_B n.
Lemma nat_to_B_sum x y: nat_to_B (x + y) = nat_to_B x +c nat_to_B y.
Lemma nat_to_B_prod x y: nat_to_B (x * y) = nat_to_B x *c nat_to_B y.
Lemma nat_to_B_pow x y: nat_to_B (x ^ y) = nat_to_B x ^c nat_to_B y.
```

¶ Let's show the following property: if x is an infinite cardinal, then $x.x = x$. We shall prove later on that $X \times X$ is equipotent to X, whenever X is an infinite set. These two properties are equivalent if the Axiom of Choice holds; the proof given here does not depend on the axiom of choice.

Given a pair of ordinals $x = (a, b)$ we define $\bar{x}$ to be the greatest of a and b. Let $x' = (\bar{x}, a, b)$. Define $x \leq^* y$ to be $x' \leq_{\text{lex}} y'$, where $\leq_{\text{lex}}$ compares the three components in lexicographic order. This is a well-ordering (the lexicographic product of three well-ordered sets is well-ordered). We call it the *canonical ordering of pairs of ordinals.* We give here a direct proof:

[2] The word "nonempty" is missing in Bourbaki

Let X be a non-empty set of pairs of ordinals. The set of all $\bar{x}$ (for $x \in X$) has a least element γ; the set of all $\mathrm{pr}_1 x$ with $\bar{x} = \gamma$, has a least element, say α; the set of all $\mathrm{pr}_2 x$ with $\bar{x} = \gamma$, and $\mathrm{pr}_1 x = \alpha$ has a least element; say β. Then (α, β) is the least element of X for $\leq^*$.

Let $p(x)$ be the property $\mathrm{Card}(x \times x) = x$. We show that p holds for any infinite cardinal by transfinite induction: we consider an infinite cardinal κ, assume $p(x)$ true for every infinite cardinal x such that $x < \kappa$, and deduce $p(\kappa)$. Note that $x \leq \mathrm{Card}(x \times x)$ is obvious.

Since $\leq^*$ is a well-ordering, it induces a well-ordering on $\kappa \times \kappa$, and we can consider its ordinal λ. This means that we have a bijection $f : \kappa \times \kappa -> \lambda$, such that $x \leq^* y$ if and only if $f(x) \leq_{\mathrm{ord}} f(y)$. Notice that, if $y \in \kappa$, $f(x) \in f(y)$ is equivalent to $f(x) <_{\mathrm{ord}} f(y)$, thus $x <^* y$. Let z be the greatest of the two components of y. Then $x <^* y$ implies that the two components of x are $\leq_{\mathrm{ord}} z$, thus $<_{\mathrm{ord}} z^+$, so that $x \in z^+ \times z^+$. This gives a bound on the cardinal of $f(y)$: let t be the cardinal of z^+. we have $\mathrm{Card}(f(y)) \leq t.t$. This is obviously $< \kappa$ if z is finite. On the other hand, κ is a limit ordinal, so that $z <_{\mathrm{ord}} \kappa$ implies $z^+ <_{\mathrm{ord}} \kappa$, so that $t < \kappa$, and $t^2 = t$. It follows: $\mathrm{Card}(f(y)) < \kappa$. Since $f(y)$ is an ordinal, it follows $f(y) <_{\mathrm{ord}} \kappa$. Thus $f(y) \in \kappa$; it follows $\lambda \subset \kappa$, and $\mathrm{Card}(\lambda) \leq \kappa$. SInce λ is equipotent to $\kappa \times \kappa$, the conclusion follows.

As a byproduct, we have: for any infinite cardinal X, there is a bijection $f : X \times X \to X$ so such that $x \leq^* y$ is equivalent to $f(x) \leq f(y)$.

```
Definition ordinal_pair x := is_pair x & is_ordinal (P x) & is_ordinal (Q x).
Definition ord_pair_le  x y:=
   ( ordinal_pair x & ordinal_pair y &
   (union2 (P x) (Q x) <o union2 (P y) (Q y)
    \/ (union2 (P x) (Q x) = union2 (P y) (Q y)
        & ((P x) <o (P y)
           \/ (P x = P y & Q x <=o Q y))))).

Lemma ordering_pair1 x: ordinal_pair x ->
   ((P x <=o Q x) &  (union2 (P x) (Q x) = Q x))
   \/ ((Q x <=o P x) &  (union2 (P x) (Q x) = P x)).
Lemma ordering_pair2 x: ordinal_pair x -> is_ordinal  (union2 (P x) (Q x)).
Lemma ordering_pair3 x y : ord_pair_le x y ->
  inc x (coarse (succ_o ((union2 (P y) (Q y))))).
Lemma ordering_pair4 x: ordinal_pair x -> ord_pair_le x x.

Lemma well_ordering_pair: worder_r ord_pair_le. (* 85 *)
Lemma infinite_product_aux k: infinite_c k ->  (* 80 *)
  (forall z,  infinite_c z -> z <o k -> z *c z = z) ->
  let lo:= graph_on ord_pair_le (product k k) in
  k *c k = k &
  exists f, function_prop f (product k k) k & bijection f &
     (forall x y, inc x (source f) -> inc y (source f) ->
       (glt lo x y <-> (W x f) <o (W y f))).
Lemma infinite_product_alt x : infinite_c x -> x *c x = x.
Lemma infinite_product_prop2 k: infinite_c k ->
  let lo:= graph_on ord_pair_le (product k k) in
  exists f, function_prop f (product k k) k & bijection f &
     (forall x y, inc x (source f) -> inc y (source f) ->
       (glt lo x y <-> (W x f) <o (W y f))).
```

Chapter 6

Properties of integers

6.1 Operations on integers and finite sets

By *operation* on a set E, one means a function $g : \mathrm{E} \times \mathrm{E} \to \mathrm{E}$, often denoted by an infix symbol such as $a + b$. There are some unary operations $\mathrm{E} \to \mathrm{E}$ such as x^+, the successor of x. The sum of two cardinals may be considered as an operation (but there is no set of cardinals). Binary operation may be generalized to more than two arguments. Given a list $x_1, x_2, \ldots, x_n$ of $n \geq 2$ terms, one can define a function F as follows

(a) $$\mathrm{F}(x_1, x_2, \ldots, x_n) = g(x_1, \mathrm{F}(x_2, \ldots, x_n)).$$

Note that if g maps $\mathrm{E} \times \mathrm{G}$ to G, and g_0 is some function $\mathrm{E} \to \mathrm{G}$, we can define $\mathrm{F}(x_1) = g_0(x_1)$, so that F maps a sequence of elements of E onto an element of G. Other variants are possible. In general, we define $\mathrm{F}(a) = a$, so that $\mathrm{F}(a, b) = g(a, b)$. We say that e is a unit of g if $g(e, x) = x$ whatever x. In this case, we may define $\mathrm{F}() = e$, $\mathrm{F}(x) = g(e, x)$, so that F is defined on $\mathrm{L}(\mathrm{E})$ the set of lists of E, otherwise, it is defined only on $\mathrm{L}_e(\mathrm{E})$, the set of non-empty lists of E. Here, we may identify a list with a function $[1, n] \to \mathrm{E}$, and a non-empty list corresponds to the case $n \neq 0$. If X is the function associated to the list $x_1, \ldots, x_n$ and X' the function associated to $x_2, \ldots, x_n$, we have $\mathrm{X}'(i) = \mathrm{X}(i + 1)$, and (a) says $\mathrm{F}(\mathrm{X}) = g(\mathrm{X}(1), \mathrm{F}(\mathrm{X}'))$. Every finite totally ordered set is uniquely order-isomorphic to an interval $[1, n]$. Thus, given a mapping $x : \mathrm{I} \to \mathrm{E}$, where I is finite and totally ordered, we may consider x as list; if l is the least element of I, and if x' is the restriction of x to $\mathrm{I} - \{l\}$, with the induced order, relation (a) states $\mathrm{F}(x) = g(x_l, \mathrm{F}(x'))$.

We say that g is associative if $g(a, g(b, c)) = g(g(a, b), c)$. This implies

(2) $$\mathrm{F}(x_1, x_2, \ldots, x_n) = g(\mathrm{F}(x_1, \ldots, x_{n-1}), x_n)$$

and in particular that, if x is as above, if g is the greatest element of I, x'' is the restriction of x to $\mathrm{I} - \{g\}$, with the induced order, we get $\mathrm{F}(x) = g(\mathrm{F}(x''), x_g)$. More generally, given any partition of the interval $[1, n]$ as $[1, n] = \bigcup \mathrm{X}_k$, where each X_k is non-empty, if $k < l$ implies $u < v$ whenever $u \in \mathrm{X}_k$ and $v \in \mathrm{X}_l$, if x'_k denotes the list of x_i with $i \in \mathrm{X}_k$ (with the induced ordering), and $x''_k = \mathrm{F}(x_k)$ then $\mathrm{F}(x) = \mathrm{F}(x'')$.

We say that g is commutative if $g(a, b) = g(b, a)$. If g is commutative and associative we have for instance $\mathrm{F}(a, b, c) = \mathrm{F}(b, a, c)$. More generally, $\mathrm{F}(x)$ becomes independent of the ordering of the elements of the list. The associativity theorem can be simplified: the condition "$k < l$ implies $u < v$ whenever $u \in \mathrm{X}_k$ and $v \in \mathrm{X}_l$" becomes unnecessary. If moreover e is a unit, the condition "X_k is non-empty" can be dropped as well.

In a previous version of our software, we introduced the notion of non-empty list, non-empty sequence, etc, so as to handle the case where there is no unit element (for instance, intersection of sets, infimum function on **N**, etc), see 11.6. The ssreflect library proposes a module bigops, that implements this kind of operations. In the example that follows, I is a finite type (a finite totally ordered set) and the operation is applied to all x_i where $i \in \mathrm{I}$ satisfies a predicate P. There is a function $p : \mathrm{I} \to \mathrm{J}$ that defines a partition $\mathrm{X}_k = p^{-1}(\{k\})$ of I. Let's compare the associativity theorem of bigops and the associativity of cardinals in Gaia:

```
(*
Lemma partition_big : forall (I J : finType) (P : pred I) p (Q : pred J) F,
    (forall i, P i -> Q (p i)) ->
      \big[*%M/1]_(i | P i) F i =
         \big[*%M/1]_(j | Q j) \big[*%M/1]_(i | P i && (p i == j)) F i.
Theorem cardinal_sum_assoc: forall f g,
  fgraph f  -> partition_fam g (domain f) ->
  cardinal_sum f  = cardinal_sum (L (domain g) (fun l =>
    cardinal_sum (restr f (V l g)))).
*)
```

Similarly, we may compare the commutativity theorems:

```
(*
Definition perm_eq (s1 s2 : seq T) := all (same_count1 s1 s2) (s1 ++ s2).
Lemma eq_big_perm : forall (I : eqType) r1 r2 (P : pred I) F,
    perm_eq r1 r2 ->
  \big[*%M/1]_(i <- r1 | P i) F i = \big[*%M/1]_(i <- r2 | P i) F i.

Theorem cardinal_sum_commutative: forall X f,
  fgraph X ->  target f = domain X -> bijective f ->
  cardinal_sum X = cardinal_sum (gcompose X (graph f)).
*)
```

There is a fundamental difference between the ssreflect theory, and the Bourbaki theorems proved so far. In one case, we start with a binary operation and extent it to finite lists of arguments, and in the other case, we start with an operation defined for many arguments, and study the case of two arguments. Note that a finite sum of integers is finite, but an infinite sum of integers is not an integer. One may define the sum and product of a finite sequence of ordinals; one can also define an infinite sum, but not an infinite product.

We start with some helper lemmas. The first two say that the sum or product of the restriction of a family $(x_i)_i$ to a singleton $\{j\}$ is x_j. The other two ones say that we have a partition of $\mathrm{X} = a \cup \{b\}$ formed of a and $\{b\}$ whenever $b \notin a$. For any X, if $b \in \mathrm{X}$, if $a = \mathrm{X} - \{b\}$ then $\mathrm{X} = a \cup \{b\}$.

```
Lemma Bsum_M0le a b: inc a Bnat -> inc b Bnat->
  a <=c (a +c b).
Lemma Bprod_M1le a b: inc a Bnat -> inc b Bnat->
  b <> \0c -> a <=c (a *c b).
Lemma Bnat_to_ell a b: inc a Bnat -> inc b Bnat ->
  a = b \/ a <c b \/ b<c  a.
Lemma Bnat_to_el a b: inc a Bnat -> inc b Bnat ->
  a <=c b \/ b <c a.
Lemma csum_trivial4 f a: fgraph f ->
  inc a (domain f) ->
```

```
  card_sum (restr f (singleton a))  = cardinal (V a f).
Lemma cprod_trivial4 f a: fgraph f ->
  inc a (domain f) ->
  card_prod (restr f (singleton a))  = cardinal (V a f).
Lemma partition_tack_on a b: ~ inc b a ->
  partition_fam (variantLc a (singleton b)) (tack_on a b).
Lemma partition_complement a b: inc b a ->
  partition_fam (variantLc (compl_singl a b) (singleton b)) a.
```

We show here

$$x_j + \sum_{i\in \mathrm{J}} x_i = \sum_{i\in \mathrm{J}\cup\{j\}} x_i, \tag{1}$$

$$x_j \cdot \prod_{i\in \mathrm{J}} x_i = \prod_{i\in \mathrm{J}\cup\{j\}} x_i, \tag{b}$$

whenever $\mathrm{J}\cup\{i\}$ is a subset of (or is equal to) the domain of the family x_i and $j \notin \mathrm{J}$. The proofs are similar; we have a partition of $\mathrm{J}\cup\{j\}$, and use the associativity formula, that says that the sum on the right hand side is a sum over the canonical doubleton, thus a sum of two terms.[1]

```
Lemma induction_sum0 f a b: fgraph f ->
  sub (tack_on a b) (domain f) -> (~ inc b a) ->
  card_sum (restr f (tack_on a b)) =
  card_sum (restr f a) +c (V b f).
Lemma induction_prod0 f a b: fgraph f ->
  sub (tack_on a b) (domain f) -> (~ inc b a) ->
  card_prod (restr f (tack_on a b)) =
  (card_prod (restr f a)) *c (V b f).
Lemma induction_sum1 f a b: fgraph f ->
  domain f = tack_on a b -> (~ inc b a) ->
  card_sum f = card_sum (restr f a) +c  (V b f).
Lemma induction_prod1 f a b: fgraph f ->
  domain f = tack_on a b -> (~ inc b a) ->
  card_prod f = card_prod (restr f a) *c (V b f).
```

Given a family of cardinals $\mathfrak{a}_i$ with sum S and product P we have $\mathfrak{a}_i \leq \mathrm{S}$ and $\mathfrak{a}_i \leq \mathrm{P}$, whenever all factors are non-zero.

```
Lemma csum_increasing6 f j: cardinal_fam f ->
  inc j (domain f) -> (V j f) <=c (card_sum f).
Lemma cprod_increasing6 f j: cardinal_fam f ->
  inc j (domain f) -> (V j f) <=c (card_prod f).
```

A *finite family of integers* is a functional graph $i \mapsto x_i$ where the index set I is finite and each x_i is finite.

```
Definition finite_int_fam f:=
  fgraph f  &
  (forall i, inc i (domain f) -> inc (V i f) Bnat) &
  finite_set (domain f).
```

[1] In a previous version we assumed that x is a cardinal. This is not needed.

Proposition 1 [3, p. 171] says that if $(a_i)_{i\in I}$ is a finite family of integers, then $\sum_{i\in I} a_i$ and $\prod_{i\in I} a_i$ are integers. As a consequence, if $J \subset I$ then $\sum_{i\in J} a_i$ and $\prod_{i\in J} a_i$ are integers. The proof is by induction on the finite set J via formulas (1) and (2).

```
Section FiniteIntFam.
Variable f: Set.
Hypothesis fif: finite_int_fam f.

Lemma finite_sum_finite_aux x:
  sub x (domain f) -> inc (card_sum (restr f x)) Bnat.
Lemma finite_product_finite_aux  x:
  sub x (domain f) -> inc (card_prod (restr f x)) Bnat.
Theorem finite_sum_finite: inc (card_sum f) Bnat.
Theorem finite_product_finite: inc (card_prod f) Bnat.

End FiniteIntFam.
```

We have obvious consequences. For instance, a finite union of finite sets is finite. As explained above, this is easy by induction. Bourbaki says that, if E is the union and S is the sum, then S is finite, and, since there is a surjection from S onto E, we have Card(E) ≤ S, so that card(E) is finite. However Card(E) ≤ S has already been stated (as corollary to Proposition 4). A finite product of finite sets is a finite set (since the cardinal of the product is the product of the cardinals). Since a^b is a product, it is finite if a and b are finite. Thus, the powerset of a finite set is finite (these results were proved in the previous chapter).

```
Lemma finite_union_finite f: fgraph f ->
  (forall i, inc i (domain f) -> finite_set (V i f))
  -> finite_set (domain f) -> finite_set(unionb f).
Lemma finite_product_finite_set f: fgraph f ->
  (forall i, inc i (domain f) -> finite_set (V i f))
  -> finite_set (domain f) -> finite_set(productb f).
```

6.2 Strict inequalities between integers

Proposition 2 [3, p. 173] says that $a < b$ if and only if there is c such that $0 < c$ and $b = c + a$ (a, b and c being integers).[2] Assume $a < b$. We know that there exists a cardinal c such that $b = c + a$; obviously c is a non-zero integer. Conversely, since $a < a + 1$, and $1 \le c$, we get $a + 1 \le a + c$ and we conclude by transitivity.

```
Lemma strict_pos_pr a: inc a Bnat ->
  (\0c <> a <-> \0c <c a).
Lemma strict_pos_pr1 a: inc a Bnat ->
   (a <> \0c <-> \0c <c a).
Theorem card_lt_pr a b: inc a Bnat -> inc b Bnat ->
  (a <c b <->  exists c, inc c Bnat & c <> \0c & a +c c = b).
```

Proposition 3 [3, p. 173] says that $\sum a_i < \sum b_i$ and $\prod a_i < \prod b_i$ for two families of integers with the same index set, if $a_i \le b_i$ for each i and $a_j < b_j$ for some j. In the case of a product, $b_i > 0$ is required. The case where the family has two elements is a consequence of the statements given above [we have $ab < a(b + c)$ if a and c are non-zero, by distributivity). In the

[2] In version 3, we replaced all tests $0 < c$ by $c \neq 0$

general case, assume $a_j < b_j$, and consider the partition $J \cup \{j\}$ of I. We have $\sum_{i\in I} a_i = A + a_j$, where $A = \sum_{i\in J} a_i$. In the same way, $\sum_{i\in I} b_i = B + b_j$, and $A \leq B$. The proof in the case of the product is similar. Note that A is finite since it is the sum of a restriction.

```
Lemma csum_Mlelt a b a' b':
  inc a Bnat -> inc b Bnat -> inc a' Bnat -> inc b' Bnat->
  a <=c a' -> b <c b' -> (a +c b) <c (a' +c b').
Lemma csum_Mlteq a a' b:
  inc a Bnat -> inc b Bnat -> inc a' Bnat ->
  a <c a' -> (a +c b) <c (a'+c  b).
Lemma cprod_Mlelt a b a' b':
  inc a Bnat -> inc b Bnat -> inc a' Bnat -> inc b' Bnat->
  a <=c a' -> b <c b' -> a' <> \0c ->
  (a *c b) <c (a' *c b').
Theorem finite_sum_lt f g:  (* 14 *)
  finite_int_fam f -> finite_int_fam g -> domain f = domain g ->
  (forall i, inc i (domain f) -> (V i f) <=c  (V i g)) ->
  (exists i, inc i (domain f) &  (V i f) <c (V i g)) ->
  (card_sum f) <c (card_sum g).
Theorem finite_product_lt f g:  (* 18 *)
  finite_int_fam f -> finite_int_fam g -> domain f = domain g ->
  (forall i, inc i (domain f) -> (V i f) <=c (V i g)) ->
  (exists i, inc i (domain f) &  (V i f) <c (V i g)) ->
  (forall i, inc i (domain g) -> (V i g) <> \0c) ->
  (card_prod f) <c (card_prod g).
```

Consequences: $a < a'$ implies $0 < a'$. If $a \neq 0$ and $1 < b$ then $a < ab$. If $a \neq 0$ and $b \geq 1$ then $a^b \geq a$ (this holds also for infinite cardinals). We have $a^b < a'^b$ if $a < a'$ and $b \neq 0$ (the first condition implies $a' > 0$). We have $a^b < a^{b'}$ if $a > 1$ and $b < b'$ (the case $a = 0$ is special, if $a = 1$, both terms are 1).

```
Lemma cprod_M1lt a b: inc a Bnat -> inc b Bnat ->
  a <> \0c -> \1c <c b -> a <c (a *c b).
Lemma cpow_nz a b: a<> \0c -> (a ^c b) <> \0c.
Lemma cpow2_nz x: \2c ^c x <> \0c.
Lemma cpow_Mltle lt1 a a' b:
  inc a Bnat ->  inc a' Bnat -> inc b Bnat->
  a <c a' -> b <> \0c ->  (a ^c b) <c (a' ^c b).
Lemma cpow_Mlelt a b b':
  inc a Bnat ->  inc b Bnat -> inc b' Bnat->
  b <c b' -> \1c <c a ->  (a ^c b) <c (a ^c b').
Lemma cpow_M1lt a b: inc a Bnat ->  inc b Bnat ->
  \1c <c b ->  a <c (b ^c a).
```

If $a + b = a + b'$ or if $ab = ab'$ then $b = b'$ (all arguments are integers; $a \neq 0$ in the case of a product).

```
Lemma csum_simplifiable_left a b b':
  inc a  Bnat -> inc b Bnat -> inc b' Bnat ->
  a +c b = a +c b' -> b = b'.
Lemma csum_simplifiable_right a b b':
  inc a  Bnat -> inc b Bnat -> inc b' Bnat ->
  b +c a = b' +c a -> b = b'.
Lemma cprod_simplifiable_left a b b':
  inc a  Bnat -> inc b Bnat -> inc b' Bnat ->
```

```
  a <> \0c -> a *c b = a *c b' -> b = b'.
  card_mult a b = card_mult a b' -> b = b'.
Lemma cprod_simplifiable_right a b b':
  inc a  Bnat -> inc b Bnat -> inc b' Bnat ->
  a <> \0c -> b *c a = b' *c a -> b = b'.
```

Given two cardinals a and b such that $a \leq b$, there exists a cardinal c such that $b = a + c$; it is called the *difference*, and denoted by $b - a$. The operation is called *subtraction*. If a and b are integers, then c is an integer is uniquely defined by this relation.

```
Lemma cdiff_wrong a b: inc a Bnat -> inc b Bnat ->
  ~ (b <=c a) -> a -c b = \0c.
Lemma BS_diff a b: inc a Bnat -> inc b Bnat ->
  inc (a -c b) Bnat.
Lemma cdiff_pr a b:
  b <=c a -> b +c (a -c b) = a.
Lemma cdiff_rpr a b:
  b <=c a -> (a -c b) +c b = a.
Lemma cdiff_pr0 a b: inc a Bnat-> inc b Bnat ->
  b <=c a -> (inc (a -c b) Bnat  & b +c (a -c b) = a).
```

We have $(a + b) - b = a$ for all integers. Uniqueness means that if $a + b = c$ then $a = c - b$ and $b = c - a$.

```
Lemma cdiff_pr1 a b: inc a Bnat-> inc b Bnat ->
  (a +c b) -c b  = a.
Lemma cdiff_pr2 a b c: inc a Bnat-> inc b Bnat ->
  a +c b = c -> c -c b = a.
```

We have $(b-a)+(b'-a') = (b+b')-(a+a')$ if the first two differences are defined. We have the trivial formulas: $a - a = 0$ and $a - 0 = a$. If a is an integer[3], then $a - 1$ is the predecessor of a. We have $(a - b) - c = a - (b + c)$ if $a \geq b + c$. Taking $c = 1$, and denoting by P and S the predecessor and successor, we have $\mathrm{P}(a - b) = a - \mathrm{S}b$. If $b \leq a$, then $a - b \leq a$; and if $b < a$, then $(a - b) - 1 < a$. If $b + 1 \leq a$ then $b < a$.

```
Lemma cdiff_n_n a: a -c a = \0c.
Lemma cdiff_n_0 a: inc a Bnat -> a -c \0c = a.

Lemma cdiff_pr4 a b a' b': inc a Bnat-> inc b Bnat ->
  inc a' Bnat-> inc b' Bnat ->
  a <=c b -> a' <=c b' ->
  (b +c b') -c (a +c a') = (b -c a) +c (b' -c a').
Lemma cdiffA a b c:
  inc a Bnat -> inc b Bnat -> inc c Bnat ->
  (b +c c) <=c a -> (a -c b) -c c = a -c (b +c c).
Lemma cpred_pr4 a: inc a Bnat ->
  cpred a = a -c \1c.
Lemma cdiff_nz1 a b: inc a Bnat -> inc b Bnat ->
  (succ b) <=c a -> a -c b <> \0c.
Lemma cdiff_A1 a b: inc a Bnat -> inc b Bnat ->
  (succ b) <=c a -> cpred (a -c b) = a -c (succ b).
Lemma cdiff_le_symmetry a b:
  b <=c a -> (a -c b) <= a.
```

[3] this is also true if $a = 0$

```
Lemma cdiff_lt_symmetry n p:  inc p Bnat ->
  n <c p -> (cpred (p -c n)) <c p.
Lemma double_diff n p: inc n Bnat -> cardinal_le p n ->
    p <=c n -> n -c (n -c p) = p.
Lemma csucc_diff a b: inc a Bnat -> inc b Bnat ->
  (succ b) <=c a -> a -c b = succ (a -c (succ b)).
```

We state here some properties of the ordering on **N**.

```
Lemma Bnat_le_R a: inc a Bnat -> a <=N a.
Lemma Bnat_le_T a b c: a <=N b -> b <=N c -> a <=N c.
Lemma Bnat_le_A  a b: a <=N b -> b <=N a -> a = b.
Lemma Bnato_to_el: forall a b, inc a Bnat -> inc b Bnat ->
  a <=N b \/ b <N a.
Lemma Bnat_zero_least a: inc a Bnat -> \0c <=N a.
Lemma Bnat_zero_least1 a: a <=N \0c -> a = \0c.
```

We show here that if $a+b \leq a+b'$, $a+b < a+b'$, $ab \leq ab'$ or $ab < ab'$ then $b \leq b'$ or $b < b'$ if inequality is strict in the assumption; in the case of a product, a must be non-zero. This is because $\leq$ is a total ordering, and the opposite relation yields a contradiction. We deduce that $(a+c)-(b+c) = a-b$, even when b is greater than a.

```
Lemma csum_le_simplifiable a b c:
  inc a Bnat -> inc b Bnat -> inc c Bnat->
  (a +c b) <=c (a +c c) ->  b <=c c.
Lemma csum_lt_simplifiable a b c:
  inc a Bnat -> inc b Bnat -> inc c Bnat->
  (a +c b) <c (a +c c) -> b <c c.
Lemma cprod_le_simplifiable a b c:
  inc a Bnat -> inc b Bnat -> inc c Bnat-> a <> \0c ->
  (a *c b) <=c  (a *c c) -> b <=c c.
Lemma cprod_lt_simplifiable a b c:
  inc a Bnat -> inc b Bnat -> inc c Bnat-> a <> \0c ->
  (a *c b) <c (a *c c) -> b <c c.

Lemma cdiff_pr5 a b c: inc a Bnat -> inc b Bnat -> inc c Bnat ->
     (a +c c) -c (b +c c) = a -c b.
Lemma cdiff_pr6 a b: inc a Bnat -> inc b Bnat ->
   (succ a) -c (succ b) = a -c b.
```

6.3 Intervals in sets of integers

Bourbaki says that, for every integer a, there is a set containing all integers $\leq a$; he denotes this as $[0,a]$. The set $[a,b]$ is used without definition, in a context where this denotes the set of all integers x such that $a \leq x \leq b$. He says that $x \mapsto x+a$ is a strictly increasing isomorphism of $[0,b]$ onto $[a,a+b]$. In fact, these sets contain only cardinals, hence are well-ordered by $\leq_{\mathrm{Card}}$. As a consequence, all isomorphisms are strictly increasing.

We use here a different approach. If $a \in \mathbf{N}$ and $b \in \mathbf{N}$, we can consider $[a,b]$ as an interval for the ordering $\leq_{\mathbf{N}}$ on **N** (ordered by $\leq_{\mathbf{N}}$). This set is the same as the previous one, and has the same ordering. We give a name to the special intervals $[0,a[$, $[1,a]$, and $[0,a]$,

```
Definition interval_Bnat a b := interval_cc Bnat_order a b.
```

```
Definition interval_co_0a a:= interval_co Bnat_order \0c a.
Definition interval_cc_0a a:= interval_Bnat \0c a.
Definition interval_cc_1a a:= interval_Bnat \1c a.
Definition interval_Bnato a b :=
  graph_on cardinal_le (interval_cc Bnat_order a b).
```

We give here some properties of intervals. We have $x \in [a,b]$ if and only if $a \leq_{\mathbf{N}} x$ and $x \leq_{\mathbf{N}} b$, where $x \leq_{\mathbf{N}} b$ is the same as $x \in \mathbf{N}$, and $b \in \mathbf{N}$ and $x \leq b$. Note that $x \leq b$ implies $x \in \mathbf{N}$. Thus we state: $x \in [0,a[$ if and only if $x < a$ and $x \in [0,a]$ if and only if $x \leq a$. These two intervals are subsets of $\mathbf{N}$. We have $[0,a+1[= [0,a]$.

```
Lemma sub_interval_Bnat a b:  sub (interval_Bnat a b) Bnat.
Lemma sub_interval_co_0a_Bnat a:  sub (interval_co_0a a) Bnat.
Lemma interval_Bnat_pr a b x: inc a Bnat -> inc b Bnat ->
  (inc x (interval_Bnat a b) <-> (a <=c x & x <=c b)-.
Lemma interval_Bnat_pr0 b x: inc b Bnat ->
  (inc x (interval_cc_0a b) <-> x <=c b).
Lemma interval_Bnat_pr1 b x: inc b Bnat ->
  (inc x (interval_cc_1a b) <->  (x <> \0c & x <=c b)).
Lemma interval_Bnat_pr1b b x: inc b Bnat ->
  inc x (interval_cc_1a b) =
   (\1c <=c x & x <=c b).
Lemma interval_co_0a_pr2 a x: inc a Bnat ->
  inc x (interval_co_0a a) = x <c a.
Lemma interval_co_0a_pr3 a x: inc a Bnat ->
  inc x (interval_co_0a (succ a)) = x <=c a.
Lemma interval_co_cc p: inc p Bnat ->
  interval_cc_0a p = interval_co_0a (succ p).
Lemma empty_interval_co0: interval_co_0a \0c = emptyset.
Lemma singleton_interval_co1:
  (inc \0o (interval_co_0a \1c) & interval_co_0a \1c = singleton \0c).
Lemma interval_cc_0a_increasing a b: inc b Bnat ->
  a <=c b ->
  sub (interval_cc_0a a) (interval_cc_0a b).
Lemma interval_cc_0a_increasing1 a: inc a Bnat ->
  sub (interval_cc_0a a) (interval_cc_0a (succ a)).
Lemma inc_a_interval_co_succ a: inc a Bnat ->
  inc a (interval_co_0a (succ a)).
Lemma interval_co_0a_increasing a: inc a Bnat ->
  sub (interval_co_0a a) (interval_co_0a (succ a)).
Lemma interval_co_0a_increasing1 a b: inc a Bnat -> inc b Bnat ->
  a <=c b -> sub (interval_co_0a a) (interval_co_0a b).
Lemma interval_co_pr4 n: inc n Bnat ->
  ( (tack_on (interval_co_0a n) n = (interval_co_0a (succ n)))
    & ~(inc n (interval_co_0a n))).
Lemma interval_bn_pr5 n: inc n Bnat ->
  let si := (interval_Bnat \1c n) in
  ( (tack_on si \0c = interval_cc_0a n)  & ~(inc \0c si)).

Lemma cardinal_c_induction5 (r:Set -> Prop) a:
  inc a Bnat -> r \0c ->
  (forall n, n <c a -> r n -> r (succ n))
  ->  (forall n, n <=c a -> r n).
```

Note that $x \leq_{[a,b]} y$ is equivalent to $a \leq x \leq y \leq b$, where $\leq$ is the ordering on cardinals, since this relation implies that x and y are finite.

```
Lemma interval_Bnato_wor a b: inc a Bnat -> inc b Bnat ->
  worder (interval_Bnato a b).
Lemma interval_Bnato_sr a b: inc a Bnat -> inc b Bnat ->
  substrate (interval_Bnato a b) = interval_Bnat a b.
Lemma interval_Bnato_gle a b x y: inc a Bnat -> inc b Bnat ->
  (gle (interval_Bnato a b) x y <-> (inc x (interval_Bnat a b) &
    inc y (interval_Bnat a b) & x <=c y)).
Lemma interval_Bnato_gle1 a b x y: inc a Bnat -> inc b Bnat ->
  (gle (interval_Bnato a b) x y <->
  (a <=c x  & a <=c y & x <=c b & y <=c b & x <=c y)).
Lemma interval_Bnato_gle2 a b x y: inc a Bnat -> inc b Bnat ->
  (gle (interval_Bnato a b) x y <-> (a <=c x  & y <=c b & x <=c y)).
```

We define here the ordered interval $[0, a[$. This is a well-ordered set (in fact, it is a segment of $\mathbf{N}$). We have $x \leq_{[0,a[} y$ if and only if $x \leq y$ and $y < a$, since these two relations imply $x \in [0, a[$ and $y \in [0, a[$.

```
Definition interval_Bnatco a :=
  graph_on cardinal_le (interval_co_0a a).

Lemma interval_Bnatco_wor a: inc a Bnat ->
  worder (interval_Bnatco a).
Lemma interval_Bnatco_sr a: inc a Bnat ->
  substrate (interval_Bnatco a) = interval_co_0a a.
Lemma interval_Bnatco_gle a x y: inc a Bnat ->
  (gle (interval_Bnatco a) x y <-> (x <=c y  & y <c a)).
Lemma segment_Bnat_order x: inc x Bnat ->
  segment Bnat_order x = interval_co Bnat_order \0c x.

Lemma cdiff_increasing2 a b c: inc a Bnat -> inc b Bnat -> inc c Bnat->
  c <=c (a +c b) ->  (c -c b) <=c a.
Lemma cdiff_increasing3 a b c: inc a Bnat ->inc b Bnat ->inc c Bnat ->
  b <=c c -> c <c (a +c b) ->  (c -c b) <c a.
```

We consider now the function $z \mapsto z + b$, $(z \in [0, a], z + b \in [a, a + b])$, that has $z \mapsto z - b$ as inverse, and hence is a bijection. Proposition 4 [3, p. 174] says that these functions are order isomorphisms.

```
Definition rest_plus_interval a b :=
  BL(fun z => z +c b)(interval_Bnat \0c a)
    (interval_Bnat b (a +c b)).

Definition rest_minus_interval a b :=
  BL(fun z => z -c b) (interval_Bnat b (a +c b))
  (interval_Bnat \0c a)

Theorem restr_plus_interval_is a b: inc a Bnat -> inc b Bnat->
  (bijection (rest_plus_interval a b)
  & bijection (rest_minus_interval a b)
  &(rest_minus_interval a b) = inverse_fun (rest_plus_interval a b)
  &
   order_isomorphism (rest_plus_interval a b)
    (interval_Bnato \0c a)
    (interval_Bnato b (a +c b))). (* 58 *)
```

If $a \leq b$, then $[a, b+1] = [a, b] \cup \{b+1\}$ and the union is disjoint. Thus $[a, b+1]$ has one more element than $[a, b]$. By induction, it has $b + 1$ elements when $a = 0$, and by application of the

isomorphism shown above, it has $(b-a)+1$ elements. This is Proposition 5 [3, p. 174]. As a consequence, the set of integers is infinite (since $[0,n]$ is subset of **N** we have $n+1 \leq \mathrm{Card}(\mathbf{N})$, so that Card(**N**) cannot be of the form n). This argument is used at the start of Chapter 6: the axiom that asserts the existence of an infinite set is equivalent to the assertion that there exists a set containing all finite cardinals.

```
Lemma interval_zero_zero:
  interval_cc_0a \0c = singleton \0c.
Lemma cardinal_interval0a a: inc a Bnat ->
  cardinal (interval_cc_0a a) = succ a.
Theorem cardinal_interval a b: a <=N b->
  cardinal (interval_Bnat a b) = succ (b -c a).
Lemma finite_set_interval_Bnat a b: a <=N b ->
  finite_set (interval_Bnat a b).
Lemma finite_set_interval_co a: inc a Bnat ->
  finite_set (interval_co_0a a).
Lemma Bnat_infinite: ~(finite_set Bnat).
```

Proposition 6 [3, p. 175] asserts that every finite totally ordered set is isomorphic to a unique interval $[1,n]$, where n is the number of elements. Bourbaki adds the condition $n \geq 1$, this is not needed. We start with a lemma that says that $[1,n]$ has n elements (note that this is also true for $n=0$). Then we pretend that if E and F are two finite equipotent totally ordered sets, there is a unique order isomorphism between them. This is because the sets are well-ordered, so that there exists a unique order morphism from E onto a segment of F; whatever the image, the function is bijective hence is an order isomorphism. The general theorem says that there is another possibility, namely that there exists an isomorphism from F onto a segment of E; it is bijective, so that its inverse is an isomorphism from E onto F. Bourbaki uses Corollary 2 to Proposition 2 of § 2, no. 2 (it says: if X is a subset of a finite set E, and $X \neq E$, then $\mathrm{Card}(X) < \mathrm{Card}(E)$. Maybe Corollary 4 was intended, since it says that an injection is bijective).

```
Lemma cardinal_interval1a a: inc a Bnat ->
  cardinal (interval_cc_1a a) = a.
Lemma isomorphism_worder_finite r r':  (* 43 *)
  total_order r -> total_order r' ->
  finite_set (substrate r) -> (substrate r) \Eq (substrate r') ->
  exists_unique (fun f => order_isomorphism f r r').
Theorem finite_ordered_interval r: total_order r ->
  finite_set (substrate r) ->
  exists_unique (fun f => order_isomorphism f r
    (interval_Bnato \1c (cardinal (substrate r)))).
```

We consider here properties of $[0,b-1]$. If we denote it by I_b, it is the interval $[0,b[$. Note that $[0,b[$ exists even when $b=0$. We can rewrite Proposition 6 as: every finite totally ordered set is isomorphic to a unique interval $[0,n[$, where n is the number of elements.

```
Lemma cardinal_interval_co_0a a: inc a Bnat -> a <> \0c ->
  cardinal (interval_cc_0a (cpred a)) = a.
Lemma interval_co_0a_pr1 a: inc a Bnat -> a <> \0c ->
  interval_cc_0a (cpred a) =  interval_co_0a a.
Lemma cardinal_interval_co_0a1 a: inc a Bnat ->
  cardinal (interval_co_0a a) = a.
Lemma emptyset_interval_00: interval_co_0a \0c = emptyset.
```

```
Theorem finite_ordered_interval1 r: total_order r ->
  finite_set (substrate r) ->
  exists_unique (fun f => order_isomorphism f r
    (interval_Bnatco  (cardinal (substrate r)))).
```

More properties of intervals.

```
Lemma partition_tack_on_intco a: inc a Bnat ->
  partition_fam (variantLc (interval_co_0a a)
    (singleton a)) (interval_co_0a (succ a)).
Lemma interval_co_0a_restr a f: inc a Bnat ->
  L (interval_co_0a a) f
    = (restr (L (interval_co_0a (succ a)) f)  (interval_co_0a a)).
Lemma partition_tack_on_int n: inc n Bnat ->
  partition_fam (variantLc (interval_cc_1a n) (singleton \0c))
    (interval_cc_0a n).
Lemma interval_int_restr a f: inc a Bnat ->
  L (interval_Bnat \1c a) f
    = restr (L (interval_Bnat \0c a) f) (interval_Bnat \1c a).
```

We show here the following. Assume[4] that $f(n)$ is a cardinal for all n. Let $\mathrm{F}(n)$ be the cardinal sum of the family $i \mapsto f(i)$ on $[0, n-1]$. Then $\mathrm{F}(n+1) = f(n) + \mathrm{F}(n)$, and there is a similar relation for the product.

```
Lemma induction_on_sum a f:  inc a Bnat ->
  let iter := fun n=> card_sum (L (interval_co_0a n)f)
    in (iter a) +c (f a) = (iter (succ a)).
Lemma induction_on_prod a f:  inc a Bnat ->
  let iter := fun n=> card_prod (L (interval_co_0a n ) f)
    in (iter a) *c (f a) = (iter (succ a)).
```

We show here that

$$\sum_{0\le i\le n} f(i) = f(0) + \sum_{1\le i\le n} f(i) = f(0) + \sum_{0\le i\le n-1} f(i+1).$$

```
Lemma fct_sum_rec0 f n: inc n Bnat ->
  card_sum (L (interval_cc_0a n) f)
  = (card_sum (L (interval_cc_1a n) f)) +c (f \0c).
Lemma fct_sum_rec1 f n: inc n Bnat -> (* 21 *)
  card_sum (L (interval_Bnat c (succ n)) f)
  card_sum (L (interval_Bnat \0c (succ n)) f)
  = (card_sum (L (interval_Bnat \0c n) (fun i=> f (succ i)))) +c (f \0c).

Lemma fct_sum_rev f n: inc n Bnat ->
  let I := (interval_co_0a (succ n)) in
  card_sum (L I f) = card_sum (L I (fun i=> f (n -c i))).
```

6.4 Finite sequences

A *finite sequence* is a family $(x_i)_{i\in \mathrm{I}}$ whose index set is a finite subset of **N** (Bourbaki says: a finite set of integers). Let f be the unique isomorphism f of the interval $[1, n]$ onto I (with

[4] This assumption is in fact not required

the natural ordering on I). Then $x_{f(k)}$ is defined for $k \in [1,n]$. It is called the *kth term of the sequence*. If $k = 1$ or $k = n$, it is called the first or last term.

6.5 Characteristic functions on sets

We define $\phi_A(x)$ to be 1 if $x \in A$ and 0 otherwise. This induces a function on every set B, the characteristic function of B.

```
Definition char_fun A B :=  BL (varianti A \1c \0c)  B (doubleton \1c \0c).

Lemma char_fun_axioms A B:
  transf_axioms (varianti A \1c \0c) B (doubleton \1c \0c).
Lemma char_fun_function: forall A B, is_function (char_fun A B).

Lemma char_fun_W A B x:
  inc x B -> W x (char_fun A B) = varianti A \1c \0c x.
Lemma char_fun_W_cardinal A B x:
  inc x B -> is_cardinal (W x (char_fun A B)).
Lemma char_fun_W_a  A B x: sub A B -> inc x A ->
  W x (char_fun A B) = \1c.
Lemma char_fun_W_b A B x: sub A B -> inc x (complement B A) ->
  W x (char_fun A B) = \0c.
```

Now some properties. We have $\phi_A = \phi_B$ if and only if A = B. The function ϕ_A (for $A \subset E$) is constant if and only if A = E or $A = \emptyset$. Proposition 7 [3, p. 176] lists additional properties.

$$\phi_{E-A}(x) = 1 - \phi_A(x)$$

$$\phi_{A\cap B}(x) = \phi_A(x)\phi_B(x)$$

$$\phi_{A\cap B}(x) + \phi_{A\cup B}(x) = \phi_A(x) + \phi_B(x)$$

```
Lemma char_fun_injective A A' B: sub A B -> sub A' B ->
  ((A=A') <-> (char_fun A B = char_fun A' B)).

Lemma char_fun_W_aa A x: inc x A ->
  W x (char_fun A A) = \1c.
Lemma char_fun_W_bb A x:  inc x A ->
  W x (char_fun emptyset A) = \0c.
Lemma char_fun_constant A B: sub A B ->
  (forall x y, inc x B -> inc y B -> W x (char_fun A B) = W y (char_fun A B))
  -> (A=B \/ A = emptyset).
Lemma char_fun_complement A B x: sub A B -> inc x B ->
  W x (char_fun (complement B A) B)
  = \1c -c (W x (char_fun A B)).
Lemma char_fun_inter A A' B x: sub A B -> sub A' B -> inc x B ->
  W x (char_fun (intersection2 A A') B)
  =  (W x (char_fun A B))  *c (W x (char_fun A' B)).
Lemma char_fun_union A A' B x: sub A B -> sub A' B -> inc x B ->
  (W x (char_fun (intersection2 A A') B))
  +c (W x (char_fun (union2 A A') B) )
  =  (W x (char_fun A B)) +c (W x (char_fun A' B)).
```

6.6 Euclidean Division

Assume that P$\{x\}$ is a property of integers, satisfied by at least one element. Since the set of integers is well-ordered, there is a least such element, hence x such that P(x) is false and P($x+1$) is true, unless P(0) is true. We give two variants of this fact.

```
Lemma least_int_prop  (prop:Set -> Prop):
  (forall x, prop x -> inc x Bnat) -> (exists x, prop x) ->
  prop \0c \/ (exists x, inc x Bnat & prop(succ x) & ~ prop x).
Lemma least_int_prop1 (prop:Set -> Prop):
  (forall x, prop x -> inc x Bnat) -> ~(prop \0c) ->
  (exists x, prop x) -> (exists x, inc x Bnat &prop(succ x) & ~ prop x).
```

Theorem 1 [3, p. 176] says that, if $b > 0$, a and b are integers, there exist unique integers q (called *quotient*) and r (called *remainder*) such that $a = bq + r$ and $r < b$. We show that the conditions are equivalent to $bq \leq a < b(q+1)$ and $r = a - bq$. Thus q is the least integer such that $a < b(q+1)$. This inequality is satisfied for $q = a$, this shows existence and uniqueness of q.

```
Definition card_division_prop a b q r :=
  a = (b *c q) +c r & r <c b.
Lemma card_division_prop_alt a b q r: inc a Bnat-> inc b Bnat ->
  inc q Bnat-> inc r Bnat -> b <> \0c ->
  (card_division_prop a b q r <->
   ( (b *c q) <=c a & a <c (b *c succ q) & r =  a -c (b *c q))).

Lemma card_division_unique a b q r q' r': inc a Bnat-> inc b Bnat ->
  inc q Bnat-> inc r Bnat -> inc q' Bnat-> inc r' Bnat -> b <> \0c ->
  card_division_prop a b q r -> card_division_prop a b q' r' ->
  (q = q' & r =r').

Lemma card_division_exists a b: inc a Bnat-> inc b Bnat ->
  b <> \0c -> exists q, exists r,
    (inc q Bnat & inc r Bnat & card_division_prop a b q r).
```

Note that the least ordinal q satisfying $a < b(q+1)$ is $\leq a$, thus is finite, thus is an integer. This gives an explicit form for the quotient; the remainder is defined by $a - bq$. If the remainder of the division of a by b is zero we say that *b divides a*. We use here a strict definition: a and b are assumed to be integers.

```
Definition card_quo a b := least_ordinal (fun q => a <c b *c (succ q)) a.
Definition card_rem a b := a -c (b *c (card_quo a b)).
Definition card_divides b a :=
  inc a Bnat & inc b Bnat & card_rem a b = \0c.

Notation "x %/c y" := (card_quo x y) (at level 40).
Notation "x %%c y" := (card_rem x y) (at level 40).
Notation "x %|c y" := (card_divides x y) (at level 40).

Lemma card_division a b (q := a %/c b) (r :=  (a %%c b)):
  inc a Bnat -> inc b Bnat -> b <> \0c ->
  (inc q Bnat & inc r Bnat & card_division_prop a b q r).  (* 32 *)
```

We state some properties of division.

```
Lemma card_quo_zero a: a %/c \0c = \0c.
Lemma card_rem_zero a: inc a Bnat -> a %%c \0c = a.
Lemma BS_quo a b: inc a Bnat-> inc b Bnat ->
 inc (a %/c b) Bnat.
Lemma BS_rem a b: inc a Bnat-> inc b Bnat ->
  inc (a %%c b) Bnat.
Lemma cdiv_pr a b: inc a Bnat-> inc b Bnat ->
  a = (b *c (a %/c b)) +c (a %%c b).
Lemma crem_pr a b: inc a Bnat-> inc b Bnat -> b <> \0c ->
  (a %%c b) <c b.
Lemma cquorem_pr a b q r:
  inc a Bnat-> inc b Bnat -> inc q Bnat -> inc r Bnat ->
  card_division_prop a b q r -> (q = a %/c b & r = a %%c b).
Lemma cquorem_pr0 a b q:
  inc a Bnat-> inc b Bnat -> inc q Bnat -> b <> \0c ->
  a = (b *c q) -> (q = a %/c b & \0c = a %%c b).
```

Note. Bourbaki says: the number q introduced above is *the integral part of the quotient of a by b*, since in the set of rational numbers $\mathbf{Q}$, there exists c such that $a = bc$, and q is the integral part of c. He reserves the term *quotient* only to the case where the remainder is zero. Moreover he says "writing a/b or $\frac{a}{b}$ will imply that b divides a". This is an abuse of notations. (It is all right to say that $a < b$ implies that a and b are integers, because we can consider as a short-hand for "$a < b$ and $a \in \mathbf{N}$ and $b \in \mathbf{N}$", but, if $d(a,b)$ is the property that b divides a, and $q(a,b)$ is the quotient, then a/b cannot be a short-hand for "$q(a,b)$ and $d(a,b)$" because this expression makes no sense (it is neither a term nor a relation). Moreover, the convention is not always respected since Bourbaki proves

$$\sum_{i=1}^{n} i = \frac{1}{2} n(n+1).$$

Now some consequences when division is exact. Bourbaki says: every multiple a' of a multiple a of b is a multiple of b. One can restate this as: if b divides a, then b divides ac.

```
Lemma cdivides_pr a b:  b %|c a -> a =  b *c (a %/c b).
Lemma cdivides_pr1 a b: inc a Bnat -> inc b Bnat ->
   b %|c (b *c a).
Lemma cdivides_pr2 a b q:
  inc a Bnat -> inc b Bnat -> inc q Bnat -> b <> \0c ->
  a = b *c q -> q = a %/c b.
Lemma cdivides_one a: inc a Bnat -> \1c %|c  a.
Lemma cquo_one a:  inc a Bnat -> inc a Bnat -> a %/d \1c = a.
Lemma cdivides_pr3 a b q:
  b %|c a -> q = a %/c b ->  a = b *c q.
Lemma cdivides_pr4 b q: inc b Bnat -> inc q Bnat -> b <> \0c ->
  (b *c q) %/c b = q.
Lemma cdivision_itself a: inc a Bnat -> a <> \0c ->
  (a %|c a & a %/c a = \1c).
Lemma cdivides_itself a: inc a Bnat -> a <> \0c ->
  a %|c a
Lemma cquo_itself a: inc a Bnat -> a <> \0c ->
   a %/c a = \1c.
Lemma cdivision_of_zero a:  inc a Bnat ->
```

```
  (a %|c \0c & \0c %/c a = \0c).
Lemma cdivides_trans a b a':
  a  %|c a'-> b  %|c a -> b  %|c a'.
Lemma cdivides_trans1 a b a':
  a  %|c a' -> b %|c a
  -> a' %/c b = (a' %/c a) *c (a  %/c b).
Lemma cdivides_trans2 a b c: inc c Bnat ->
  b  %|c a-> b %|c (a *c c).
```

The first lemma says $(ac)/(bc) = a/b$ even when division is not exact. If b divides a and a', it divides the sum and the difference.

```
Lemma cquo_simplify a b c:
  inc a Bnat -> inc b Bnat -> inc c Bnat -> b <> \0c -> c <> \0c ->
  (a *c c) %/c (b *c c) =  a %/c b.
Lemma cdivides_and_sum a a' b:  b %|c a ->  b %|c a'
  -> (b %|c  (a +c a') &
    (a +c a') %/c b = (a %/c b) +c (a' %/c b)).
Lemma cdistrib_prod2_sub a b c: inc a Bnat -> inc b Bnat -> inc c Bnat
  -> c <=c b ->
  a *c (b -c c) = (a *c b) -c (a *c c).

Lemma cdistrib_prod2_sub a b c: inc a Bnat -> inc b Bnat -> inc c Bnat
  -> cardinal_le c b ->
  a *c(card_sub b c) =  (a *c b) -c ( a *c c).
Lemma cdivides_and_difference a a' b:
  a' <=c a ->   b %|c a ->  b %|c a'
  ->  (b %|c  (a -c a') & (a' %/c b) <=c (a %/c b) &
    (a -c a') %/c b = (a %/c b) -c (a' %/c b)).
```

6.7 Expansion to base b

Proposition 8 [3, p. 177] is *Let b be an integer > 1. For each integer $k > 0$ let E_k be the lexicographic product of the family $(\mathrm{J}_h)_{0\le h\le k-1}$ of intervals all identical with $[0, b-1]$; For each $r = (r_0, r_1, \ldots, r_{k-1}) \in \mathrm{E}_k$, let $f_k(r) = \sum_{h=0}^{k-1} r_h b^{k-h-1}$; then the mapping f_k is an isomorphism of the ordered set E_k onto the interval $[0, b^k - 1]$*. Bourbaki notes that (if $a > 0$) there is a least integer k such that $a < b^k$ hence a unique sequence r_h such that

$$a = \sum_{h=0}^{k-1} r_h b^{k-h-1} \tag{1}$$

subject to the conditions $0 \le r_h \le b-1$ for $0 \le h \le k-1$ and $r_0 > 0$.

Discussion. We say that (1) is a BE-expansion, and it is a normalized expansion if either $k = 0$ (the sum is empty) or $r_0 > 0$. The quantity r_h is the digit of index k, and r_0 is the leading digit. We can restate the theorem as: every integer has an expansion to base b for some k, and a unique normalized expansion. Two numbers expressed in base b with k digits can be compared using only the value of the digits, starting with the leading digits. We can complete the theorem as follows: one can add or remove zero leading digits in the expansion, hence given two numbers with k and k' digits, one can add leading zeroes to the smallest sequence, then apply the theorem, or else remove leading zeroes in order to get normalized expansions, and then the number that has the smallest number of digits is the smallest number.

Note that the conditions $0 \le r_h$ and $0 \le h$ are redundant, since negative integers have not yet been introduced. The conditions can be restated as $r_h < b$ for $h < k$, and the interval $[0, b-1]$ is the interval $[0, b[$, this is the set of all integers $< b$. The condition $0 \le h \le k-1$ is equivalent to $0 \le k-h-1 \le k-1$, and the sum can be rewritten as $\sum_{h=0}^{k-1} r_{k-h-1} b^h$. If $s_k = r_{k-h-1}$, we get

$$a = \sum_{h=0}^{k-1} s_h b^h \tag{2}$$

subject to the conditions $0 \le s_h \le b-1$ for $0 \le h \le k-1$ and $s_{k-1} \ne 0$ is the normalization condition. We call this is LE-expansion.[5] Associated to $f_k(r)$ is the function $g_k(s)$.

If $\psi(s)$ is the sequence $(s_1, \dots, s_k)$, then

$$g_{k+1} = s_0 + b.g_k(\psi(s)) \tag{3}$$

(Bourbaki has a similar formula with f and ϕ). This is a recursive definition; one can convert it into an iterative one: as long as there are digits, multiply by b and add the next digit. We start with s_{k-1} and terminate with s_0. This means that we consider digits from left to right, r_0 then r_1, then r_2, etc. This is called the Horner scheme for the formula (1), and is used by every computer program to read numbers. If s is represented as a list, then s_0 is the head of the list and $\psi(s)$ is its tail, and (3) is the natural way to associate a value to the list.

A consequence of (3) is that s_0 and $g_k(\psi(s))$ are the remainder and quotient of the Euclidean division of a by b, and this shows uniqueness by induction. Computers use this method for printing numbers, i.e., finding the sequence s_h given a; the number k is not known a priori. In practice, one has either fixed-size numbers, say $< 2^{32}$, case where an apriori bound can be found; or else $a = \sum_0^{\mathrm{K}-1} \mathrm{S}_h \mathrm{B}^h$, for some B, case where $k \le n\mathrm{K}$ for some n, which is the size of the expansion of B in base b. The digits are computed one after the other, stored in a buffer. After that, the number is normalized (useless zeroes are removed).

Assume $s_0 < b$, $s_0' < b$; then $s_0 + bg \le s_0 + bg'$ if and only if either $g = g'$ and $s_0 \le s_0'$ or $g < g'$. This condition is equivalent to $(g, s_0) \le (g', s_0')$, where (g, s_0) is in the substrate of some lexicographic product $\mathrm{F}_k \times \mathrm{J}$ of two sets. By induction, we can identify F_k with the set of sequences $(s_1, \dots, s_k)$. Because s_0 comes after g, this set is the lexicographic product *in reverse order* of the sets J_h. Thus, the ordering of the sequence r_h is the lexicographic product of the sets J_h.

Consider now the sequence $\Psi(s) = (s_0, \dots, s_{k-1})$, then

$$g_{k+1} = g_k(\Psi(s)) + s_k.b^k. \tag{4}$$

It happens that s_k and $g_k(\Psi(s))$ are the quotient and remainder of the Euclidean division of g_{k+1} by b^k. This is an alternate way to show uniqueness of the expansion (one could use it to print a number in a computer program, the drawback being that one has to compute all b^k in decreasing order). Note that $s_k b^k \le g_{k+1} < (s_k+1)b^k$. This shows that two numbers with distinct leading digits compare as their leading digits, and shows the theorem. We shall use this approach since Ψ is just the restriction.

We define an *expansion* to be a family of k terms, all less than b, where k and b are integers, $b \ge 2$. The domain of the family is the interval $[0, k[$, it is the set of integers i with $i < k$. The associated *value* is $\sum f_i b^i$. It is an integer. If we have an expansion of length $k+1$, the restriction to $[0, k[$ is an expansion. The values are the same, up to the quantity $f_k b^k$. Similarly, we can extend an expansion from size k to size $k+1$.

[5] According to Wikipedia, little-endian storage means: increasing numeric significance with increasing memory addresses; big-endian is its opposite, most-significant byte first.

```
Lemma b_power_k_large a b: inc a Bnat ->  inc b Bnat ->
  \1c <c  b -> a <> \0c -> exists k,
    inc k Bnat & (b ^c k) <=c a & a <c (b ^c (succ k)).

Definition is_expansion f b k :=
  inc b Bnat & inc k Bnat & \1c <c b &
  fgraph f & domain f = interval_co_0a k &
  forall i, inc i (domain f) -> (V i f) <c b.

Definition expansion_value f b :=
  card_sum (L (domain f) (fun i=> (V i f) *c  (b ^c i))).
```

We assume locally that (f, k, b) and $(g, k'+1, b)$ are expansions.

```
Section Base_b_expansion.
Variables f b k: Set.
Variable Exp: is_expansion f b k.
Variable Expg: is_expansion g b (succ k').

Lemma is_expansion_prop0 i:
   (inc i (domain f)) <-> i <c k.
Lemma is_expansion_prop1 i:
  i <c k -> inc (V i f) Bnat.
Lemma is_expansion_prop2:
  finite_int_fam  (L (domain f) (fun i=> (V i f) *c  (b ^c i))).
Lemma is_expansion_prop3:
  inc (expansion_value f b) Bnat.
Lemma is_expansion_prop4: is_cardinal k' -> inc k' Bnat.
Lemma is_expansion_prop6: is_cardinal k' -> inc (V k' g) Bnat.
Lemma is_expansion_prop7: is_cardinal k' ->
  (expansion_value g b) =
   (expansion_value (restr g (interval_co_0a k')) b)
   +c  (V k' g  *c (b ^c k')).
End  Base_b_expansion.
```

Denote by $s(f)$ or by $s_k(f)$ the sum $\sum_{i<k} f_i$. We have $s_k(f) < b^k$, and $s_{k+1}(f) = s_k(f) + f_k b^k$. As a consequence the quotient and remainder of the division of $s_{k+1}(f)$ by b^k are f_k and $s_k(f)$. This shows uniqueness of the expansion, namely that $s_k(f) = s_k(g)$ implies $f_i = g_i$ for all $i < k$.

```
Lemma is_expansion_prop8 f b k x:
  let h:=  L (interval_co_0a (succ k)) (fun i=> Yo (i=k) x (V i f)) in
 is_expansion f b k ->
   inc x Bnat -> x <c b ->
    (is_expansion h b (succ k) &
      expansion_value h  b =
      (expansion_value f b) +c ((b ^c k)  *c x)).
Lemma is_expansion_prop9 f b k:  is_expansion f b k ->
    (expansion_value f b) <c (b ^c k).
Lemma is_expansion_prop10 f b k:  is_cardinal k ->
  is_expansion f b (succ k) ->
  card_division_prop (expansion_value f b) (b ^c k) (V k f)
  (expansion_value (restr f (interval_co_0a k)) b).
Lemma is_expansion_unique f g b k:
  is_expansion f b k -> is_expansion g b k ->
```

```
  expansion_value f b = expansion_value g b -> f = g. (* 47 *)
Lemma is_expansion_prop11 f g b k: is_cardinal k ->
  is_expansion f b (succ k) -> is_expansion g b (succ k) ->
  (V k f) <c (V k g) ->
  (expansion_value f b) <c (expansion_value g b).
```

Consider the two following properties. $P(f,g)$ says that there exists an index i in the range of g, not in the range of f such that g_i is not zero. It obviously implies $s(f) < s(g)$. Condition $Q(f,g)$ first says that there is an index n such that $f_i = 0$ and $g_i = 0$ for $i \geq n$, provided that these expressions are defined (and f_i and g_i are defined for $i < n$). Such an index exists if $P(f,g)$ and $P(g,f)$ are false. We have then $s(f) = s_n(f)$ and $s(g) = s_n(g)$. The Bourbaki claim is then that $s(f)$ and $s(g)$ can be compared lexicographically (replacing i by $n-i$). Condition Q says moreover that there exists k such that $f_i = g_i$ for $k < i < n$, so that $s(f)$ and $s(g)$ compare the same as $s_k(f)$ and $s_k(g)$. The case $k = -1$ is special, since it says that $s(f) = s(g)$. Thus, assuming $s(f) \neq s(g)$, there is a least such k [in other terms: if $s(f) \neq s(g)$ there is a greatest index k such that $f_k \neq g_k$]. Now condition Q says $f_k < g_k$. We have shown above that this implies $s_k(f) < s_k(g)$. The theorem is now $s(f) < s(g)$ if and only one of P or Q is true. Notice that the five cases $P(f,g)$, $P(g,f)$, $Q(f,g)$, $Q(g,f)$, $R(f,g)$ are mutually exclusive (here R is the condition that there is n, such that $f_i = g_i$ for $i < n$, and for all indices $i \geq n$ for which f_i and g_i are defined, the value is zero; it implies $s(f) = s(g)$).

```
Lemma is_expansion_restr1 f b k l:
  is_expansion f b k -> l <=c  k ->
  is_expansion (restr f  (interval_co_0a l)) b l.
Lemma is_expansion_restr2 f b k l: (* 35 *)
  is_expansion f b k -> l <=c k ->
  (forall i, l <=c i -> i <c k -> V i f = \0c) ->
  expansion_value (restr f  (interval_co_0a l)) b = expansion_value f b.

Lemma is_expansion_prop12 f g b kf kg l n:
  n <=c kf -> n <=c kg ->  l <c n ->
  (forall i, n <=c i -> i <c kf -> V i f = \0c) ->
  (forall i, n <=c i -> i <c kg -> V i g = \0c) ->
  (forall i, l <c i ->  i <c n -> V i f = V i g) ->
  is_expansion f b kf -> is_expansion g b kg ->
  (V l f) <c  (V l g) ->
  (expansion_value f b)  <c (expansion_value g b).  (* 99 *)
Lemma is_expansion_prop13 f g b kf kg l:
  kf <=c l -> l <c kg ->
  is_expansion f b kf -> is_expansion g b kg ->
  V l g <> \0c ->
  (expansion_value f b) <c (expansion_value g b).  (* 26 *)

Lemma is_expansion_prop13 f g b kf kg l:
  cardinal_le kf l -> cardinal_lt l kg ->
  is_expansion f b kf -> is_expansion g b kg ->
  V l g <> \0c ->
  cardinal_lt (expansion_value f b)  (expansion_value g b).

Lemma is_expansion_prop14 f g b kf kg:
  is_expansion f b kf -> is_expansion g b kg ->
  (expansion_value f b) <c  (expansion_value g b) ->
  (exists l,  kf <=c l & l <c kg & V l g <> \0c)
 \/ (
```

```
  exists l, exists n,
  (n <=c kf & n <=c kg & l <c n &
  (forall i, n <=c i -> i <c kf -> V i f = \0c) &
  (forall i, n <=c i -> i <c kg -> V i g = \0c) &
  (forall i, l <c i ->  i <c n -> V i f = V i g) &
   (V l f) <c  (V l g))).  (* 91 *)
```

If $a < b^k$ there is an expansion of length k (proof by induction, the highest term is the quotient of the division by b^{k-1}). Since $a < b^a$, there is at least one expansion.

```
Lemma is_expansion_exists1 a b k:
  inc b Bnat -> \1c <c  b -> inc k Bnat ->
  inc a Bnat -> a <c (b ^c k) ->
  exists f,  (is_expansion f b k & expansion_value f b = a). (* 32 *)
Lemma is_expansion_exists a b:  inc a Bnat -> inc b Bnat ->
  \1c <c  b -> exists k, exists f,
    (is_expansion f b k & expansion_value f b = a).
```

We study some properties of the modulo operator. First, the quantities $a\mathrm{B}+b$ and b are equal modulo B. Then then show that the sum and products of quantities equal mod B are equal mod B. We the show that if $a = 1$ mod B, then $a^n = 1$ mod B.

```
Definition eqmod a b B:= a %%c B = b %%c B.
Lemma card_plus_permute24 s1 s2 r1 r2:
  ((s1 +c s2) +c r1) +c r2 = (s1 +c r1) +c  (s2 +c r2).

Section ModuloProps.
Variable B: Set.
Hypothesis Bn: inc B Bnat.
Hypothesis Bnz: B <> \0c.

Lemma crem_prop a b: inc a Bnat -> inc b Bnat ->
  eqmod ((B *c a) +c b) b B.
Lemma crem_sum a b: inc a Bnat -> inc b Bnat ->
  eqmod (a +c b) ((a %%c B) +c (b %%c B)) B.
Lemma crem_mult a b: inc a Bnat -> inc b Bnat ->
  eqmod (a *c b) ((a %%c B) *c (b %%c B)) B.
Lemma eqmod_plus a b a' b': inc a Bnat -> inc b Bnat ->
  inc a' Bnat -> inc b' Bnat ->
  eqmod a a' B -> eqmod b b' B -> eqmod (a +c b) (a' +c b') B.
Lemma eqmod_mult a b a' b': inc a Bnat -> inc b Bnat ->
  inc a' Bnat -> inc b' Bnat ->
  eqmod a a' B -> eqmod b b' B -> eqmod (a *c b) (a' *c b') B.
Lemma eqmod_rem a: inc a Bnat ->  eqmod a (a %%c B) B.
Lemma eqmod_succ  a a': inc a Bnat -> inc a' Bnat ->
  eqmod a a' B -> eqmod (succ a) (succ a') B.
Lemma eqmod_pow1  a n: inc a Bnat -> inc n Bnat ->
  eqmod a \1c B -> eqmod (a ^c n) \1c B.
Lemma eqmod_pow2 a b n: inc a Bnat -> inc b Bnat -> inc n Bnat ->
  eqmod a \1c B ->  eqmod (b *c (a ^c n)) b B.
```

Assume $b = 1$ mod B. Then $\sum a_i b^i = \sum a_i$ modulo B

```
Lemma eqmod_pow3  f b k:  is_expansion f b k ->
   eqmod b \1c B -> eqmod (expansion_value f b) (card_sum f) B.
End ModuloProps.
```

Define $5 = 4 + 1$, so that $2 + 3 = 5$. Define $10 = 5 + 5$ so that $10 = 3 \times 3 + 1$. We deduce that 10 is 1 mod 3.

```
Definition card_five := succ card_four.
Definition card_ten :=  card_five +c card_five.
Notation "\10c" := card_ten.

Lemma BS5 : inc card_five Bnat.
Lemma BS10 : inc \10c Bnat.
Lemma card_plus_3_2:  \3c +c \2c = card_five.
Lemma card_mult_3_3: \10c = succ (\3c *c \3c).
Lemma card_mult_10_3: eqmod \10c \1c \3c.
```

We can now state that two quantities $\sum a_i 10^i$ and $\sum a_i$ have the same remainder modulo 3.

```
Definition is_base_ten_expansion  f k :=
  inc k Bnat & fgraph f & domain f = interval_co_0a k &
  forall i, inc i (domain f) -> (V i f) <c \10c.
Lemma divisibiliy_by_three  f k: is_base_ten_expansion f k ->
  let g:= (L (domain f) (fun i=> (V i f) *c (\10c ^c i))) in
  eqmod (card_sum g) (card_sum f) \3c.
```

6.8 Combinatorial analysis

The next result is Proposition 9 [3, p. 179], known in French as the shepherd's principle. If f is a function from a set with cardinal $\mathfrak{a}$ onto a set with cardinal $\mathfrak{b}$, and if all sets $f^{-1}\langle\{x\}\rangle$ have the same cardinal $\mathfrak{c}$, then $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$. Bourbaki assumes f surjective; in fact if x is in the target but not in the range, then $\mathfrak{c} = 0$, and the source is empty.

```
Theorem shepherd_principle f c: is_function f ->
  (forall x, inc x (target f) -> cardinal (inv_image_by_fun f (singleton x))=c)
  -> cardinal (source f) = (cardinal (target f)) *c c. (* 29 *)
```

The following definition and lemmas have been moved here from the next chapter. See explanations on page 142.

```
Definition induction_defined0 (h: Set -> Set -> Set)  (a: Set) :=
  transfinite_defined Bnat_order
  (fun u => Yo(source u = \0c) a (h (cpred (source u))(W (cpred (source u)) u))).

Definition induction_defined (s: Set -> Set)  (a: Set) :=
  transfinite_defined Bnat_order
  (fun u => Yo(source u = \0c) a (s (W (cpred (source u)) u))).

Lemma segment_Bnat_order1 n: inc n Bnat -> segment Bnat_order  n = n.

Lemma induction_defined_pr0 h a (f :=  induction_defined0 h a):
     source f = Bnat &  surjection f & W \0c f = a &
     forall n, inc n Bnat -> W (succ n) f = h n (W n f).
Lemma induction_defined_pr s a (f :=  induction_defined s a):
     source f = Bnat &  surjection f & W \0c f = a &
     forall n, inc n Bnat -> W (succ n) f = s (W n f).
```

```
Lemma integer_induction0 h a: exists_unique
   (fun f =>
   source f = Bnat & surjection f &
    W \0c f = a
    &  forall n, inc n Bnat -> W (succ n) f = h n (W n f)).
Lemma integer_induction s a: exists_unique (fun f =>
  source f = Bnat &  surjection f & W \0c f = a &
  forall n, inc n Bnat -> W (succ n) f = s (W n f)).
```

We use here the previous definition as follows: for any element a and any function $s(x,y)$ we define a term T (depending on s and a) such that $T(0) = a$ and $T(n+1) = s(n, T(n))$.

```
Definition induction_term s a := fun n => W n (induction_defined s a).
Lemma induction_term0 s a:
  induction_term s a \0c =a.
Lemma induction_terms s a n:
  inc n Bnat ->
  induction_term s a (succ n) = s n (induction_term s a n).
```

6.8.1 Factorial

Bourbaki defines the *factorial* of n, denoted by $n!$, as $\prod_{i<n}(i+1)$. It satisfies $0! = 1$ and $(n+1)! = n!(n+1)$. There is a unique function satisfying this property.

```
Definition factorial n :=
  card_prod (L (interval_co_0a n) succ).

Lemma factorial_succ n: inc n Bnat ->
  factorial (succ n) = (factorial n) *c (succ n).
Lemma factorial0: factorial \0c = \1c.
Lemma factorial1: factorial \1c = \1c.
Lemma factorial2: factorial \2c = \2c.
Lemma factorial_nz n: inc n Bnat -> factorial n <> \0c.
Lemma BS_factorial n: inc n Bnat -> inc (factorial n) Bnat.
Lemma factorial_prop f: f \0c = \1c ->
  (forall n, inc n Bnat -> f (succ n) =  (f n) *c (succ n)) ->
  forall x, inc x Bnat -> f x = factorial x.
```

We show how the factorial function could have been defined by induction.

```
Lemma factorial_induction n: inc n Bnat ->
 factorial n = induction_term  (fun a b=> b *c (succ a)) \1c n.
```

6.8.2 Number of injections

Proposition 10 [3, p. 170] says that the number of injections from a set with m elements to a set with n elements is $A_{nm} = n!/(n-m)!$. Note that, if such an injection exists, we have $m \leq n$; otherwise the number is zero.

We first prove that the quantity A_{nm} is well-defined (by induction on $n-m$). Thus $A_{nm}(n-m)! = n!$. From this we deduce $A_{nm}(n-m) = A_{n,m+1}$.

```
Definition number_of_injections b a :=
 (factorial a) %/c (factorial (a -c b)).

Lemma quotient_of_factorials a b:
  inc a Bnat -> inc b Bnat -> b <=c a ->
  (factorial b) %|c (factorial a).
Lemma quotient_of_factorials1 a b:
  inc a Bnat -> inc b Bnat -> b <=c a ->
  (factorial (a -c b)) %.%|c (factorial a).

Lemma number_of_injections_pr a b:
  inc a Bnat -> inc b Bnat -> b <=c a ->
  (number_of_injections b a) *c (factorial (a -c b)) = factorial a.
Lemma number_of_injections_base a: inc a Bnat ->
  number_of_injections \0c a = \1c.
Lemma number_of_injections_rec a b:
  inc a Bnat -> inc b Bnat -> b <c a ->
  (number_of_injections b a) *c (a -c b) =
  number_of_injections (succ b) a.
Lemma number_of_injections_int a b:
 inc a Bnat -> inc b Bnat ->
 inc (number_of_injections b a) Bnat.
```

Consider an injective function f from A into B, which are sets with cardinals a and b. Let c be the cardinal of the complement of the image, we have $a + c = b$. We deduce $b - a = c$ when the target is finite.

```
Lemma cardinal_complement_image1 f: injection f ->
  (cardinal (complement (target f) (image_of_fun f)))
  +c (cardinal (source f))
  = cardinal (target f).
Lemma cardinal_complement_image f: injection f ->
  finite_set (target f) ->
  ((cardinal (source f)) <=c (cardinal (target f)) &
    cardinal (complement (target f) (image_of_fun f)) =
    (cardinal (target f)) -c (cardinal (source f))).
```

The proof of the proposition is as follows. If $m = 0$, then $A_{nm} = 1$; and there is a unique function from the empty set into E, this function is injective. Consider now $A = A' \cup \{a\}$, where A' has m elements. Let G_1 and G_2 be the sets of injective functions from A and A' to F, and H_1 and H_2 their cardinals. By induction, using the recurrence formula for A_{nm} we must show $H_1 = H_2(n - m)$. Given $f \in G_1$, its restriction $R(f)$ to A' is obviously injective, hence is in G_2. Given f' in G_2, and $b \in F$, there is a unique f with $f' = R(f)$ and $f(a) = b$. This function is an injection if and only if b is not in the range of f'. In other terms, $f \mapsto f(a)$ is a bijection from $R^{-1}\langle\{f'\}\rangle$ onto the complementary of the range of f' that has $n - m$ elements. Hence $R^{-1}\langle\{f'\}\rangle$ has $n - m$ elements; the conclusion follows from the shepherd's principle.

```
Definition set_of_injections  E F :=
  Zo (set_of_functions E F)(injection).

Lemma number_of_injections_prop E F:
  finite_set F ->
  cardinal E <=c cardinal F ->
  cardinal (set_of_injections E F)  =
    number_of_injections (cardinal E) (cardinal F). (* 128 *)
```

An injection from E into itself is a bijection when E is finite. Since $A_{nn} = n!$, we deduce that $n!$ is the number of *permutations* of E.

```
Lemma number_of_permutations E:  finite_set E ->
  cardinal (set_of_permutations E)  = (factorial (cardinal E)).
```

6.8.3 Number of coverings

Proposition 11 [3, p. 180] says "let E be a finite set with p elements, and let $(p_i)_{1\le i\le h}$ be a finite sequence of integers such that $\sum_{i=1}^{h} p_i = n$. Then the number of coverings $(X_i)_{1\le i\le h}$ of E by mutually disjoints sets X_i such that $\mathrm{card}(X_i) = p_i$ for $1 \le i \le h$ is equal to $n!/(\prod_{i=1}^{h} p_i!)$."

In this section, p will denote a finite integer family (a mapping $I \to \mathbf{N}$, where I is a finite set). We shall replace the condition $1 \le i \le p$ by $i \in I$ everywhere. To say that $(X_i)_{i\in I}$ is a covering of E by mutually disjoints sets is the same as partition_fam X E, by abuse of language we shall call this a "partition" of E. Notice that X_i is a subset of E, so that the family X is an element of the set of functional graphs $I \to \mathfrak{P}(E)$. If the domain of X is I and $\mathrm{card}(X_i) = p_i$ whenever $i \in I$, we say that X is a partition with p elements and write $X \in C_{p}E$. The proposition gives a formula for the cardinal of $C_{p}E$.

Note that $\sum p_i$ is finite. Thus, if $C_{p}E$ is non-empty, then E is finite. We start with a lemma that says that, if $\mathrm{Card}(E) = \sum p_i$ then $C_{p}E$ is non-empty. The proof given here is a rather long (over 500 lines).

```
Definition partition_with_pi_elements p E f :=
  domain f = domain p &
  (forall i, inc i (domain p) -> cardinal (V i f) = V i p) &
  partition_fam f E.
Definition set_of_partitions p E :=
  Zo(set_of_gfunctions (domain p) (powerset E))
    (partition_with_pi_elements p E).
Lemma set_of_partitions_rw p E f:
  inc f  (set_of_partitions p E) <-> partition_with_pi_elements p E f.
Lemma fif_cardinal i p:
  finite_int_fam p -> inc i (domain p) -> is_cardinal (V i p).
Lemma pip_prop0 p E f: partition_with_pi_elements p E f ->
  forall i, inc i (domain f) ->  sub (V i f) E.

Lemma number_of_partitions1 p E:
  finite_int_fam p -> card_sum p = cardinal E ->
  nonempty (set_of_partitions p E). (* 45 *)
```

Define Q(E) to be the set of permutations of E. Assume $g \in Q(E)$ and $f \in C_{p}E$. Denote by $\psi(f,g)$ the mapping $i \mapsto g\langle f_i\rangle$; it belongs to $C_{p}E$ (same argumentation as before). Thus we have a function $\phi : g \mapsto \psi(f,g)$, from Q to $C_{p}E$. We pretend this is surjective (lemma 4) (if X_i is a partition with p_i elements, then F_i is equipotent to X_i; this gives a bijection h_i from F_i to X_i, hence a function g_i that extends h_i, with target E, and a function g that coincides with g_i on F_i).

```
Definition set_of_partitions_aux f g:=
  L (domain f) (fun i => image_by_fun g (V i f)).

Lemma number_of_partitions3 p E f g:
```

```
  partition_with_pi_elements p E f -> inc g  (set_of_permutations E) ->
  inc (set_of_partitions_aux f g) (set_of_partitions p E).  (* 26 *)

Lemma number_of_partitions4 p E f:
  finite_int_fam p -> card_sum p = cardinal E ->
  partition_with_pi_elements p E f ->
  surjection (BL (set_of_partitions_aux f)
    (set_of_permutations E) (set_of_partitions p E)).  (* 72 *)
```

This function ϕ is not injective: lemma 5 says $\phi(g) = \phi(h)$ if and only if $h^{-1} \circ g$ is a bijection that leaves each $f(i)$ invariant. Fix h; for each g and i, the restriction w_i of $h^{-1} \circ g$ to $f(i)$ can be considered as a function from $f(i)$ to itself; it is in fact a bijection, hence the family $(w_i)_i$ is an element of the product of the sets $Q(f(i))$ (this is lemma 6).

Consider now the function Φ that associates to each permutation k of E the family of restrictions to $A_i = f(i)$. If k leaves A_i invariant, the restriction of k to A_i is a bijection, so that, if we consider as target of Φ the product of the permutations of A_i, Φ is well-defined for $h^{-1} \circ g$, for each $g \in \phi^{-1}(\{h\})$. This is a bijection (lemma 7). The main result follows: the target of Φ has cardinal $\prod p_i!$; the source of Φ is $\phi^{-1}\langle\{h\}\rangle$ if we take $k = h^{-1} \circ g$; it then suffices to apply the shepherd's principle.

```
Lemma number_of_partitions5 p E f g h:
  finite_int_fam p -> card_sum p = cardinal E ->
  partition_with_pi_elements p E f ->
  inc h (set_of_permutations E) -> inc g (set_of_permutations E) ->
  ((set_of_partitions_aux p E f g = set_of_partitions_aux p E f h) <->
  (forall i, inc i (domain p) ->
      image_by_fun ((inverse_fun h) \co g) (W i f) = (W i f))). (* 38 *)

Lemma number_of_partitions6 p E f h:
  finite_int_fam p -> card_sum p = cardinal E ->
  partition_with_pi_elements p E f ->
  inc h (set_of_permutations E) ->
  bl_axioms (fun g=> L (domain p)(fun i=>  (restriction2
    ((inverse_fun h) \co g)
    (V i f) (V i f))))
  (Zo  (set_of_permutations E)
    (fun g => (set_of_partitions_aux f g = set_of_partitions_aux f h)))
  (productb (L (domain p)(fun i=> (set_of_permutations (V i f))))).


Lemma number_of_partitions7 p E f h: (* 130 *)
  finite_int_fam p -> card_sum p = cardinal E ->
  partition_with_pi_elements p E f ->
  inc h (set_of_permutations E) ->
  bijection(BL (fun g=> L (domain p)(fun i=> (restriction2
    ((inverse_fun h) \co g)
    (V i f) (V i f))))
  (Zo  (set_of_permutations E)
    (fun g => (set_of_partitions_aux f g = set_of_partitions_aux f h)))
  (productb (L (domain p)(fun i=> (set_of_permutations (V i f)))))).
```

We give here the long and the short version of the theorem.

```
Theorem number_of_partitions p E:
```

```
  finite_int_fam p -> card_sum p = cardinal E ->
  let num:= factorial (cardinal E) in
    let den := card_prod (L (domain p) (fun z => factorial (V z p)))
      in (
        num = cardinal (set_of_partitions p E) *c  den &
        finite_c num & finite_c den & den <> \0c &
        finite_set (set_of_partitions p E)).  (* 63 *)

Theorem number_of_partitions_bis p E:
  finite_int_fam p -> card_sum p = cardinal E ->
  cardinal (set_of_partitions p E) =
  (factorial (cardinal E)) %/c
           (card_prod (L (domain p) (fun z => factorialC (V z p)))).
```

We consider here the special case where the family has two elements m and p, so that $n = m + p$.

```
Lemma number_of_partitions_p2 E m p:
 inc m Bnat -> inc p Bnat -> cardinal E = (m +c p) ->
 let num := factorial (m +c p) in
 let den := (factorial m) *c (factorial p) in
 let x := cardinal (set_of_partitions (variantLc m p) E) in
   inc x Bnat & num = x *c den &
      inc num Bnat & inc den Bnat & den <> \0c.
```

6.8.4 The binomial coefficient

Bourbaki defines the *binomial coefficient* $b_{np} = \binom{n}{p}$ as the number of subsets of p elements in a set of n elements, after showing that

$$b_{n,p} = \frac{n!}{p!(n-p)!} \text{ if } p \le n, \qquad b_{n,p} = 0 \text{ otherwise.} \tag{1}$$

He shows in Proposition 13 [3, p. 181]) that it satisfies the following relation.

$$\binom{n}{0} = 1, \quad \binom{0}{p+1} = 0, \qquad \binom{n+1}{p+1} = \binom{n}{p+1} + \binom{n}{p}. \tag{2}$$

Our initial implementation used induction on the type nat in order to define the function binom below via (2); we showed showed that it satisfies (1). The ssreflect library provides a function bin, recursively defined by bin_rec, with exactly the same properties. We show here the power of the library by giving the proof od the following relation

$$\sum_{i=0}^{n} \binom{n}{i} a^{n-i} b^i = (a+b)^n. \tag{3}$$

```
(*
 Fixpoint binom (n p:nat) {struct n} : nat :=
   match n, p with
     | 0, 0 => 1
     | 0, S m => 0
     | S q, 0 => 1
     | S q, S m =>  (binom q (S m)) + (binom q m)
  end.
```

```
Fixpoint bin_rec (m n : nat) {struct m} :=
  match m, n with
  | m'.+1, n'.+1 => bin_rec m' n + bin_rec m' n'
  | _, 0 => 1
  | 0, _.+1 => 0
  end.
Theorem pascal : forall a b n,
  (a + b) ^ n = \sum_(i < n.+1) (bin n i * (a ^ (n - i) * b ^ i)).
Proof.
move=> a b; elim=> [|n IHn]; first by rewrite big_ord_recl big_ord0.
rewrite big_ord_recr big_ord_recl /= expnS {}IHn muln_addl !big_distrr.
rewrite big_ord_recl big_ord_recr /= !bin0 !binn !subn0 !subnn !mul1n !muln1.
rewrite -!expnS addnA; congr (_ + _); rewrite -addnA -big_split; congr (_ + _).
apply: eq_bigr => i _ /=; rewrite 2!(mulnCA b) (mulnCA a) (mulnA a) -!expnS.
by rewrite -leq_subS ?ltn_ord // -muln_addl -binS.
Qed.
*)
```

We define here a function c_{np} by induction on n; the definition is a bit obscure since we define by induction an auxiliary term f_n, where f_n is a functional graph on **N**, then say $c_{np} = f_n(p)$.

```
Definition binom n m :=
  V m (induction_term
      (fun _ T: Set => L Bnat (fun z => variant \0c \1c
          (V z T +c V (cpred z) T) z))
       (L Bnat (variant \0c \1c \0c))
       n).
Lemma binom00:  binom \0c \0c = \1c.
Lemma binom0Sm m: inc m Bnat -> binom \0c (succ m) = \0c.
Lemma binomSn0 n: inc n Bnat-> binom (succ n) \0c = \1c.
Lemma binomSnSm n m: inc n Bnat-> inc m Bnat ->
  binom (succ n) (succ m) = (binom n (succ m)) +c (binom n m).
Lemma BS_binom n m: inc n Bnat -> inc m Bnat ->
  inc (binom n m) Bnat.
```

Let $f(n,p)$ be the product of c_{np} by $p!(n-p)!$. Note that if $p > n$, then $n-p$ is zero by convention and $(n-p)!$ is one. We have then $(n-p)! = \text{if}(p < n, n-p, 1)\cdot(n-(p+1))!$. This gives us an induction property for f, from which we deduce $f(n,p) = \text{if}(p \le n, n!, 0)$. We restate this as: if $p > n$ the binomial coefficient is zero, else $f(n,p) = n!$. This formula shows that $p!(n-p)!$ divides $n!$.

```
Lemma binom_alt_pr n m: inc n Bnat -> inc m Bnat ->
  (binom n m) *c  (factorial m) *c (factorial (n -c m)) =
  Yo (m <=c n) (factorial n) \0c.   (* 108 *)
Lemma binom_bad n m: inc n Bnat -> inc m Bnat ->
  n <c m -> binom n m = \0c.
Lemma binom_good n m: inc n Bnat -> inc m Bnat ->
  m <=c n ->
  (binom n m) *c  (factorial m) *c (factorial (n -c m)) = (factorial n).
```

We then have

$$\binom{n}{p} = \binom{n}{n-p} = \frac{n!}{p!(n-p)!} \quad \text{when } p \le n. \tag{4}$$

```
Lemma binom_pr0 n p: inc n Bnat -> inc p Bnat ->
  p <=c n ->
  let num := (factorial n) in
  let den:=  (factorial p) *c (factorial (n -c p)) in
  den %|c num & binom n p = num %/c den.

Lemma binom_pr1 n p: inc n Bnat -> inc p Bnat ->
  p <=c n ->
  binom n p = (factorial n) %/c ((factorial p) *c (factorial (n -c p))).

Lemma binom_symmetric n p: inc n Bnat -> inc p Bnat ->
    p <=c n -> binom n p = binom n (n -c p).
```

We show here

$$\binom{n}{0} = 1, \qquad \binom{n}{1} = n, \qquad \binom{n+1}{2} = \frac{n(n+1)}{2}. \tag{5}$$

If $p \leq n$, the binomial coefficient is non-zero; if $p = n$ it is one, and if $p \leq n+1$ it is a strictly increasing function of n.

```
Lemma binom0 n: inc n Bnat -> binom n \0c = \1c.
Lemma binom1 n: inc n Bnat -> binom n \1c = n.
Lemma binom2a  n: inc n Bnat ->
  \2c *c (binom (succ n) \2c) = n *c (succ  n).
Lemma binom2 n: inc n Bnat ->
 binom (succ n) \2c = (n *c (succ n)) %/c \2c.
Lemma binom_nn n: inc n Bnat -> binom n n = \1c.
Lemma binom_pr3 n p:  inc n Bnat -> inc p Bnat ->
  p <=c n -> binom n p <> \0c.

Lemma finite_sum4_lt a b: inc a Bnat -> inc b Bnat ->
     a <> \0c -> b <c (b +c a).
Lemma binom_monotone1 k n m:
  inc k Bnat -> inc n Bnat -> inc m Bnat ->
  k <> \0c -> k <=c (succ n) -> n <c m ->
  (binom n k) <c (binom m k).   (* 26 *)
Lemma binom_monotone2 k n m:
  inc k Bnat -> inc n Bnat -> inc m Bnat ->
  k <> \0c -> k <=c (succ n) ->  k <=c (succ m) ->
  (n <c m <->  (binom n k) <c (binom m k)).
```

The last lemma of the previous section says that that $c_{m+p,p}$ is the number of partitions of a set E with $n = m + p$ elements into two sets with m and p elements. For completeness, we show that $c_{n,p}$ is the number of partitions of E into two sets with $n - p$ and p elements (if $p > n$, this number is zero, there is no partition of E into two subsets with p and k elements, whatever k). Let Q_p be the set of all subsets A of E that have p elements. If $\mathrm{A} \in \mathrm{Q}_p$ then $\mathrm{E} - \mathrm{A} \in \mathrm{Q}_{n-p}$ and $\mathrm{A} \mapsto \mathrm{E} - \mathrm{A}$ is a bijection. Moreover $(\mathrm{A}, \mathrm{E} - \mathrm{A})$ is a partition with $(p, n - p)$ elements. The cardinal of Q_p is b_{np}, the Bourbaki definition of the binomial coefficient. Thus $b_{np} = c_{np}$ whenever $p \leq n$. The relation also holds if $p > n$ since Q_p is empty and $c_{np} = 0$.

```
Lemma number_of_partitions_p3 E m p: inc m Bnat -> inc p Bnat ->
  cardinal E = m +c p ->
  cardinal (set_of_partitions (variantLc m p) E) =
  binom (m +c p) m.
```

```
Lemma number_of_partitions_p4 E n m: inc n Bnat -> inc m Bnat ->
  cardinal E = n ->
  cardinal (set_of_partitions (variantLc m (n -c m)) E) =
  binom n m.

Definition subsets_with_p_elements p E:=
  Zo (powerset E)(fun z=> cardinal z =p).
Lemma cardinal_complement1 n p E A: inc n Bnat -> inc p Bnat ->
  cardinal E = n -> cardinal A = p -> sub A E ->
  cardinal (complement E A) = n -c p.

Lemma subsets_with_p_elements_pr n p E: inc n Bnat -> inc p Bnat ->
  cardinal E = n ->
  binom n p = cardinal (subsets_with_p_elements p E).   (* 52 *)
Lemma subsets_with_p_elements_pr0 n p: inc n Bnat -> inc p Bnat ->
  binom n p = cardinal (subsets_with_p_elements p n).
Lemma bijective_complement n p E: inc n Bnat -> inc p Bnat ->
  p <=c n -> cardinal E = n ->
  bijection (BL (complement E)
    (subsets_with_p_elements p E)(subsets_with_p_elements (n -c p) E)).
```

The following formula is true whenever a_i are permutable elements of a ring

$$\sum_{\sum p_i=n} \frac{n!}{p_1!\cdots p_k!} a_1^{p_1}...a_k^{p_k} = (a_1+a_2+\cdots+a_k)^n. \tag{6}$$

Here $\sum p_i$ means $p_1+p_2+\cdots+p_k$. If we denote the sum $p_1+p_2+\cdots+p_{k-1}$ by $\sum p_j$, factor out powers of a_k and use associativity, the previous expression becomes

$$\sum_m \left(\sum_{\sum p_j=m} \frac{m!}{p_1!\cdots p_{k-1}!} a_1^{p_1}...a_{k-1}^{p_{k-1}} \right) \binom{n}{m} a_k^{n-m}$$

We can proceed by induction on k, starting with $k=2$, for which we use induction on n. We show here an alternate method, valid only when the quantities a_i are integers. In the special case $a_i=1$ the formula reduces to

$$\sum_{\sum p_i=n} \frac{n!}{p_1!\cdots p_k!} = k^n, \qquad \sum_p \binom{n}{p} = 2^n. \tag{7}$$

Let E be a finite set of cardinal n, I an index set, and $\mathrm{A}_{n\mathrm{I}}$ be the set of all mappings p with $\sum_{i\in \mathrm{I}} p(i) = n$. This relation implies that $p(i) \in [0,n]$. Thus we complete the definition of $\mathrm{A}_{n\mathrm{I}}$ by requiring that the target of p is the interval $[0,n]$. We shall compute the cardinal of this set later on.

Let $\mathrm{B}_{n\mathrm{I}}$ be the set of graphs of elements of $\mathrm{A}_{n\mathrm{I}}$. We have that $p \in \mathrm{B}_{n\mathrm{I}}$ if and only if p is a functional graph defined on I and $\sum p_i = n$. There is a similar property for graphs with $p \in \mathrm{B}_{n\mathrm{I}}$ instead of $\sum p_i = n$. Thus we can rewrite (6) and (7) with $\sum p_i \leq n$.

```
Definition set_of_functions_sum_eq F n:=
  Zo (set_of_functions F (interval_cc_0a n))
  (fun z=> (card_sum (P z)) = n).
Definition set_of_functions_sum_le E n:=
  Zo (set_of_functions E (interval_cc_0a n))
  (fun z=> (card_sum (P z)) <=c n).
```

```
Definition set_of_graph_sum_eq F n:=
  fun_image (set_of_functions_sum_eq F n) graph.
Definition set_of_graph_sum_le F n:=
  fun_image (set_of_functions_sum_le F n) graph.

Lemma setof_suml_aux F n f: inc n Bnat ->
  (inc f (set_of_graph_sum_le F n) <->
  (fgraph f & domain f = F & (card_sum f) <=c n &
    (forall i, inc i (domain f) -> is_cardinal (V i f)))).
Lemma setof_sume_aux F n f: inc n Bnat ->
  (inc f (set_of_graph_sum_eq F n) <->
  (fgraph f & domain f = F & card_sum f = n &
    (forall i, inc i (domain f) -> is_cardinal (V i f)))).
```

Consider now a finite sequence of integers a_i (We shall see in the next chapter that if a_i is infinite, both terms in (6) are equal to the greatest element element of the family, the case where the number of terms in infinite can also be handled, bu the result has no great interest). We let k be the cardinal of the index set I, and consider a set F whose cardinal is $\sum a_i$. We know that there is a partition F_i of F with $\mathrm{Card}(\mathrm{F}_i) = a_i$. If f is a function $\mathrm{E} \to \mathrm{F}$, we consider the set $f_i = f^{-1}\langle \mathrm{F}_i \rangle$ of all elements of E such that $f(x) \in \mathrm{F}_i$. Denote by $\phi(f)$ the mapping $i \mapsto \mathrm{Card}\, f_i$, and by $\Phi(f)$ the mapping $i \mapsto f_i$. We have $\phi(f) \in \mathrm{B}_{n\mathrm{I}}$ and $\Phi(f) \in \mathrm{C}_{\phi(f),\mathrm{E}}$, where $\mathrm{C}_{p\mathrm{E}}$ denotes the set of all partitions with p_i elements of E.

If P is in $\mathrm{C}_{p\mathrm{E}}$ we consider a family of bijections $h_i : [1, p_i] \to \mathrm{P}_i$. If we take $\mathrm{P} = \Phi(f)$, the function $f \circ h_i$ maps $[1, p_i]$ to F_i. Its restriction to F_i is a element of $\mathscr{F}([1, p_i], \mathrm{F}_i)$. Denote it by $\Psi(f)(i)$. Now $\Psi(f) \in \prod \mathscr{F}([1, p_i], \mathrm{F}_i)$. The mapping $f \to (\Phi(f), \Psi(f)$ is a bijection (for fixed $\phi(f)$), as can be easily shown (the proof is a bit technical, thus long).

The case $a_i = 1$ is a bit easier. Here each F_i has one element, and we can identity Φ and Ψ (given a partition E_i of E, if $\mathrm{F}_i = \{y_i\}$, f maps E_i into F_i if and only if $f(x) = y_i$ for $x \in \mathrm{E}_i$). This makes the proof much shorter.

```
Lemma sum_of_gen_binom E F n: inc n Bnat -> cardinal E = n ->
  card_sum (L (set_of_graph_sum_eq F n)
              (fun p => cardinal (set_of_partitions p E)))
  = (cardinal F) ^c n.  (* 115 *)
Lemma sum_of_gen_binom0 E n a: (* 251 *)
  inc n Bnat ->  cardinal E = n -> finite_int_fam a ->
  (card_sum a) ^c n =
     card_sum  (L (set_of_graph_sum_eq (domain a) n)
       (fun p =>
         (cardinal (set_of_partitions p E)) *c
            (card_prod (L  (domain a) (fun i=> ((V i a) ^c (V i p))))))).
```

We consider now a special case where F is the canonical doubleton. Its cardinal is 2. To each $m \in [0, n]$ we can associate the function defined on F that maps the first element to m and the second to $n-m$. This is a bijection onto $\mathrm{B}_{n\mathrm{F}}$, and we can use it to perform a change of variables in the sum, and we can apply number_of_partitions_p4, and we get the binomial coefficient (Formula (7b)). We give an alternate proof: with count the number of subset of E, according to their cardinal p.

We prove the Pascal formula (3) by induction. If we replace n by $n+1$, isolate the term

$p = 0$, write $p = k+1$ and use the binomial relation we get

$$\sum_{i=0}^{n} \binom{n}{k} a^{k+1} b^{n+1-k-1} + \sum_{i=0}^{n} \binom{n}{k+1} a^{k+1} b^{n+1-k-1} + b^{n+1}.$$

We re-introduce p in the second sum, factor out a and b and we get

$$a \left[\sum_{i=0}^{n} \binom{n}{k} a^k b^{n-k} \right] + \left[\sum_{i=0}^{n} \binom{n}{p} a^p b^{n-p} \right] b.$$

It suffices to apply the induction hypothesis. The proof is similar to that given above, just much longer.

```
Lemma sum_of_gen_binom2 n: inc n Bnat ->
  card_sum (L  (interval_cc_0a n) (binom n)) = \2c ^c n. (* 47 *)
Lemma sum_of_binomial n: inc n Bnat ->  (* 35 *)
  card_sum (L (interval_cc_0a n) (binom n)) = \2c ^c n.

Lemma sum_of_binomal2 a b n:
  is_cardinal a -> is_cardinal b -> inc n Bnat ->
  is_cardinal a -> is_cardinal b -> inc n Bnat ->
  card_sum (L (interval_cc_0a n)
         (fun p => (binom n p) *c (a ^c p) *c (b ^c (n -c p))))
  = (a +c b) ^c n.    (* 105 *)
```

6.8.5 Number of increasing functions

Consider two finite totally ordered sets E and F. Denote by $\mathscr{S}(\mathrm{E},\mathrm{F})$ the set of strictly increasing mappings from E into F, and by $\mathscr{A}(\mathrm{E},\mathrm{F})$ the set of increasing mappings from E into F. We pretend that

$$\mathrm{Card}(\mathscr{S}(\mathrm{E},\mathrm{F})) = \binom{\mathrm{Card}(\mathrm{F})}{\mathrm{Card}(\mathrm{F})}.$$

$$\mathrm{Card}(\mathscr{A}(\mathrm{E},\mathrm{F})) = \binom{n+p-1}{p} \text{ if } \mathrm{Card}(\mathrm{E}) = p \text{ and } \mathrm{Card}(\mathrm{F}) = n.$$

The first relation is a consequence of the the fact that that the mapping $f \mapsto \mathrm{R}(f)$ is a bijection from $\mathscr{S}(\mathrm{E},\mathrm{F})$ onto the set of subsets with p elements of F, where $\mathrm{R}(f)$ denotes the range of f. It requires Theorem 3 (§ 2, no. 5) (uniqueness of isomorphisms between well-ordered sets). [note: if u and v are two strictly increasing functions with the same source range, target, they must be equal. For otherwise there would exists a least element x such that $u(x) \neq v(x)$ since the source is finite and totally ordered. We have $u(x) = v(a)$ for some a. If $a < x$ we get $v(a) = u(a) = u(x)$, hence $a = x$ by injectivitity of u. The case $a = x$ is also excluded, so that $x < a$ and $v(x) < v(a)$, thus $v(x) < u(x)$. By symmetry $u(x) < v(x)$, absurd.

```
Definition set_of_incr_functions r r' :=
  (Zo (set_of_functions (substrate r) (substrate r'))
    (fun z => increasing_fun z r r')).
Definition set_of_strict_incr_functions r r' :=
  (Zo (set_of_functions (substrate r) (substrate r'))
    (fun z => strict_increasing_fun z r r')).

Lemma cardinal_set_of_increasing_functions1 r r' f:
```

```
  total_order r -> strict_increasing_fun f r r' ->
  (order_morphism f r r' & cardinal (substrate r)  = cardinal (image_of_fun f)).

Lemma cardinal_set_of_increasing_functions2 r r':
  total_order r -> total_order r' ->
  finite_set (substrate r)  ->  finite_set (substrate r')  ->
  bijection (BL (fun z  => range (graph z))
    (set_of_strict_incr_functions r r')
    (subsets_with_p_elements (cardinal (substrate r)) (substrate r'))). (* 91 *)

Lemma cardinal_set_of_increasing_functions r r':
  total_order r -> total_order r' ->
  finite_set (substrate r)  ->  finite_set (substrate r')  ->
  cardinal (set_of_strict_incr_functions r r')
  = binom (cardinal (substrate r')) (cardinal (substrate r)).
```

We shall give some variants of this property. We start with the case where E is a segment of **N**, say an interval $[0, p]$. If f is a function with values in an ordered set F, if $f(x) \leq f(x+1)$ for $x < p$, then $f(x) \leq_f (y)$ whenever $x \leq y \leq p$. The same holds if $\leq$ is replaced by $<$; the proof is obvious by induction. In the case of a strictly increasing function, we have an injection, since the source is totally ordered.

```
Lemma increasing_prop0 p f r: inc p Bnat -> order r ->
  (forall i, i <=c p -> inc (f i) (substrate r)) ->
  (forall n,  n <c p -> gle r (f n) (f (succ n))) ->
  (forall i j, i <=c j -> j <=c p ->
    gle r (f i) (f j)). (* 21 *)
Lemma increasing_prop1 p f: inc p Bnat ->
  (forall i, i <=c p -> inc (f i) Bnat) ->
  (forall n, n <c p -> (f n) <=c (f (succ n))) ->
  (forall i j, i <=c j -> j <=c p -> (f i) <=c (f j)).
Lemma strict_increasing_prop0 p f r: inc p Bnat -> order r ->
  (forall n,  n <c p -> glt r (f n) (f (succ n))) ->
  (forall i j, i <c j -> j <=c p ->
    glt r (f i) (f j)).  (* 29 *)

Lemma increasing_prop p f r: inc p Bnat -> is_function f ->
  source f =  interval_co_0a (succ p) -> order r -> substrate r = target f ->
  (forall n, n <c p -> gle r (W n f) (W (succ n) f)) ->
  increasing_fun f (interval_Bnato \0c p) r.

Lemma strict_increasing_prop p f r: inc p Bnat -> is_function f ->
  source f =  interval_co_0a (succ p) -> order r -> substrate r = target f ->
  (forall n, n <c p -> glt r (W n f) (W (succ n) f)) ->
  (injection f &
    strict_increasing_fun f (interval_Bnato \0c p) r).
```

Denote by I_p the interval $[0, p[$. Assume now that f is strictly increasing $\mathrm{I}_p \to \mathbf{N}$. Then $x \leq f(x)$ on I_p by induction on x. The function $i \mapsto f(i) - i$ is decreasing. Assume f maps I_p into I_{n+p}. When $i < p$ we have $f(i) - i \leq f(p-1) - (p-1) < n + p - p + 1$, hence $f(i) - i \in \mathrm{I}_{n+1}$.

```
Lemma strict_increasing_prop1 f p:
  inc p Bnat ->  (forall i, i <c p -> inc (f i) Bnat)
  -> (forall i j, i <c j -> j <c p -> (f i) <c (f j)) ->
  inc p Bnat ->  (forall i, i <c p -> inc (f i) Bnat)
```

```
  -> (forall i j, i <c j -> j <c p -> (f i) <c (f j)) ->
Lemma strict_increasing_prop3 f p n:
    (forall i j, i <=c j -> j <c p -> ((f i) -c i) <=c ((f j) -c j)).
  inc p Bnat -> inc n Bnat -> (forall i, i <c p -> inc (f i) Bnat)
  -> (forall i j, i <c j -> j <c p -> (f i) <c (f j)) ->
  (forall i, i <c p -> (f i) <c (n +c p)) ->
  (forall i, i <c p -> ((f i) -c i) <=c n).
```

If f is a mapping, denote by $s(f)$ the mapping $i \mapsto f(i) - i$, and by $a(f) : i \mapsto f(i) + i$. We have shown that s is a mapping from $\mathscr{S}(\mathrm{I}_p, \mathrm{I}_{n+p})$ into $\mathscr{A}(\mathrm{I}_p, \mathrm{I}_{n+1})$. It is a bijection with inverse a. Thus we get

$$\mathrm{Card}(\mathscr{A}(\mathrm{I}_p, \mathrm{I}_{n+1})) = \mathrm{Card}(\mathscr{S}(\mathrm{I}_p, \mathrm{I}_{n+p})) = \binom{n+p}{p}.$$

```
Lemma cardinal_set_of_increasing_functions3 n p:
  inc n Bnat -> inc p Bnat ->
  cardinal (set_of_incr_functions (interval_Bnatco p)
         (interval_Bnato \0c n))
   = binom (n +c p) p. (172 *)
```

We compute the cardinal of $\mathscr{A}(\mathrm{E}, \mathrm{F})$ in the general case. If F is empty, there is a unique function $\mathrm{E} \to \mathrm{F}$ if E is empty, and none if E is non-empty. Assume that E has p elements and F has $n > 0$ elements. There is a unique order isomorphism f between between E and $[0, p[$, and g between F and $[0, n-1]$. Assume $h : \mathrm{E} \to \mathrm{F}$ increasing. Then $k = g \circ h \circ f^{-1} : [0, p[\to [0, n[$. Since $h = g^{-1} \circ k \circ f$, we have that h is increasing if and only if k is increasing, thanks to the additional lemmas.This gives an isomorphism between $\mathscr{A}(\mathrm{E}, \mathrm{F})$ and $\mathscr{A}(\mathrm{I}_p, \mathrm{I}_n)$, and these two sets have the same number of elements.

```
Lemma increasing_compose f g r r' r'':
  increasing_fun f r r' ->  increasing_fun g r' r'' ->
  (g \coP f &
    (forall x, inc x (source f) -> W x (g \co f) = W (W x f) g) &
    increasing_fun (g \co f) r r'').

Lemma increasing_compose3 f g h r r' r'' r''':
  strict_increasing_fun f r r' ->  increasing_fun g r' r'' ->
  strict_increasing_fun h r'' r''' ->
  let res:=  (h \co g) \co f in
    (inc res (set_of_functions (source f) (target h)) &
      (forall x, inc x (source f) -> W x res = W (W (W x f) g) h) &
      increasing_fun res r r''').

Lemma cardinal_set_of_increasing_functions4 r r':
  let n :=  (cardinal (substrate r')) in
    let p := (cardinal (substrate r)) in
      total_order r -> total_order r' ->
      finite_set (substrate r)  ->  finite_set (substrate r')  ->
      cardinal (set_of_incr_functions r r')
      = binom ((n +c p) -c \1c) p. (* 116 *)
```

We compute now the number a_n of pairs (i, j) such that $1 \le i \le j \le n$, and the number b_n of pairs satisfying $1 \le i < j \le n$. This is the number of increasing (resp. strictly increasing) mappings of a set with two elements into the interval $[1, n]$, which has n elements. Our

previous results show

$$a_n = \frac{n(n+1)}{2} = \binom{n+1}{2}, \qquad b_n = \frac{n(n-1)}{2} = \binom{n}{2}.$$

The Bourbaki proof of these relations (Proposition 14 [3, p. 181]) is different. He notices that $a_n = b_n + n$ since $i \leq j$ is equivalent to $i < j$ or $i \leq j$. A subset of $[1, n]$ is of cardinal two if and only if it is a doubleton $\{i, j\}$ with $i \neq j$, and we may assume $i < j$; hence b_n is the number of subsets of cardinal two of $[1, n]$. The link between a_n and b_n is given by the following trivial relation:

$$\binom{n+1}{2} = \frac{n(n+1)}{2} = \binom{n}{2} + n.$$

```
Lemma binom_2plus n: inc n Bnat ->
   binom (succ n) \2c =  (n *c (succ n)) %/c \2c.
Lemma binom_2plus0 n: inc n Bnat ->
  binom (succ n) \2c = (binom n \2c) +c n.

Lemma cardinal_pairs_lt n: inc n Bnat ->
  cardinal (Zo (product Bnat Bnat)
    (fun z => \1c <=c (P z) &  (P z) <c (Q z) &  (Q z) <=c n)) =
  (binom n \2c).   (* 55 *)

Lemma cardinal_pairs_le n: inc n Bnat ->
  cardinal(Zo (product Bnat Bnat)
    (fun z=> \1c <=c  (P z) & (P z) <=c (Q z) & (Q z) <=c n)) =
     (binom (succ n) \2c).   (* 26 *)
```

A corollary is the following formula

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2} = \binom{n+1}{2}$$

This formula is obvious by induction on the type nat; the prof by induction on **N** is a bit longer. Let s_n be the sum and s'_n be the sum $\sum_{i \leq n}(n-i)$. By re-ordering indices, we have $s'_n = s_n$: moreover $s_n + s'_n = \sum_{i \leq n} n$, which is $n(n+1)$, and we get the result by division by two; finally, we follow Bourbaki, showing that s_n is the cardinal of the set E of all pairs (i, j) with $1 \leq i \leq j \leq n$, since E is the union of the sets $[1, j] \times \{j\}$, i.e., the disjoint union of the intervals $[1, j]$, which are of cardinal j.

```
Lemma fct_sum_const1 f n m:
  inc n Bnat ->  (forall i, i <c n -> f i = m) ->
  card_sum (L (interval_co_0a n) f) = n *c m.
Lemma sum_of_i n: inc n Bnat ->
  card_sum (L (interval_co_0a n) id) =
   binom n \2c.
Lemma sum_of_i3 n: inc n Bnat ->
  card_sum (L (interval_co_0a n) id)) =
   binom n \2c.  (* 31 *)
Lemma sum_of_i2 n:  inc n Bnat ->
  card_sum (L (interval_Bnat \1c n) id) =
  (binom (succ n) \2c).   (* 27 *)
```

6.8.6 Number of monomials

Consider a set E, a law of composition of E, and elements x, y, z, etc, of E. Consider a combination of these variables, where x appears 3 times, z appears twice and y appears once. If the law is associative and commutative, the combination is equal to $x\cdot(x\cdot(x\cdot(y\cdot(z\cdot z))))$. This is called a monomial, and denoted by x^3yz^2. The total number of terms (here six) is called the degree. Assume that we have a second law of composition $a+b$, and that the usual rules apply. This means that $(a+b)^n$ can be expanded as sum monomials: $(a+b)^n = \sum \gamma_{ij} a^i b^j$. It happens that $\gamma_{ij} = \binom{n}{i}$ if $i+j=n$ (this is the explanation of the term "binomial coefficient"). More generally, $(\sum x_i)^n = \sum_{\mathrm{I}\in \mathrm{S}_n} \Gamma_{\mathrm{I}} x^{\mathrm{I}}$, where I is a mapping $i \mapsto n_i$, x^{I} denotes the monomial $x_1^{n_1} x_2^{n_2}\cdots x_p^{n_p}$. The total degree of the monomial is $\sum n_i = n$. Taking $a=b=1$, gives $\sum_p \binom{n}{p} = 2^n$. Taking $a=-1$ and $b=1$ gives $\sum_p (-1)^p \binom{n}{p} = 0$ (The first result has already been proved, the second is the object of Exercice 5.2). We have also $\sum_{\mathrm{I}\in \mathrm{S}_n} \Gamma_{\mathrm{I}} = p^n$ (it can be shown, by induction of the number of variables, that Γ_{I} is the number of coverings of a set with n elements by subsets with n_i elements). The cardinal of the set S_n is the object of the next theorem. We compute it by induction on both n and p.

Let E be a set with h elements, $\bar{\mathrm{A}}_n$ and $\bar{\mathrm{B}}_n$ be the sets of functions u with $\sum_{i\in \mathrm{E}} u(i) \le n$ and $\sum_{i\in \mathrm{E}} u(i) = n$ respectively. Let A_{nh} and B_{nh} the cardinals of these sets. Proposition 15 [3, p. 182]) says

$$\mathrm{A}_{nh} = \binom{n+h}{h} \qquad \mathrm{B}_{nh} = \binom{n+h-1}{h-1}.$$

We have $\mathrm{A}_{nh} = \mathrm{B}_{nh} + \mathrm{A}_{n-1,h}$ since $\bar{\mathrm{A}}_n$ is the disjoint union of $\bar{\mathrm{B}}_n$ and $\bar{\mathrm{A}}_{n-1}$ If $x \notin \mathrm{E}$, every function u such that $\sum u(i) \le n$ can be uniquely extended to $\mathrm{E}\cup\{x\}$ in such a way as $\sum_{i\in \mathrm{E}\cup\{x\}} u(i) = n$. This gives $\mathrm{B}_{n,h+1} = \mathrm{A}_{nh}$. The formulas follow by induction (they are trivial for $h=0$ and $n=0$). One difficulty of the Bourbaki's proof is that he has not yet defined the set of integers; as a consequence, he adds the condition that the target of u is the interval $[0,n]$, so that $\bar{\mathrm{A}}_{n-1}$ is not a subset of $\bar{\mathrm{A}}_n$; it is nevertheless isomorphic to the complement of $\bar{\mathrm{B}}_n$ in $\bar{\mathrm{A}}_n$. There are two other solutions: we may consider functions with target **N**, or graphs of functions. Since we already introduced the set of graphs of functions such that $\sum u_i = n$, we use graphs. We first show that the sets have the same number of elements.

```
Lemma sof_sum_eq_equi F n: inc n Bnat ->
  (set_of_functions_sum_eq F n) \Eq (set_of_graph_sum_eq F n).
Lemma sof_sum_le_equi: forall F n, inc n Bnat ->
  (set_of_functions_sum_le F n) \Eq (set_of_graph_sum_le F n).

Lemma set_of_functions_sum0 f:
  (forall a, inc a Bnat -> f \0c a = \1c) ->
  (forall a, inc a Bnat -> f a \0c = \1c) ->
  (forall a b, inc a Bnat -> inc b Bnat ->
     f (succ a) (succ b) = (f (succ a) b) +c (f a (succ b))) ->
  forall a b, inc a Bnat -> inc b Bnat ->  f a b = (binom (a +c b) a).

Lemma set_of_functions_sum1 E x n:
  inc n Bnat -> ~ (inc x E) ->
  (set_of_functions_sum_le E n)
  \Eq (set_of_functions_sum_eq (tack_on E x) n). (* 55 * )

Lemma set_of_functions_sum2 E n: inc n Bnat ->
 cardinal(set_of_graph_sum_le E (succ n))
 =  (cardinal (set_of_graph_sum_eq E (succ n)))
```

```
   +c  (cardinal (set_of_graph_sum_le E n)).  (* 21 *)

Lemma set_of_functions_sum3 E:
  cardinal (set_of_functions_sum_le E \0c) = \1c.  (* 18 *)
Lemma set_of_functions_sum4: forall n, is_cardinal n->
    cardinal (set_of_functions_sum_le emptyset n) = \1c.

Lemma set_of_functions_sum_pr n h:
  inc n Bnat -> inc h Bnat ->
  let intv:= fun h => (interval_co_0a h) in
    let sle:= fun n h => set_of_graph_sum_le (intv h) n in
      let seq := fun n h =>  set_of_graph_sum_eq (intv h) n in
        let A:= fun n h => cardinal (sle n h) in
          let B:=  fun n h => cardinal (seq n h) in
            (A n h =  B n (succ h) & A n h = (binom (n +c h) n)).
```

We give now a variant of the theorem[6]. We pretend that the number of functions y defined on $[0,p]$ with values in $[0,n]$ and such that $\sum y_i \leq n$ is $A_{n,p+1} = \binom{n+p+1}{p+1}$. This is the previous result for $h = p+1$ (if $h = 0$, there is a unique function defined on a set with h elements, the empty function, and the sum is zero). The quantity $A_{n,p+1}$ is the number of subsets of $[0,n+p]$ with $p+1$ elements. Consider the sequence x_i, defined by induction (see next chapter) via $x_0 = y_0$ and $x_{i+1} = y_{i+1} + x_i + 1$. All we have to do is prove that the mapping $y \mapsto \{x_0, x_1, \ldots, x_p\}$ is bijective (as a function with values in the subset of $\mathfrak{P}([0,n+p])$ formed of sets with $p+1$ elements). The idea is that the function x is strictly increasing, and uniquely defined by its range. Since $A_{n,p+1}$ is the number of strictly increasing functions $[0,p] \to [0,n+p]$; all we have to to is to prove that the mapping $y \mapsto x$ is a bijection (as a function into $\mathscr{S}([0,p],[0,n+p])$.

The inverse mapping of $y \mapsto x$ is defined by $y_{i+1} = x_{i+1} - (x_i + 1)$. It is easier to consider $y_{i+1} = z_{i+1} - z_i$, where $z_i = x_i - i$. It happens that z_i is the sum of the restriction of y to the interval $[0,i]$ (there is no need to define it by induction) and is obviously increasing. Since $A_{n,p+1}$ is the cardinal of $\mathscr{A}([0,p],[0,n])$, the set of increasing functions $[0,p] \to [0,n]$, all we have to do is show that $y \mapsto z$ is a bijection $C_{pn} \to C'_{pn}$ where C_{pn} is the set of functions $y : [0,p] \to [0,n]$ such that $\sum y_i \leq n$ and $C'_{pn} = \mathscr{A}([0,p],[0,n])$. We first show that $A_{n,p+1}$ is the cardinal of C'_{pn}.

```
Definition set_of_graph_sum_le_int p n :=
  set_of_graph_sum_le (interval_cc_0a p) n.

Definition set_of_increasing_functions_int p n :=
  (Zo (set_of_functions (interval_cc_0a p)(interval_cc_0a n))
    (fun z => increasing_fun z
      (interval_Bnato \0c p)
      (interval_Bnato \0c n))).

Lemma card_set_of_increasing_functions_int  p n:
  inc p Bnat -> inc n Bnat ->
  cardinal (set_of_increasing_functions_int p n) =
  binom (succ (n +c p)) (succ p).
```

If $R(f,i)$ denoted the restriction of the function f to the interval $[0,i]$ we have $R(R(f,i),j) = R(f,j)$ if $j \leq i$. If $S(f,i)$ is the sum of the restriction of $R(f,i)$ we have $S(f,0) = f(0)$ and $S(f,i+1) = f(i+1) + S(f,i)$.

[6] Suggested by Jean-Baptiste Pomet

```
Lemma double_restrc f n p: fgraph f -> inc p Bnat ->
  n <c p ->
  domain f = interval_cc_0a p ->
  restr (restr f (interval_cc_0a (succ n)))
     (interval_cc_0a n) =
     restr f (interval_cc_0a n).
Lemma induction_on_sum3 f m:
  fgraph f -> inc m Bnat ->
  domain f = interval_cc_0a m ->
  (forall a, inc a (domain f) -> is_cardinal (V a f)) ->
  (card_sum (restr f (interval_cc_0a \0c))
    = (V \0c f)
    & (forall n, n <=c m ->
      (card_sum (restr f (interval_co_0a n))) +c (V n f)
      = card_sum (restr f (interval_co_0a (succ n))))).  (* 29 *)
```

Given a function y, we consider z such that $z_i = \mathrm{S}(y,i)$. We first show that z maps $[0,p]$ into $[0,n]$, and then that $z \in \mathrm{C}'pn$ if $y \in \mathrm{C}_{pn}$ (note that $i \mapsto \mathrm{S}(y,i)$ is increasing). We then show that $y \mapsto z$ is injective and surjective. The key relation is $z_0 = y_0$ and $z_{i+1} = y_{i+1} + z_i$ (trivial consequence of induction_on_sum3); it says that y is uniquely defined from z. Moreover, given an increasing function z', if $y_0 = z'_0$ and $y_{i+1} = z'_{i+1} - z'_i$, the same formula is satisfied by z', hence $z = z'$, thus proving surjectivity.

```
Definition csum_to_increasing_fun y :=
  fun i => card_sum (restr y (interval_cc_0a i)).

Definition csum_to_increasing_fct y n p :=
  BL (csum_to_increasing_fun y)
  (interval_cc_0a p) (interval_cc_0a n).

Lemma csum_to_increasing1 y n p:
  inc n Bnat -> inc p Bnat ->
  inc y (set_of_graph_sum_le_int p n) ->
  bl_axioms (csum_to_increasing_fun y)
    (interval_cc_0a p)
    (interval_cc_0a n).

Lemma csum_to_increasing2 n p:
  inc n Bnat -> inc p Bnat ->
  bl_axioms (fun y=> (csum_to_increasing_fct y n p))
  (set_of_graph_sum_le_int p n)
  (set_of_increasing_functions_int p n).  (* 46 *)

Lemma csum_to_increasing4 n p:
  inc n Bnat -> inc p Bnat ->
  injection (BL (fun y=> (csum_to_increasing_fct y n p))
  (set_of_graph_sum_le_int p n)
  (set_of_increasing_functions_int p n)).  (* 41 *)

Lemma csum_to_increasing5 n p:
  inc n Bnat -> inc p Bnat ->
  surjection (BL (fun y=> (csum_to_increasing_fct y n p))
  (set_of_functions_sum_le_int p n)
  (set_of_increasing_functions_int p n)). (* 86 *)

Lemma csum_to_increasing6 n p:
```

```
inc p Bnat -> inc n Bnat ->
cardinal (set_of_graph_sum_le_int p n) =
binom (succ (n +c p)) (succ p).
```

Chapter 7

Infinite sets

7.1 The set of natural integers

Bourbaki defines an *infinite set* as a set that is not finite. If such a set exists and $\mathfrak{a}$ is its cardinal, then all integers n satisfy $n < \mathfrak{a}$, since otherwise we would have $\mathfrak{a} \leq n$, which implies $\mathfrak{a}$ finite. This means that the set **N** of cardinals n such that n is finite contains all integers. By extensionality, it does not depend on $\mathfrak{a}$. Bourbaki has an axiom that asserts the existence of an infinite set, and deduces (Theorem 1, [3, p. 184]) that the set **N** of integers exists. Its cardinal is denoted by $\aleph_0$. We have already seen that the set is infinite: for any integer n, the interval $[0, n]$ is a subset of **N**; taking cardinals yields $n+1 \leq \aleph_0$; since $n < n+1$ we get $n \neq \aleph_0$.

```
Lemma infinite_Bnat: infinite_set Bnat.
```

Bourbaki defines a *sequence* as a family whose index set I is a subset of **N**. It is called an infinite sequence if I is infinite. Remember that a *finite sequence* is a family where I is finite and contains only integers; this means that I is a finite subset of **N**.

Let's quote Bourbaki [3, p. 184] "Let $\mathrm{P}\{n\}$ be a relation and let I denote the set of integers n such that $\mathrm{P}\{n\}$ is true. I is then a subset of **N**. A sequence $(x_n)_{n\in\mathrm{I}}$ is then sometimes written $(x_n)_{\mathrm{P}\{n\}}$, and x_n is called the *nth term* in the sequence." Example. Assume that $\mathrm{P}\{n\}$ is the relation $n \in \mathbf{Z}$ where **Z** denotes the set of rational integers as a subset of **N**. According to the quote, $(x_n)_{n\in\mathbf{N}}$ and $(x_n)_{n\in\mathbf{Z}}$ are the same sequences. This may be confusing since they are obviously different families. In section 6.4, Bourbaki assumes that $\mathrm{P}\{n\}$ implies that n is an integer; this is missing here. Other example: we consider the property "n even and $n < 10$". This is a finite sequence and the 4th term is 6; in the French version, we can read "x_4 est le terme d'indice 4" and "x_6 est le 4-ème terme". In English this is translated as "x_4 is the 4th term" and "x_6 is the 4th term". This may be confusing.

According to Bourbaki, if the property is $n \geq k$, the sequence is written as $(x_n)_{k\leq n}$ or $(x_n)_{n\geq k}$ or even (x_n) if $k = 0$ or $k = 1$. This last notation is obviously ambiguous. The sum of such a family may be denoted a $\sum_{n=k}^{\infty} x_n$.

Two sequences $(x_n)_{n\in\mathrm{I}}$ and $(y_n)_{n\in\mathrm{I}}$ with the same index set are said to *differ only in the order of their terms* if there exists a permutation f of the index set I such that $x_{f(n)} = y_n$ for all $n \in \mathrm{I}$. This makes sense even if I is not a subset of the integers. By commutativity, two sequences that differ only in the order of their terms have same sum and product.

A *multiple sequence* is a family whose index set is a subset of a product $\mathbf{N}^p$ (p is a integer).

Let f be a bijection of $\mathbf{N}$ onto a set I. For each family $(x_i)_{i\in \mathrm{I}}$, the sequence $n \mapsto x_{f(n)}$ is said to be obtained by *arranging the family* $(x_i)_{i\in \mathrm{I}}$ *in the order defined by* f.

7.2 Definition of mappings by induction

If we instantiate Criterion C60 (see page 50) to the well-ordered set $\mathbf{N}$ we get another criterion; it assert that, for any term T, there exists a unique surjective function f such that

(TIND) $$\forall n, n \in \mathbf{N} \implies f(n) = \mathrm{T}\{f^{(n)}\}$$

where $u^{(x)}$ denotes the restriction of u to the segment $]\leftarrow, x[$. Recall that $]\leftarrow, x[$ is the set of all elements y such that $y < x$; this is just the interval $[0, x[$. In Bourbaki, the operator $\mathrm{T}\{u\}$ is *defined* for any set u. In practice, it is *used* only when u is of the form $u_n = f^{(n)}$, this is the restriction of some unknown function to a segment of our well-ordered set.

In the case of induction on $\mathbf{N}$, the argument n of f is an integer, and the source u_n is the set of all integers $< n$. This is a set with n elements. We deduce $\mathrm{M}(u_n) = n$, where $\mathrm{M}(u)$ is the cardinal of the source of u_n. If we use von Neumann integers, we can simplify the definition by noting that the source of u_n is n. Consider the two definitions of T:

```
T  u := Yo (M u = \0c) a (s (W (prec (M u)) u))
T' u := Yo (M u = \0c) a (h (cpred (M u)) (W (cpred (M u)) u))
```

If $n = 0$, then $\mathrm{M}(u_n) = 0$, and by definition of Yo we get $f(0) = a$. Assume $n = m+1$. We have $u_n(m) = f(m)$ and $\mathrm{M}(u_n) \neq 0$. By definition of Yo, we get $f(m+1) = s(f(m))$ (in the case T) and $f(m+1) = h(m, f(m))$ in the case T′. We deduce:

There exists a unique surjective function f defined on $\mathbf{N}$ such that

(IND) $$f(0) = a \quad \text{and} \quad f(n+1) = s(f(n)).$$

There exists a unique surjective function f defined on $\mathbf{N}$ such that

(IND0) $$f(0) = a \quad \text{and} \quad f(n+1) = h(n, f(n)).$$

It is easy to show by induction on n, that if E is any set, $a \in \mathrm{E}$ and $s(\mathrm{E}) \subset \mathrm{E}$ then $f(n) \in \mathrm{E}$ for all n. In the case of T′, the condition becomes $h(n, x) \in \mathrm{E}$ whenever $n \in \mathbf{N}$ and $x \in \mathrm{E}$.

Our previous implementation started with proving existence of (IND0) below, then using the choose function. The following definitions have been introduced in the previous chapter.

```
(*
Definition induction_defined0 (h: Set -> Set -> Set)  (a: Set) :=
 transfinite_defined Bnat_order
 (fun u => Yo(source u = \0c) a (h (cpred (source u))(W (cpred (source u)) u))).

Definition induction_defined (s: Set -> Set)  (a: Set) :=
 transfinite_defined Bnat_order
 (fun u => Yo(source u = \0c) a (s (W (cpred (source u)) u))).
*)
```

In the case of general transfinite induction on a set E, we proceed in a similar way; let $m = \mathrm{M}'(u)$ be the supremum of the source of u. We have obviously $m \leq n$. In the case $m < n$,

n is called the successor of m, and we can define $f(n)$ as some function of $f(m)$. If n is not a successor, an alternate method must be used (in some case $\sup_{i<n} f(i)$ is used).

Bourbaki first shows (IND), then deduces a variant of (IND0) as follows. He considers a function $h : \mathbf{N} \times \mathrm{E} \to \mathrm{E}$, where E is some fixed set. Denote by F the set $\mathbf{N} \times \mathrm{E}$. Consider the function $\psi : \mathrm{F} \to \mathrm{F}$ defined by $y \mapsto (\mathrm{pr}_1 y + 1, h(y))$ and the surjective function g with values in F defined by induction as $g(0) = (0, a)$ and $g(n+1) = \psi(g(n))$. Define $a_n = \mathrm{pr}_1(g(n))$ and $b_n = \mathrm{pr}_2(g(n))$. From $g(n) \in \mathrm{F}$, one deduces $a_n \in \mathbf{N}$, $b_n \in \mathrm{E}$ and $g(n) = (a_n, b_n)$, hence the two recurrence relations $a_{n+1} = a_n + 1$ and $b_{n+1} = h(a_n, b_n)$. Obviously $a_n = n$, thus $b_{n+1} = h(n, b_n)$, and the function associated to b_n satisfies (IND0).

Consider the following example. Given a sequence of cardinals (x_i), we consider $h(n, y) = x_{n+1} + y$ or $h(n, y) = x_{n+1}^y$. Using (IND0), there exist two sequences satisfying $y_0 = x_0$ and $y_{n+1} = x_{n+1} + y_n$ or $z_0 = x_0$ and $z_{n+1} = x_{n+1}^{z_n}$. Let's try to apply the Bourbaki method. There is a set E stable by cardinal sum that contains the range E_0 of the sequence (x_i). If E_0 is a subset of $\mathbf{N}$, just take $\mathbf{N}$, otherwise let $\mathfrak{a}$ be the supremum of E_0 (see page 78). This is an infinite cardinal, and we shall see in the next section that the set of all cardinals $\leq \mathfrak{a}$ is stable by cardinal sum. As a consequence, we can use the Bourbaki construction and $y_n \in \mathrm{E}$. Doing the same for z_n is tricky. For any set E we define $p(\mathrm{E})$ to be the set of all x^y for $x \in \mathrm{E}$ and $y \in \mathrm{E}$, and for any cardinal $\mathfrak{a}$, we define $\mathrm{E}_\mathfrak{a}$ to be the set of all cardinals $\leq \mathfrak{a}$ and $s(\mathfrak{a})$ to be the supremum of $p(\mathrm{E}_\mathfrak{a})$. Let f be the function defined by induction via s (where $f(0)$ is any cardinal such that $\mathrm{E}_{f(0)}$ contains the range of the sequence x_i). Finally, let E be the union of the sets $\mathrm{E}_{f(i)}$. Since f is increasing, if x and y are two elements of E, there is an i such that $x \in \mathrm{E}_{f(i)}$ and $y \in \mathrm{E}_{f(i)}$ hence $x^y \in \mathrm{E}_{f(i+1)} \subset \mathrm{E}$. This is a very large set (we could reduce a bit its size by defining $p(\mathrm{E})$ to be the set of all x^y for $x \in \mathrm{E}_0$ and $y \in \mathrm{E}$).

In the first version of the Software we had

```
M := fun u => supremum Bnat_order (source u)
T:= fun u => Yo (u = empty_function) a (s (W (M u) u))
```

We recognize here the quantity M′; it is defined whenever u has nonempty source, which is equivalent to say that u is not the empty function.

The Bourbaki definition is much more complicated. He starts with

$$\mathrm{D}(u) = \mathscr{E}_x(x \in \mathbf{N} \text{ and } (\exists y)((x, y) \in \mathrm{pr}_1(\mathrm{pr}_1(u)))).$$

He says that, if u is a function defined on a subset of $\mathbf{N}$, then $\mathrm{D}(u)$ is the domain. In fact, $\mathrm{D}(u)$ is by definition the intersection of the domain of the graph of u and $\mathbf{N}$. If u is a function, $\mathrm{D}(u)$ is the intersection of its source and $\mathbf{N}$. As mentioned above, in the proof we need only to consider the case where the source is a subset of $\mathbf{N}$, and $\mathrm{D}(u)$ could be replaced by the source of u. Bourbaki defines $\mathrm{M}(u)$ as being the least upper bound of $\mathrm{D}(u)$, and in a footnote says that one could change the definition of the least upper bound, in order to give a meaning to this term even when $\mathrm{D}(u)$ is unbounded. Another idea would be to consider $\mathrm{Card}(\mathrm{D}(u)) - 1$; a possible definition of $x-1$ could be x when $x = 0$ or is an infinite cardinal, the usual definition otherwise. Let ϕ be the empty function, and consider the relation

$$\mathrm{R}\{y, u\}: \quad (u = \phi \text{ and } y = a) \text{ or } (u \neq \phi \text{ and } y = \mathrm{S}\{u(\mathrm{M}(u))\}).$$

Let $\mathrm{T}\{u\}$ be the term $\tau_y(\mathrm{R}\{y, u\})$. If $u = \phi$ then $\mathrm{T}\{u\}$ is a, otherwise it is $\mathrm{S}\{u(\mathrm{M}(u))\}$; our equivalent of the if-then-else construct is Yo. The argument u of the term T will always be (this is obvious by induction) the restriction of f to the interval $[0, n-1[$. If $n = 0$, the restriction is ϕ hence $f(0) = a$, and if $n = m+1$, the restriction has source $[0, m]$, whose supremum is m, so that $f(m+1) = \mathrm{S}\{f(m)\}$.

We give here six definitions[1]. In the the first cases, the function f is assumed to be surjective, and in the other cases, the target will be a given set E. The function will satisfy one of (IND) or (IND0) or

(IND1') $\qquad f(0) = a \quad \text{and} \quad f(n+1) = g(n, f(n)) \quad \text{if} \quad n < m.$

We call this partial induction. In order to get uniqueness, we either have to restrict the source of f to the interval $[0, m]$ or specify a value $f(n)$ for $n > m$. We consider

(IND1) $\qquad f(0) = a \quad \text{and} \quad f(n+1) = g(n, f(n)) \quad \text{if} \quad n < m \quad \text{and} \quad f(n) = a \quad \text{otherwise.}$

```
Definition induction_defined1 (h: Set -> Set -> Set) a p :=
   induction_defined0 (fun n x => Yo (cardinal_lt n p) (h n x) a) a.

Definition change_target_fun f t := corresp (source f) t (graph f).

Definition induction_defined_set s a E:=
  change_target_fun (induction_defined s a) E.

Definition induction_defined_set0 h a E:=
  change_target_fun (induction_defined0 h a) E.

Definition induction_defined_set1 h a p E:=
  change_target_fun (induction_defined1 h a p) E.
```

We state some theorems that say that such a function exists and is unique.

```
Lemma induction_defined_pr1 h a p (f := induction_defined1 h a p):
    inc p Bnat ->
    ( source f = Bnat & surjection f &
      W \0c f = a &
      (forall n,  n <c p -> W (succ n) f = h n (W n f))&
      (forall n, inc n Bnat -> ~ (n <=c p) -> W n f = a)).
Lemma integer_induction1 h a p: inc p Bnat ->
  exists_unique (fun f=>  source f = Bnat & surjection f &
    W \0c f = a &
     (forall n,  cardinal_lt n p -> W (succ n) f = h n (W n f))&
     (forall n, inc n Bnat -> ~ (cardinal_le n p) -> W n f = a)).
```

We now show that the target of the function defined by induction is a subset of E under some conditions. It follows that there is a variant where the target is E. We shall not prove uniqueness, it is obvious.

```
Lemma change_target_pr f E (g:= change_target_fun f E):
   is_function f -> sub (target f) E
  -> (is_function g & source g = source f & target g = E
     & forall x, inc x (source f) -> W x g = W x f).
Lemma integer_induction_stable E g  a:
  inc a E -> (forall x, inc x E -> inc (g x) E) ->
  sub (target (induction_defined g a)) E.
Lemma integer_induction_stable0 E h a:
  inc a E -> (forall n x, inc x E -> inc n Bnat -> inc (h n x) E) ->
  sub (target (induction_defined0 h a)) E.
```

[1] two of them are shown above

```
Lemma integer_induction_stable1 E h a p:
  inc p Bnat ->
  inc a E -> (forall n x, inc x E -> cardinal_lt n p -> inc (h n x) E) ->
  sub (target (induction_defined1 h a p)) E.

Lemma induction_defined_pr_set E g a:
  let f := induction_defined_set g a E in
    inc a E -> (forall x, inc x E -> inc (g x) E) ->
     (is_function f & source f = Bnat & target f = E & W \0c f = a &
     forall n, inc n Bnat -> W (succ n) f = g (W n f)).
Lemma induction_defined_pr_set0 E h a:
  let f := induction_defined_set0 h a E in
    inc a E -> (forall n x, inc x E -> inc n Bnat -> inc (h n x) E) ->
     (is_function f & source f = Bnat & target f = E & W \0c f = a &
     forall n, inc n Bnat -> W (succ n) f = h n (W n f)).
Lemma induction_defined_pr_set1 E h a p:
  let f := induction_defined_set1 h a p E in
    inc p Bnat ->
    inc a E -> (forall n x, inc x E -> n <c p -> inc (h n x) E) ->
    (is_function f & source f = Bnat & target f = E & W \0c f = a &
      (forall n, n <c p -> W (succ n) f = h n (W n f))&
      (forall n, inc n Bnat -> ~ (n <=c p) -> W n f = a)).
```

7.3 Properties of infinite cardinals

Bourbaki claims (Lemma 1, [3, p. 186]) that every infinite set E has a subset F equipotent to **N**; the argument being that there exists a well-ordering on E. If E is isomorphic to a segment of **N**, then E is equipotent to **N** since all other segments are of the form $]\leftarrow, x[$, hence $[0, x[$, thus are finite. Otherwise, **N** is isomorphic to a segment of E, hence equipotent to a subset of E.

Our proof is simpler: since **N** is the least infinite cardinal, we have $\mathbf{N} \subset \mathrm{Card}(E)$, from wich the result follows trivially.

```
Lemma infinite_greater_countable1 E:
  infinite_set E -> (cardinal Bnat) <=c (cardinal E).
Lemma infinite_greater_countable E:
  infinite_set E -> exists F, sub F E & cardinal F = cardinal Bnat.
```

Bourbaki claims that $\mathbf{N} \times \mathbf{N}$ is equipotent to **N**: the relation $\mathrm{Card}(\mathbf{N}) \leq \mathrm{Card}(\mathbf{N} \times \mathbf{N})$ is a consequence of $\{0\} \times \mathbf{N} \subset \mathbf{N} \times \mathbf{N}$; moreover there is an injection from $\mathbf{N} \times \mathbf{N}$ into **N**. He uses expansion to base 2. Assume $x = \sum x_i 2^i$ and $y = \sum y_i 2^i$, then $\sum (2x_i + y_i)4^i$ is an injective function of x and y. We use here a different function: consider

$$f(n,m) = n + \binom{n+m+1}{2} = n + g(n+m).$$

The function g is the binomial coefficient with indices $a+1$ and 2, it is also $g(a) = a(a+1)/2$; it satisfies $g(a+1) = g(a) + a + 1$, hence $g(n+m) \leq f(n,m) < g(n+m+1)$. This relation shows that $n+m$ is uniquely defined by $f(n,m)$, from which injectivity of f follows. Consider x and the least a such that $x < g(a)$. Then $x = f(n,m)$, where n and m are the unique integers satisfying $n+m+1 = a$ and $x = n + g(a-1)$. This shows that f is bijective.

```
Lemma equipotent_N2_N: (product Bnat Bnat) \Eq Bnat. (* 87 *)
```

We show here Theorem 2 ([3, p. 186]): for every infinite cardinal $\mathfrak{a}$, we have $\mathfrak{a} = \mathfrak{a}^2$. The proof is by induction (in reality, Zorn's lemma, since there is no induction for infinite cardinals).

We consider an infinite set E, and study bijections of the form $\psi : A \to A \times A$, where $A \subset E$. This is the same as studying the set $\mathfrak{M}$ of functions defined on a subset A of E that are injective, whose target is $E \times E$, and whose range is $A \times A$. Bourbaki says "It is immediately seen that $\mathfrak{M}$ is inductive". (The example in section 3.4 is the set of functions without these conditions). The proof is similar to the one that states that there exists an isomorphism between well-ordered sets. The proof is tedious but straightforward (50 lines of proof).

Since E is infinite there exists a subset D equipotent to **N**, hence (previous theorem) a bijection between D and $D \times D$. This gives us a element ψ_0 of $\mathfrak{M}$. Let $\mathfrak{M}_0$ be the set of elements of $\mathfrak{M}$ that extend ψ_0. This set is inductive as well. We consider a maximal element with source F and cardinal $\mathfrak{b}$. We have Card(F) ≥ Card(D), so that $\mathfrak{b}$ is infinite. We have $\mathfrak{b} = \mathfrak{b}^2$. If $\mathfrak{c} \leq \mathfrak{b}$ then $\mathfrak{b} \leq \mathfrak{c} + \mathfrak{b} \leq 2\mathfrak{b} \leq \mathfrak{b}^2 = \mathfrak{b}$, hence $\mathfrak{c} + \mathfrak{b} = \mathfrak{b}$. In particular $\mathfrak{b} = 2\mathfrak{b} = 3\mathfrak{b}$.

Our theorem is true if $\mathfrak{b}$ = Card(E). Assume otherwise $\mathfrak{b} < \mathfrak{a}$. The cardinal $\mathfrak{c}$ of the complementary of F in E is $\geq \mathfrak{b}$ (a consequence of the theorem will be $\mathfrak{c} = \mathfrak{a}$). Hence, there exists a subset Y of E disjoint from F with cardinal $\mathfrak{b}$. Let $Z = Y \cup F$. The relation $\mathfrak{b} = 3\mathfrak{b}^2$ implies that Y is equipotent to $(Z \times Z) - (F \times F)$ hence Z is equipotent to $Z \times Z$. This contradicts maximality. .

```
Theorem equipotent_inf2_inf a: infinite_c a ->
  a ^c \2c = a. (* 227 *)
```

By induction, $\mathfrak{a}^n = \mathfrak{a}$ if $\mathfrak{a}$ is an infinite cardinal and $n \geq 1$ is an integer. As a consequence, if $(\mathfrak{a}_i)_{i\in I}$ is a finite family of non-zero cardinals, if the largest one is an infinite cardinal $\mathfrak{a}$, then the product is $\mathfrak{a}$. If $\mathfrak{a}_i \leq \mathfrak{a}$ then $\sum \mathfrak{a}_i \leq \mathfrak{a}$ and we have equality if one of the cardinals is $\mathfrak{a}$. The cardinals can be zero, and the index set can be infinite, provided that Card(I) ≤ *a*. Finally, if $\mathfrak{a}$ and $\mathfrak{b}$ are two non-zero cardinals, one of them being infinite, then the sum and the product is the greatest of them.

Note. Bourbaki writes "$\sup(\mathfrak{a}, \mathfrak{b})$" instead of "the greatest of them"; our function sup takes three arguments, one of them being the order; in this case, there is no order (because there is no set containing all cardinals). We know that the supremum of any family of cardinals exists, so that the supremum of two cardinals is well-defined, but we have no notation for this operation. Note also that using a lemma of the form: if $\mathfrak{a}$ is infinite or $\mathfrak{b}$ is infinite, then $\mathfrak{a} + \mathfrak{b} = \sup(\mathfrak{a}, \mathfrak{b})$ is uneasy, since this means that $\mathfrak{a} + \mathfrak{b}$ is at least $\mathfrak{a}$, at least $\mathfrak{b}$, and at most any $\mathfrak{c}$ that is at least $\mathfrak{a}$ and at least $\mathfrak{b}$. It is much easier to say: if $\mathfrak{a}$ is infinite and $\mathfrak{b} \leq \mathfrak{a}$ then $\mathfrak{a} + \mathfrak{b} = \mathfrak{a}$, and use commutativity of addition when needed. Note that $\mathfrak{b}$ infinite implies $\mathfrak{a}$ infinite, and $\mathfrak{b}$ can be zero (since $\mathfrak{a}$ is infinite, it is clearly non-zero). In the case of a product, we need the condition $\mathfrak{b} \neq 0$.

```
Lemma square_of_infinite a: infinite_c a ->
  a *c a = a.
Lemma power_of_infinite a n: infinite_c a -> inc n Bnat ->
  n <> \0c -> a ^c  n = a.
Lemma power_of_infinite1 a n: infinite_c a -> inc n Bnat ->
   (a ^c n) <=c a.
Lemma finite_family_product a f: fgraph f ->
  finite_set (domain f) -> infinite_c a ->
  (forall i, inc i (domain f) -> (V i f) <=c a) ->
```

```
  (forall i, inc i (domain f) -> V i f <> \0c) ->
  (exists j,  inc j (domain f) & (V j f) = a) ->
  card_prod f = a.
Lemma product2_infinite a b: b <=c a ->
  infinite_c a -> b <> \0c -> a *c b = a.
Lemma product2_infinite1 a b: b <=c a ->
  infinite_c a -> (a *c b) <=c a.
Lemma product2_infinite2 a b: finite_c b ->
  infinite_c a -> (a *c b) <=c a.
Lemma product2_infinite4 a b c:
   a <=c c -> b <=c c -> infinite_c c -> a *c b <=c c.
Lemma notbig_family_sum a f: fgraph f ->
  infinite_c a ->(cardinal (domain f)) <=c a ->
  (forall i, inc i (domain f) -> (V i f) <=c a) ->
  (card_sum f) <=c a.
Lemma notbig_family_sum1 a f: fgraph f ->
  infinite_c a -> (cardinal (domain f)) <=c a ->
  (forall i, inc i (domain f) ->  (V i f) <=c a) ->
  (exists j,  inc j (domain f) & (V j f) = a) ->
  (card_sum f) = a.
Lemma sum2_infinite1 a: infinite_c a -> a +c a = a.
Lemma sum2_infinite a b: b <=c a ->
  infinite_c a -> a +c b = a.
Lemma sum2_infinite2 a b c: is_cardinal c -> infinite_c a ->
  b <c a -> a = b +c c -> a = c.


Lemma infinite_pow x y: infinite_c x ->  is_cardinal y -> y <> \0o ->
   infinite_c (x ^c y).
Lemma infinite_pow2 x y: infinite_c x ->   infinite_c y ->
   infinite_c (x ^c y).
```

7.4 Countable sets

A countable set is one that is equipotent to a subset of **N**. Proposition 2 [3, p. 188] says that an infinite countable set is equipotent to **N**. We rewrite this as: a countable set is finite or equipotent to **N**.

Proposition 1 [3, p. 188] says that a subset of a countable set is countable; the product of a finite family of countable sets is countable; the union of a countable family of countable sets is countable.

Proposition 3 [3, p. 189] says that an infinite set E has a partition $(X_\iota)_{\iota\in I}$ where X_ι is infinite countable and I is equipotent to E. Proposition 4 [3, p. 189] says that if f is a function from E onto F, such that F is infinite and $\overset{-1}{f}\langle\{x\}\rangle$ is countable for any $x \in$ F, then F is equipotent to E.

```
Definition countable_set E:= equipotent_to_subset E Bnat.

Lemma cardinal_comp_singl_inf  E x:
  infinite_set E -> cardinal E = cardinal (complement E (singleton x)).
Lemma countable_prop E:
  countable_set E <-> (cardinal E) <=c (cardinal Bnat).
Lemma countable_finite_or_N E: countable_set E ->
  finite_c (cardinal E) \/ cardinal E = cardinal Bnat.
Lemma countable_finite_or_N_b E: countable_set E ->
```

```
  finite_set E \/ equipotent E Bnat.
Lemma countable_finite_or_N_c E: countable_set E ->
  infinite_set E -> equipotent E Bnat.
Theorem countable_subset E F: sub E F -> countable_set F  ->
  countable_set E.
Theorem countable_product f: fgraph f ->
  finite_set (domain f) ->
  (forall i, inc i (domain f) -> countable_set (V i f)) ->
  countable_set (productb f).
Theorem countable_union f:  fgraph f ->
  countable_set (domain f) ->
  (forall i, inc i (domain f) -> countable_set (V i f)) ->
  countable_set (unionb f).
Lemma card_bnat_not_zero: cardinal Bnat <> \0c.
Theorem infinite_partition E: infinite_set E ->
  exists f, partition_fam f E & (domain f) \Eq E &
    (forall i, inc i (domain f) -> (infinite_set (V i f) &
      countable_set (V i f))). (* 41 *)
Theorem countable_inv_image f: surjection f ->
  (forall y, inc y (target f) ->
    countable_set (inv_image_by_fun f (singleton y))) ->
  infinite_set (target f) ->
  (source f) \Eq (target f). (* 44 *)
```

Proposition 5 [3, p. 189] says that the set $\mathfrak{F}$ of finite subsets of an infinite set E is equipotent to E. The proof of Bourbaki is not clear. He defines $\mathfrak{F}_n$ as the set of all subsets with n elements of E and claims $\mathrm{Card}(\mathfrak{F}_n) \leq \mathrm{Card}(E)$. Thus, the cardinal of the union of these sets is at most $\sum_{n\in\mathbf{N}} \mathrm{Card}(E) = \mathrm{Card}(E)$. Thus $\mathrm{Card}(\mathfrak{F}) \leq \mathrm{Card}(E)$; equality holds because the set of singletons is equipotent to E and is a subset of $\mathfrak{F}$.

The Bourbaki claim is: for every $X \in \mathfrak{F}_n$ there is a bijection from $[1,n]$ onto X, so that the cardinal of $\mathfrak{F}_n$ is at most the cardinal of the set of functions from $[1,n]$ into X which is $\mathrm{Card}(E^n) = \mathrm{Card}(E)$. Our proof is as follows.

For every $X \in \mathfrak{F}_n$ there is a bijection $[1,n] \to X$, hence an injective function from $[1,n]$ into E with range X, but it is not unique. Let K be the set of injections from $[1,n]$ into E. Let f be the function that associates to each element of K its range. The target of this function is clearly $\mathfrak{F}_n$. Let Q be the set of permutations of $[1,n]$, and c it cardinal. This is a non-zero integer. We pretend that the cardinal of $f^{-1}\langle\{x\}\rangle$ is c. We take an element g in this set (it exists, by the remark above). For every permutation h of $[1,n]$, we consider $g \circ h$. This operation is a bijection from Q onto $f^{-1}\langle\{x\}\rangle$. Surjectivity of this operation uses the fact that for any k there exists g such that $k = g \circ h$, if the ranges are the same (since h is injective) and this function is surjective. It is bijective since it is an endomorphism of a finite set. We can now apply the shepherd's principle. The product of the cardinal a of $\mathfrak{F}_n$ and c is the cardinal b of the set of injections, that is smaller than the cardinal d of the set of functions from $[1,n]$ into E. If $n = 0$, we clearly have $a \leq \mathrm{Card}(E)$; otherwise $d = \mathrm{Card}(E)$. Hence $ac = b \leq \mathrm{Card}(E)$. If a is finite, we have $a \leq \mathrm{Card}(E)$; but if a is infinite, we have $a = ac$ (since c is non-zero finite). This implies $a \leq \mathrm{Card}(E)$.

As a corollary, the set of finite sequences with value into E is equipotent to E; in fact, this set is the union of the sets of functions from I into E (that has the same cardinal as E^I) for all finite subsets I of **N**. Since E^I and I are equipotent and since the set of finite subsets of **N** is countable, the result is immediate.

```
Theorem infinite_finite_subsets E: infinite_set E ->
```

```
  (Zo (powerset E) (fun z => finite_set z)) \Eq  E. (* 177 *)

Lemma infinite_finite_sequence E: infinite_set E ->  (* 48 *)
 (Zo(set_of_sub_functions Bnat E) (fun z=> finite_set (source z))) \Eq E.
```

A set is said to have *the power of the continuum* if it is equipotent to $\mathfrak{P}(\mathbf{N})$. In this case, its cardinal is $2^{\aleph_0}$, and the set is not countable.

7.5 Stationary sequences

A sequence $(x_n)_{n\in\mathbf{N}}$ is *stationary* if there exists an integer m such that $x_n = x_m$ for $n \geq m$. We define here the notion of increasing and decreasing sequences. It is the graph of an increasing function where the source is **N** with its natural order. Note that a decreasing sequence is increasing for the opposite order.

```
Definition stationary_sequence f :=
  fgraph f & domain f = Bnat &
  exists m,  inc m Bnat & forall n, inc n Bnat -> m <=c n ->
    V n f = V m f.
Definition increasing_sequence f r:=
  fgraph f & domain f = Bnat & sub (range f) (substrate r) &
  forall n m, inc n Bnat  -> inc m Bnat -> n <=c m ->
    gle r (V n f) (V m f).

Definition decreasing_sequence f r:=
  fgraph f & domain f = Bnat & sub (range f) (substrate r) &
  forall n m, inc n Bnat  -> inc m Bnat -> n <=c m ->
    gle r  (V m f) (V n f).
```

Proposition 6 [3, p. 190] says that, if E is an ordered set, each non-empty set has a maximal element if and only if each increasing sequence is stationary. We start with a lemma: a function f such that $f(n) \leq f(n+1)$ is increasing (by induction on m, we have $f(n) \leq f(n+m)$). By definition increasing_fun f r r' says that the target of f is the substrate of r; in our case, it is merely a subset, so that the definition will not be used: we show that the graph of f is an increasing sequence.

The Proposition is shown as follows. Given an increasing sequence, its range is non-empty. It it has a maximal element x_n and $m \geq n$ then $x_n \leq x_m$ implies $x_m = x_n$. Conversely, assume that we have a set A that has no maximal element. For each x, the subset T_x of elements of A greater than x is non-empty. This means that the product $\prod \mathrm{T}_x$ is non-empty, hence there is a function $f : \mathrm{A} \to \mathrm{A}$ such that $f(x) > x$ and a sequence $x_{n+1} = f(x_n)$. This sequence is strictly increasing, absurd.

As a consequence a totally ordered set E is well-ordered if and only if each decreasing sequence is stationary (to show that it is well-ordered, we consider the opposite order; thus every non-empty set has a minimal element, this element is the least element, since all subsets of E are directed). Moreover, an increasing sequence in a finite ordered set has a maximal element.

```
Lemma increasing_prop f r: order r ->
  is_function f -> source f = Bnat -> sub (target f) (substrate r) ->
  (forall n, inc n Bnat -> gle r (W n f) (W (succ n) f)) ->
  increasing_sequence (graph f) r.
```

```
Lemma decreasing_prop f r: order r ->
  is_function f -> source f = Bnat -> sub (target f) (substrate r) ->
  (forall n, inc n Bnat -> glt r (W (succ n) f)  (W n f)) ->
  decreasing_sequence (graph f) r.
Theorem  increasing_stationary r: order r ->
  ((forall X, sub X (substrate r) -> nonempty X ->
     exists a,  maximal_element (induced_order r X) a) <->
  (forall f, increasing_sequence f r -> stationary_sequence f)). (* 47 *)
Theorem  decreasing_stationary r: total_order r ->
  ( (worder r) <->
   (forall f, decreasing_sequence f r -> stationary_sequence f)). (* 32 *)
Theorem  finite_increasing_stationary r: order r ->
  finite_set (substrate r) ->
  (forall f, increasing_sequence f r -> stationary_sequence f).
```

Proposition 7 [3, p. 190] says that if E is noetherian (every non-empty set has maximal element) and if F is a subset of E such that for $a \in$ E, if $\forall x, x > a \implies x \in$ F then $a \in$ F; then F = E.

```
Theorem noetherian_induction r F:  order r ->
  (forall X, sub X (substrate r) -> nonempty X ->
    exists a,  maximal_element (induced_order r X) a) ->
  sub F (substrate r) ->
  (forall a, inc a (substrate r) -> (forall x, glt r a x -> inc x F)
    -> inc  a F)
  -> F = substrate r.
```

Chapter 8

Ordinal numbers

What follows is not part of the main text of Bourbaki, but comes from exercises. Ordinal sums are defined in Ex1.3 (10.1), order-types in in Ex2.13 (10.2) and ordinal numbers in Ex2.14 (10.2). Ex2.20 (10.2) defines "pseudo-ordinals", and an identification between pseudo-ordinals and ordinals. These pseudo-ordinals are the von Neumann ordinals introduced in section 4.1. In what follows, the term "ordinal" has to be understood as "von Neumann ordinal". The Cantor Normal Form and properties of multiplication comes from Cantor [5].

8.1 Order sums and products

In Exercise 2.13 (10.2), Bourbaki says "The order-type of the ordinal sum of the family of sets [...] is called the *ordinal sum* of the order-types [...] and is denoted by $\sum_{\iota\in I}\lambda_\iota$. The order-type of the lexicographic product of the family of sets [...] is called the *ordinal product* of the family [...] and is denoted by $\mathsf{P}_{\iota\in I}\lambda_\iota$.".

The first definition makes the term "ordinal sum" ambiguous. In what follows, we shall use the terms "order sum" and "order product" for some operations on orderings, and denote them by $\sum_r$, $\prod_r$, and in the case of two arguments, we shall use the notation $a+_r b$ and $a\cdot_r b$. We shall use the subscript t in the case of order-types, and the subscript o in the case of ordinals. With these conventions, the definition and characteristic properties of $\sum_t$ will be:

$$\sum_t \mathrm{E}_i = \mathrm{Ord}(\sum_r \mathrm{E}_i), \qquad \mathrm{Ord}(\sum_r \mathrm{E}_i) = \sum_t \mathrm{Ord}(\mathrm{E}_i). \tag{1}$$

Consider a family of sets $(\mathrm{X}_\iota)_{\iota\in I}$, an ordering $\leq$ on I and an ordering $\leq_\iota$ on each X_ι. The product $\prod_{\iota\in I}\mathrm{X}_\iota$ is the set of all families $x=(x_\iota)_{\iota\in I}$, where $x_\iota\in\mathrm{X}_\iota$ for any $\iota\in I$. There are two possible orderings on the product: the simple ordering defined in Section 2.4, and the lexicographic order defined in Section 3.6. We shall consider here the lexicographic ordering. We assume the index set I well-ordered, so that if $x\neq y$ there is a least λ such that $x_\lambda\neq y_\lambda$. We say $x<y$ if $x_\lambda <_\lambda y_\lambda$. The condition $x\leq y$ can be restated as "either $x=y$, or there is an index j such that $x_j <_j y_j$, and whenever $i<j$ we have $x_i=y_i$".

In the lemmas that follow, orprod will be a prefix for the product order, and oprod will be a prefix for the ordinal product. The same conventions will be used for orprod2 and orsum. The prefix OS indicates that the conclusion of the lemma is that some object is an ordinal.

```
Lemma prod_of_substrates_pr i z g:
```

```
  inc i (domain g) -> inc z (prod_of_substrates g) ->
  inc (V i z) (substrate (V i g)).
Lemma orprod_gle g x x':
  order_prod_a r g ->
  gle (order_prod r g) x x' =
  (inc x (prod_of_substrates g) & inc x' (prod_of_substrates g) &
    (x= x' \/ exists j, inc j (substrate r)  &
      glt (V j g) (V j x) (V j x') &
      forall i, glt r i j -> V i x = V i x')).
```

The sum (or disjoint union) of a family (X_ι) indexed by a set I is denoted by $\sum_{\iota\in I} X_\iota$. It is the set of all x which are pairs (x_1, x_2) with $x_2 \in I$ and $x_1 \in E_{x_2}$. Given an ordering $\leq$ on I and a family of orderings $\leq_\iota$ on each X_ι we can define an ordering on the sum as follows: $x \leq y$ when either $x_2 < y_2$, or $x_2 = y_2 = \lambda$ and then $x_1 \leq_\lambda y_1$. In Exercise 1.3 (10.1) it is assumed $X_j \neq \emptyset$, but this forbids zero in a sum.

```
Definition sum_of_substrates g := disjoint_union (fam_of_substrates g).
Definition orsum_ax r g:=
  order r & substrate r = domain g & order_fam g.

Definition order_sum_r r g x x' :=
    (glt r (Q x) (Q x') \/ (Q x = Q x' & gle (V (Q x) g) (P x) (P x'))).
Definition order_sum r g :=
  graph_on (order_sum_r r g)  (sum_of_substrates g).
```

These lemmas characterize the substrate of the order sum.

```
Lemma du_index_pr1 g x: inc x (sum_of_substrates g) ->
  (inc (Q x) (domain g) &  inc (P x) (substrate (V (Q x) g)) & is_pair x).
Lemma inc_disjoint_union1 g x y:
  inc y (domain g) -> inc x (substrate (V y g)) ->
  inc (J x y) (sum_of_substrates g).
```

The disjoint union of the family $\alpha \mapsto E_1$ and $\beta \mapsto E_2$ is $E_1 \times \{\alpha\} \cup E_2 \times \{\beta\}$. In the special case where α and β are TPa and TPb, we call this the canonical disjoint union. This is the substrate of the ordinal sum $E_1 + E_2$. The substrate of the lexicographic product $E_1 \cdot E_2$ is known as product2; it is the set of functional graphs x such that $x_\alpha \in E_2$ and $x_\beta \in E_1$, it is set-isomorphic to $E_2 \times E_1$ (note that this is set-isomorphic to $E_1 \times E_2$, but these set-isomorphisms are not used, we shall consider only order-preserving isomorphisms).

```
Definition canonical_du2 a b :=
  disjoint_union (variantLc a b).

Lemma canonical_du2_rw a b:
  canonical_du2 a b =
  union2 (product a (singleton TPa)) (product b (singleton TPb)).
Lemma canonical_du2_pr a b x:
  inc x (canonical_du2 a b) <-> (is_pair x &
  ((inc (P x) a & Q x = TPa) \/ (inc (P x) b & Q x = TPb))).
Lemma canonical_du2_pr1 a b x:
  inc x (canonical_du2 a b) ->
  ((inc (P x) a & Q x = TPa) \/ (inc (P x) b & Q x = TPb)).
Lemma canonical_du2_pr2 a b x:
  inc x (canonical_du2 a b) -> (Q x = TPa \/ Q x = TPb).
```

```
Lemma canonical_du2_pra a b x:
  inc x a -> inc (J x TPa) (canonical_du2 a b).
Lemma canonical_du2_prb a b x:
  inc x b -> inc (J x TPb) (canonical_du2 a b).
Lemma canonical2_substrate r r':
  fam_of_substrates (variantLc r r') = Lvariantc (substrate r) (substrate r').
```

We show here that this definition induces an order on the disjoint union.

```
Section OrderSumBasic.
Variables r g: Set.
Hypothesis osa: orsum_ax r g.

Lemma orsum_or: order (order_sum r g).
Lemma orsum_sr:
  substrate (order_sum r g) = sum_of_substrates g.

Lemma orsum_gle x x':
  gle (order_sum r g) x x' <->
  (inc x (sum_of_substrates g) & inc x' (sum_of_substrates g) &
    order_sum_r r g x x').
Lemma orsum_gle1 x x':
  gle (order_sum r g) x x' ->
  (glt r (Q x) (Q x') \/ (Q x = Q x' & gle (V (Q x) g) (P x) (P x'))).
Lemma orsum_gle2  a b a' b':
  gle (order_sum r g) (J a b) (J a' b')  ->
  (glt r b b' \/ (b = b' & gle (V b g) a a')).
Lemma orsum_gle_id x x':
  gle (order_sum r g) x x' -> gle r (Q x) (Q x').
End OrderSumBasic.
```

We consider now the case of the sum and product of two sets. This operation is non-commutative, and we shall use our canonical doubleton as ordering. Recall that we have two distinguished elements, TPa and TPb, let's call them α and β. The canonical doubleton is the set $\{\alpha,\beta\}$, ordered by $\alpha < \beta$. This is a well-ordering. Note the ordering of the product: it is so that $x + x = x \cdot 2$.

```
Definition order_prod2 r r' :=
  order_prod canonical_doubleton_order (variantLc r' r).

Definition order_sum2 r r' :=
  order_sum canonical_doubleton_order (variantLc r r').

Lemma order_sp_axioms r r':
  order r -> order r' -> order_fam  (variantLc r r').

Section OrderSum2Basic.
Variables r r': Set.
Hypotheses (or: order r) (or': order r').

Lemma orsum2_axioms: orsum_ax canonical_doubleton_order (variantLc r r').
Lemma orprod2_axioms: orprod_ax canonical_doubleton_order (variantLc r' r).
Lemma orsum2_or: order (order_sum2 r r').
Lemma orprod2_or: order (order_prod2 r r').
Lemma orsum2_sr:
```

```
  substrate (order_sum2 r r') = canonical_du2 (substrate r) (substrate r').
Lemma orprod2_sr:
  substrate (order_prod2 r r') = product2 (substrate r') (substrate r).
```

The ordering on $E_1 + E_2$ is defined by $x \leq_s y$ if and only if either $\mathrm{pr}_2 x = \mathrm{pr}_2 y = \alpha$ and $\mathrm{pr}_1 x \leq \mathrm{pr}_1 y$ (in E_1), or $\mathrm{pr}_2 x = \mathrm{pr}_2 y = \beta$ and $\mathrm{pr}_1 x \leq \mathrm{pr}_1 y$ (in E_2), or $\mathrm{pr}_2 x = \alpha$ and $\mathrm{pr}_2 y = \beta$. Note that $u = \beta$ can be replaced by $u \neq \alpha$.

In the case of a product $E \cdot I$, we have $x < y$ if either $x_\alpha < y_\alpha$ in I or if $x_\alpha = y_\alpha$, and $x_\beta < y_\beta$ in E.

```
Lemma orsum2_gle x x':
  gle (order_sum2 r r') x x' <->
  (inc x (canonical_du2 (substrate r) (substrate r')) &
    inc x' (canonical_du2 (substrate r) (substrate r')) &
    ((Q x = TPa & Q x' = TPa & gle r (P x) (P x'))
      \/ (Q x <> TPa & Q x' <> TPa & gle r' (P x) (P x'))
      \/ (Q x = TPa & Q x' <> TPa))).
Lemma orsum2_gle_spec  x x':
  inc x (substrate r) -> inc x' (substrate r') ->
  glt (order_sum2 r r') (J x TPa) (J x' TPb).
Lemma orprod2_gle a b:
  gle (order_prod2 r r') a b <->
  (inc a  (product2 (substrate r') (substrate r))
    & inc b  (product2 (substrate r') (substrate r))
    & ( (V TPa a = V TPa b & gle r (V TPb a) (V TPb b))
      \/ (glt r' (V TPa a)(V TPa b)))).  (* 26 *)
End OrderSum2Basic.
```

Exercise 2.10 (10.2) says that $E \cdot I$ is isomorphic to the sum $\sum_{\iota \in I} E_\iota$ where each E_ι is equal to E. The isomorphism is simply $x \mapsto (x_\beta, x_\alpha)$.

```
Lemma order_prod_pr:
  (order_prod2 r r') \Is (order_sum r' (cst_graph (substrate r') r)).  (* 35 *)
```

If I and each E_i are well-ordered, so is the ordinal sum. If moreover I is finite, then the product is also well-ordered (for the converse, see Exercises 2.9 (10.2) and 2.11 (10.2)). If I and each E_i are totally ordered, so is the ordinal sum; the converse is true (provided that E_i is non-empty). The ordinal sum or product of two well-ordered sets are well-ordered.

```
Lemma orsum2_totalorder r r':
  total_order r -> total_order r' -> total_order (order_sum2 r r').

Lemma orsum_wor r g:
  worder r -> substrate r = domain g -> fgraph g ->
  (forall i, inc i (domain g) -> worder (V i g)) ->
  worder (order_sum r g).  (* 42 *)
Lemma orprod_wor r g:
  order_prod_a r g ->
  (forall i, inc i (domain g) -> worder (V i g)) ->
  finite_set (substrate r) ->
  worder (lexicographic_order r  g). (* 138 *)
Lemma finite_set_scdo:  finite_set (substrate canonical_doubleton_order).
Lemma orprod2_wor r r':
  worder r -> worder r' ->  worder (order_prod2 r r').
Lemma orsum2_wor r r':
  worder r -> worder r' -> worder (order_sum2 r r').
```

8.2 Order types

In Exercise 2.13 (10.2), Bourbaki defines $\mathrm{Is}(\Gamma,\Gamma')$ as the relation "Γ is an ordering (on E) and Γ' is an ordering (on E$'$), and there exists an isomorphism of E, ordered by Γ, onto E$'$, ordered by Γ'". Exercise 2.13(a) says that "$\mathrm{Is}(\Gamma,\Gamma')$ is an equivalence relation on every set whose elements are orderings" (see section 4.1.1.) The order-type of Γ, denoted $\mathrm{Ord}(\Gamma)$, is $\tau_\Delta(\mathrm{Is}(\Gamma,\Delta))$; it is some ordering isomorphic to Γ. We shall admit the existence of an order-type. Since this is rarely used, we shall introduce this axiom in a separate file, see Section 8.22.

```
(*
Parameter order_type: Set -> Set.
Axiom order_type_exists:
  forall x, order x -> order_isomorphic x (order_type x).
Axiom order_type_unique:
  forall x y, order_isomorphic x y -> (order_type x = order_type y).
*)
```

In what follows, we shall consider only ordinals. Given a well-ordering r, we denote by $\mathrm{ord}(r)$ the von Neumann ordinal associated to r; this set is naturally ordered by $o(\mathrm{ord}(r))$ which is order-isomorphic to r.

```
Lemma ordinal_p8 r r': worder r -> worder r' ->
  r \Is r' -> ordinal r = ordinal r'.
Lemma ordinal_p9 r E: worder r -> is_ordinal E ->
  r \Is (ordinal_o E) -> ordinal r = E.
Lemma ordinal_p10 x: is_ordinal x -> worder (ordinal_o x).
Lemma ordinal_p11 x: is_ordinal x -> order (ordinal_o x).
```

We list some trivial lemmas of ordinal ordering.

```
Lemma worder_invariance r r':
  r \Is r' -> worder r -> worder r'. (* 28 *)
Lemma ordinal_pr51 x: is_ordinal x -> order (ordinal_o x).
Lemma ordinal_pr52 x: is_ordinal x -> total_order  (ordinal_o x).
Lemma ordinal_le_order_r: order_r ordinal_le.
Lemma order_le_reflexive x: order x -> order_le x x.
```

Exercise 2.14(c) (page 332) says "Let α be an ordinal. Show that the relation "ξ is an ordinal and $\xi \leq \alpha$" is collectivizing in ξ, and that the set O_α of ordinals $< \alpha$ is a well-ordered set such that $\mathrm{Ord}(\mathrm{O}_\alpha) = \alpha$. We shall often identify O_α with α." The set of ordinals $\leq \alpha$ is denoted by O'_α. With Bourbaki's notations, the relation "ξ is an ordinal and $\xi \leq \alpha$" is equivalent to "ξ is the order-type of a segment of α", while $\xi < \alpha$ is equivalent to "ξ is the order-type of a strict segment S_x of α". These two relations are thus collectivizing. Moreover, the mapping $\xi \to x$ is an order-isomorphism, $\mathrm{O}_\alpha \to \alpha$. As a consequence O_α is well-ordered and its ordinal is α. (See page 390 for the initial implementation using Bourbaki ordinals.)

In the case of von Neumann ordinals, we have the trivial result $\xi <_{\mathrm{ord}} \alpha \iff \xi \in \alpha$ and $\xi \leq_{\mathrm{ord}} \alpha \iff \xi \in \alpha^+$. Note that the restriction of $\leq_{\mathrm{ord}}$ to α is $o(\alpha)$, thus it is a well-ordering and its ordinal is α.

```
Lemma set_ord_le_rw a x: is_ordinal a ->
```

```
  x <=o a = inc x (succ_o a).
Lemma set_ord_lt_rw a x: is_ordinal a ->
  x <o a = inc x a.
Lemma set_ord_lt_prop3 a: is_ordinal a ->
  ordinal (graph_on ordinal_le a) = a.
```

We have shown that a total ordering of a finite set is a well-ordering, and that two totally ordered finite sets are isomorphic if they have the same cardinal. Thus, to each finite cardinal is associated a unique ordinal. Let n be an integer, and consider the interval $[0,n[$ (with the ordering induced by that of $\mathbf{N}$). It has n elements, and its ordinal is called the n-th ordinal. If we define a cardinal as the least ordinal equipotent to it, it follows that $n = \mathrm{ord}([0,n[)$. If we consider von Neumann ordinals, we have $[0,n[= n$ (the trick is that elements of $[0,n[$ are all cardinals $< n$, while elements of n are all ordinals $< n$; now $< n$ implies finite, and all finite ordinals are cardinals).

```
Lemma finite_ordinal1 n: inc n Bnat ->
   n = ordinal (interval_Bnatco n).
Lemma finite_ordinal2 n: inc n Bnat ->
   n = interval_co_0a n.
```

Consider the set of ordinals $< \omega$, ordered by $\leq_{\mathrm{ord}}$; this is the set of all integers, ordered by $\leq_{\mathbf{N}}$. The ordinal of this set is ω. Finally, we have already noticed that ω is a cardinal. Since $\omega = \mathbf{N}$, we can restate this as $\mathrm{Card}(\mathbf{N}) = \mathbf{N}$, and as: $x \in \mathbf{N} \iff x < \omega$.

```
Lemma ord_omega_pr: \omega = ordinal Bnat_order.
Lemma cardinal_Bnat: cardinal Bnat = Bnat.
Lemma lt_omega_pr x: (x <o \omega) <-> inc x Bnat.
Lemma omega_nz: \omega <> \0o.
Lemma ord_lt_1omega: \1o <o \omega.
Lemma ord_le_2omega: \2o <=o \omega.
```

The empty set is an ordered set. Its ordinal is zero. Since any set isomorphic to the empty set is empty, the ordinal zero must be the empty set (for the same reason, the cardinal zero is the empty set).

```
Lemma emptyset_or: order emptyset.
Lemma emptyset_sr : substrate emptyset = emptyset.
Lemma emptyset_wor: worder emptyset.
Lemma empty_substrate_zero  x: order x -> substrate x = emptyset ->
   x = emptyset.
Lemma ordinal_o_emptyset: ordinal_o emptyset = emptyset.
Lemma ordinal0_emptyset: \0o = emptyset.
Lemma ordinal0_pr: ordinal emptyset = \0o.
Lemma ordinal0_pr1 x: order x -> substrate x = emptyset ->
  ordinal x = \0o.
Lemma ordinal0_pr2: forall x, worder x -> ordinal x = \0o
  -> substrate x = emptyset.
Lemma succo_nz x : succ_o x <> \0o.
Lemma inc0_ord x: is_ordinal x -> x <> \0o -> inc \0o x.
```

There is a unique ordering on a singleton, and has the form $\{(x,x)\}$. Thus all singletons are order-isomorphic. They are well-ordered; their ordinal is 1.

```
Lemma singleton_wor  x:
```

```
  let E := singleton (J x x) in
   substrate E = singleton x & worder E.
Lemma singleton_order_is x y:
  (singleton (J x x)) \Is (singleton (J y y)).
Lemma singleton_order_is1 x:
  order x -> is_singleton (substrate x) ->
  exists v, x = singleton (J v v).
Lemma singleton_order_is2 x y:
  order x -> order y ->
  is_singleton (substrate x) -> is_singleton (substrate y)
  -> x \Is y.
Lemma ordinal1_pr x: ordinal (singleton (J x x)) = \1o.
Lemma singleton_ordinal x:
  order x -> is_singleton (substrate x) ->
  ordinal x = \1o.
Lemma worder_singleton1 r e:
 order r -> substrate r = singleton e -> worder r.
```

8.3 Operations on ordinals

The ordinal of the order sum (resp. order product) of the natural orderings of a family of ordinals will be called the *ordinal sum* (resp. *ordinal product*) of the family. It is defined if the index set is well-ordered, and finite in the case of a product. Compare the following definition with (1):

$$\sum_o \mathrm{E}_i = \mathrm{ord}(\sum_r o(\mathrm{E}_i)). \tag{2}$$

```
Definition ord_sum r g :=
   ordinal (order_sum r (L (domain g) (fun z => (ordinal_o (V z g))))).
Definition ord_prod r g :=
   ordinal (order_prod r (L (domain g) (fun z => (ordinal_o (V z g))))).
Definition ord_sum2 a b := ordinal (order_sum2 (ordinal_o a) (ordinal_o b)).
Definition ord_prod2 a b := ordinal (order_prod2 (ordinal_o a) (ordinal_o b)).
Definition ordinal_fam g :=
  fgraph g &  (forall x, inc x (domain g) -> is_ordinal (V x g)).

Notation "x +o y" := (ord_sum2 x y) (at level 50).
Notation "x *o y" := (ord_prod2 x y) (at level 40).

Lemma OS_sum r g:
  worder r -> substrate r = domain g -> ordinal_fam g
  -> is_ordinal (ord_sum r g).

Lemma OS_prod r g:
  worder r -> substrate r = domain g -> ordinal_fam g
  -> finite_set (substrate r)
  -> is_ordinal (ord_prod r g).
```

We state some properties of the sum or product of two order types.

```
Lemma ordinal_sp_axioms a b:
  is_ordinal a -> is_ordinal b -> ordinal_fam  (variantLc a b).
Lemma variantLc_comp a b f:
```

```
 (variantLc (f a) (f b)) =
  (L (domain (variantLc a b)) (fun z => f (V z (variantLc a b)))).
Lemma osum2_rw a b:
  a +o b = ord_sum canonical_doubleton_order (variantLc a b).
Lemma oprod2_rw a b:
  a *o b = ord_prod canonical_doubleton_order (variantLc b a).
Lemma OS_sum2  a b: is_ordinal a -> is_ordinal b -> is_ordinal (a +o b).
Lemma OS_prod2 a b: is_ordinal a -> is_ordinal b -> is_ordinal (a *o b).
```

The second relation of (1) follows from the fact that, if one replaces an ordering by an isomorphic ordering, then the sum or product is isomorphic to the initial one. We give some variants of this property.

```
Lemma orsum_invariant1 r r' f g g':
  order r -> substrate r = domain g -> fgraph g ->
  order r' -> substrate r' = domain g' -> fgraph g' ->
  order_isomorphism f r r' ->
  (forall i, inc i (substrate r) -> (V i g) \Is (V (W i f) g'))
  -> (order_sum r g) \Is (order_sum r' g'). (* 75 *)

Lemma orprod_invariant1  r r' f g g':
  worder r -> substrate r = domain g -> fgraph g ->
  order r' -> substrate r' = domain g' -> fgraph g' ->
  order_isomorphism f r r' ->
  (forall i, inc i (substrate r) -> (V i g) \Is (V (W i f) g'))
  -> (order_prod r g) \Is (order_prod r' g').   (* 120 *)

Lemma orsum_invariant2 r g g':
  order r -> substrate r = domain g -> fgraph g ->
  substrate r = domain g' -> fgraph g' ->
  (forall i, inc i (substrate r) -> (V i g) \Is (V i g'))
  ->   (order_sum r g) \Is (order_sum r g').

Lemma orprod_invariant2 r g g':
  worder r -> substrate r = domain g -> fgraph g ->
  substrate r = domain g' -> fgraph g' ->
  (forall i, inc i (substrate r) -> order_isomorphic (V i g)  (V i g'))
  ->   order_isomorphic(order_prod r g) (order_prod r g').

Lemma orsum_invariant3 r g:
  worder r -> substrate r = domain g -> worder_fam g ->
  ordinal (order_sum r g) =
    ord_sum r (L (substrate r) (fun i => ordinal (V i g))).

Lemma orprod_invariant3 r g:
  worder r -> substrate r = domain g -> worder_fam g ->
  finite_set (substrate r) ->
  ordinal (order_prod r g) =
    ord_prod r (L (substrate r) (fun i => ordinal (V i g))).

Lemma orsum_invariant4 r1 r2 r3 r4:
  r1 \Is r3 -> r2 \Is r4 ->
  (order_sum2 r1 r2) \Is (order_sum2 r3 r4).
Lemma orprod_invariant4 r1 r2 r3 r4:
  r1 \Is r3 -> r2 \Is r4 ->
  (order_prod2 r1 r2) \Is (order_prod2 r3 r4).
```

```
Lemma orsum_invariant5 a b c: worder a -> worder b ->
  (order_sum2 a b) \Is c  ->
  (ordinal a) +o (ordinal b) = ordinal c.
Lemma orprod_invariant5 a b c: worder a -> worder b ->
  (order_prod2 a b) \Is c ->
  (ordinal a) *o (ordinal b) = ordinal c.
```

Assume that λ and μ are two ordinals. We have $\sum_{\iota\in I}\mu_\iota = \mu\lambda$ whenever $\mu_\iota = \mu$ for all indices $\iota \in I$ and I is isomorphic to λ.

```
Lemma oprod_pr1 a b c:
  is_ordinal a -> is_ordinal b -> order_isomorphic (ordinal_o b) c ->
  a *c b = ord_sum c (cst_graph (substrate c) a).
```

We show here

$$\sum_{\emptyset} \mathrm{E}_i = 0, \quad \prod_{\emptyset} \mathrm{E}_i = 1, \quad \sum_{i\in\{a\}} \mathrm{E}_i = \mathrm{E}_a, \quad \prod_{i\in\{a\}} \mathrm{E}_i = \mathrm{E}_a. \tag{3}$$

```
Lemma osum_emptyset r x:
  order r -> substrate r = domain x -> fgraph x ->
  substrate r = emptyset ->
  ord_sum r x = \0o.
Lemma oprod_emptyset r x:
  worder r -> substrate r = domain x -> fgraph x ->
  substrate r = emptyset ->
  ord_prod r x = \1o.

Lemma osum_singleton r x e:
  order r -> substrate r = domain x -> fgraph x ->
  substrate r = singleton e -> is_ordinal (V e x) ->
  ord_sum r x = V e x.  (* 39 *)
Lemma oprod_singleton r x e:
  order r -> substrate r = domain x -> fgraph x ->
  substrate r = singleton e ->  is_ordinal (V e x) ->
  ord_prod r x = V e x. (* 57 *)
```

We show here that an ordinal sum remains unchanged if zero terms are removed. In a similar fashion, one can remove ones in a product. In fact, consider $\prod \mathrm{E}_i$, assume E_j is one (or an ordering whose substrate is a singleton) for $j \in \mathrm{I}-\mathrm{J}$; we know that the restriction to J is a bijection from $\prod_{\mathrm{I}} \mathrm{E}_i$ to $\prod_{\mathrm{J}} \mathrm{E}_i$. It is clearly an order isomorphism. We deduce

$$0 + x = x + 0 = x; \qquad 1 \cdot x = x \cdot 1 = x; \qquad x^+ = x + 1. \tag{4}$$

```
Lemma osum_unit1 r g j:  (* 41 *)
  orsum_ax r g -> sub j (domain g) ->
  (forall i, inc i (complement (domain g) j) -> V i g = emptyset) ->
  (order_sum r g) \IS (order_sum (induced_order r j) (restr g j)).
Lemma oprod_unit1 r g j:  (* 112 *)
  orprod_ax r g -> sub j (domain g) ->
  (forall i, inc i (complement (domain g) j)
     ->  is_singleton (substrate (V i g))) ->
  (order_prod r g) \Is (order_prod (induced_order r j) (restr g j)).
```

```
Lemma osum_unit2 r g j:
  worder r -> substrate r = domain g -> ordinal_fam g ->
  sub j (domain g) ->
  (forall i, inc i (complement (domain g) j) -> V i g = \0o) ->
  ord_sum r g = ord_sum (induced_order r j) (restr g j). (* 30 *)
Lemma oprod_unit2 r g j:
  worder r -> substrate r = domain g -> ordinal_fam g ->
  finite_set (substrate r) ->
  sub j (domain g) ->
  (forall i, inc i (complement (domain g) j) -> V i g = \1o) ->
  ord_prod r g = ord_prod (induced_order r j) (restr g j). (* 33 *)

Lemma unit_helper x y j:
  let g :=  (variantLc x y) in
    let r := canonical_doubleton_order  in
      is_ordinal x -> is_ordinal y -> sub j (domain g) ->
      (worder r & substrate r = domain g & ordinal_fam g
        & worder (induced_order r j)
        & substrate (induced_order r j) = j
        & substrate (induced_order r j) = domain (restr g j)
        & fgraph (restr g j)).

Lemma osum2_unitl x: is_ordinal x -> \0o +o x = x.
Lemma osum2_unitr x: is_ordinal x -> x +o \0o = x.
Lemma oprod2_unitr x: is_ordinal x -> x *o \1o  = x.
Lemma oprod2_unitl x: is_ordinal x -> \1o *o x = x.

Lemma ord_succ_pr x:
  is_ordinal x -> x +o \1o = succ_o x.  (* 47 *)
```

We show associativity of the ordinal sum and product:

(5a) $$\sum_{\iota\in I} E_\iota = \sum_{\lambda\in L}\Big(\sum_{\iota\in J_\lambda} E_\iota\Big), \qquad I = \sum_{\lambda\in L} J_\lambda.$$

(5b) $$\prod_{\iota\in I} E_\iota = \prod_{\lambda\in L}\Big(\prod_{\iota\in J_\lambda} E_\iota\Big), \qquad I = \sum_{\lambda\in L} J_\lambda.$$

We give two proofs; in the first proof we consider order types (thus show that two ordered sets are isomorphic by providing the isomorphism), and in the second proof we show that some ordinals are the same. In both cases, I is the ordinal sum of the family J_λ. In the second proof, this set has to be well-ordered (and finite in the case of a product); thus we assume that L and J_λ are well-ordered (and finite in the case of a product).

```
Lemma orsum_assoc_iso r g r' g':
  orsum_ax r g ->  orsum_ax r' g' ->
  r = order_sum r' g' ->
  let order_sum_assoc_aux :=
    fun l =>
    order_sum (V l g') (L (substrate (V l g')) (fun i => V (J i l) g)) in
  let  order_sum_assoc :=
    order_sum r' (L (domain g')  order_sum_assoc_aux)
  in order_isomorphism (BL (fun x=> J (J (P x) (P (Q x))) (Q (Q x)))
      (sum_of_substrates g) (substrate (order_sum_assoc)))
  (order_sum r g) (order_sum_assoc).    (* 103 *)
```

```
Lemma orprod_assoc_iso r g r' g':
  orprod_ax r g ->  orsum_ax r' g' ->
  r = order_sum r' g' ->
  worder r' ->
  (forall i, inc i (domain g') -> worder (V i g')) ->
  let order_sum_assoc_aux :=
    fun l =>
    order_prod (V l g') (L (substrate (V l g')) (fun i => V (J i l) g)) in
  let  ordinal_prod_assoc :=
    order_prod r' (L (domain g') order_sum_assoc_aux)
  in order_isomorphism (BL
     (fun z  => L (domain g') (fun l  =>
       L (substrate (V l g')) (fun j => V (J j l) z)))
      (prod_of_substrates g) (substrate (ordinal_prod_assoc)))
  (order_prod r g) (ordinal_prod_assoc). (* 128 *)

Lemma osum_assoc1  r g r' g':
  worder r -> substrate r = domain g  ->
  worder r' -> substrate r' = domain g' ->
  ordinal_fam g -> worder_fam g' ->
  r = order_sum r' g' ->
  let order_sum_assoc_aux :=
    fun l =>
    ord_sum (V l g') (L (substrate (V l g')) (fun i => V (J i l) g)) in
  ord_sum r g = ord_sum r' (L (domain g') (order_sum_assoc_aux)).  (* 35 *)
Lemma oprod_assoc1 r g r' g':
  worder r -> substrate r = domain g  ->
  worder r' -> substrate r' = domain g' ->
  ordinal_fam g -> worder_fam g' ->
  r = order_sum r' g' ->
  finite_set (substrate r) ->finite_set (substrate r') ->
  (forall i, inc i (domain g') ->  finite_set (substrate (V i g'))) ->
  let order_prod_assoc_aux :=
    fun l =>
    ord_prod (V l g') (L (substrate (V l g')) (fun i => V (J i l) g)) in
  ord_prod r g = ord_prod  r' (L (domain g') order_prod_assoc_aux). (* 37 *)
```

We may consider a set with three elements, well-order it, and use this to define $a+b+c$ or $a\cdot b\cdot c$, the sum or product of three terms. We could then partition our set, taking apart the least or greatest element, and apply twice the associativity theorem, then get

$$a+(b+c)=(a+b)+c \text{ and } a\cdot(b\cdot c)=(a\cdot b)\cdot c. \tag{6}$$

This is rather difficult; we prefer a direct proof; there is no difficulty here, except that the proofs are rather long, especially for the sum. We have similarly

$$a\cdot(b+c)=a\cdot b+a\cdot c. \tag{7}$$

Direct proof is easy. Note that there is a natural bijection between $(b+c)\cdot a$ and $b\cdot a+c\cdot a$ but it is not always order preserving.

```
Lemma osum_assoc2 a b c:
  order a -> order b -> order c ->
  (order_sum2 a (order_sum2 b c))
   \Is (order_sum2 (order_sum2 a b) c).   (* 193 *)
```

```
Lemma oprod_assoc2 a b c:
  order a -> order b -> order c ->
  (order_prod2 a (order_prod2 b c))
  \Is (order_prod2 (order_prod2 a b) c).  (* 95 *)

Lemma osum_distributive x y z:
  order x -> order y -> order z ->
  (order_prod2 z (order_sum2 x y))
   \Is (order_sum2 (order_prod2 z x) (order_prod2 z y)).   (* 149 *)

Lemma osumA a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  a +o (b +o c) = (a +o b) +o c. (* 20 *)
Lemma oprodA a b c:
  is_ordinal a ->is_ordinal b -> is_ordinal c ->
   a *o (b *o c) = (a *o b) *o c.  (* 20 *)
Lemma osum_prodD  a b c:  (* 23 *)
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  c *o (a +o b) = (c *o a) +o (c *o b).
```

If zero is a factor of a product, then the product is zero. Conversely, if the product is zero then at least one factor is zero (we state the theorem in the case of two ordinals, but it is true for any orderings).

```
Lemma oprod_zero r g:
   orprod_ax r g ->
    (exists i, inc i (domain g) & substrate (V i g) = emptyset)
  -> order_prod r g = emptyset.

Lemma oprod0r x: is_ordinal x -> x *o \0o = \0o.
Lemma oprod0l x: is_ordinal x -> \0o *o x = \0o.
Lemma oprod2_nz a b: is_ordinal a -> is_ordinal b ->
   a <> \0o -> b <> \0o -> a *o b <> \0o.
Lemma osum_x_xy x y: is_ordinal x -> is_ordinal y ->
  x +o (x *o y) = x*o (\1o +o y).
```

We consider here a functional graph f, defined on I_n, the interval $[0, n[$ ordered by $\geq_{\mathbf{N}}$, and define the sums and products of f, denoted S_{fn} and P_{fn}. If $n = 0$, the sum is zero, the product is on, if $n = 1$ both quantities are $f(0)$.

```
Definition ord_sum_expansion f n :=
  ord_sum (opposite_order (interval_Bnatco n)) f.
Definition ord_prod_expansion f n :=
  ord_prod (opposite_order (interval_Bnatco n)) f.
Definition expansion_ax f n:=
  ordinal_fam f & domain f = interval_co_0a n.

Lemma opbnat_sr n: inc n  Bnat ->
  substrate (opposite_order (interval_Bnatco n)) =  interval_co_0a n.
Lemma opbnat_wor n: inc n  Bnat ->
  worder (opposite_order (interval_Bnatco n)).


Lemma OS_sum_expansion n f:
  inc n Bnat -> expansion_ax f n ->
```

```
  is_ordinal (ord_sum_expansion f n).
Lemma ord_sum_expansion0 n f:
  n = \0c ->  expansion_ax f n ->
  ord_sum_expansion f n = \0o.
Lemma ord_prod_expansion0 n f:
  n = \0c ->  expansion_ax f n ->
  ord_prod_expansion f n = \1o.
Lemma ord_sum_expansion1 n f:
  n = \1c ->  expansion_ax f n ->
  ord_sum_expansion f n = V \0c f.
Lemma ord_prod_expansion1 n f:
  n = \1c ->  expansion_ax f n ->
  ord_prod_expansion f n = V \0c f.
```

The first two theorem are a variant of the associativity theorem. Our ordering of I_n is such that $\mathrm{S}_{f,n+1} = f(n) + \mathrm{S}_{fn}$ and $\mathrm{P}_{f,n+1} = \mathrm{P}_{fn} \cdot f(n)$.

```
Lemma ord_sum_expansion_A n m f:  (* 214 *)
  inc n Bnat ->  inc m Bnat -> expansion_ax f (n +c m) ->
  let fs := L (interval_co_0a m) (fun z => V (z +c n) f) in
  ord_sum_expansion f (n +c m) =
       (ord_sum_expansion fs m)
    +o (ord_sum_expansion (restr f (interval_co_0a n)) n).
Lemma ord_prod_expansion_A n m f:  (* 225 *)
  inc n Bnat ->  inc m Bnat -> expansion_ax f (n +c m) ->
  ord_prod_expansion f (n +c m) =
   (ord_prod_expansion (restr f (interval_co_0a n)) n)
   *o (ord_prod_expansion (L (interval_co_0a m) (fun z => V (z +c n) f)) m).
Lemma ord_sum_expansion_r  n f:
  inc n Bnat ->  expansion_ax f (succ n) ->
  ord_sum_expansion f (succ n) =
  (V n f) +o (ord_sum_expansion (restr f (interval_co_0a n)) n).
Lemma ord_prod_expansion_r  n f:
  inc n Bnat ->  expansion_ax f (succ n) ->
  ord_prod_expansion f (succ n) =
  (ord_prod_expansion (restr f (interval_co_0a n)) n) *o (V n f).
```

8.4 Basic properties of ordering

Since $a < b^+$ is $a \in b \cup \{b\}$, we have

$$\alpha < \beta \iff \alpha^+ \leq \beta \text{ and } \alpha < \beta^+ \iff \alpha \leq \beta. \tag{8}$$

```
Lemma ord_succ_lt a: is_ordinal a -> a <o (succ_o a).
Lemma ord_lt_succ a b: a <o (succ_o b) <-> a <=o b.
Lemma ord_succ_lt2 a b: a <o b <-> (succ_o a) <=o b.
Lemma ord_le_succ_succ x y: x <=o y <-> ((succ_o x) <=o (succ_o y)).
Lemma ord_lt_succ_succ x y: x <o y <-> ((succ_o x) <o (succ_o y)).
```

Some consequences

```
Lemma ord_succ_inj a b: is_ordinal a -> is_ordinal b ->
   (succ_o a  = succ_o b) -> a = b.
```

```
Lemma ord_lt_12: \1o <o \2o.
Lemma ord_lt_01: \0o <o \1o.
Lemma ord_lt_02: \0o <o \2o.

Lemma ord_le_1_succ  x: is_ordinal x -> \1o <=o (succ_o x).
Lemma ord2_lt_pr x: \2o <=o x <-> \1o <o x.
Lemma ord2_trichotomy x: is_ordinal x ->
  x = \0o \/ x= \1o \/ \2o <=o x.
Lemma ord2_trichotomy1 x: \2o <=o  x -> (x <> \0o & x <> \1o).
Lemma limit_positive x: limit_ordinal x -> \0o <o x.
Lemma limit_nz x: limit_ordinal x -> x <> \0o.
```

We state here some properties of the supremum of a set of ordinals. Recall that sup(X) is the least ordinal such that $x \leq \sup(X)$ whenever $x \in X$.

Since any ordinal α is a set of ordinals, whenever α is an ordinal, we may consider $\sup(\alpha)$. If α is a successor, say $\alpha = \beta^+$, then $\sup(\alpha) = \beta$. Otherwise, $\sup(\alpha) = \alpha$. In this case, if α is not zero, it is called a *limit ordinal*. More generally, if a non-empty set has no greatest element, its supremum is a limit ordinal.

If $X \subset Y$, then $\sup(X) \leq \sup(Y)$, more generally if X is cofinal in Y and conversely, then $\sup(X) = \sup(Y)$.

```
Definition mutually_cofinal x y :=
   ordinal_set x & ordinal_set y &
   (forall a, inc a x -> exists b, inc b y & a <=o b) &
   (forall a, inc a y -> exists b, inc b x & a <=o b).

Lemma ord_sup_pr7 a (b:= \osup a): is_ordinal a ->
  a = b \/ a = succ_o b.
Lemma ord_sup_pr8 X:
  ordinal_set X ->  ~(inc (\osup X) X) ->
  forall x, inc x X -> x <o (\osup X).
Lemma ord_sup_pr9 X:
  ordinal_set X -> nonempty X ->
  inc (\osup X) X \/ limit_ordinal (\osup X).

Lemma ord_sup_pr10  x y:
  sub x y ->  ordinal_set y ->
  (\osup x) <=o (\osup y).
Lemma ord_sup_pr11 x y:
  mutually_cofinal x y -> \osup x = \osup y.
Lemma ord_sup_pr12 X f g:
  (forall x, inc x X -> f x = g x) ->
  \osup (fun_image X f) = \osup (fun_image X g).
Lemma ord_sup_pr13  x y:
  sub x y ->  ordinal_set y ->
  (forall a, inc a y -> exists b, inc b x & a <=o b) ->
  \osup x = \osup y.
Lemma ord_sup_pr14 A x: ordinal_set A -> x <o (\osup A) ->
  exists z, inc z A & x <=o z.
```

Taking the supremum of the set of all ordinals yields a contradiction; thus there is no set containing all ordinals. The supremum of a countable set of countable ordinals is a *countable ordinal*, that is, an ordinal x such that $\mathrm{Card}(x) \leq \mathrm{Card}(\mathbf{N})$.

```
Definition countable_ordinal x :=
```

```
   is_ordinal x & cardinal x <=c (cardinal Bnat).

Lemma ordinal_not_collectivizing E:
   (forall x, is_ordinal x -> inc x E) -> False.
Lemma countable_ordinal_sup E:
  countable_set E -> (forall x, inc x E -> countable_ordinal x)->
  countable_ordinal (\osup E).
```

8.5 Normal ordinal functional symbols

Whenever T is a term (that depends on a parameter x) such that T$\{x\}$ is an ordinal whenever x is an ordinal), we say that T is an *ordinal functional symbol*. In the definition of normal OFS that follows, we shall assume f strictly increasing, so that $x <_{\mathrm{ord}} y$ implies $f(x) <_{\mathrm{ord}} f(y)$; taking $y = x+1$ show that $f(x)$ is an ordinal.

We start with some properties to the interval $[a, b[$, where a and b are two ordinal numbers. If $a = 0$ this is O_b, the set of ordinals $< b$, thus is b. If $a < b$, the supremum of $[a, b[$ is the predecessor of b (is b if b is a limit ordinal).

```
Definition ordinal_interval a b := Zo b (fun z => a <=o  z).

Lemma ointv_pr a b z:
  is_ordinal b ->
  (inc z (ordinal_interval a b) <-> (a <=o z & z <o b)).
Lemma ointv_pr0 b z:
  is_ordinal b -> (inc z (ordinal_interval \0o b) <-> z <o b).
Lemma ointv_pr1  b: is_ordinal b ->
  ordinal_interval \0o b = b.
Lemma ointv_pr2 a b z:
  inc z (ordinal_interval a b) -> is_ordinal z.
Lemma ointv_sup a b: a <o b ->
  \osup (ordinal_interval a b) =  \osup b.
Lemma ointv_sup1 a b: a <o b -> limit_ordinal b ->
  \osup (ordinal_interval a b) = b.
```

We say that f is a *normal ordinal functional symbol* (defined for $x \leq \alpha_0$) if f satisfies

$$\alpha_0 \leq x < y \implies f(x) < f(y), \tag{9a}$$

$$(\forall i \in \mathrm{I}, \alpha_0 \leq x_i) \implies \sup_{i\in\mathrm{I}}(f(x_i)) = f(\sup_{i\in\mathrm{I}} x_i). \tag{9b}$$

Note that (9a) implies $f(x)$ is an ordinal if $\alpha_0 \leq x$. Instead of a family $(x_i)_{i\in\mathrm{I}}$, we consider a set X and rewrite condition (9b) as follows

$$(\forall x \in \mathrm{X}, \alpha_0 \leq x) \implies \sup_{x\in\mathrm{X}}(f(x)) = f(\sup_{x\in\mathrm{X}} x). \tag{9c}$$

(we assume X and I non-empty). Let $y > \alpha_0$ be a limit ordinal, and consider $\mathrm{X} = [\alpha_0, y[$, so that we have $\sup(\mathrm{X}) = y$. Then (9c) implies

$$\sup_{\alpha_0\leq x<y}(f(x)) = f(y). \tag{9d}$$

Conversely, any functional that satisfies (9a) and (9d) satisfies also (9b). In effect, the result is clear if the supremum $\bar{x}$ of X is in X. Otherwise $\bar{x}$ is a limit ordinal. We can rewrite (9c), and notice that X is cofinal in $\bar{x}$ (more precisely, in the set of all x such that $\alpha_0 \leq x$ and $x < \bar{x}$).

```
Definition normal_ofs1 f u:=
  (forall x y, u <=o x -> x <o y -> (f x) <o (f y)) &
  (forall X, (forall x, inc x X -> u <=o x) -> nonempty X ->
    \osup (fun_image X f) = f (\osup X)).
Definition normal_ofs2 f u:=
  (forall x y, u <=o x -> x <o y ->  (f x) <o (f y)) &
  (forall x, limit_ordinal x -> u <o x ->
    f x = \osup (fun_image (ordinal_interval u x) f)).
Definition normal_ofs f :=
  (forall x y, x <o y -> (f x) <o (f y)) &
  (forall x, limit_ordinal x -> f x = \osup (fun_image x f)).

Lemma normal_ofs1_o f u:
  (forall x y, u <=o x -> x <o y ->  (f x) <o (f y))
  ->  (forall x, u <=o x -> is_ordinal (f x)).
Lemma normal_ofs_o  f:
  normal_ofs f ->  (forall x, is_ordinal x -> is_ordinal (f x)).

Lemma normal_fs_equiv f u:  (* 84 *)
  normal_ofs1 f u <-> normal_ofs2 f u.
Lemma normal_fs_equiv1 f:
  normal_ofs1 f \0o <-> normal_ofs f.
Lemma normal_fs_equiv2  f a:
  is_ordinal a -> normal_ofs f-> normal_ofs1 f a.
```

If f is an OFS, then $f(x)$ is limit if x is limit.

```
Lemma normal_fs_limit f x: normal_ofs f ->  limit_ordinal x ->
  limit_ordinal (f x).
Lemma normal_compose f g: (* 30 *)
 normal_ofs f -> normal_ofs g -> normal_ofs (f \o g).
```

Assume that f is an OFS. Assume first f strictly increasing. Then (by induction) for any x, $f(x) \geq x$. Thus, there is a least y such that $f(y) \geq x$. Assume f normal. Then either $x < f(0)$ or there is z such that $f(z) \leq x < f(z+1)$. This is obvious if $x = f(y)$, so assume $x < f(y)$. The result is clear if y is zero or a successor. Assume y limit. Then $x < f(y)$ says $x < f(z)$ for some $z < y$, contradiction.

If f if normal, it has arbitrary large fix points, for the supremum of the sequence x_i, defined by $x_{n+1} = f(x_n)$ is a fix-point $\geq x_0$.

```
Lemma ordinal_worder5  x (p: Set -> Prop):
 \1o <=o x -> ~(p x) ->
 let y := least_ordinal (fun z => (~ (\1o <=o z -> p z))) x in
   is_ordinal y & (\1o <=o y) & ~(p y) &
    (forall z, inc z (ordinal_interval \1o y) -> p z).

Lemma ord_strict_incr_unbounded x f:
   (forall x y, x <o y -> (f x) <o (f y)) ->
   is_ordinal x -> x <=o (f x).
Lemma ord_strict_incr_unbounded1 x f:
   (forall x y, x <o y -> (f x) <o (f y)) ->
   is_ordinal x -> exists y,
      is_ordinal y & x <=o (f y)
      & forall z, is_ordinal z -> x <=o (f z) -> y <=o z.
Lemma normal_ofs_bounded x f: is_ordinal x -> normal_ofs f->
```

```
  x <o f \0o \/ exists y, is_ordinal y & f y <=o x & x <o f (succ_o y).
Lemma normal_ofs_critical x f:  (* 35 *)
  normal_ofs f -> is_ordinal x ->
  exists y, x <=o y & f y  = y.
```

8.6 Operations and Ordering

We show that $\sum a_i \leq \sum b_i$ whenever $a_i \leq b_i$. This is true for order-types as well as ordinals. We then deduce that the sum over J is not greater than the sum over I when $J \subset I$.

```
Lemma osum_increasing1 r f g:
  orsum_ax r f -> orsum_ax r g ->
  (forall x, inc x (domain f) -> order_le (V x f) (V x g)) ->
  order_le (order_sum r f) (order_sum r g).  (* 68 *)
Lemma osum_increasing2 r f g:
  worder r ->
  fgraph f -> fgraph g -> substrate r = domain f -> substrate r = domain g ->
  (forall x, inc x (domain f) -> (V x f) <=o (V x g)) ->
  (ord_sum r f) <=o (ord_sum r g).
Lemma ord_sum_increasing3 r f j:  (* 36 *)
  orsum_ax r f ->
  sub j (domain f) ->
  order_le (order_sum (induced_order r j) (restr f j)) (order_sum r f).
Lemma osum_increasing4 r f j:  worder r ->
  substrate r = domain f -> sub j (domain f) -> ordinal_fam f ->
  (ord_sum (induced_order r j) (restr f j)) <=o (ord_sum r f). (* 23 *)
```

We show here the same for the product. Note that the product over J is not greater than the product over I when $J \subset I$, provided that factors in $I - J$ are non-zero.

```
Lemma oprod_increasing1 r f g:
  orprod_ax r f -> oprod_ax r g ->
  (forall x, inc x (domain f) -> order_le (V x f) (V x g)) ->
  order_le (order_prod r f) (order_prod r g).  (* 71 *)
Lemma oprod_increasing2 r f g: worder r ->
  fgraph f -> fgraph g -> substrate r = domain f -> substrate r = domain g ->
  (forall x, inc x (domain f) -> (V x f) <=o (V x g)) ->
  finite_set (substrate r) ->
  (ord_prod r f) <=o (ord_prod r g).
Lemma oprod_increasing3 r f j:  (* 50 *)
  orprod_ax r f ->
  sub j (domain f) ->
  (forall x, inc x  (complement (domain f) j) ->
     substrate (V x f) <> emptyset) ->
  order_le (order_prod (induced_order r j) (restr f j)) (order_prod r f).
Lemma oprod_increasing4 r f j:  worder r ->  (* 27 *)
  substrate r = domain f -> sub j (domain f) -> ordinal_fam f ->
  finite_set (substrate r) ->
  (forall x, inc x (complement (domain f) j)  -> V x f <> \0o) ->
  (ord_prod (induced_order r j) (restr f j)) <=o (ord_prod r f).
```

We deduce the following relations. We prove it when arguments are ordinals, but the result holds in the case of order-types. In the case of a product, some arguments have to ne

non-zero.

(10a) $a \le b$ and $a' \le b' \implies a + a' \le b + b', \qquad a \le a + b, \qquad b \le a + b.$

(10b) $a \le b$ and $a' \le b' \implies a \cdot a' \le b \cdot b', \qquad a \le a \cdot b, \qquad b \le a \cdot b.$

```
Lemma osum_Mlele a b c d:
  a <=o b -> c <=o d -> (a +o c) <=o (b +o d)
Lemma osum_Mle0 a b:
  is_ordinal a -> is_ordinal b -> a <=o (a +o b).
Lemma osum_M0le a b:
  is_ordinal a -> is_ordinal b -> b <=o (a +o b)
Lemma osum2_nz x y:
  is_ordinal x -> is_ordinal y -> x <> \0o  -> (x +o y) <> \0o.

Lemma oprod_Mlele a b c d:
  a <=o b -> c <=o d -> (a *o c) <=o (b *o d)
Lemma oprod_Mle1 a b:
  is_ordinal a -> is_ordinal b ->  b <> \0o -> a <=o (a *o b).
Lemma oprod_M1le a b:
  is_ordinal a -> is_ordinal b -> a <> \0o -> b <=o (a *o b).
```

As a consequence, if $a \le b$, we have

(10c) $c + a \le c + b, \quad a + c \le b + c, \quad c \cdot a \le c \cdot b, \quad a \cdot c \le b \cdot c.$

```
Lemma osum_Mleeq a b c:
  a <=o b -> is_ordinal c -> (a +o c) <=o (b +o c).
Lemma osum_Meqle a b c:
  a <=o b -> is_ordinal c -> (c +o a) <=o (c +o b).
Lemma oprod_Mleeq a b c:
  a <=o b -> is_ordinal c -> (a *o c) <=o (b *o c).
Lemma oprod_Meqle a b c:
  a <=o b -> is_ordinal c -> (c *o a) <=o (c *o b).
```

We show that $a + b = a$ implies $b = 0$, since a is naturally isomorphic to a segment of $a + b$, so that we have a segment of $a + b$ isomorphic to the whole set. If n is an integer, we have $n + \omega = n$ by considering the function f defined on $n + \omega$ by: if $x \in n$ then $f(x) = x$ else $f(x) = x + n$. If moreover n is non-zero, we prove $n \cdot \omega = \omega$ by considering $f(x) = an + b$ (with $a \in \omega$ and $b \in n$). We shall see below a shorter proof.

```
Lemma osum_a_ab a b:
  is_ordinal a -> is_ordinal b -> a +o b = a -> b = \0o.  (* 45 *)

(*
Lemma osum_int_omega n:
  inc n Bnat -> n +o \omega = \omega. (* 84 *)
Lemma oprod_int_omega  n:
  inc n Bnat ->  n <> \0c -> n *o \omega = \omega.  (* 107 *)
*)
```

It follows

$$(a + b)^+ = a + b^+, \qquad a \cdot b^+ = a \cdot b + a.$$

Since integers are ordinals, it follows, by induction, that the cardinal sum and product of two integers are the ordinal sum and product. Since $n <_{\mathrm{ord}} \omega$ is equivalent to $n \in \mathbf{N}$, it follows that the sum and product of two ordinals $< \omega$ is $< \omega$. We have $a \cdot 2 = a + a$.

```
Lemma oprod2_succ x y:
 is_ordinal x -> is_ordinal y ->  x *o (succ_o y) =  (x *o y) +o x.
Lemma osum2_succ x y:
   is_ordinal x -> is_ordinal y -> succ_o (x +o y) = x +o (succ_o y).
Lemma osum2_2int a b:
  inc a Bnat -> inc b Bnat -> a +o b = a +c b.
Lemma oprod2_2int a b:
  inc a Bnat -> inc b Bnat -> a *o b = a *c b.
Lemma osum2_lt_omega a b:
  a <o \omega -> b <o \omega -> (a +o b) <o \omega.
Lemma oprod2_lt_omega a b:
  a <o \omega -> b <o \omega -> (a *o b) <o \omega.
Lemma osum_11_2: \1o +o \1o = \2o.
Lemma ord_double x: is_ordinal x -> x *o \2o = x +o x.
```

The sum and product of two countable ordinals is countable. We add some other trivial properties. Cantor says that an ordinal is of the *first class* if it is finite, of the *second class* if it is countable and infinite.

```
Lemma osum2_countable x y:
   countable_ordinal x -> countable_ordinal y -> countable_ordinal (x +o y).
Lemma oprod2_countable x y:
   countable_ordinal x -> countable_ordinal y -> countable_ordinal (x *o y).
Lemma countable_ordinal_leomega x:
  x <=o \omega ->  countable_ordinal x.
Lemma countable_one: countable_ordinal \1o.
Lemma countable_succ x:
   countable_ordinal x -> countable_ordinal (succ_o x).
Lemma countable_fun_image z f:
  countable_set z -> countable_set (fun_image z f).
Lemma countable_sub a b:
  sub a b -> countable_set b -> countable_set a.
```

8.7 Ordinal Subtraction

Consider two ordinals a and b such that $b \leq a$. There exists a unique ordinal c such that $a = b + c$ (it is the ordinal of the complement of b in a). This quantity is written $(-b) + a$ by Bourbaki, but we prefer the notation $a - b$. In particular, every non-zero ordinal x can uniquely be written as $x = 1 + y$. Note that $a = d + b$ has in general no solution d (for instance if $b = 1$, the ordinal a must be a successor), and may have multiple solutions (if n is an integer $n + \omega = n$).

Assume now $b + x = b + y$. If $x \leq y$ we have $y = x + z$, for some z, hence $b + x = (b + x) + z$ by associativity, thus $z = 0$ and $x = y$. This implies uniqueness of $a - b$.

```
Definition ord_diff b a :=
  ordinal (induced_order (ordinal_o b) (complement b a)).
Notation "x -o y" := (ord_diff x y) (at level 50).

Lemma odiff_pr a b: a <=o b
  -> (is_ordinal (b -o a) & b = a +o (b -o a)).  (* 71 *)
Lemma ord_diff_wrong a b: b <=o a -> b -o a = \0c.
Lemma OS_diff a b: is_ordinal a -> is_ordinal b -> is_ordinal (a -o b).
Lemma ord_rev_pred x: is_ordinal x -> x <> \0o ->
```

```
  exists y, is_ordinal y & x = \1o +o y.
Lemma odiff_pr2  a b: a <o b -> (b -o a) <> \0o.
Lemma osum2_simpl a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  c +o a = c +o b -> a = b.
Lemma odiff_pr1 a b:
   is_ordinal a -> is_ordinal b ->
   (a +o b) -o a = b.
Lemma odiff_smaller a b: a <=o b -> (b -o a) <=o b.
```

We have

(10d) $$a < b \implies c + a < c + b \text{ and } c \cdot a < c \cdot b \text{ (when } c \neq 0).$$

The first result is a consequence of $c + a = c + b \implies a = b$. The second result is equivalent to $c \cdot a = c \cdot b \implies a = b$ for $c \neq 0$. In particular, $c \cdot a = c$ implies $a = 1$.

```
lemma osum_Meqlt a b c:
  a <o b -> is_ordinal c -> (c +o a) <o (c +o b).
Lemma oprod_Meqlt a b c:
  a <o b -> is_ordinal c -> c <> \0o -> (c *o a) <o (c *o b).
Lemma oprod_Meq1lt b c:
   is_ordinal c -> is_ordinal b -> c <> \0o ->
   b <> \0o -> b <> \1o -> c <o (c *o b).
Lemma oprod2_simpl a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c -> c <> \0o ->
  c *c a = c *o b -> a = b.
Lemma oprod_a_ab x y:
  is_ordinal x -> is_ordinal y -> x <> \0o -> x *o y = x -> y = \1o.
```

We have

(10e) $$(\mu + \alpha < \mu + \beta \text{ or } \alpha + \mu < \beta + \mu \text{ or } \mu\alpha < \mu\beta \text{ or } \alpha\mu < \beta\mu) \implies \alpha < \beta.$$

Proof: if the conclusion were false, we would have $\beta \leq \alpha$. We can add or multiply μ, use transitivity, and we get $x < x$ for some x.

```
Lemma osum_Meqltr a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  (c +o a) <o (c +o b) -> a <o b.
Lemma osum_Mlteqr a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  (a +o c) <o (b +o c) -> a <o b.
Lemma oprod_Meqltr a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  (c *o a) <o (c *o b) -> a <o b.
Lemma oprod_Mlteqr a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  (a *o c) <o (b *o c) -> a <o b.
```

8.8 Ordinal Division

Let b be a non-zero ordinal, and a any ordinal. Let q be the greatest ordinal such that $b \cdot q \leq a$. We can write $a = b \cdot q + r$, for some r with $r < q$. The existence of q follows from (11c)

below (however, this requires a property of multiplication that is shown by using Euclidean division).

For this reason, we show existence of division as follows: Assume $a < b \cdot c$. (Since b is non-zero, such a c exists). Consider now the well-orderings associated to a and the product. There an element x in the product so that a is isomorphic to the segment S_x. The two components of x correspond to the quotient and remainder, and we have thus $r < b$ and $q < c$.

```
Definition ord_div_pr0 a b q r :=
  is_ordinal q & is_ordinal r & a = (b *o q) +o r & r <o b.
Definition ord_div_pr1 a b c q r :=
  ord_div_pr0 a b q r & q <o c.
Lemma ord_div_nonzero_b a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  a <o (ord_prod2 b c)  -> b <> \0o.
Lemma ord_div_nonzero_b_bis  a b:
  is_ordinal a -> is_ordinal b -> b <> \0o ->
  exists c, is_ordinal c & a <o (b *o c).
Lemma odivision_unique a b cr q' r':
  is_ordinal a -> is_ordinal b ->
  ord_div_pr0 a b q r -> ord_div_pr0 a b q' r' ->
  (q = q' & r = r').   (* 27 *)
Lemma odivision_exists  a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  a <o (ord_prod2 b c) ->
  exists q, exists r, ord_div_pr1 a b c q r. (* 195 *)
Lemma odivision_exists1 a b:
  is_ordinal a -> is_ordinal b ->  b <> \0o ->
  exists q, exists r, ord_div_pr0 a b q r.
```

The functional symbol $b \to a+b$ is normal. Consider a limit ordinal c. Let S be $\sup_{b<c}(a+b)$. We have clearly $S \leq a+c$. Since c is non-zero we have $a \leq S$, so that $S = a+e$ for some e. The previous relation says $e \leq c$. We pretend that equality holds, since otherwise we would have $e < c$, $e+1 < c$, $a+e+1 \leq S$, absurd.

The functional symbol $b \to a \cdot b$ is normal for $a \neq 0$. We proceed as above. Let S be $\sup_{b<c}(a \cdot b)$. We have $S \leq a \cdot c$. We may write $S = a \cdot q + r$, and have $S < a \cdot (q+1)$. From $q < c$ we deduce $a \cdot (q+1) \leq S$, absurd.

Consider the case $a < \omega$ and $c = \omega$. Let T_a be the set of all $a+b$ for $b < \omega$. Note that T_a is cofinal in T_0, so that the two sets have the same supremum, and $a + \omega = \omega$. In the same way, $a \cdot \omega = \omega$ if a is non-zero.

```
Lemma osum_normal a: is_ordinal a -> (* 40 *)
  normal_ofs (fun u => a +o u).
Lemma oprod_normal a: is_ordinal< a ->  (* 28 *)
  a <> \0o -> normal_ofs (fun u =>  a *o u).

Lemma osum_int_omega n:
  inc n Bnat -> n +o \omega = \omega.
Lemma oprod_int_omega  n:
  inc n Bnat ->  n <> \0c -> n *o \omega = \omega.
```

We deduce an example of $a+b \neq b+a$ with $a = 1$ and $b = \omega$, an example of $(a+b) \cdot c \neq a \cdot c + b \cdot c$ with $a = b = 1$ and $c = \omega$ and an example of $a \cdot b \neq b \cdot a$ with $a = 2$ and $b = \omega$.

```
Lemma osum2_nc: let x := \1o in let y := \omega in
   x +o y <> y +o x.
Lemma oprod2_nc: let x := \2o in let y := \omega in
   x *o y <> y *o x.
Lemma osum2_nD:
  let a:= \1o in let b:= \1o in let c:= \omega in
  (a +o b) *o c <> (a *o c) +o (b *o c).
```

We shall need later on the property that $1+x=x$ whenever x is infinite (in this case, $x=\omega+y$ for some y and $1+x=1+\omega+y=\omega+y$). If x is an ordinal and y is limit, then $x+y$ is limit (assume $z<x+y$; the result is clear if $z\leq x$; otherwise $z=x+t$, with $t<y$, so that $t+1<y$).

```
Lemma osum1inf x: \omega <=o x -> \1o +o x = x.
Lemma osum_limit x y: is_ordinal x -> limit_ordinal y ->
  limit_ordinal (x +o y).  (* 22 *)
```

8.9 Indecomposable ordinals

An ordinal c is called *indecomposable* if it is non-zero and never the sum of two ordinals a and b such that $a<c$ and $b<c$. We shall see later on that indecomposable ordinals are powers of ω.

Assume $c=a+b$. Then $a\leq c$ and $b\leq c$. If c is indecomposable, then either $c=a$ or $c=b$. If c is a successor, then c is indecomposable if and only if $c=1$. The ordinal ω is indecomposable since $x<\omega$ implies x finite, and ω is not the sum of two finite ordinals. Any indecomposable ordinal is thus 1, or at least ω.

```
Definition ord_indecomposable z :=
   z <> \0o & (forall x y, x <o z -> y <o z -> x +o y <> z).
Lemma indecomposable_pr x a b:
  is_ordinal x -> is_ordinal a -> is_ordinal b ->
  ord_indecomposable x -> a +o b = x -> (a = x \/ b = x).
Lemma indecomp_example x: is_ordinal x -> x <> \0o ->
  ~ (ord_indecomposable (succ_o x)).
Lemma indecomp_one: ord_indecomposable \1o.
Lemma indecomp_omega: ord_indecomposable \omega.
Lemma limit_infinite1 x: limit_ordinal x -> \omega <=o x.
Lemma indecomp_omega1 x: ord_indecomposable x -> is_ordinal x ->
  x = \1o \/ \omega <=o x.
```

We prove here the following claims: (a) A non-zero ordinal x is indecomposable if and only if $y<x$ implies $y+x=x$; (b) if $y>0$, and $x\neq 1$, then x indecomposable if and only if $y\cdot x$ is indecomposable and (c) if x is indecomposable and $y<x$ then y divides x and the quotient is indecomposable.

Result (a) is follows from existence of subtraction. Assume now $z<y\cdot x$, x indecomposable, $x>1$. By division we have $z=y\cdot q+r$, with $r<y$ and $q<x$. Thus $z+y\cdot x\leq y\cdot q+r+y\cdot x\leq y\cdot q+y+y\cdot x=y\cdot(q+1+x)$. If $q+1<x$, this is $y\cdot x$. We deduce $z+y\cdot x=y\cdot x$. Otherwise, we have $q+1=x$; this is absurd since $1<x$. This shows (b). Finally, assume x is indecomposable and $y<x$. Statement (c) is true if $x=y\cdot x$. Otherwise we have $x<y\cdot x$, thus $x=y\cdot q+r$ with $q<x$ and $r<y$. Since x is indecomposable, one of the two terms y and r must be x. Since

$r < y < x$ we get $x = y \cdot q$. If $q > 1$, point (b) implies that q is indecomposable. The result is also true if $q = 1$.

```
Lemma indecomposable_prop x: is_ordinal x -> x <> \0o ->
  (ord_indecomposable x <-> (forall y, y <o x -> y +o x = x)).
Lemma indecomposable_prop1 x y: ord_indecomposable x ->
  y <o x -> y +o x = x.
Lemma indecomposable_prod  x y: is_ordinal x -> x <> \0o ->
  x <> \1o -> is_ordinal y -> y <> \0o ->
  (ord_indecomposable x <-> ord_indecomposable (y *o x)).  (* 39 *)
Lemma indecomposable_division x y: ord_indecomposable x ->
  y <> \0o -> y <o x ->
  exists z, ord_indecomposable z & is_ordinal z & x = y *o z.
```

Let a be a non-zero ordinal. Then $a \cdot \omega$ is an indecomposable ordinal $> a$; if b is indecomposable and $a < b$, then $b = a \cdot c$ for some indecomposable ordinal c, that cannot be 1, hence must be $\geq \omega$. Thus $a \cdot \omega$ is the least indecomposable ordinal $> a$. We deduce $(a+1) \cdot \omega = a \cdot \omega$ (these quantities are the least indecomposable $> a$ or $> a+1$ (note that $a \cdot \omega$ cannot be equal to $a+1$).

```
Lemma indecomposable_prod2 a (b:= a *o \omega):
  is_ordinal a -> a <> \0o ->
  (ord_indecomposable b & a <o b &
  forall c, ord_indecomposable c -> a <o c -> b <=o c).
Lemma indecomposable_prod3 a: is_ordinal a -> a <> \0o ->
  (succ_o a) *o \omega = a *o \omega.
```

8.10 Definition by transfinite induction

One can define addition of integers by induction via the formula $x + (y+1) = (x+y)+1$. This formula holds for ordinal numbers as well, and can be used to define $x + y$ whenever y is a successor. Otherwise y is zero, and $x + 0 = x$, or y is a limit ordinal, case where $x + y = \sup_{z<y} x + z$. The construction that follows unifies the cases y limit and y successor, and can be used for the product and exponentiation as well. It works if $y \geq 2$, so that the cases $y = 0$ and $y = 1$ are special, and defined by two functions w_0 and w_1. The construction works only for $x \geq \alpha_0$, where α is 0 for the sum, 1 for the product and 2 for the exponential. Our function is defined via an auxiliary function g. We shall usually assume

(11a) $$x \leq w_1(x), \qquad x < g(x,y) \qquad (\text{if } x \geq \alpha_0, y \geq \alpha_0).$$

and define f satisfying, for $x \geq \alpha_0$ and all $y \geq 2$,

(11b) $$f(x,0) = w_0(x), \quad f(x,1) = w_1(x), \quad f(x,y) = \sup_{0<z<y} g(f(x,z),x).$$

Uniqueness is obvious.

```
Definition ord_induction_prop w0 w1 g u f:=
  (forall x, u <=o x -> f x \0o = w0 x) &
  (forall x, u <=o x -> f x \1o = w1 x) &
  (forall x y, u <=o x -> \1o <o y ->
    f x y = union (fun_image (ordinal_interval \1o y)
```

```
                        (fun z => g (f x z) x))).
Lemma ord_induction_unique w0 w1 g u f f':
   ord_induction_prop w0 w1 g u  f ->  ord_induction_prop w0 w1 g u f' ->
   forall x y, u <=o x -> is_ordinal y -> f x y = f' x y.
```

We shall define f by transfinite induction. Given any term $p(\mathrm{F})$, and any well-ordered set E, there exists a (unique) surjective function F defined on E, such that $\mathrm{F}(y) = p(\mathrm{F}_y)$, where F_y denotes the restriction of F to the set of elements less than y. Let b be an ordinal, and take for E the set b ordered by $\leq_{\mathrm{ord}}$. If $y \in \mathrm{E}$, then y is an ordinal, and the set of elements less than y is y. In other words, y is the source of F_y. Assume that the term p is independent of b. Consider two ordinals b and b' and two associated functions F and F$'$. We have $\mathrm{F}(y) = \mathrm{F}'(y)$ whenever $y < b$ and $y < b'$. This allows us to define $f(y)$ as $\mathrm{F}(y)$ for any ordinal y.

In this particular case, we fix x, and define $p(\mathrm{F})$ as follows. Let y be the source of F. If $y = 0$ then $p(\mathrm{F}) = w_0(x)$, if $y = 1$ then $p(\mathrm{F}) = w_1(x)$, and otherwise the supremum of $g(\mathrm{F}(z), x)$ taken over all non-zero elements z in y.

```
Definition ord_induction_sup (g: Set -> Set -> Set) x y f :=
  \osup (fun_image (ordinal_interval \1o y) (fun z => g (f z) x)).

Definition ord_induction_prop w0 w1 g u f:=
  (forall x, u <=o x -> f x \0o = w0 x) &
  (forall x, u <=o x -> f x \1o = w1 x) &
  (forall x y, u <=o x -> \1o <o y ->
    f x y =  ord_induction_sup g x y (f x)).

Definition ord_induction_p  (w0 w1: Set-> Set) g x f :=
  (Yo (source f = \0o) (w0 x)
    (Yo (source f = \1o) (w1 x)
       (ord_induction_sup g x (source f) (W^~f)))).
Definition ord_induction_aux w0 w1 g x b :=
  transfinite_defined (ordinal_o b) (ord_induction_p w0 w1 g x).

Definition ord_induction_defined w0 w1 g :=
  fun x y => W y (ord_induction_aux w0 w1 g x (succ_o y)).

Lemma ord_induction_p0 w0 w1 g x b:
   is_ordinal b ->
  let f := (ord_induction_aux w0 w1 g x b) in
     (is_function f & source f = b &
       (inc \0o b -> W \0o f = w0 x) &
       (inc \1o b -> W \1o f = w1 x) &
       (forall y, inc y b -> \1o <o y ->
         W y f = ord_induction_sup g x y (W ^~ f))). (* 35 *)

Lemma ord_induction_p1 w0 w1 g x:
  ord_induction_defined w0 w1 g x \0o = w0 x.
Lemma ord_induction_p2 w0 w1 g x:
  ord_induction_defined w0 w1 g x \1o = w1 x.
Lemma ord_induction_p3 w0 w1 g a b1 b2 x:
  is_ordinal b1 -> is_ordinal b2 ->
  inc x b1 -> inc x b2 -> is_ordinal x ->
  let f1 := (ord_induction_aux w0 w1 g a b1) in
  let f2 := (ord_induction_aux w0 w1 g a b2) in
     W x f1 = W x f2. (* 29 *)
Lemma ord_induction_exists w0 w1 g u:
```

```
ord_induction_prop w0 w1 g u (ord_induction_defined w0 w1 g).
```

Let's assume for a moment that f is defined for $x \geq \alpha_0$ via w_0, w_1 and g. We assume first that relations (11a) are satisfied.

```
Section OrdinalInduction.
Variables (u: Set) (w0 w1: Set -> Set) (f g : Set -> Set -> Set).
Hypothesis axiom_w1: (forall x, u <=o x -> x <=o (w1 x)).
Hypothesis axiom_g1: (forall x y, u <=o x -> u <=o y -> x <o (g x y)).
Hypothesis fv: f = ord_induction_defined w0 w1 g.
```

We assume here $x \geq \alpha_0$ and $y \geq 1$.

We have $f(x,y) \geq x$. This is clearly true for $y = 1$. Otherwise, we may consider the least y for which this is false. Then for $0 < z < y$, we have $x \leq f(x,z)$, hence $f(x,z) < g(f(x,z),x)$. Thus $g(f(x,z),x)$ is an ordinal $\geq x$; the supremum of these quantities is then an ordinal $\geq x$. As a consequence we get: $f(x,y) \geq u$, $f(x,y)$ is an ordinal, $g(f(x,y),x)$ is an ordinal.

By construction, if $y < z$, we have $g(f(x,y),x) \leq f(x,z)$. This implies $f(x,y) < f(x,z)$. Assume $x \geq 1$, so that $f(x,1) \geq 1$. Since f is strictly increasing, we deduce $f(x,y) \geq y$. Finally, we show that f is normal. Let y be a limit ordinal, T the set of all $f(x,t)$ where $1 \leq t < y$. Since f is increasing, we have $\sup \mathrm{T} \leq f(x,y)$. We have $f(x,y) \leq \sup \mathrm{T}$, since $f(x,y) = \sup g(f(x,t),x)$ and, for each $t < y$, we have $g(f(x,t),x) \leq f(x,t^+) \leq \sup \mathrm{T}$, (remember that $t < y$ implies $t^+ < y$).

```
Lemma ord_induction_p4  x y:
  u <=o x -> \1o <=o y -> x <=o (f x y). (* 20 *)
Lemma ord_induction_p41 x y:
  u <=o x -> \1o <=o y -> u <=o (f x y).
Lemma ord_induction_p5 x y:
  u <=o x -> \1o <=o y -> is_ordinal (f x y).
Lemma ord_induction_p6 x y:
  u <=o x -> \1o <=o y -> is_ordinal (g (f x y) x).
Lemma ord_induction_p7 x y y':
  u <=o x -> \1o <=o y -> y <o y' ->
  (g (f x y) x) <=o (f x y').
Lemma ord_induction_p8 x y y':
  u <=o x -> \1o <=o y ->  y <o y' ->  (f x y) <o (f x y').
Lemma ord_induction_p9 x y:
  u <=o x -> \1o <=o x -> \1o <=o y -> y <=o (f x y).
Lemma ord_induction_p10 x:  u <=o x ->
  normal_ofs1 (f x) \1o. (* 30 *)
```

Assume now that a and b are ordinals. There exists a unique ordinal y such that

$$f(a,y) \leq b < f(a,y+1). \tag{11c}$$

Uniqueness is obvious since f is strictly increasing. There is a necessary condition, namely $f(a,1) \leq b$, which can be restated as $w_1(a) \geq b$. We also assume $a > 0$ (thus $b > 0$). Consider the least y such that $b < f(a,y+1)$ (it exists as f is a strictly increasing function of y). If $z < y$ we have $f(a,z+1) \leq b$. If y is a successor, then $f(a,y) \leq b$ is obvious; otherwise $f(a,y)$ is a supremum of quantities that are $\leq b$.

Note: if $g(x,y) = x + y$, then $f(x,y) = x \cdot y$ and y is the quotient of b by a.

```
Lemma ord_induction_p11 x b y y':
  u <=o x -> \1o <=o y -> \1o <=o y' ->
    (f x y) <=o b -> b <o (f x (succ_o y))  ->
    (f x y') <=o b -> b <o (f x (succ_o y'))  ->
    y = y'.  (* 22 *)
Lemma ord_induction_p12 x b: (* 47 *)
  u <=o x -> (w1 x) <=o b -> \1o <=o x ->
  exists y,  \1o <=o y & y <=o b & (f x y) <=o b & b <o (f x (succ_o y)).
```

These results correspond to Exercise 2.17(d) (page 334). Assume

(11d) $\alpha_0 \le x < x' \implies w_1(x) < w_1(x')$, $\alpha_0 \le x < x', \alpha_0 \le y \le y' \implies g(x,y) \le g(x',y')$.

We have

(11e) $$f(x,y+1) = g(f(x,y),x)$$

because the least upper bound becomes a greatest element. It follows that f is increasing; if moreover $g(x,y) < g(x,y')$ for $\alpha_0 \le y < y'$, then $f(x,y+1) < f(x',y+1)$ for all ordinals y.

```
Hypothesis axiom_w2: (forall a a', u <=o a -> a <o a' -> (w1 a) <o (w1 a')).
Hypothesis axiom_g2:  (forall a b a' b',
    u <=o a -> u <=o b ->  a <=o a' -> b <=o b' ->
   (g a b) <=o (g a' b')).


Lemma ord_induction_p15 x y x' y':  (* 41 *)
  u <=o x -> x <=o x' -> \1o <=o y -> y <=o y' -> f x y <=o f x' y'.
Lemma ord_induction_p16  x y: u <=o x -> \1o <=o y ->
  f x (succ_o y) =  g (f x y) x. (* 22 *)
Lemma ord_induction_p17 x x' y:
   (forall a b b', u <=o a -> u <=o b -> b <o b' -> g a b <o g a b') ->
   u <=o x -> x <o x' -> is_ordinal y ->
   f x (succ_o y) <o f x' (succ_o y).
```

These results correspond to Exercise 2.17(e) (page 334). Assume that $g(x,y)$ is a normal function of y. In particular $g(x,y) < g(x,y')$ when $y < y'$. Assume further that (11a) and (11d) are true, that $w(x) = x$ and that g is associative: $g(x,g(y,z)) = g(g(x,y),z)$. Then, whenever $x \ge \alpha_0$, $y \ge 1$, $z \ge 1$, we have

(12) $$g(f(x,y),f(x,z)) = f(x,y+z), \qquad f(f(x,y),z) = f(x,yz).$$

If g is addition, then f is multiplication; these relation express the distributivity law and the associativity of multiplication. If g is multiplication, then we get relations (13) below; for simplicity, we shall write $x \cdot y$ and x^y instead of $g(x,y)$ and $f(x,y)$ (see relations 13 below). We have $x^1 = w_1(x) = x$ and $x^{y+1} = x^y \cdot x$ by (11e). This shows that (12) is true for $z = 1$. Assume that it is true for z, and consider $z+1$. We have $x^y \cdot x^{z+1} = x^y \cdot (x^z \cdot x) = (x^y \cdot x^z) \cdot x = x^{y+z} \cdot x = x^{y+z+1}$. We have $x^{y\cdot(z+1)} = x^{y\cdot z+y} = x^{y\cdot z} \cdot x^y = (x^y)^z \cdot (x^y)^1 = (x^y)^{z+1}$ by application of (12a). This shows that (12) is true for any successor.

Finally, assume that z is a limit ordinal, and the relation true for $t < z$. Let's show the first relation (the proof for the second one is similar). Since f is normal (in its second argument), x^z is the supremum of all x^t for $1 \le t < z$. Since g is normal, $g(x^y,x^z)$ is the supremum of all $g(x^y,x^t)$ for $t < z$. We can apply (12), so that it becomes the sup of all x^{y+t}. This is x^{y+z} by normality of f and the sum.

```
Hypothesis axiom_w3:  (forall a, w1 a = a).
Hypothesis axiom_g3:  (forall a, u <=o a -> normal_ofs1 (fun b => g a b) u).
Hypothesis axiom_g4:  (forall a b c, u <=o a -> u <=o b -> u <=o c ->
     g (g a b) c = g a (g b c)).

Lemma ord_induction_p18 a b c:
  u <=o a -> \1o <=o b -> \1o <=o c ->
    f a (b +o c) = g (f a b) (f a c). (* 75 *)
Lemma ord_induction_p19 a b c: (* 68 *)
  u <=o a -> \1o <=o b -> \1o <=o c ->   f a (b *o c) = f (f a b) c.
```

Assume here that (11d) holds, so that f is increasing in both its arguments. It is strictly increasing in its second argument. Fix b and consider

(*) $$\exists x, \quad f(x,y) = b.$$

In Exercise 6.13(b) (page 370), Bourbaki says that there exists "at most a finite number" of ordinals y satisfying this property. This can be formalized as: there exists a finite set E_b such that $y \in \mathrm{E}_b$ is equivalent to (*).

For each y such that (*) holds, we consider the least solution x, and call it x_y. Consider two solutions y and z; assume $y < z$. If $x_y \le x_z$ then $f(x_y, y) \le f(x_z, y) < f(x_z, z)$ absurd. Assume for a moment that there is a set E_b, containing all solutions of (*). This set is well-ordered for $\le_{\mathrm{ord}}$ since it is a set of ordinals. It is also well-ordered for $\ge_{\mathrm{ord}}$ for it is isomorphic to the set of all x_y (for $y \in \mathrm{E}_b$) which is a set of ordinals. This it has to be finite (see below). Note that if y is a solution of (*) then $y \le b$, unless $f(0,y) = b$, since $f(x,y) \ge y$ if $x > 0$. We know that $f(0,y)$ is strictly increasing, so that there exists c such that $f(0,y) \ge y$ if $y \le c$ (see below). Thus (*) implies $y \le \sup(b,c)$. This remark shows existence of the set E_b.

We prove here the first auxiliary lemma: if E is well-ordered by $\le$ and $\ge$, then it is finite. Let S be the subset of E formed of all x such that $]\leftarrow, x[$ is finite. If E is empty, there is nothing to do; otherwise E has a least element, which is in S so that S has a greatest element y. Let $\mathrm{F} =]\leftarrow, y[\cup \{y\}$. This is a finite segment. Since it cannot be of the form $]\leftarrow, x[$, it has to be E itself. Thus E is finite. This is Exercise 4.3 (page 344).

Assume now that $w(x)$ is a strictly increasing for $x \ge \alpha_0$. Then, for any y we have $w(x) + y \le w(x+y)$ (the least y for which such an inequality is false must be zero, a successor or a limit ordinal, all cases being impossible). Set $b = \alpha_0 + 1$. We have $w(b) + b \cdot n \le w(b + b \cdot n)$, for any integer n. In particular $b \cdot n \le w(b \cdot n^+)$. Since multiplication is a normal ordinal symbol, it follows, if $c = b \cdot \omega$ that $c \le w(c)$, hence, for all x with $c \le x$, we have $x \le w(x)$. This is Exercise 6.13(a).

```
Lemma worder_has_empty_seg r: worder r ->
  nonempty (substrate r) -> exists x,
    (inc x (substrate r) & segment r x = emptyset).
Lemma well_ordered_opposite r:
  worder r -> worder (opposite_order r) -> finite_set (substrate r).
Lemma osum_increasing5 a w: is_ordinal a ->  (* 79 *)
  (forall x y, a <=o x -> x <o y -> w x <o w y) ->
  ((forall x y, a <=o x -> is_ordinal y -> (w (x) +o y) <=o w (x +o y))
  & (exists b, a <=o b & forall x, b <=o x -> x <=o w x)).
```

Bourbaki hints that $f(x, y+1) \ge w_1(x) + y$ in Exercise 6.13(e). In particular, this implies $f(x, y+1) \ge x + y$. This inequality is true for $y = 0$; moreover $f(x, y+1) = g(f(x,y), x) > f(x,y)$

so that the equality is true by induction on y for all integers. In particular, for $y = 1$ we get $f(x,2) > x$, so that $f(x,y) > x$, whenever $y \geq 2$. Taking the supremum for all finite y yields $x + \omega \leq f(x,\omega)$ and $x + y \leq f(x,y)$ whenever y is infinite.

```
Lemma ord_inductionp_p20 b:  (* 81 *)
  is_ordinal b -> exists E, finite_set E &
  forall y, (inc y E) <-> (exists x, u <=o x  & \1o <=o y & f x y = b).

Lemma ord_induction_p21a x y:  u <=o x ->  y <o \omega->
   x +o y  <=o (f x (succ_o y)).
Lemma ord_induction_p21b x: u <=o x ->
   x +o \omega  <=o (f x \omega).  (* 19 *)
Lemma ord_induction_p21c x y:  u <=o x  -> \omega <=o y ->
   x +o y  <=o (f x y).  (* 44 *)
Lemma ord_induction_p21d x y:  u <=o x -> \2o <=o  y ->
   x <o (f x y).
```

We say that y is *critical* if $f(x,y) = y$ for all x such that $x < y$. For instance, if f is addition, then y is critical if, and only if, it is indecomposable. Assume $y = z + 1$. From $y = g(f(z,z),z)$ we deduce $f(z,z) = z$ which is impossible (unless $y = 1$, or $y = 2$; in order to avoid these particular cases, Bourbaki assumes moreover y infinite).

Assume $f(x,y) = y$ whenever $x \in$ A where A is a set whose supremum is y. Assume y infinite, so that $y \leq f(x,y)$ for all x. Since f is increasing, it follows that y is critical. Note that A is nonempty, which implies $\alpha_0 \leq y$.

Consider the sequence defined by $z_{n+1} = f(z_n, z_n)$, with $z_0 = \alpha_0 + 2$. It is obvious by induction that this is a sequence of ordinals and is strictly increasing. Let y be the supremum. This is a critical ordinal. In fact, y is infinite, $> \alpha_0$ and a limit ordinal. It thus remains to show that $x < y$ and $z < y$ imply $f(x,z) \leq y$. By definition of y, it suffices to show $f(x_n, x_m) \leq y$. Assume $n \leq p$ and $m \leq p$. Then $f(x_n, x_m) \leq f(x_p, x_p) = x_{p+1} \leq y$.

Let A be a set of critical elements and y its supremum. Then y is critical. The case $y \in$ A is obvious. Otherwise y is a limit ordinal. All we have to do is to show that $f(x,z) \leq y$ whenever $x < y$ and $z < y$. Let t be the supremum of x and z. We have $t + 1 < y$ since y is limit. Thus, there exists $u \in$ A such that $x \leq t < u$ and $z \leq u$. We have $f(x,z) \leq f(t,u) = u \leq y$.

Finally, a critical ordinal is indecomposable, for if $x < y$ we have $x + y \leq f(x,y)$ (remember that y is infinite).

```
Definition critical_ordinal y :=
  is_ordinal y & infinite_o y & u <o y &
   forall x, u <=o x -> x <o y  -> f x y = y.

Lemma critical_limit  y:
   critical_ordinal y -> limit_ordinal y. (* 17 *)
Lemma is_critical_pr y:
  \omega <=o y -> u <o y ->
  (forall x, u <=o x ->  x <o y -> f x y <=o y)
  -> critical_ordinal y.
Lemma sup_critical A y:
   \omega <=o y -> ordinal_set A -> \osup A = y ->
   (forall x, inc x A -> u <=o x) ->
   (forall x, inc x A -> f x y = y) ->
   critical_ordinal y. (* 17 *)
Lemma sup_critical2:
```

```
  let A:= target (induction_defined0 (fun _ z => f z z) (u +o \2o)) in
  is_ordinal u -> critical_ordinal (\osup A). (* 74 *)
Lemma sup_critical3 A: nonempty A ->
   (forall x, inc x A -> critical_ordinal x) ->
   critical_ordinal (\osup A). (* 36 *)
Lemma critical_indecomposable y:
  critical_ordinal y -> ord_indecomposable y.
End OrdinalInduction.
```

These results correspond to Exercise 2.17(c) (page 334). If we define by induction a function with $w(x) = x+1$ and $g(x,y) = x+1$ we get addition. If we define by induction a function with $w(x) = x$ and $g(x,y) = x+y$ we get multiplication. The non-trivial point is to show that $x+y = \sup_{1\leq z<y}(x+z^+)$ and $x\cdot y = \sup_{1\leq z<y}(x\cdot z^+)$. The result is clear if y is of the form z^+. We use the relations $x+z^+ = (x+z)^+$ and $x\cdot z^+ = x\cdot z+z$. There are two cases to consider. If y is a successor, the result is clear. Otherwise, we use normality of addition or multiplication. We have to show $\sup_{t<y} x+t = \sup_{1\leq z<y} x+z^+$. Note that $t<y$ implies $t^+<y$, since y is limit.

```
Lemma ord_induction_p13 a b:
  is_ordinal a -> is_ordinal b ->
  ord_induction_defined id succ_o (fun u v:Set => succ_o u)  a b
  = a +o b.   (* 69 *)

Lemma ord_induction_p14 a b:
  ordinal_le \1o a -> is_ordinal b ->
  ord_induction_defined (fun z:Set=> \0o) id ord_sum2  a b =
  a *o b. (* 64 *)
```

8.11 Ordinal power

This section corresponds to Exercise 2.18 (page 336). We define a function (for $x \geq 2$) by ordinal induction with $w_0(x) = 1$, $w_1(x) = x$ and $g(x,y) = x\cdot y$. We denote it x^y. All assumptions of the previous section are fulfilled. We add the following specifications: $1^x = 1$, $0^0 = 1$ and $0^y = 0$ for non-zero y.

```
Definition ord_pow' := ord_induction_defined (fun z:Set => \1o) id ord_prod2.
Definition ord_pow a b :=
  Yo (a = \0o)
     (Yo (b = \0o) \1o \0o)
     (Yo (a = \1o) \1o (ord_pow' a b)).
Notation "x ^o y" := (ord_pow x y) (at level 30).

Lemma ord_pow_axioms:
  let w1 := fun x: Set => x in let g := ord_prod2 in let u := \2o in
    (forall x, u <=o x -> x <=o (w1 x))
    & (forall x y, u <=o x -> u <=o y -> x <o (g x y))
    & (forall a a', u <=o a -> a <o a' -> (w1 a) <o (w1 a'))
    &  (forall a b a' b',
      u <=o a -> u <=o b ->  a <=o a' -> b <=o b' -> (g a b) <=o (g a' b'))
    &  (forall a, w1 a = a)
    &  (forall a, u <=o a -> normal_ofs1 (fun b => g a b) u)
    &  (forall a b c, u <=o a -> u <=o b -> u <=o c
       -> g (g a b) c = g a (g b c)).
```

We start with trivial properties.

```
Lemma opow00: \0o ^o \0o = \1o.
Lemma opow0x x: x <> \0o -> \0o ^o x = \0o.
Lemma opow1x x: \1o ^o x = \1o.
Lemma opowx0 x: x ^o \0o = \1o.
Lemma opowx1 x: x ^o \1o = x.
Lemma opow2x x y: \2o <=o  x -> x ^o y = ord_pow' x y.
```

We apply the previous results. The quantity x^y is an ordinal. It is a normal ordinal functional (as a function of y) for $x \geq 2$ and $y \geq 1$. The function is strictly increasing in y for $x \geq 2$; it is increasing (in both arguments) if $x > 0$. We shall see below that n^ω is independent of n when n is an integer ≥ 2, so that the function is not strictly increasing in x for fixed y. We have, for all ordinals (compare with (12)):

$$a^{b+c} = a^b \cdot a^c \qquad a^{b\cdot c} = (a^b)^c. \tag{13}$$

```
Lemma OS_pow x y: is_ordinal x -> is_ordinal y ->
  is_ordinal (x ^o y).
Lemma opow_normal  x: \2o <=o x ->
  normal_ofs1 (fun y => x ^o y) \1o.
Lemma opow_increasing0  x y: \1o <=o x ->
  \1o <=o y  -> x <=o (x ^o y).
Lemma opow_increasing1 x y: \2o <=o x ->
  is_ordinal y  -> y <=o (x ^o y).
Lemma opow_increasing2 x y y': \2o <=o  x ->
  y <o y' -> (x ^o y) <o (x ^o y').
Lemma opow_increasing2b a b c: \2o <=o a ->
  is_ordinal b -> is_ordinal c ->
  ((b <o c) <-> ( (a ^o b) <o (a ^o c))).
Lemma opow_regular a b c: \2o <=o a ->
  is_ordinal b -> is_ordinal c -> a ^o b = a ^o c -> b = c.
Lemma opow_increasing3 x y:
  is_ordinal x -> is_ordinal y -> x ^o y = \0o ->
  (x = \0o & y <> \0o).

Lemma opow_nz ab: is_ordinal a -> is_ordinal b ->
  a <> \0o -> \0o <o (a ^o b).
Lemma opow2_nz  a b: \2o <=o a -> is_ordinal b -> \0o <o (a ^o b).
Lemma omega_pow_nz  x: is_ordinal x -> \omega ^o x <> \0o.
Lemma opow_increasing4 x x' y y':
  x <> \0o ->  x <=o x' -> y <=o  y' ->
 (x ^o y) <=o (x' ^o y').
Lemma opow_sum a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
    a ^o (b +o c) = (a ^o b) *o  (a ^o c).
Lemma opow_prod a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  a ^o (b *o c) = (a ^o b) ^o c.
Lemma opow_succ x y: is_ordinal x -> is_ordinal y ->
  x ^o (succ_o y) = (x ^o y) *o x.
```

If a and b are integers, then the ordinal power a^b is equal to the cardinal power and is an integer (by induction on b, since $^{c+1} = a^c \cdot a$). Thus, if both arguments are $< \omega$ so is the power. If both arguments are countable, so is the power (by ordinal induction; let b be the least for which the power is is not countable; then b has to be a limit ordinal; by normality a^b is then the supremum of countable ordinals indexed by a countable set).

```
Lemma opow_2int a b:
  inc a Bnat -> inc b Bnat ->
  a ^o b = a ^c b.
Lemma opow_2int1 a b:
  a <o \omega -> b <o \omega -> (a ^o b) <o \omega.
Lemma opow_countable x y:
   countable_ordinal x -> countable_ordinal y -> countable_ordinal (x ^o y).
```

Assume $a \geq 2$. Then $a^b \geq ab$. The result is true for $b = 0$, $b = 1$, and also if $b = c + 1$ since $a^{c+1} = a^c \cdot a \geq a^c + a$, since $a \geq 2$ and $a^c \geq 1$. The result is true when b is a limit ordinal by normality of multiplication.

Moreover there exists a unique triple (x, y, z) such that $b = a^x \cdot y + z$, where $z < a^x$ and $0 < y < a$. In the case $b < a$, the triple is $(0, b, 0)$. In the case $a \leq b$, we use (11c). It asserts exists of a unique x such that $a^x \leq b < a^{x+1}$. Then y and z are the quotient and remainder in the division of b by a^x.

```
Lemma opow_increasing5 a b: is_ordinal b ->
  \2o <=o a -> (a *o b) <=o (a ^o b).  (* 52 *)
Lemma opow_increasing6 a b c:
  is_ordinal c -> c <> \0o ->  a <=o b -> (c ^o a) <=o (c ^o b).
Lemma opow_omega_increasing a b:
   a <=o b -> (\omega ^o a) <=o (\omega ^o b).
Lemma opow_omega_increasing2 a b:
   a <o b -> (\omega ^o a) <o (\omega ^o b).

Definition ord_ext_div_pr (a b  x y z: Set) :=
  is_ordinal x & is_ordinal y & is_ordinal z &
  b =  ((a ^o x) *o y) +o z
  & z <o (a ^o x) & y <o a & y <> \0o.

Lemma ord_ext_div_unique a b x y z x' y' z':
  \2o <=o a -> \1o <=o b ->
  ord_ext_div_pr a b x y z ->  ord_ext_div_pr a b x' y' z' ->
  (x=x' & y=y' & z=z'). (* 74 *)
Lemma ord_ext_div_exists a b:
  \2o <=o a -> \1o <=o b ->
  exists x, exists y, exists z,  ord_ext_div_pr a b x y z. (* 23 *)
```

8.12 Cantor Normal Form

The aim of this section is to show that any ordinal has an expansion to base b, and if $b = \omega$, one can perform computations like addition and multiplication. We shall deduce that indecomposable ordinals are powers of ω.

8.12.1 The simple normal form

Consider an integer $b \geq 2$. Any integer m can be written as $m = \sum b^i g_i$, with $g_i < b$. One can remove non-zero terms in the sum and write $m = \sum b^{f_i} g_i$, where f_i is strictly decreasing and $0 < g_i < b$. The same holds for ordinals.

In Exercise 6.12, Bourbaki says that any ordinal α can be written uniquely as

$$\alpha = \gamma^{\lambda_1}\mu_1 + \gamma^{\lambda_2}\mu_2 + \ldots + \gamma^{\lambda_k}\mu_k, \tag{14a}$$

where $0 < \mu_i < \gamma$ and the sequence λ_i is strictly decreasing, provided that $\gamma \geq 2$. A case of interest is when $\gamma = \omega$:

$$\alpha = \omega^{\lambda_1}\mu_1 + \omega^{\lambda_2}\mu_2 + \ldots + \omega^{\lambda_k}\mu_k. \tag{14b}$$

Here we have the condition $\mu_i < \omega$, which says that μ_i is finite, thus of the form $1+1+\ldots+1$. This means that we can write

$$\alpha = \omega^{\beta_1} + \omega^{\beta_2} + \cdots + \omega^{\beta_m}, \tag{14c}$$

where the sequence β_i is decreasing. The greatest exponent in the list is called the degree.

In what follows, we shall consider functional graphs whose domain is some interval $[0, n[$ for $n \in \mathbf{N}$. We shall call this a CNF. Later on, we shall distinguish between CNFb, corresponding to (14a), CNFr, corresponding to (14c), CNFp, corresponding to the product form (18d) below, and CNFq, corresponding to (14b), without the restriction $\mu_i > 0$.

```
Definition CNF_graph f :=
 fgraph f & (exists n, inc n Bnat & domain f = interval_co_0a n).

Lemma CNF_domain_rw  X: CNF_graph X ->
  forall i, (inc i (domain X) <-> i <c (cardinal (domain X))).
Lemma CNF_domain1 X:  CNF_graph X -> inc (cardinal (domain X)) Bnat.
Lemma CNF_domain2 X i:  CNF_graph X -> inc i (domain X) -> inc i Bnat.
Lemma CNF_domain_rw  X i: CNF_graph X ->
  (inc i (domain X) <-> i <c (cardinal (domain X))).
Lemma inc0_int01: inc \0c (interval_co_0a \1c).
```

Given a CNF X of length n and an integer m, we can consider X_r, the restriction of X to $[0, m[$, and X_s, the restriction to $[m, n[$, shifted so that the index set becomes $[0, n-m[$. Given X_1 and X_2, one can merge these CNFs into a CNF Y. We have then $Y_r = X_1$ and $Y_s = X_2$.

```
Definition CNF_r X n:=  restr X (interval_co_0a n).
Definition CNF_s X m k:= (L (interval_co_0a m) (fun z  => V (z +c k) X)).
Definition CNF_s0 X m:= (L (interval_co_0a m) (fun z  => V (succ z) X)).
Definition CNF_m X1 k X2 m :=
  L (interval_co_0a (k +c m))
    (fun z => Yo (z <c k) (V z X1) (V (z -c k) X2)).

Lemma CNF_m_rs X1 X2
  (k := cardinal (domain X1)) (m := (cardinal (domain X2))) :
  CNF_graph X1 -> CNF_graph X2 ->
  (CNF_r (CNF_m X1 k X2 m) k = X1
   & CNF_s (CNF_m X1 k X2 m) m k = X2).  (* 30 *)
Lemma CNF_s_s0 X m : CNF_s X m \1c = CNF_s0 X m.
Lemma CNF_r_cd X n:
  CNF_graph X -> n <=c (cardinal (domain X)) ->
  cardinal  (domain (CNF_r X n)) = n.
```

We study now (14c). Let f_0, f_1, etc, be the β_i in increasing order (so that $f_0 = \beta_m$). We call this a CNFr . The sum of the ω^{f_i} will be denoted by $s(f)$. We start with easy properties. In particular, we consider what happens when the sequence has length zero or one. We also consider what happens if we split X into two subsequences, one of them having a single term.

```
Definition CNFr_ax f:=
```

```
  CNF_graph f &
  (forall i, inc i (domain f) -> is_ordinal (V i f)) &
  (forall i, inc i (domain f) -> inc (succ i) (domain f) ->
     (V i f) <=o (V (succ i) f)).
Definition CNFrv f :=
  ord_sum_expansion (L (domain f) (fun i => (\omega ^o (V i f))))
  (cardinal (domain f)).
Lemma CNFr_axioms f : CNFr_ax f ->
  expansion_ax (L (domain f) (fun i => \omega ^o V i f)) (cardinal (domain f)).
Lemma OS_CNFr f: CNFr_ax f -> is_ordinal (CNFrv f).
Lemma CNFr0 f: CNFr_ax f -> cardinal (domain f) = \0c -> CNFrv f = \0o.
Lemma CNFr1 f: CNFr_ax f -> cardinal (domain f) = \1c ->
   CNFrv f = \omega  ^o (V \0c f).


Lemma CNFr_Al f n (sf := CNF_s0 f n):
  CNFr_ax f ->  inc n Bnat -> cardinal (domain f) = succ n ->
  (CNFr_ax sf &
    is_ordinal (CNFrv sf)  &
    is_ordinal (V \0c f) & is_ordinal (\omega ^o V \0c f) &
    CNFrv f = (CNFrv sf) +o  (\omega ^o V \0c f)).   (* 41 *)
Lemma CNFr_Ar f n (sf := CNF_r f n):
  CNFr_ax f ->  inc n Bnat -> cardinal (domain f) = succ n ->
  (CNFr_ax sf &
    is_ordinal (CNFrv sf)  &
    is_ordinal (V n f) & is_ordinal (\omega ^o V n f) &
    CNFrv f = (\omega ^o V n f) +o (CNFrv sf)).  (* 35 *)
```

Let n be the degree of a CNF X. Then $\omega^n \le s(\mathrm{X}) < \omega^{n+1}$. The first relation is trivial, the second follows from $s(\mathrm{X}) \le \omega^n \cdot k$, where k is the number of terms.

These two inequalities say that if $s(\mathrm{X}) = s(\mathrm{Y})$ then X and Y have the same degree. Write $s(\mathrm{X}) = \omega^n + s(\mathrm{X}_r)$ and $s(\mathrm{Y}) = \omega^n + s(\mathrm{Y}_r)$. We can simplify and get $s(\mathrm{X}_r) = s(\mathrm{Y}_r)$. Uniqueness of the CNF follows by induction.

We prove existence by considering the least ordinal (assumed to exist) that cannot be written in the form (14c), there is a least one a, and we can write $a = \omega^b \cdot d + c$. Since d is non-zero, it has the form $1 + e$, so that $a = \omega^b + (\omega^b \cdot e + c) = \omega^b + c_1$. Note that $a = \omega^b \cdot e + (\omega^b + c)$ because e is finite and $e + 1 = 1 + e$. Note that $c_1 \neq a$, since after simplification on the left by $\omega^b \cdot e$ we would have $\omega^b + c = c$, that implies $\omega \le c$, absurd. Thus c_1 can be written in the form (14c). Let f be its degree. We have $\omega^f \le c_1$. We deduce $\omega^f < \omega^{b+1}$. It follows $f < b + 1$, thus $f \le b$.

```
Lemma CNFr_mon1 f n (x := CNFrv f):
  CNFr_ax f ->  inc n Bnat -> cardinal (domain f) = succ n ->
  (is_ordinal (V n f) & \omega ^o V n f <=o  x & x <> \0o ).
Lemma CNFr_mon2 f n m:
  CNFr_ax f ->  inc n Bnat -> cardinal (domain f) = succ n ->
  V n f <o m -> CNFrv f <o \omega ^o m.
Lemma CNFr_unique f g: ( * 56 *)
  CNFr_ax f -> CNFr_ax g -> CNFrv f = CNFrv g ->
  f = g.
Lemma CNFr_exists x: is_ordinal x ->
  exists f, (CNFr_ax f  & x = CNFrv f). (* 100 *)
```

8.12.2 Indecomposable ordinals

We say that x can be *neglected* before y and write it as $x \ll y$ if $x + y = y$. This implies $x = y = 0$ or $x < y$. Note that, if y is indecomposable, then $x < y$ implies $x \ll y$.

We shall use here the property that any non-zero ordinal x has the form $1 + y$ (where y is $x - 1$), so that $z \cdot x = z + y'$ and $z^x = z \cdot y''$ for some y' and y''.

Obviously $x \ll y$ implies $x \ll y + z$, as well as $x \ll y \cdot z$ if z is non-zero. If $x \ll y$ and $x' \ll y$ then $x + x' \ll y$. If $b < \omega$, we have $b \ll \omega$ since ω is indecomposable. It follows $b \ll \omega^e$ and $b \ll \omega^e \cdot d$, provided that e and d are non-zero.

Assume $a < e$; we have $\omega^a \ll \omega^e$ and $\omega^a \cdot b \ll \omega^e \cdot d$.

```
Definition ord_negl a b := a+o b = b.
Notation "x <<o y" := (ord_negl x y) (at level 60).

Lemma ord_negl_lt a b: is_ordinal a -> is_ordinal b ->
  a <<o b -> ((a = \0o & b = \0o) \/ (a <o b)).
Lemma ord_negl_sum a b c: is_ordinal a -> is_ordinal b -> is_ordinal c
  -> a <<o b -> a <<o (b +o c).
Lemma ord_negl_prod a b c: is_ordinal a -> is_ordinal b ->
   is_ordinal c -> c <>  \0o -> a <<o b -> a <<o (b *o c).
Lemma ord_negl_sum_assoc  a b c:
  is_ordinal a -> is_ordinal b -> is_ordinal c ->
  a <<o c -> b <<o c -> (a +o b) <<o c.
Lemma ord_negl_p1 b:  b <o \omega -> b <<o \omega.
Lemma ord_negl_p2 b e:
  b <o \omega -> e <> \0o ->  is_ordinal e ->
  b <<o ((\omega ^o e)).
Lemma ord_negl_p3 b e d:
  b <o \omega -> e <> \0o -> d <> \0o -> is_ordinal e -> is_ordinal d ->
  b <<o ((\omega ^o e) *o d).
Lemma ord_negl_p4 e e':
   e <o e' -> (\omega ^o e) <<o (\omega ^o e').
Lemma ord_negl_p5 e c e' c':
  c <o \omega ->  is_ordinal c' -> e <o e' -> c' <> \0o ->
  ((\omega ^o e) *o c) <<o ((\omega ^o e') *o c').
```

Let f and g be two CNFr, with $m+1$ and $p+1$ terms. Assume $e < f_m$. Then $\omega^e \ll s(f)$. By induction $s(g) \ll s(f)$ if $g_p < f_m$. A consequence of this property is uniqueness of the CNFr (but we already know this).

```
Lemma ord_negl_p6 e f n:
  CNFr_ax f ->  inc n Bnat -> cardinal (domain f) = succ n ->
  e <o (V n f) -> (\omega ^o e) <<o (CNFrv f).
Lemma ord_negl_pg f n g p:
  CNFr_ax f ->  inc n Bnat -> cardinal (domain f) = succ n ->
  CNFr_ax g ->  inc p Bnat -> cardinal (domain g) = succ p ->
  V p g <o V n f ->  CNFrv g <<o CNFrv f.
```

Consider the expansion of an indecomposable ordinal x. Write it as $x = \omega^n + r$. This is possible since x is non-zero. Since x is indecomposable, it is equal to one of the two terms. It cannot be r by uniqueness of the decomposition. Thus $x = \omega^n$. Conversely, let $x = \omega^n$. Assume $0 < y < x$; if m is the degree of y it follows $m < n$ hence $y \ll x$. Thus x is indecomposable.

Any indecomposable ordinal is either 1 or a limit ordinal. Thus ω^n is limit for non-zero n.

It follows that if n is the degree of x, then ω^n is the greatest indecomposable ordinal $\leq x$. As a consequence, if E is a set of indecomposable ordinals, then the least upper bound of E is an indecomposable ordinal. This is Exercise 2.16 (d) and (e) (page 334).

```
Lemma indecomposable_prop2 x: is_ordinal x ->
  ord_indecomposable x -> exists y, is_ordinal y & x = \omega ^o y.
Lemma indecomposable_prop3 y: is_ordinal y -> (* 26 *)
  (ord_indecomposable (\omega ^o y)).
Lemma indecomposable_limit x: ord_indecomposable x -> is_ordinal x ->
   x = \1o \/ limit_ordinal x.
Lemma indecomposable_limit2 n: is_ordinal n -> n <> \0o ->
   limit_ordinal (\omega ^o n).

Lemma indecomposable_sup1 x: is_ordinal x -> x <> \0o ->
  exists y, (ord_indecomposable y & y <=o x &
    forall z, ord_indecomposable z -> z <=o x -> z <=o y).
Lemma indecomposable_sup E:
  ordinal_set E -> nonempty E ->
  (forall x, inc x E -> ord_indecomposable x) ->
  ord_indecomposable (\osup E).
```

¶ Application. If a and b are two ordinals, and $c = a + b$, we have $\mathrm{Card}(c) = \mathrm{Card}(a +_{\mathrm{ord}} b) = \mathrm{Card}(a) +_{\mathrm{Card}} \mathrm{Card}(b)$. Assume that c is a limit ordinal, and let b be the least monomial in the CNF of c. Let n be the exponent, so that $b = \omega^n$. If $n = 0$ then $b = 1$ and c is not limit. Let's assume $a \neq 0$. Then we can write $a = d + \omega^m$, and $m \geq n$. This implies $a \geq b$. We have shown than ω^m is a limit ordinal; it follows that a is limit. Finally, since c is infinite, its cardinal is the greatest of $\mathrm{Card}(a)$ and $\mathrm{Card}(b)$, thus is $\mathrm{Card}(a)$.

```
Lemma osum_cardinal x y:
  is_ordinal x -> is_ordinal y ->
  cardinal (x +o y) = cardinal x +c (cardinal y).
Lemma cantor_of_limit x: limit_ordinal x ->
  exists a, exists n, is_ordinal a & is_ordinal n & n <> \0o &
   x = a +o (\omega ^o n)
   & (a = \0o \/
      (limit_ordinal a & (\omega ^o n) <=o a
       & cardinal x = cardinal a)).
```

8.12.3 The general normal form

We consider now (14a). We assume $b \geq 2$, n integer and X is a functional graph defined on the interval $[0, n[$ so that each X_i is a pair (e_i, c_i) where e_i is an ordinal, $c_i < b$ and $c_i \neq 0$. We call this a CNFb. We consider a variant where $c_i \neq 0$ is omitted. We will call it a CNFq. The relation $c_i < b$ implies that c_i is an ordinal. We denote by $s(\mathrm{X})$ the sum of all $b^{e_i} \cdot c_i$.

```
Definition cantor_mon b X i := (b ^o (P (V i X)))  *o (Q (V i X)).

Definition CNFq_ax b X :=
  CNF_graph X
  & \2o <=o b
  & (forall i, inc i (domain X) ->
    (is_pair (V i X) & is_ordinal (P (V i X)) & Q (V i X) <o b))
```

```
  & (forall i, inc i (domain X) -> inc (succ i) (domain X) ->
    P (V i X) <o (P (V (succ i) X))).
Definition CNFb_ax b X :=
  CNFq_ax b X &   forall i, inc i (domain X) ->  Q (V i X) <> \0o.

Definition CNFbv b X :=
  ord_sum_expansion (L (domain X) (cantor_mon b X)) (cardinal (domain X)).
Definition CNFv X :=  CNFbv \omega X.
```

We start with basic properties.

```
Lemma CNFq_p0 b X i: CNFq_ax b X  ->
  inc i (domain X) ->
  (is_ordinal  (P (V i X)) & is_ordinal  (Q (V i X)) &
    Q (V i X) <o b & is_ordinal (cantor_mon b X i)& is_pair (V i X)).
Lemma CNFb_p0 b X i: CNFb_ax b X  ->
  inc i (domain X) ->
  (is_ordinal  (P (V i X)) & is_ordinal  (Q (V i X)) & (Q (V i X)) <> \0o &
    Q (V i X) <o b & is_ordinal (cantor_mon b X i)
    &  (cantor_mon b X i) <> \0o & is_pair (V i X)).
Lemma CNFb_axioms b X : CNFq_ax b X ->
  expansion_ax (L (domain X)  (cantor_mon b X)) (cardinal (domain X)).
Lemma OS_CNFq b X: CNFq_ax b X -> is_ordinal (CNFbv b X).
Lemma CNFq0 b X: CNFq_ax b X -> cardinal (domain X) = \0c -> CNFbv b X = \0o.
Lemma CNFq1 b X: CNFq_ax b X -> cardinal (domain X) = \1c ->
   CNFbv b X = cantor_mon b X \0c.
```

Given a sequence X, we can split in into X_r and X_s; conversely two sequences can be merged. We give here three helper lemmas.

```
Lemma CNFq_p2 X n:
  CNF_graph X -> n <=c (cardinal (domain X)) ->
  let Xr :=  CNF_r X n in
  (CNF_graph Xr & domain Xr = interval_co_0a n
    &forall i, inc i (interval_co_0a n) -> (inc i (domain X) & V i Xr = V i X)).
Lemma CNFq_p3 X k m:
  CNF_graph X -> inc k Bnat -> inc m Bnat -> k +c m <=c (cardinal (domain X)) ->
  let Xs := (CNF_s X m k) in
  (CNF_graph Xs & domain Xs = interval_co_0a m
    &forall i, inc i (interval_co_0a m) ->
        (inc (i +c k) (domain X) & V i Xs = V (i +c k) X)).
Lemma CNFq_p4 X1 X2 k m:
  CNF_graph X1 -> CNF_graph X2 ->
  k <=c (cardinal (domain X1)) -> m <=c (cardinal (domain X2)) ->
  let Xm := (CNF_m X1 k X2 m) in
  (CNF_graph Xm & domain Xm = interval_co_0a (k +c m)
    & (forall i, inc i (domain Xm) -> i <c k \/
      (exists j, j <c m & i = j +c k))
    &(forall i, i <c k ->
        (inc i (domain Xm) & inc i (domain X1) & V i Xm = V i X1))
    &(forall i,  i <c m ->
        (inc (i +c k) (domain Xm)  &inc i (domain X2)
      & V (i +c k) Xm = V i X2))).  (* 28 *)
```

We show here that split and merge preserve the conditions.

```
Lemma CNFq_axr b X n:
  CNFq_ax b X -> n <=c (cardinal (domain X)) ->
  CNFq_ax b (CNF_r X n).
Lemma CNFb_axr b X n:
  CNFb_ax b X -> n <=c (cardinal (domain X)) ->
  CNFb_ax b (CNF_r X n).
Lemma CNFq_axs b X k m:
  CNFq_ax b X -> inc k Bnat -> inc m Bnat -> k +c m <=c (cardinal (domain X)) ->
  CNFq_ax b (CNF_s X m k).
Lemma CNFb_axs b X k m:
  CNFb_ax b X -> inc k Bnat -> inc m Bnat -> k +c m <=c (cardinal (domain X)) ->
  CNFb_ax b (CNF_s X m k).
Lemma CNFq_axm b X1 X2 k m:
  CNFq_ax b X1 -> CNFq_ax b X2 ->
  (forall i j, inc i (domain X1) -> inc j (domain X2) ->
      P (V i X1) <o P (V j X2)) ->
  k <=c (cardinal (domain X1)) -> m <=c (cardinal (domain X2)) ->
  CNFq_ax b (CNF_m X1 k X2 m). (* 28 *)
Lemma CNFb_axm b X1 X2 k m:
  CNFb_ax b X1 -> CNFb_ax b X2 ->
  (forall i j, inc i (domain X1) -> inc j (domain X2) ->
      P (V i X1) <o P (V j X2)) ->
  k <=c (cardinal (domain X1)) -> m <=c (cardinal (domain X2)) ->
  CNFb_ax b (CNF_m X1 k X2 m).
```

If X splits as X_r and X_s then $s(X) = s(X_s) + s(X_r)$. We consider the special case when one piece has a single term.

```
Lemma CNF_domain_n b X n:
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
  inc n (domain X).

Lemma CNFq_A b X n m:
  CNFq_ax b X ->  inc n Bnat -> inc m Bnat ->  cardinal (domain X) = (n +c m)->
  CNFbv b X = CNFbv b (CNF_s X m n) +o CNFbv b (CNF_r X n).  (* 31 *)
Lemma CNFq_Al b X n (sX := CNF_s0 X n):
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
  (CNFq_ax b sX &
    is_ordinal (CNFbv b sX)  &
    is_ordinal (cantor_mon b X \0c) &
    CNFbv b X = (CNFbv b sX) +o  (cantor_mon b X \0c)).
Lemma CNFq_Al0 b X n:
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
  Q (V \0c X) = \0o ->
  CNFbv b X = CNFbv b (CNF_s0 X n).
Lemma CNFq_Ar b X n (sX := CNF_r X n):
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
  (CNFq_ax b sX &
    is_ordinal (CNFbv b sX)  &
    is_ordinal (cantor_mon b X n)  &
    CNFbv b X = (cantor_mon b X n) +o (CNFbv b sX)).
Lemma CNFq_Ar0 b X n:
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
  Q (V n X) = \0o ->
  CNFbv b X = CNFbv b (CNF_r X n).
Lemma CNFq_A1 b X n m:
  CNFq_ax b X ->  inc n Bnat -> inc m Bnat ->
```

```
  cardinal (domain X) = succ (n +c m)
  -> CNFbv b X = CNFbv b (CNF_s X m (succ n))
    +o (cantor_mon b X n) +o CNFbv b (CNF_r X n).
```

By induction, if X is a CNFq with degree $< e$, then $s(X) < b^e$ (this holds also if all exponents of X are $< e$).

```
Lemma CNFq_pg0 b X:  CNFq_ax b X -> (is_ordinal b & b <> \0o).
Lemma CNFq_pg1 b X n:
  CNFq_ax b X -> inc n Bnat -> cardinal (domain X) = succ n ->
  CNFbv b X <o (b ^o (succ_o (P (V n X)))).  (* 37 *)
Lemma CNFq_pg2 b X n a:
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n->
  (P (V n X)) <o a ->  CNFbv b X <o (b ^o a).
Lemma CNFq_pg3 b X a:
  CNFq_ax b X ->  is_ordinal a
  -> (forall i, inc i (domain X) ->  (P (V i X)) <o a)
  -> CNFbv b X <o (b ^o a).
Lemma CNFq_pg b X n:
  CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
  CNFbv b (CNF_r X n) <o (b ^o (P (V n X))). (* 20 *)
```

If X is a CNFb of degree n, and the associated coefficient is non-zero, then $s(X) \geq b^n$. This implies uniqueness. Existence can be proved as follows: any ordinal x can be written as $x = b^e \cdot c + r$, with $r < x$. By induction, $r = s(X_1)$ for some X_1, and $b^e \cdot c = s(X_2)$ for some X_2 that has a single term. Thus $x = s(X)$, where X is the merge of X_1 and X_2. This is a CNF, as all exponents of X_1 are $< e$. We first establish a lemma that says that exponents are in increasing order, so that it suffices to show that the degree n of X_1 is $< e$. But $s(X_1) \geq b^n$ and $r < b^e$.

```
Lemma CNFq_pg4 b X n:
  CNFb_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n->
  b ^o (P (V n X)) <=o CNFbv b X.
Lemma CNF_singleton b c e (X:= L (interval_co_0a \1c)  (fun _ => (J e c))):
  is_ordinal e -> c <o b ->  \2c <=o  b ->
  (CNFq_ax b X & CNFv X = b ^o e *o c &
    e <> \0o -> CNFb_ax b X).
Lemma CNF_exponents_sM b X: CNFq_ax b X ->
   forall i j, inc i (domain X) -> inc j (domain X) -> i <c j ->
     P (V i X) <o P (V j X).
Lemma CNF_exponents_M b X: CNFq_ax b X ->
   forall i j, inc i (domain X) -> inc j (domain X) -> i <=c j ->
     P (V i X) <=o P (V j X).
Lemma CNF_exp_bnd b X e:
  \2c <=o b -> is_ordinal e ->
  CNFb_ax b X -> CNFbv b X <o b ^o e ->
  (forall i, inc i (domain X) -> P (V i X) <o e).

Lemma CNFb_unique b X Y:  (* 79 *)
  CNFb_ax b X -> CNFb_ax b Y -> CNFbv b X = CNFbv b Y ->
  X = Y.
Lemma CNFb_exists a b:
  is_ordinal a -> \2c <=o  b ->
  exists X, CNFb_ax b X &  a = CNFbv b X. (* 48 *)
```

¶ From now on, we shall consider only the case where the base b is ω. Such a CNFb will be called a CNF. We are often interested in the case where the number of terms is non-zero.

Let $\mathfrak{S}_e$ be the set of all functional graphs X, defined on some interval $[0, n[$ such that each X_i is a pair (a, b) with $a < \omega$ and $b < e$. This can be restated as $X_i \in \omega \times e$. If X is a CNF, with value x, then $x < b^{x+1}$, hence $X \in \mathfrak{S}_{x+1}$. If x is any ordinal, then there is a unique CNF X such that $s(X) = x$ and it is in $\mathfrak{S}_{x+1}$. We call it *the CNF* of X. The CNF of zero is the empty function. the CNF of a non-zero ordinal has at least one term, thus has a degree. If $x < \omega^k$, the degree is $< k$.

```
Definition  CNF_axn X n:=
    CNFb_ax \omega X & inc n Bnat & cardinal (domain X) =  succ n.

Definition set_of_CNF_lt e :=
  unionb (L Bnat
    (fun n => set_of_gfunctions (interval_co_0a n) (product e \omega))).
Definition the_CNF x :=
    select (fun X => CNFb_ax \omega X & CNFv X = x)
       (set_of_CNF_lt (succ_o x)).
Definition the_CNF_len x := (cpred (cardinal (domain (the_CNF x)))).
Definition the_CNF_degree x := P (V (the_CNF_len x) (the_CNF x)).

Lemma set_of_CNF_lt_pr1 X :
  CNFb_ax \omega X  -> inc X (set_of_CNF_lt (succ_o (CNFv X))).
Lemma the_CNF_pr0 X: CNFb_ax \omega X  ->  the_CNF (CNFv X) = X.
Lemma the_CNF_pr1 x (X:= the_CNF x): is_ordinal x ->
  (CNFb_ax \omega X & CNFv X = x).
Lemma the_CNF_zero: the_CNF \0o = emptyset.
Lemma the_CNF_nz x (n:= the_CNF_len x):
  is_ordinal x -> x <> \0o ->
  (inc n Bnat & cardinal (domain (the_CNF x)) = succ n).
Lemma CNF_degree_pr1 X n:
   CNF_axn X n -> is_ordinal (P (V n X)).
Lemma the_CNF_nz2 X n (x := CNFv X):
  CNF_axn X n ->
  (is_ordinal x & x <> \0o & (the_CNF_degree x) = P (V n X)).
Lemma the_CNF_mon1 x:
  is_ordinal x -> x <> \0o ->
  \omega ^o the_CNF_degree x <=o x.
Lemma the_CNF_mon2 x:
  is_ordinal x -> x <> \0o ->
  x <o \omega ^o (succ_o (the_CNF_degree x)).
Lemma the_CNF_nz3 a: is_ordinal a -> a <> \0o
  -> is_ordinal (the_CNF_degree a).
Lemma the_CNF_nz4 a e:
  is_ordinal e ->  a <o (\omega ^o e) -> a <> \0o ->
  (the_CNF_degree a) <o e.
```

8.12.4 Cantor normal form and operations

Let's consider α and its CNF (14b) and β with its CNF

$$\beta = \omega^{\kappa_1}\nu_1 + \omega^{\kappa_2}\nu_2 + \ldots + \omega^{\kappa_m}\nu_m. \tag{14d}$$

The objective of this section is to compute the CNF of $\alpha+\beta$, $\alpha\cdot\beta$ and α^β.

We restate ord_negl_pg in the case of (14b): assume that x has degree n. Then $\omega^e \cdot c \ll x$ if $e < n$ and $c < \omega$. It follows that

$$\lambda_1 < \kappa_1 \implies \alpha + \beta = \beta. \tag{15b}$$

```
Lemma ord_negl_p7 e c X n:
  CNF_axn X n ->c <o \omega -> e <o (P (V n X)) ->
  c <o \omega -> e <o (P (V n X)) ->
   ((\omega ^o e) *o c)  <<o (CNFv X).
Lemma ord_negl_p8 Xp p Xn n:
  CNF_axn Xp p ->  CNF_axn Xn n -> (P (V p Xp)) <o (P (V n Xn))  ->
  (CNFv Xp) <<o  (CNFv Xn). (* 30 *)
```

Let $\bar{Y}$ the sequence obtained by modifying a single coefficient of Y. If the new coefficient c satisfies $0 < c < \omega$, we get a CNF. If we change the leading coefficient, we get $s(\bar{Y}) = m + s(Y')$, where Y' is the restriction of Y, and m is a monomial. If the coefficient is obtained by adding c to the leading coefficient of Y, we get in fact $s(\bar{Y}) = \omega^n \cdot c + s(Y)$.

```
Definition CNF_change_n f n x :=
  L (domain f) (fun z => Yo (z=n) (J (P (V n f)) x) (V z f)).

Lemma CNF_change_n_pr0 f n x i: inc i (domain f) ->
  P (V i (CNF_change_n f n x)) =  P (V i f).
Lemma CNF_change_n_pr b f n x:
  x <> \0o ->  x <o b -> inc n (domain f) ->
  CNFb_ax b f ->
  CNFb_ax b (CNF_change_n f n x).
Lemma CNF_change_nv  X n c (Xc:= CNF_change_n X n (c +o (Q (V n X)))):
  c <o \omega ->   c <o \omega -> CNF_axn X n ->
  (CNFb_ax \omega Xc & CNFv Xc = \omega ^o (P (V n X)) *o c +o CNFv X). (* 34 *)
```

The sum of two CNFs X and Y can be computed as follows. Assume that Y is of degree n. Split X in to two parts such that $s(X) = s(X_s) + s(X_r)$, where exponents of X_s are $\geq n$, those of X_r are $< n$. We have $s(X) + s(Y) = s(X_s) + s(Y)$. The case $X_s = 0$ has been studied above.

If n is not an exponent of X, then the merge of X_s and Y is the CNF of the sum. Otherwise, we have $s(X_s) = s(X_s') + m$ and $s(X) + s(Y) = s(X_s') + m + s(Y)$, where $m = \omega^n \cdot c$ for some c. Let $\bar{Y}$ be as above, so that $m + s(Y) = s(Y')$. Then the merge of X_s' and Y' is the CNF of the sum. We write this as

(15b) $\lambda_i > \kappa_1 > \lambda_{i+1} \implies \alpha+\beta = \omega^{\lambda_1}\mu_1 + \omega^{\lambda_2}\mu_2 + \ldots + \omega^{\lambda_i}\mu_i + \omega^{\kappa_1}\nu_1 + \omega^{\kappa_2}\nu_2 + \ldots + \omega^{\kappa_m}\nu_m.$

(15c) $\kappa_1 = \lambda_{i+1} \implies \alpha+\beta = \omega^{\lambda_1}\mu_1 + \omega^{\lambda_2}\mu_2 + \ldots + \omega^{\lambda_i}\mu_i + \omega^{\kappa_1}(\mu_{i+1}+\nu_1) + \omega^{\kappa_2}\nu_2 + \ldots + \omega^{\kappa_m}\nu_m.$

```
Lemma CNF_sum_pr1 X Y n m p (X1 := CNF_m Y (succ p) (CNF_s X m n) m):
  inc n Bnat ->  inc m Bnat ->  inc p Bnat ->
  cardinal (domain X) =  (n +c m) ->
  cardinal (domain Y) =  succ p ->
  CNFb_ax \omega X ->
  CNFb_ax \omega Y ->
  P (V p Y) <o P (V n X)  -> (n = \0c \/  P (V (cpred n) X) <o  P (V p Y)) ->
  (CNFb_ax \omega X1 & CNFv X +o CNFv Y = CNFv X1). (* 49 *)
Lemma CNF_sum_pr2 X Y n m p
  (X1 := CNF_m (CNF_change_n Y p (Q (V n X) +o Q (V p Y)))
          (succ p) (CNF_s X m (succ n)) m):
  inc n Bnat -> inc m Bnat -> inc p Bnat ->
  cardinal (domain X) =  (succ n +c m) ->
```

```
  cardinal (domain Y) =  succ p ->
  CNFb_ax \omega X ->
  CNFb_ax \omega Y ->
  P (V p Y) = P (V n X)  ->
  (CNFb_ax \omega X1 & CNFv X +o CNFv Y = CNFv X1). (* 75 *)
```

We consider here the special case where $X'_s = 0$. This corresponds to $m = 0$ in the previous lemma, or $\kappa_1 = \lambda_1$ in (15c). It can be used to compute $\alpha + \alpha$, and by induction $\alpha \cdot \nu$ for any integer ν.

(15d) $$\kappa_1 = \lambda_1 \implies \alpha + \beta = \omega^{\kappa_1}(\mu_1 + \nu_1) + \omega^{\kappa_2}\nu_2 + \ldots + \omega^{\kappa_m}\nu_m.$$

(16a) $$\alpha \cdot \nu = \omega^{\lambda_1}\mu_1 \cdot \nu + \omega^{\lambda_2}\mu_2 + \ldots + \omega^{\lambda_k}\mu_k.$$

```
Lemma CNF_sum_pr3 X Y n p
  (X1 := (CNF_change_n Y p (Q (V n X) +o Q (V p Y)))):
  CNF_axn X n -> CNF_axn Y p -> P (V p Y) = P (V n X)  ->
  (CNFb_ax \omega X1 & CNFv X +o CNFv Y = CNFv X1).
Lemma CNF_prod_pr1 X n k
  (X1 := CNF_change_n X n ((Q (V n X)) *o k)):
  CNF_axn X n  -> k <> \0o -> k <o \omega ->
  (CNFb_ax \omega X1 & CNFv X *o k = CNFv X1). (* 43 *)
```

Let's say that x is a multiple of ω if it can be written as $x = \omega \cdot x'$. The following are equivalent for non-zero ordinals: x is a multiple of ω, x is limit and the least exponent of x is non-zero. Let's first notice that $x = y + \omega^n$, for some ordinal y, where n is the least exponent in the CNF. If $n = 0$, then x is a successor, this is not limit. Otherwise, ω^n is indecomposable, thus limit and x is limit. If n is non-zero, then ω^n is a multiple of ω, and by induction, x will be. The converse is left as an exercise (a multiple of ω is never a successor, consider for instance a CNFr).

```
Lemma CNF_limit X n:
  CNF_axn X n -> P (V \0c X) <> \0o  ->
  exists z, is_ordinal z & CNFv X = \omega *o z. (* 35 *)
Lemma CNF_limit0 X n:
  CNF_axn X n ->
  (exists y, is_ordinal y &  (CNFv X) = y +o \omega ^o  P (V \0c X)).
Lemma CNF_limit1 X n:
  CNF_axn X n ->  P (V \0c X) = \0o  ->  ~ (limit_ordinal (CNFv X)).
Lemma CNF_limit2 X n:
  CNF_axn X n ->  P (V \0c X) <> \0o  ->  (limit_ordinal (CNFv X)).
Lemma CNF_limit3 x (X := the_CNF x) (n := the_CNF_len x) :
  limit_ordinal x ->
  (CNF_axn X n & CNFv X = x & P (V \0c X) <> \0o).
Lemma CNF_limit4 x: limit_ordinal x ->
  exists z, is_ordinal z & x = \omega *o z.
```

If we take the supremum (16a) over all integers ν we find that $\alpha \cdot \omega = \omega^{\lambda_1} \cdot \omega$, so that $\alpha \cdot \beta_1 = \omega^{\lambda_1} \cdot \beta_1$ for any limit ordinal β_1. The RHS is as β_1, but with exponents shifted by λ_1.

```
Definition CNF_se X e :=
    L (domain X) (fun i => J (e +o P (V i X))  (Q (V i X))).
```

```
Lemma CNF_se_p X e (Xs:= (CNF_se X e)):
   CNFb_ax \omega X -> is_ordinal e ->
   (CNFb_ax \omega Xs & CNFv Xs = \omega ^o e *o CNFv X). (* 51 *)
Lemma CNF_prod_pr2 X n:
  CNF_axn X n ->
  CNFv X  *o \omega = \omega ^o (P (V n X)) *o \omega.  (* 29 *)
Lemma CNF_prod_pr2bis x: is_ordinal x -> x <> \0o ->
  x *o \omega = \omega ^o (the_CNF_degree x) *o \omega.
Lemma CNF_prod_pr3 x y: is_ordinal x -> x <> \0o -> limit_ordinal y ->
  x *o y = \omega ^o (the_CNF_degree x) *o y.
```

Let's write $\beta = \beta_1 + \nu$, where β_1 is limit and ν integer. We have then $\alpha\cdot\beta = \alpha\cdot\beta_1 + \alpha\cdot\nu$. These terms has the form $s(A)$ and $s(B)$, where all exponents of A are greater than the exponents of B. It follows that the sum is $s(C)$ where C is the merge of A and B (We have given above the CNF of both terms; addition is trivial: it suffices to merge the CNFs, we use CNF_sum_pr1 with $n = 0$, the shift of X is X).

Thus, we get, in the case where the least exponent is non-zero

$$\kappa_m \neq 0 \implies \alpha\cdot\beta = \omega^{\lambda_1}\cdot\beta = \omega^{\lambda_1+\kappa_1}\nu_1 + \omega^{\lambda_1+\kappa_2}\nu_2 + \cdots + \omega^{\lambda_1+\kappa_m}\nu_m, \tag{16b}$$

and otherwise

$$\kappa_m = 0 \implies \alpha\cdot\beta = \omega^{\lambda_1+\kappa_1}\nu_1 + \omega^{\lambda_1+\kappa_2}\nu_2 + \ldots + \omega^{\lambda_1+\kappa_{m-1}}\nu_{m-1} + \omega^{\lambda_1}\mu_1\cdot\nu_m + \omega^{\lambda_2}\mu_2 + \ldots + \omega^{\lambda_k}\mu_k. \tag{16c}$$

```
Lemma CNF_prod_pr4 X n Y p:
  CNF_axn X n -> CNF_axn Y p ->  P (V \0c Y) <> \0o ->
  CNFv X *o CNFv Y =  (\omega ^o (P (V n X))) *o CNFv Y.
Lemma CNF_prod_pr5 X n Y p (Z := CNF_se Y (P (V n X))):
  CNF_axn X n -> CNF_axn Y p ->  P (V \0c Y) <> \0o ->
  (CNFb_ax \omega Z & CNFv X *o CNFv Y =  CNFv Z).
Lemma CNF_prod_pr6 X n Y m
  (X1 := CNF_change_n X n (Q (V n X) *o (Q (V \0c Y))))
  (Y1:= CNF_se (CNF_s Y m \1c) (P (V n X)))
  (Z := CNF_m X1 (succ n) Y1 m):
  CNF_axn X n -> CNF_axn Y m ->  P (V \0c Y) = \0o ->
  (CNF_axn Z (n +c m) & CNFv X *o CNFv Y = CNFv Z). (* 65 *)
```

¶ We prove here the following remark of Cantor [5, §19G], that $\omega^p + 1$ is not the product of two non-trivial ordinals. The case $p = 1$ is trivial. Assume $\omega^p + 1 = \alpha\cdot\beta$. Obviously no factor can be zero. Note that (16b) cannot be applied as the least exponent is $\lambda_1 + \kappa_m > 0$. This means that (16c) applies. Counting the number of terms shows that one of α or β has a single term. It is then one.

```
Lemma oprod2_one a b: is_ordinal a -> is_ordinal b ->
  a *o b = \1o -> (a = \1o & b = \1o).
Lemma oprod2_two a b: is_ordinal a -> is_ordinal b ->
  a *o b = \2o -> (a = \1o \/ b = \1o).
Lemma CNF_succ_pow1 n c:
  \0o <o n ->  c <> \0o -> c <o \omega ->
  the_CNF (succ_o ((\omega ^o n) *o c)) =
     L (interval_co_0a \2c) (fun z => Yo (z = \0c) (J \0o \1o) (J n c)).
Lemma CNF_succ_pow n: \0o <o n ->
  the_CNF (succ_o (\omega ^o n)) =
     L (interval_co_0a \2c) (fun z => Yo (z = \0c) (J \0o \1o) (J n \1o)).
```

```
Lemma succ_pow_omega_irred p a b:
  is_ordinal p -> is_ordinal a -> is_ordinal b ->
  succ_o (\omega ^o p) = a *o b ->
  (a = \1o \/ b = \1o).  (* 67 *)
```

Let's compute α^β. If both arguments are finite, by induction, the value is finite, and equal to the cardinal power. Taking the supremum for $\beta < \omega$ yields $\alpha^\omega = \omega$.

(17a) $$2 \leq \alpha < \omega \implies \alpha^\omega = \omega.$$

If $\lambda_k > 0$, we get $\alpha^2 = \omega^{\lambda_1} \cdot \alpha$ and by induction

(17b) $$\lambda_k \neq 0 \implies \alpha^{n+1} = \omega^{\lambda_1 \cdot n} \cdot \alpha.$$

Assume α infinite, so that $\lambda_1 > 0$. Using either (16b) or (16c) shows the following:

(17c) $$n > 0, \lambda_1 > 0 \implies \alpha^n = \omega^{\lambda_1 \cdot n} \cdot \mu_1 + \ldots$$

```
Lemma opow_int_omega n:
  \2o <=o n -> n <o \omega -> n ^o \omega = \omega.  (* 29 *)

Lemma CNF_pow_pr1 X n k (x:=  CNFv X):
  CNF_axn X n -> P (V \0c X) <> \0o -> inc k Bnat ->
  x ^o (succ_o k) =  (\omega ^o ((P (V n X)) *o k)) *o x.
Lemma CNF_pow_pr2 X n Y m k:  (* 54 *)
  CNF_axn X n ->  P (V \0c X) <> \0o -> inc k Bnat ->  k <> \0c ->
  CNFb_ax \omega Y -> inc m Bnat -> cardinal (domain Y) = m ->
  (CNFv X) ^o k  = (CNFv Y) ->
  (m <> \0c & (V (cpred m) Y) = J (P (V n X) *o k) (Q (V n X))).
```

Cantor deduces the relation (17d) below, by taking the supremum for $n < \omega$. There is a simpler proof. We have $\omega^{\lambda_1} \leq \alpha \leq \omega^{\lambda_1+1}$ so that $\omega^{\lambda_1 \cdot \omega} \leq \alpha^\omega \leq \omega^{(\lambda_1+1)\cdot\omega}$. We have $(\lambda_1 + 1) \cdot \omega = \lambda_1 \cdot \omega$, so that we get

(17d) $$\lambda_1 > 0 \implies \alpha^\omega = \omega^{\lambda_1 \cdot \omega}.$$

If $\kappa_m \neq 0$ then $\beta = \omega\beta'$, for some β'; it follows

(17e) $$\kappa_m \neq 0, 2 \leq n < \omega, \lambda_1 > 0 \implies \alpha^\beta = \omega^{\lambda_1 \cdot \beta}, \qquad n^\beta = \omega^{\beta'}.$$

If $\kappa_m = 0$, then $\beta = \beta_1 + \mu_m$, and $\alpha^\beta = \alpha^{\beta_1} \cdot \alpha^{\mu_m}$. For the first factor we can apply (17e). For the second factor, we apply (17c), unless α is finite. The exact expression is much too complicated, and we shall not give it.

```
Lemma CNF_pow_pr3 X n:
   CNF_axn X n -> P (V n X) <> \0o ->
  (CNFv X) ^o \omega  = \omega ^o  (P (V n X) *o \omega).

Lemma CNF_pow_pr4 X n Y m:
  inc n Bnat -> CNF_ax \omega X (succ n) ->  P (V n X) <> \0o ->
  inc m Bnat -> CNF_ax \omega Y (succ m)  ->
  P (V \0c Y) <> \0o ->
  let sX := CNF X (succ n) in
  let sY := CNF Y (succ m) in
  sX ^o sY  = \omega ^o  (P (V n X) *o sY).
```

```
Lemma CNF_pow_pr5 x Y m:
  \2o <=o x -> x <o \omega ->
  inc m Bnat -> CNF_ax \omega Y (succ m)  ->
  P (V \0c Y) <> \0o ->
  let sY := CNF Y (succ m) in
  exists z, is_ordinal z & sY = \omega *o z & x ^o sY  = \omega ^o z.
```

8.12.5 The product form

Assume that n is an ordinal and a a non-zero integer. We have $a\cdot(\omega^n+1)=\omega^n+a$ and

$$(\omega^n+1)\cdot a=\omega^n\cdot a+1. \tag{18a}$$

The first relation is obvious (and will not be needed), the second is a consequence of (16a).

```
Definition CNFp_value1 x := succ_o ((\omega ^o (P x)) *o (Q x)).
Definition CNFp_value2 x := (succ_o (\omega ^o (P x))) *o (Q x).
Lemma CNFp_pr1 x:
  (is_pair x & is_ordinal (P x) &  Q x <> \0o & Q x <o \omega) ->
  P x <> \0o -> CNFp_value1 x = CNFp_value2 x.
```

We consider now a sequence X_i, of pairs of non-zero ordinals (n,a) where a is finite. We associate to it the product p_X of all $(\omega^n+1)\cdot a$. We consider an additional pair (e,c) (where e might be zero), and consider $\omega^e\cdot c\cdot p_X$.

We start with trivial results.

```
Definition CNFp_ax1 X :=
  CNF_graph X &
  (forall i, inc i (domain X) ->
    (is_pair (V i X) & is_ordinal (P (V i X)) & Q (V i X) <o \omega &
      P (V i X) <> \0c & Q (V i X) <> \0c)).
Definition CNFp_ax X p :=
  CNFp_ax1 X &
  is_pair p & is_ordinal (P p) & Q p <> \0o & Q p <o \omega.
Definition CNFpv1 X :=
  ord_prod_expansion (L (domain X)  (fun z => (CNFp_value1 (V z X))))
      (cardinal (domain X)).
Definition CNFpv X p := ((\omega ^o (P p)) *o (Q p)) *o (CNFpv1 X).

Lemma CNFp_aux X:
  CNFp_ax1 X ->
   expansion_ax (L (domain X) (fun z : Set => CNFp_value1 (V z X)))
   (cardinal (domain X)).
Lemma OS_CNFp1 X:  CNFp_ax1 X -> is_ordinal (CNFpv1 X).
Lemma OS_CNFp X p:  CNFp_ax X p-> is_ordinal (CNFpv X p).
Lemma CNFp_0 X: (cardinal (domain X) = \0c) -> CNFpv1 X = \1o.
Lemma CNFp_1 X:  CNFp_ax1 X ->  cardinal (domain X) = \1c ->
  CNFpv1 X = CNFp_value1 (V \0c X).
Lemma CNFp_A X n m :
  CNFp_ax1 X ->  inc n Bnat -> inc m Bnat ->  cardinal (domain X) = (n +c m)->
  CNFpv1 X = CNFpv1 (CNF_r X n) *o CNFpv1  (CNF_s X m n). (* 25 *)
Lemma CNFp_r X n:
 CNFp_ax1 X ->  inc n Bnat -> cardinal (domain X) = succ n ->
 CNFpv1 X = CNFpv1 (CNF_r X n)  *o (CNFp_value1 (V n X)).
```

Assume $k \geq 2$ in (14b); we can write $\lambda_1 = \lambda_2 + p$ with $p > 0$. Write $\alpha = \omega^{\lambda_2+p} \cdot c + r$. Then $r \cdot (\omega^p \cdot c + 1) = (r \cdot \omega^p) \cdot c + r = \alpha$ according to (16b). By induction,

(18b) $$\alpha = \omega^{\lambda_k} \cdot \mu_k \cdot (\omega^{\lambda_{k-1}-\lambda_k}\mu_{k-1} + 1) \cdots (\omega^{\lambda_2-\lambda_3}\mu_2 + 1) \cdot (\omega^{\lambda_1-\lambda_2}\mu_1 + 1)$$

or

(18c) $$\alpha = \omega^{\lambda_k} \cdot \mu_k \cdot (\omega^{\lambda_{k-1}-\lambda_k} + 1) \cdot \mu_{k-1} \cdots (\omega^{\lambda_2-\lambda_3} + 1) \cdot \mu_2 \cdot (\omega^{\lambda_1-\lambda_2} + 1) \cdot \mu_1.$$

Renaming the exponents yields

(18d) $$\alpha = \omega^{\nu_k} \cdot \mu_k(\omega^{\nu_{k-1}} + 1) \cdot \mu_{k-1} \cdots (\omega^{\nu_2} + 1) \cdot \mu_2 \cdot (\omega^{\nu_1} + 1) \cdot \mu_1.$$

This form exists (if α is non-zero), and is unique provided all exponents are non-zero (with the possible exception of ν_k), since (14b) and (18d) are equivalent.

```
Lemma CNFp_p2 X n
   (a:= P (V n X) -o P (V (cpred n) X)) (b := Q (V n X)):
   CNF_axn X n -> n <> \0c ->
   (is_ordinal a & is_ordinal b & a <> \0o & b <> \0o & b <o \omega &
    CNFv X = (CNFv (CNF_r X n))  *o (CNFp_value1 (J a b))).  (* 37 *)
Lemma CNFp_p3 X n:  CNF_axn X n ->
  exists Y, exists p,
  (CNFp_ax Y p
   & cardinal (domain Y) = n
   & (forall i, inc i (domain Y) ->
     (is_pair (V i Y) & P (V i Y) = (P (V (succ i) X)) -o (P (V i X))
       & Q (V i Y) = Q (V (succ i) X)))
   & CNFv X = CNFpv Y p & p = V \0c X).  (* 72 *)
Lemma CNFp_exists x: is_ordinal x -> x <> \0c ->
  exists Y, exists p,  CNFp_ax Y p & x = CNFpv Y p.
Lemma CNFp_p4 Y p (n:= cardinal (domain Y)): CNFp_ax Y p  ->
  exists X,
    CNF_axn X n
   & (forall i, inc i (domain Y) ->
     (is_pair (V i Y) & P (V i Y) = (P (V (succ i) X)) -o (P (V i X))
       & Q (V i Y) = Q (V (succ i) X)))
   & p = V \0c X & CNFv X = CNFpv Y p. (* 76 *)
Lemma CNFp_unique Y p  Y' p' :
   CNFp_ax Y p  ->  CNFp_ax Y' p' -> CNFpv Y p = CNFpv Y' p' ->
   (Y = Y' & p = p').
```

We now study the question when (S): $\alpha + \beta = \beta + \alpha$ or (P): $\alpha \cdot \beta = \beta \cdot \alpha$. Let ($S_1$) (respectively ($P_1$)) be the assumption that there exist two integers n and m (i.e., two finite ordinals) such that $\alpha = \gamma \cdot n$, $\beta = \gamma \cdot m$ (respectively: $\alpha = \gamma^n$ and $\beta = \gamma^m$). Then (S_1) implies (S) and (P_1) implies (P). Condition (P_1) has been studied by Cantor [5, §19K].

Conversely, assume (S). Consider the CNF of these numbers. If $\lambda_1 < \kappa_1$, then $\alpha + \beta = \beta$. It follows $\beta = \beta + \alpha$, which implies $\alpha = 0$. In this case, we can chose $\gamma = \beta$, $n = 0$ and $m = 1$. In the case $\lambda_1 = \kappa_1$ relation (15d) shows that the expansions are the same, with the possible exception of the leading coefficient. In other terms, $\alpha = \omega^k \cdot n + r$ and $\beta = \omega^k \cdot m + r$. Let $\gamma = \omega^k + r$; relation (16a) says $\gamma \cdot n = a$ and $\gamma \cdot m = b$.

```
Lemma osum2_commutes a b: (* 94 *)
  is_ordinal a -> is_ordinal b ->
  ((a +o b = b +o a) <-> (exists c, exists n, exists m,
    is_ordinal c & inc n Bnat & inc m Bnat & a = c *o n & b = c *o m)).
```

The case of a product is a bit more complicated. The conditions (P_2) that says that one factor is zero, and (P_3) that says both arguments are integers imply (P). There are some other cases, for instance $\alpha = \omega^2 + \omega$ and $\beta = \omega \cdot \alpha$ satisfy (P). This pair does not satisfy (P_1) for, if $\alpha = \gamma^n$, then either $n = 1$, γ is of degree two, and β is of degree $3 = 2m$, absurd, or $n = 2$ and α is the square of an ordinal of degree one, which is equally absurd, as $(\omega + c)^2 = \omega^2 + \omega \cdot c + c$.

Let (P_4) be the assumption that the pair (α,β) is such that α is a limit ordinal (i.e., $\lambda_k \neq 0$), there exists an ordinal γ, two integers n and m such that the degree of α is $\gamma \cdot n$, while $\beta = \omega^{\gamma \cdot m} \cdot \alpha$. The example above satisfies this condition.

Assume (P_4) holds. The pair $(\omega^{\gamma \cdot n}, \omega^{\gamma \cdot m})$ satisfies (P), and, by (16b), it follows that (P) holds for (α,β). Conversely, assume that α and β are limit ordinals satisfying (P) so that we can apply (16b) twice. We get $k = m$, $\nu_i = \mu_i$ and $\lambda_1 + \kappa_i = \kappa_1 + \lambda_i$. Taking $i = 1$ shows that the degrees of α and β satisfy (S). Write $\lambda_1 = \gamma \cdot n$ and $\mu_1 = \gamma \cdot m$. This shows that the pair (α,β) satisfies (P_4).

Assume $\kappa_m \neq 0$ and $\lambda_k = 0$. We may consider (16b) and (16c). Counting the number of terms yields $m = 1$. This means that β is an integer. Looking at the coefficients, we see that $\beta = 1$, so that (P_1) holds with $\gamma = \alpha$, $n = 1$ and $m = 0$.

Assume finally that α and β are successors. If they are finite, they commute. Assume β finite; so that $\alpha \cdot \beta$ is given by (16a). Using (16c) for $\beta \cdot \alpha$ and considering trailing coefficients shows $\beta = 1$.

```
Definition oprod_comm x y :=  (x *o y = y *o x).

Lemma oprod2_comm1 X n Y m:
  CNF_axn X n -> CNF_axn Y m -> P (V \0c Y) = \0o -> P (V \0c X) <> \0o ->
  oprod_comm (CNFv X) (CNFv Y) ->  CNFv Y = \1o.  (* 23 *)
Lemma oprod2_comm2 X n mu (dX := P (V n X)) (y := (\omega ^o mu) *o (CNFv X)):
  CNF_axn X n ->  P (V \0c X) <>  \0o -> is_ordinal mu ->
  dX +o mu = mu +o dX -> oprod_comm (CNFv X) y. (* 20 *)

Definition oprod2_comm_P4 x y :=
  exists n, exists X, exists gamma, exists c1, exists c2,
    inc n Bnat & CNF_ax \omega X (succ n)  & P (V \0c X) <>  \0o &
      is_ordinal gamma & inc c1 Bnat & inc c2 Bnat &
      x = CNF X (succ n) &
      P (V n X) = gamma *o c1 &
      y = \omega ^o (gamma *o c2) *o x.
Lemma oprod2_comm3 x y:  oprod2_comm_P4 x y ->  oprod_comm x y.
Lemma oprod2_comm4 X n Y m (x := CNFv X) (y := CNFv Y):
  CNF_axn X n -> CNF_axn Y m -> P (V \0c X) <>  \0o -> P (V \0c Y) <> \0o ->
  oprod_comm x y ->
  (oprod2_comm_P4 x y \/ oprod2_comm_P4 y x). (* 72 *)
Lemma oprod2_comm5 X Y n m x y:
  inc n Bnat -> inc m Bnat ->
  CNF_ax \omega X (succ n) ->  CNF_ax \omega Y (succ m) ->
  (x *o y = y *o x)  ->
  x = CNF \omega X (succ n) -> y = CNF \omega Y (succ m) ->
  P (V \0c X) = \0o ->  P (V \0c Y) = \0o ->
  n = \0c ->
  (x = \1c \/ (finite_o x & finite_o y)). (* 44 *)
```

Assume now our numbers infinite and successors. We can apply (16c), and use uniqueness of the CNF. This yields $m + k - 1$ pairs of equations. Let $p = m + k - 2$. This is the number

of non-zero exponents, and the number of equations concerning non-zero exponents. Concerning coefficients, we have $p+1$ equations and $p+2$ unknowns. Thus, there is generically one free parameters, a coefficient. Consider for example $m=5$ and $k=3$. Solving the equations is trivial; there are 3 redundant equations for coefficients and 2 for the exponents. In fact, any β is a solution, and the equations are equivalent to $\alpha=\beta^2$. The case $m=6$ and $k=3$ is a bit more difficult. If $c=\lambda_5$ and $d=\lambda_4$, then one equation is $d+d+d=d+d+c+c$. After simplification, this gives $d=c+c$. In this case, we have 3 free parameters, two coefficients a and b, and exponent c. The ordinals α and β satisfy the following properties: the leading coefficient is a, the trailing coefficient is b, other coefficients are ab. The i-th exponent (in increasing order, starting with zero) is $c\cdot i$. Let $\gamma=\omega^c a+b$. It is easy to check that $\beta=\gamma^2$, and one could verify that $\alpha=\gamma^5$.

In the general case we shall use the product form (18c). The important property we use here is that $\lambda_k=0$. Thus, we may write, with new notations:

$$\alpha=\mu_{k+1}\cdot(\omega^{\lambda_k}+1)\cdot\mu_k\cdots(\omega^{\lambda_2}+1)\cdot\mu_2\cdot(\omega^{\lambda_1}+1)\cdot\mu_1.$$

$$\beta=\nu_{m+1}\cdot(\omega^{\kappa_m}+1)\cdot\nu_m\cdots(\omega^{\kappa_2}+1)\cdot\nu_2\cdot(\omega^{\kappa_1}+1)\cdot\nu_1.$$

The product of two such product is such a product. The exponents are the exponents of α or β; and the coefficients are the coefficients of α or β, with an exception: there is $\mu_1\cdot\nu_{m+1}$ or $\nu_1\cdot\mu_{k+1}$. Since k and m are non-zero, it follows that $\mu_1=\nu_1$. We also have $\lambda_1=\kappa_1$, $\mu_2=\nu_2$, etc. In fact, if $k=m$, we get $\alpha=\beta$. More generally, if $k>m$, there exists γ that has the same form, such that $\alpha=\gamma\cdot\beta$. Moreover, $\beta\cdot\gamma=\gamma\cdot\beta$. The proof is straightforward, the relation (P_1) follows by induction on $k+m$.

```
Definition CNF_mp X Y pY n m :=
   L (interval_co_0a (succ n +c m))
     (fun z => (Yo (z = n) (J (P (V n X)) (Q (V n X) *o (Q pY)))
        (Yo (z <c n) (V z X) (V (z -c (succ n)) Y)))).
Lemma CNFp_pg X pX n Y pY m (Z := CNF_mp X Y pY n m): (* 100 *)
  CNFp_ax X pX -> CNFp_ax Y pY ->
  cardinal (domain X)  = succ n -> cardinal (domain Y)  = m -> inc n Bnat ->
  ( CNFp_ax Z pX &
  (Q pX *o (CNFpv1 X)) *o ((Q pY) *o (CNFpv1 Y)) = (Q pX)*o CNFpv1 Z).

Definition CNFp_ax4 X p n x :=
  inc n Bnat & CNFp_ax X p & cardinal (domain X) = succ n &
  P p = \0o & x = Q p *o CNFpv1 X.
Lemma CNFp_ph X pX n Y pY m x y
  (Z1 := CNF_mp X Y pY n (succ m))
  (Z2 := CNF_mp Y X pX m (succ n)):
  CNFp_ax4 X pX n x  -> CNFp_ax4 Y pY m y ->
  oprod_comm x y ->
   (Z1 = Z2 &
    ( n = m -> x = y) &
    ( m <c n -> exists Z, exists pZ, exists p, exists z,
       CNFp_ax4 Z pZ p z & x = z *o y &  z *o y = y *o z & p <c n)). (* 153 *)

Lemma CNFp_ph X pX n Y pY m x y
  (Z1 := CNF_mp X Y pY n (succ m))
  (Z2 := CNF_mp Y X pX m (succ n)):
  CNFp_ax4 X pX n x  -> CNFp_ax4 Y pY m y ->
  oprod_comm x y ->
   (Z1 = Z2 &
```

```
    ( n = m -> x = y) &
    ( m <c n -> exists Z, exists pZ, exists p, exists z,
       CNFp_ax4 Z pZ p z & x = z *o y &  z *o y = y *o z & p <c n)).

Definition oprod2_comm_P1 x y :=
  exists c, exists n, exists m,
    is_ordinal c & inc n Bnat & inc m Bnat & x = c ^o n & y = c ^o m.
Lemma oprod2_comm6 X pX n x Y pY m y:
  CNFp_ax4 X pX n x  -> CNFp_ax4 Y pY m y ->
  oprod_comm x y -> oprod2_comm_P1 x y.
```

Thus (P) is equivalent to (P_1) or (P_2) or (P_3) or (P_4).

```
Theorem oprod2_comm x y: is_ordinal x -> is_ordinal y ->
   ((oprod_comm x y) <->
   (x = \0o \/ y = \0o \/ (oprod2_comm_P4 x y \/  oprod2_comm_P4 y x) \/
   (finite_o x & finite_o y) \/ oprod2_comm_P1 x y)).  (* 77 *)
```

8.12.6 Natural sum and products of ordinals

Consider two ordinals α and β and the sum $\gamma = \alpha + \beta$. Consider two disjoint sets A and B, ordered by $\leq_A$ and $\leq_B$ such that the order-type of $\leq_A$ is α, and the order-type of $\leq_B$ is β. Let $C = A \cup B$, ordered by $\leq_C$ which is $x \leq_A y$ or $x \leq_B y$ or ($x \in A$ and $y \in B$)". Then the order-type of $\leq_C$ is γ. Consider $x \leq y$ defined by "$x \leq_A y$ or $x \leq_B y$". This is an ordering of C, which is in general only partial. Consider $\alpha \oplus \beta$ to be the supremum of all λ, where λ is a well-ordering of C that extends this ordering. We have obviously $\gamma \leq \lambda$. One can show that there is a well-ordering on C whose ordinal is γ. The quantity $\alpha \sharp \beta$ is called the *natural sum* of the two ordinals. On the product $D = A \times B$ one can consider the relation $x \leq y$ defined by "$\mathrm{pr}_1 x \leq_A \mathrm{pr}_1 y$ and $\mathrm{pr}_2 x \leq_B \mathrm{pr}_2 y$". This is the usual order-product, and is in general not total. Let $\leq_D$ be the lexicographic product. This is a a well-ordering that extends $\leq$ (and whose order-type δ is $\alpha \cdot \beta$). The supremum of all μ, where μ is a well-ordering of C that extends this ordering is called the *natural product*. It is obviously $\geq \delta$. Let's denote it by $\alpha \otimes \beta$.

One can show that these operations are commutative and associative, and the distributivity law also holds. Moreover $\alpha \oplus 0 = \alpha \otimes 1 = \alpha$. These functions are monotone in the sense that $\delta \oplus \alpha > \delta \oplus \beta$ and $\delta \otimes \alpha > \delta \otimes \beta$ are equivalent to $\alpha > \beta$. Finally, the product of two powers of ω is a power of ω. The natural sum and product are the "smallest" (in some sense) operations that satisfy these properties (see [6]). One has $\omega^a \otimes \omega^b = \omega^c$ where $c = a \oplus b$. Thus,

$$\left(\sum_i \omega^{a_i} \cdot c_i\right) \otimes \left(\sum_j \omega^{b_j} \cdot d_j\right) = \sum_{ij} \omega^{a_i \oplus b_j} \cdot c_i d_j$$

provided that c_i and d_i are integers. One could use this formula as the definition of the natural product (one has to be careful as $\sum$ is non-commutative: the RHS has to be understood as the sum of all $\omega^k \cdot e_k$ where k has the form $a_i \oplus b_j$, for some i, j, listed in decreasing order, and e_k is the sum of all corresponding $c_i d_j$ (this is a natural integer).

In what follows, we consider only the natural sum, and we have

$$\left(\sum_i \omega^{a_i} \cdot c_i\right) \oplus \left(\sum_j \omega^{a_j} \cdot d_j\right) = \sum_i \omega^{a_i} \cdot (c_i + d_i).$$

Given two ordinals α and β, one can consider their CNF $\alpha = \sum_i \omega^{a_i} \cdot c_i$, $\beta = \sum_j \omega^{b_j} \cdot d_j$, and merge the two lists of exponents, adding zero coefficients where required.

We are interested in the following two properties: if $\alpha < \omega^n$ and $\beta < \omega^n$ then $\alpha \oplus \beta < \omega^n$. Given γ, the equation $\alpha \oplus \beta = \gamma$ has a finite number of solutions (the product of the coefficients of the CNF of γ).

We define $e(X)$ to be the sequence of exponents of X. If $x < \omega^e$, all exponents of the CNF of x are $< e$.

Assume X of length n, e_i the i-coefficient. Then $i \mapsto e_i$ is a strictly increasing bijection from $[0, n[$ onto $e(X)$. In particular $e(X)$ has n elements. If a an ordinal, one of the following holds: it is an exponent, it is greater than all exponents, it is smaller than all exponents, or it is between two exponents.

```
Definition CNF_exponents X:= (fun_image (domain X) (fun z => P (V z X))).
Definition CNF_coefficients X := fun_image (domain X) (fun z => Q (V z X)).

Lemma CNF_exponents_rw X x:
  inc x (CNF_exponents X) <-> (exists j, inc j (domain X) & P (V j X) = x).
Lemma CNFq_exists3 a e (X := the_CNF a):
  is_ordinal e ->  a <o (\omega ^o e) ->
  (forall z, inc z (CNF_exponents X) -> z <o e).
Lemma CNF_exponents_card b X:
  CNFq_ax b X-> cardinal (domain X) = cardinal (CNF_exponents X).
Lemma CNF_exponents_of b X (e := CNF_exponents X):
  CNFq_ax b X -> (finite_set e & ordinal_set e).
Lemma CNF_exponents_compare_expo b X a (e:= CNF_exponents X) :
  CNFq_ax b X -> is_ordinal a ->
  (inc a e \/ (forall t, inc t e -> t <o a) \/ (forall t, inc t e -> a <o t)
   \/ (exists i, inc i (domain X) & inc (succ i) (domain X) &
           P (V i X) <o a & a <o (P (V (succ i) X)))).

Lemma CNFq_axm_expo X1 X2
  (k := cardinal (domain X1)) (m := (cardinal (domain X2))) :
  CNF_graph X1 -> CNF_graph X2 ->
  CNF_exponents  (CNFq_m X1 k X2 m)
      = union2 (CNF_exponents X1)(CNF_exponents X2). (* 21 *)
```

Given a CNFq X and an ordinal a, there exists X_a such that $s(X_a) = s(X)$ and $e(X_a) = e(X) \cup \{a\}$. Moreover, if e is an exponent of X, it is an exponent of X_a with the same coefficient. We restate this as: forall i, there is j such that $X(i) = X_a(j)$.

This is trivial if $a \in e(X)$. If a is greater than all (or less than all) exponents, it suffices to extend X. Otherwise, we split X into X_r and X_s, extend one piece, and merge. By induction, there is X_A such that $s(X_A) = s(X)$ and $e(X_A) = e(X) \cup A$, for every finite set of ordinals A.

```
Definition CNF_ext_coeffs X Y:=
  (forall i, inc i (domain X) -> exists j, inc j (domain Y) & V i X = V j Y).
Lemma CNF_exponents_add_expo b X a (e:= CNF_exponents X) :
  CNFq_ax b X -> is_ordinal a ->
  exists Y,  CNFq_ax b Y & CNF_exponents Y = tack_on e a &
    CNFbv b X = CNFbv b Y & CNF_ext_coeffs X Y.  (* 177 *)
Lemma CNF_exponents_add_expos b  X A (e:= CNF_exponents X) :
  CNFq_ax b X -> ordinal_set A -> finite_set A ->
  exists Y, CNFq_ax b Y & CNF_exponents Y = union2 e A &
    CNFbv b X = CNFbv b Y & CNF_ext_coeffs X Y.
```

Assume $s(X) = s(Y)$ and $e(X) = e(Y)$. Then X = Y. The proof is by induction on the length n of $e(X)$. If this length is zero, then X and Y are the empty functions. Otherwise, write X as a

monomial (e, c) and a rest X′. Note that e is the greatest element of $e(X)$ and $e(X') = e(X) - \{e\}$. Writing Y as a monomial (e', c') and a rest Y′ gives $e = e'$ and $e(X') = e(Y')$. We have $s(X) = \omega^e \cdot c + s(X')$; similarly for Y. If c' is the exponent associated to Y, $c'' = \min(c, c')$, we can write $s(X) = \omega^e \cdot c'' + \omega^e \cdot c_x + s(X')$ and $s(Y) = \omega^e \cdot c'' + \omega^e \cdot c_y + s(Y')$. In the expresion $s(X) = s(Y)$ we can simplify by $= \omega^e \cdot c''$. Since $s(X') < \omega^e$ and $s(Y') < \omega^e$ and one of c_x c_y is zero, it follows that both are zero. Thus $c = c'$ and $s(X') = s(Y')$. By induction X′ = Y′ and X = Y.

It also follows by induction that if $s(X) = 0$ then all coefficients have to be zero.

If L is any set with n elements, we may consider the set $\mathfrak{S}_L$ of functional graphs $[0, n[\to L \times b$. For any CNFq X, we have $X \in \mathfrak{S}_{e(X)}$. If L is a finite set of ordinals there is a unique X′ in $\mathfrak{S}_{e(X) \cup L}$ such that $s(X) = s(X')$. We call it the expansion of X. Given two CNF X and Y, taking for L the union $e(X) \cup e(Y)$, we get X′ and Y′, such that $s(X) = s(X')$, $s(Y) = s(Y')$, and $e(X') = e(Y') = L$. We call this the common extension of X and Y.

```
Definition set_of_CNFq b el:=
  set_of_gfunctions (interval_co_0a (cardinal el)) (product el b).
Definition CNFq_extension b x e :=
  let ne := union2 (CNF_exponents x) e in
  select (fun y =>   CNFq_ax b y &  CNF_exponents y = ne
       &  CNFbv b x = CNFbv b y & CNF_ext_coeffs x y)
     (set_of_CNFq b ne).

Lemma set_of_CNFq_inc b X : CNF_ax X ->
   inc X (set_of_CNFq b (CNF_exponents X)).
Lemma CNFq_exponents_r b X n (e:= P (V n X)) (el := (CNF_exponents X)):
   CNFq_ax b X ->  inc n Bnat -> cardinal (domain X) = succ n ->
   (inc e el &
     (forall a, inc a el -> a <=o e) &
     (CNF_exponents (CNF_r X n) = compl_singl el e)). (* 30 *)
Lemma CNFq_extensionality1 b  X Y:
   CNFq_ax b X -> CNFq_ax b Y ->  CNF_exponents X = CNF_exponents Y ->
   (domain X = domain Y &
    (forall i, inc i (domain X) -> P (V i X) = P (V i Y))). (* 42 *)
Lemma CNFq_zero b X: CNFq_ax b X ->  CNFbv b X = \0o ->
   forall i, inc i (domain X) -> Q (V i X) = \0o. (* 33 *)
Lemma CNFq_extensionality b X Y:   (* 89 *)
   CNFq_ax b X -> CNFq_ax b Y ->  CNFbv b X = CNFbv b Y ->
   CNF_exponents X = CNF_exponents Y ->
  X = Y.
Lemma set_of_CNFq_inc X : CNFq_ax \omega X ->
   inc X (set_of_CNFq (CNF_exponents X)).
Lemma CNFq_extension_pr b X A (B := union2 (CNF_exponents X) A)
  (Y := CNFq_extension b X A) :
  CNFq_ax b X -> ordinal_set A -> finite_set A ->
  (CNFq_ax b Y &  CNF_exponents Y = B &
      CNFbv b X = CNFbv b Y & CNF_ext_coeffs X Y).

Lemma CNFq_extension2_pr b X Y
   (X' := CNFq_extension b X (CNF_exponents Y))
   (Y' := CNFq_extension b Y (CNF_exponents X)):
  CNFq_ax b X -> CNFq_ax b Y ->
  (CNFq_ax b X' & CNFq_ax b Y' &
    CNFbv b X = CNFbv b X' &  CNFbv b Y = CNFbv b Y'
    & (domain X' = domain Y')
    & CNF_exponents X' = CNF_exponents Y'
```

```
    & CNF_exponents X' = union2 (CNF_exponents X)  (CNF_exponents Y)
    & CNF_ext_coeffs X X' & CNF_ext_coeffs Y Y').
```

Consider two CNFq X and Y. Let's assume that X and Y have the same exponents e_i, with coefficients x_i and y_i. We consider the CNF Z whose exponents is e_i and whose coefficients are $x_i + y_i$. Let z be the supremum of the coefficients of Z. Then all x_i are $\leq z$. This will be called the natural sum of two CNFq.

We shall assume the base b is indecomposable, so that the result is a CNFq in base b. We shall moreover assume $b = \omega$. In this case, x_i and y_i are integers, thus $x_i + y_i = y_i + x_i$. In this case the natural sum is commutative and has the empty CNF as unit.

```
Definition CNFq_sum X Y :=
  L (domain X) (fun i => J (P (V i X)) (Q (V i X) +o Q (V i Y))).

Lemma CNF_sum_ax X Y:
   CNFq_ax \omega  X -> CNFq_ax \omega Y -> domain X = domain Y ->
   CNFq_ax \omega (CNFq_sum X Y).
Lemma CNF_sumC X Y:
   CNFq_ax \omega X -> CNFq_ax \omega Y -> CNF_exponents X = CNF_exponents Y ->
   CNFq_sum X Y = CNFq_sum Y X.
Lemma CNF_sum_zero X Y:
   CNFq_ax \omega  X -> (forall i, inc i (domain X) -> Q (V i Y) = \0o) ->
   CNFq_sum X Y = X.
Lemma CNF_sum_bounded1 X Y:
  CNFq_ax \omega X -> CNFq_ax \omega Y -> domain X = domain Y ->
  (forall i, inc i (domain X) ->
  Q (V i X) <=o \osup (CNF_coefficients (CNFq_sum X Y))).
```

Given two ordinals x and y, we consider X and Y their CNF, X′ and Y′ their common extension; Z the natural sum of X′ and Y′, and $z = s(\mathrm{Z})$. Let's note that $e(\mathrm{Z}) = e(\mathrm{X}) \cup e(\mathrm{Y})$, so that all coefficients of Z are non-zero, and Z is the CNF of z. It is obvious that any exponent of X is an exponent of Z, and any coefficient of X is at most c, the supremum of the coefficients of Z. Since $e(\mathrm{Z})$ and c are finite, there is a finite set F_Z (depending only on Z), such that $\mathrm{X} \in \mathrm{F}_\mathrm{Z}$.

This quantity z will be called the *natural sum* of x and y and denoted by $x\sharp y$. We have $x\sharp y = y\sharp x$ and $x\sharp 0 = x$. One could easily show that the operation is associative.

If $x < \omega^e$ and $y < \omega^e$, then all exponents of X and Y are $< e$, so that the same holds for X′, Y′ and Z; thus $x\sharp y < \omega^e$. We have $x \leq x\sharp y$ (proof by induction).

```
Definition ord_natural_sum x y :=
  let X := the_CNF x in let Y := the_CNF y in
  CNFv (CNFq_sum (CNFq_extension X (CNF_exponents Y))
    (CNFq_extension Y (CNF_exponents X))).
Notation "x +#o y" := (ord_natural_sum x y) (at level 50).

Lemma natural_sum_p1 x y :
  is_ordinal x -> is_ordinal y ->
  let X := the_CNF x in let Y := the_CNF y in
    let s := (CNFq_sum (CNFq_extension \omega X (CNF_exponents Y))
                       (CNFq_extension \omega Y (CNF_exponents X))) in
  (CNFb_ax \omega s
   & (forall i, inc i (domain X) ->
     Q (V i X) <=o \osup (CNF_coefficients s))). (* 27 *)
Lemma OS_natural_sum x y :
```

```
  is_ordinal x -> is_ordinal y -> is_ordinal (x  +#o y).
Lemma natural_sum_p2 x y (s := (the_CNF (x +#o y))):
   is_ordinal x -> is_ordinal y ->
    (sub (CNF_exponents (the_CNF x)) (CNF_exponents s)
     & (forall i, inc i (domain  (the_CNF x)) ->
        Q (V i  (the_CNF x)) <=o \osup (CNF_coefficients s))).

Lemma CNF_sup_coef_pr X (s := \osup (CNF_coefficients X)) :
   CNFq_ax \omega X ->
   (s = \0c \/ exists i, inc i (domain X) & s = Q (V i X)).
Lemma natural_sum_p3 x y (s := (the_CNF (x +#o y)))
  (E:= product (CNF_exponents s) (succ_o (\osup (CNF_coefficients s))))
  (I1 := interval_cc_0a (cardinal (domain s)))
  (F:= unionb (L I1 (fun z => set_of_gfunctions (interval_co_0a z) E))):
  is_ordinal x -> is_ordinal y ->
  (inc (the_CNF x) F & finite_set F).
Lemma ord_natural_sumC x y :
 is_ordinal x -> is_ordinal y ->   (x +#o y) = (y +#o x).
Lemma ord_natural_sum0 x  :
  s_ordinal x -> (x +#o \0o) = x.
Lemma ord_natural_small x y e :
  is_ordinal e -> x <o (\omega ^o e) -> y <o (\omega ^o e) ->
  (x +#o y) <o  (\omega ^o e).

Lemma CNF_M0le b X Y :
  CNFq_ax b X -> CNFq_ax b Y -> domain X = domain Y ->
  (forall i, inc i (domain X) ->
     (P (V i X) = P (V i Y) & Q (V i X) <=o Q (V i Y)))
  -> CNFbv b X <=o CNFbv b Y.  (* 27 *)
Lemma ord_natural_M0le x y :
  is_ordinal x ->  is_ordinal y -> x <=o  (x +#o y).
```

Let n be any ordinal, and $\mathrm{E} = \omega^n$. This can be considered as the set of all ordinals x such that $x < \omega^n$. We just proved that E is stable by natural sum. Each element z of E can be written as $z = x \sharp y$ (consider $y = 0$). The number of decompositions is finite (we have shown that the CNF X of x have to be in some finite set F_z. If G_z is the set of $s(\mathrm{X})$ for $\mathrm{X} \in \mathrm{G}_z$, this is a finite set and $x \in \mathrm{G}_z$. By symmetry of addition we have also $y \in \mathrm{G}_z$. Note that, if the coefficients of the CNF of z are e_i, then there are $\prod(e_i + 1)$ possibilities for x; our proof says that there are at most $(\sup e_i + 1)^n$, where n is the number of coefficients in z). One could also show that, for fixed x, there is a unique y such that $z = x \sharp y$.

```
Lemma ord_natural_finite_cover x e (E:= \omega ^o e): (* 34 *)
  is_ordinal e -> inc e E ->
  let n := cardinal (Zo (coarse E) (fun z => ord_natural_sum (P z) (Q z)= x)) in
  inc n Bnat & n <> \0c.
```

8.13 Numbers equal to their degree

Recall that the degree a of x is the greatest exponent in the Cantor Normal Form. Write $x = \omega^a \cdot b + c$. We have $\omega^a \geq a$, so that $x \geq a$. In case of equality, we have $b = 1$ and $c = 0$. This implies

(19a) $$x = \omega^x.$$

Such numbers are called ϵ-ordinals. They are infinite. In fact, they are > ω.

```
Definition ordinal_epsilon x := is_ordinal x & \omega ^o x = x.
Lemma ord_eps_p1 x: is_ordinal x -> x <=o (\omega ^o x).
Lemma ord_eps_p2 a b c (x := ((\omega ^o a) *o b) +o c0 :
  is_ordinal a -> is_ordinal b -> is_ordinal c -> b <> \0c ->
  (a <=o x & (x = a -> (c = \0c & b = \1c & ordinal_epsilon x))).
Lemma ord_eps_p3 x: ordinal_epsilon x -> \omega <=o x.
Lemma ord_eps_p4 x: ordinal_epsilon x -> \omega <o x.
```

Define $\mathrm{E}(x)$ as the limit of the sequence x_k such that $x_0 = x$ and $x_{k+1} = \omega^{x_k}$. Since $t \to \omega^t$ is a normal function of t, it follows that $\mathrm{E}(t)$ is an ϵ-ordinal. It is the least ϵ-ordinal $\geq x$.

```
Definition ord_eps_fct a := induction_defined (fun z => \omega ^o z) a.
Definition epsilon_fct a := \osup (target (ord_eps_fct a)).


Lemma ord_eps_p5  a (f :=  ord_eps_fct a) (e := epsilon_fct a) : (* 55 *)
 is_ordinal a -> (
      source  f = Bnat & surjection f & W \0c f = a &
      (forall n, inc n Bnat -> W (succ n) f = \omega ^o (W n f)) &
      (forall n, inc n Bnat -> is_ordinal (W n f)) &
      (forall n, inc n Bnat -> W n f <=o (W (succ n) f)) &
      (ordinal_set (target f)) &
      (is_ordinal e) &
      (a <=o e)  &
      (ordinal_epsilon e) &
      (forall b, ordinal_epsilon b -> a <=o b -> e <=o b)).

Lemma ord_eps_p5bis a (e := epsilon_fct a) :
 is_ordinal a ->
      (a <=o e  &
       ordinal_epsilon e &
      (forall b, ordinal_epsilon b -> a <=o b -> e <=o b)).
```

We deduce that $\mathrm{E}(1)$ is the the least ϵ-ordinal, and $\mathrm{E}(x+1)$ is the least ϵ-ordinal $> x$. The least upper bound of family of ϵ-ordinals is an ϵ-ordinal, since exponentiation is normal.

```
Lemma ord_eps_p6 (e:= epsilon_fct \1o) :
  ordinal_epsilon e &
  forall x, ordinal_epsilon x -> e <=o x.
Lemma ord_eps_p7 x (e := epsilon_fct (succ_o x)):
   is_ordinal x ->
  (ordinal_epsilon e & x <o e &
  (forall y, ordinal_epsilon y -> x <o y ->  e <=o y)).
Lemma ord_eps_p8 X:
   (forall x, inc x X -> ordinal_epsilon x) -> nonempty X ->
   ordinal_epsilon (\osup X).
```

We define now a function ϵ_n by induction.

(19b)
$$\epsilon_0 = \mathrm{E}(1), \qquad \epsilon_x = \sup_{y<x} \mathrm{E}(\epsilon_y + 1)$$

The proofs of the following properties are similar to those of section 8.10.

```
Definition ord_eps_induction_aux b :=
  transfinite_defined (ordinal_o b)
  (fun f => Yo (source f = \0o) (epsilon_fct \1o)
    (\osup (fun_image (source f) (fun z => epsilon_fct (succ_o (W z f))))).
Definition epsilon_fam y:=  W y (ord_eps_induction_aux (succ_o y)).


Lemma epsilon_fam_pr0 (f := epsilon_fam)  (* 78 *)
  (f \0o = epsilon_fct \1o) &
  (forall y, \0o <o y ->
    f y = \osup (fun_image y (fun z => epsilon_fct (succ_o (f z))))).
Lemma epsilon_fam_pr1 x: is_ordinal x ->
  is_ordinal (epsilon_fam x).
Lemma epsilon_fam_pr1bis a b: is_ordinal a -> inc b a ->
  is_ordinal (epsilon_fct (succ_o (epsilon_fam b))).
Lemma epsilon_fam_pr2bis y y': y <o y' ->
  (epsilon_fct (succ_o (epsilon_fam y))) <=o  (epsilon_fam y').
Lemma epsilon_fam_pr2 y y': y <o y' ->
  (epsilon_fam y) <o (epsilon_fam y').
Lemma epsilon_fam_pr2ter y y': y <=o y' ->
  (epsilon_fam y) <=o (epsilon_fam y').
Lemma epsilon_fam_pr3 y: is_ordinal y ->
  y <=o (epsilon_fam y).
Lemma epsilon_fam_pr4: normal_ofs epsilon_fam.  (* 21 *)
Lemma epsilon_fam_pr5 b: epsilon_fct \1o <=o b -> (* 45 *)
  exists_unique (fun y=> is_ordinal y &  (epsilon_fam y) <=o b &
   b <o (epsilon_fam (succ_o y))).
```

If x is countable, then all x_i that appear in the definition of $\mathrm{E}(x)$ are countable, so that $\mathrm{E}(x)$ is countable. It follows that ϵ_x is countable.

```
Lemma countable_epsilon x:
   countable_ordinal x -> countable_ordinal (epsilon_fct x).
Lemma countable_epsilon2 x: is_ordinal x -> (* 23 *)
  ( countable_ordinal x <-> countable_ordinal (epsilon_fam x)).
```

We have $\epsilon_{x+1} = \mathrm{E}(\epsilon_x + 1)$, and for any ordinal y such that $y \geq \epsilon_0$, there is a unique x such that $\epsilon_x \leq y < \epsilon_{x+1}$. Note that $\mathrm{E}(\epsilon_x + 1)$ is the least ϵ-ordinal z such that $\epsilon_x < z$. If y is an ϵ-ordinal, it follows that $\epsilon_x = y$. On the other hand by induction ϵ_x is an ϵ-ordinal (for non-zero x, it is a limit of ϵ-ordinals).

We have thus shown: an ordinal y is an ϵ-ordinal if only if it is of the form ϵ_x. Cantor [5, §20F] deduces: the set of countable ϵ-ordinals is order-isomorphic to the set Ω of infinite countable ordinals, and has cardinal $\aleph_1$. The non-trivial point here is $\omega + \Omega = \Omega$ (there is an isomorphism between countable ordinals and infinite countable ordinals). This is left as an exercise for the reader.

```
Lemma epsilon_fam_pr6 y: is_ordinal y ->
  epsilon_fam (succ_o y) =  epsilon_fct (succ_o (epsilon_fam y)).
Lemma epsilon_fam_pr7  y: is_ordinal y ->
  ordinal_epsilon (epsilon_fam y).
Lemma epsilon_fam_pr8 y: ordinal_epsilon y ->
  exists x, is_ordinal x & y = epsilon_fam x.
```

We study now critical ordinal for the sum, product and exponentiation. As noticed above, y is critical for the sum if and only if y is infinite and indecomposable, thus, is of the form

ω^n, for some $n > 0$. Exercise 14b (page 332) studies the case of a product.

Let's say that y is CP if $1 \leq x < y$ implies $x \cdot y = y$. Let H be the property that, for all z such that $2 \leq z \leq y$ there exists an indecomposable ordinal t such that $y = z^t$. Let H$'$ be the property that $y = \omega^{\omega^n}$ for some n and H$''$ the property that $y = \omega^m$ for some indecomposable m. These properties are equivalent. Since m is indecomposable if, and only if, it has the form ω^n, properties H$'$ and H$''$ are clearly equivalent. Obviously, H implies H$''$. Conversely, assume H$''$ and fix z. If $z = y$ it suffices to take $t = 1$. If z is finite, then $z^{\omega \cdot m} = \omega^m = y$. Assume z infinite of degree k so that $k < m$. There exists q indecomposable such that $m = k \cdot q$ hence $z^q = \omega^{k \cdot q} = y$. If H$''$ holds then CP holds also, for if k is the degree of x, relation (16b) says $x \cdot y = \omega^{k+m}$, and $k + m = m$.

Assume that CP holds, consider x with $x < y$. Write $y = x^a \cdot b + c$. Assume first $x^a \cdot b < y$. Then $x^a \cdot b(x^a \cdot b + c) = x^a \cdot b + c$. Write $x^a \cdot b = 1 + u$, so that $x^a \cdot b \cdot (u + c) = c$. The LHS is at least x^a, contradicting $c < x^a$. Thus $y = x^a \cdot b$. Assume $x^a < y$. Then $x^a \cdot y \cdot y = y$. After simplification by x^a, we get $x \cdot^a b = b$. This is $y = b$, contradicting $b < x$. Thus $y = x^b$. Assume $d < b$, so that $x^d < y$. We have $x^d \cdot y = y$, which is $d + b = b$, and this says that b is indecomposable.

```
Lemma critical_product_pr: (* 80 *)
  let CP := critical_ordinal \1o ord_prod2 in
  let p1 := fun y => (exists n, is_ordinal n & y = \omega ^o (\omega ^o n)) in
  let p2 := fun y => infinite_o y & is_ordinal y &
       (forall z, \1o <o z -> z <=o y ->
          exists t, is_ordinal t & ord_indecomposable t & y = z ^o t) in
  forall y, p1 y <-> p2 y & CP y <-> p1 y.
```

We first show the following properties (see Cantor, [5, Theorem 20G]): if x is an ϵ-number and $a < x$, then $a + x = x$, $a \cdot x = x$ and $a^x = x$ (we assume $a > 0$ for the product, $a \geq 2$ for the exponentiation). This says that x is critical for the sum, product and exponentiation. The first two properties are obvious. We have $a^x = a^{\omega \cdot x} = (a^\omega)^x$. The case a finite is trivial as $a^\omega = \omega$. Otherwise, let n be the degree of a, so that $a^\omega = \omega^{n\omega}$. Then $a^x = \omega^{n \cdot x}$. But $n < x$, so that $n \cdot x = x$.

Assume $2^x = x$. It is clear that x cannot be a successor, since 2^{n+1} is at least $n + n$, and this cannot be $n + 1$. In particular, $x = \omega \cdot y$ for some y, and $\omega^y = \omega \cdot y$. Note that y cannot be zero, so let's write $y = 1 + t$. We get $\omega^t = 1 + t$. Since ω^t is indecomposable, one term is ω^t. Assume it is 1, so that $t = 0$, $y = 1$ and $x = \omega$. Otherwise, $t = \omega^t$ so t is an ϵ-ordinal and $1 + t = t$. It trivially follows $t = x$. In summary: if $2^x = x$, then x is critical for exponentiation and is ω or an ϵ-ordinal.

```
Lemma ord_epsilon_p9 x a: ordinal_epsilon x -> a <o x->
  (a +o x = x &
  (\1o <=o a -> a *o x = x) &
  (\2o <=o a -> a ^o x = x)). (* 36 *)
Lemma ord_epsilon_p10 x:
  ordinal_epsilon x -> critical_ordinal \2o ord_pow x.

Lemma ord_epsilon_p11 x: is_ordinal x -> (\2o ^o x = x) ->
  x = \omega  \/ ordinal_epsilon x. (* 48 *)
Lemma ord_epsilon_p12 x: is_ordinal x -> (\2o ^o x = x) ->
  critical_ordinal \2o ord_pow x.
```

8.14 Initial ordinals

We shall define in this section a sequence of ordinals, which are called "initial" ordinals and denoted ω_α, and the cardinal of these sets are called "alephs" and denoted by $\aleph_\alpha$. With our definition of a cardinal, it happens that an initial ordinal is a cardinal, and thus $\omega_\alpha = \aleph_\alpha$. Conversely, any infinite cardinal is an aleph; so that an initial ordinal is just an infinite cardinal, and $\omega_1 = \aleph_1$ is the least non-countable cardinal. It satisfies $\aleph_1 \leq 2^{\aleph_0}$. The continuum hypothesis (CH) asserts that these numbers are equal, the generalized continuum hypothesis (GCH) asserts more generally that $\aleph_{n+1} = 2^{\aleph_n}$. These statements are undecidable.

Let's consider first the Bourbaki point of view (Exercise 6.10). He denotes by ω_0 the ordinal of **N**. For each integer n, we may consider the interval $[0, n[$, ordered by $\leq_{\mathbf{N}}$. This is a well-ordered set, and if $\bar{n}$ denotes its ordinal, we have $\bar{n} <_{\mathrm{Ord}} \omega$. In fact, any ordinal x such that $x <_{\mathrm{Ord}} \omega$ is of the form $\bar{n}$. Finally $n \leq_{\mathbf{N}} m$ is equivalent to $\bar{n} \leq_{\mathrm{Ord}} \bar{m}$. This means that $n \to \bar{n}$ is an order-isomorphism between **N** and the set of ordinals $< \omega$. [In our framework, $\mathbf{N} = \omega$ and $\bar{n} = n$].

Given a cardinal $\mathfrak{a}$, we can consider the set $\mathrm{W}(\mathfrak{a})$ of all ordinals ξ such that $\mathrm{Card}(\xi) < \mathfrak{a}$, as well as the set $\mathrm{W}'(\mathfrak{a})$ of all ordinals ξ such that $\mathrm{Card}(\xi) \leq \mathfrak{a}$. These objects are sets: by the axiom of choice, there is a well-ordering on $\mathfrak{a}$; let α be the ordinal of this ordering. Let ξ be any ordinal such that $\mathrm{Card}(\xi) < \mathfrak{a}$. We have $\xi <_{\mathrm{Ord}} \alpha$, for otherwise we would have $\alpha \leq_{\mathrm{Ord}} \xi$. This gives an injection $\alpha \to \xi$, so that $\mathrm{Card}(\alpha) \leq_{\mathrm{Card}} \mathrm{Card}(\xi)$. Since $\mathrm{Card}(\alpha) = \mathfrak{a}$, we get a contradiction. Note that $\mathrm{W}'(\alpha)$ is a subset of $\mathrm{W}(2^\alpha)$, thus is a set.

Using von Neumann cardinals simplifies the reasoning. Notice first that, if x is a cardinal, that $\mathrm{Card}(y) <_{\mathrm{Card}} x$ is equivalent to $y <_{\mathrm{ord}} x$, since $\mathrm{Card}(x)$ is the least ordinal equipotent to x, and $a \leq_{\mathrm{ord}} b$ implies $\mathrm{Card}(a) \leq_{\mathrm{Card}} \mathrm{Card}(b)$. Thus $\mathrm{W}(x)$ is just x. There is no simple formula for W'.

```
Definition set_ordinal_card_le y :=
   Zo (\2c ^c y) (fun z => (cardinal z) <= c y).
Lemma set_ordinal_card_lt_rw y: is_cardinal y ->
  forall x, inc x y <-< (is_ordinal x & (cardinal x) <c y).
Lemma set_ordinal_card_le_rw y: is_cardinal y ->
  forall x, inc x (set_ordinal_card_le y)
    <-> (is_ordinal x & (cardinal x) <=c y).
```

We define by transfinite induction on the ordinal α a function f_α such that $f_\alpha(x)$ is the least upper bound of $\mathrm{T}_\alpha(x) = \bigcup_{y<x} \mathrm{W}'(f_\alpha(y))$. We have $f_\alpha(x) = f_\beta(x)$ if $x \leq \alpha$ and $x \leq \beta$. In particular is independent of α.

```
Definition aleph_aux1 f x :=
  union (fun_image x (fun z => (set_ordinal_card_le (cardinal (f z))))).
Definition aleph_aux2 b :=
  transfinite_defined (ordinal_o (succ_o b))
  (fun f => Yo (source f = \0o) \omega
      (\osup (aleph_aux1 (fun z => W z f) (source f)))).

Lemma aleph_aux_pr1 f x y: is_ordinal x ->
 (inc y (aleph_aux1 f x) <->
    (is_ordinal y & exists z, is_ordinal z & z <o x &
       (cardinal y) <=c (cardinal (f z)))).
Lemma aleph_aux_pr2 f g x:
  is_ordinal x -> (forall y, y <o x -> f y = g y) ->
```

```
  aleph_aux1 f x = aleph_aux1 g x.
Lemma aleph_aux_pr3 b (f:=aleph_aux2 b): is_ordinal b -> (* 20 *)
  (is_function f & source f = (succ_o b) &
       (W \0o f = \omega) &
       (forall x, x <=o b ->  x <> \0o ->
         W x f = (\osup (aleph_aux1 (fun z => W z f) x)))).
```

Set $\omega_x = f_x(x)$. This is ω for $x = 0$ and, otherwise, the least upper bound of $\mathrm{T}(x)$, the union of of $\mathrm{W}'(y)$ for $y < x$. This quantity will also be denoted by $\aleph_x$. Note that ω, **N** and $\aleph_0$ are equal, but have different meanings: it is an ordinal, the set of integers, and the least infinite cardinal. We gives names to ω_1 and $\aleph_1$.

Since $\omega_0 \in \mathrm{W}'(\omega_0)$, we have $\omega_0 \in \mathrm{T}(x)$. This show that ω_x is infinite. If $y < x$, then ω_y and $\omega_y + 1$ have the same cardinal. This shows that ω_x is strictly increasing. In particular $x \leq \omega_x$.

```
Definition omega_fct x := W x (aleph_aux2 x).
Notation "\aleph" := omega_fct (only parsing).

Definition omega1 :=  omega_fct \1o.
Definition aleph1 :=  \aleph \1o.
Definition omega2 :=  omega_fct \2o.
Definition aleph2 :=  \aleph \2o.
Definition aleph0 :=  \omega.

Lemma aleph_pr1: omega_fct \0o = \omega.
Lemma aleph_pr2: cardinal (omega_fct \0o) = cardinal Bnat.
Lemma aleph_pr3 x: is_ordinal x -> is_ordinal (omega_fct x).  (* 20 *)
Lemma aleph_pr4 x: is_ordinal x -> x <> \0o ->      (* 16 *)
  omega_fct x = (\osup (aleph_aux1 omega_fct x)).
Lemma aleph_pr5 x: is_ordinal x ->
  \omega <=o (omega_fct x).
Lemma aleph_pr6_lt_lto x y: x <o y ->
  (omega_fct x) <o (omega_fct y).
Lemma aleph_pr6_le_leo x y: x <=o y ->
  (omega_fct x) <=o (omega_fct y).
Lemma aleph_pr6_leo_le x y: is_ordinal x -> is_ordinal y ->
  (omega_fct x) <=o (omega_fct y) -> x <=o y.
Lemma aleph_pr6_lto_lt x y: is_ordinal x -> is_ordinal y ->
  (omega_fct x) <o (omega_fct y) -> x <o y.
Lemma aleph_pr6_eq x y: is_ordinal x -> is_ordinal y ->
  (omega_fct x) = (omega_fct y) -> x = y.
Lemma aleph_pr6h x: is_ordinal x -> x <=o (omega_fct x).
```

Let x be an infinite cardinal, and y the least ordinal such that $x \leq \omega_y$. Assume $x < \omega_y$ so that y is non-zero, there is $t < y$ and z such that $x \leq z$ and $\mathrm{Card}(z) \leq \mathrm{Card}(\omega_t)$. We deduce $x \leq \omega_t$, contradicting minimality of y. Thus, any infinite cardinal is an initial ordinal. Conversely, the cardinal of ω_x is infinite, since $\omega_x \geq \omega$, thus has the form ω_z; assume $z < x$; since ω_x^+ has same cardinal as x, we deduce $\omega_x^+ \in \mathrm{T}_f(x)$ thus $\omega_x^+ \leq \omega_x$, absurd. It follows $z = x$; this says that ω_x is equal to its cardinal, thus is a cardinal.

The quantity $\aleph_\alpha = \mathrm{Card}(\omega_\alpha)$ is called the *aleph of index* α. Since ω_α is a cardinal we get $\aleph_\alpha = \omega_\alpha$. In what follows, we shall use $\aleph_\alpha$ instead of ω_α, whenever convenient.

Thus, any initial ordinal is an infinite cardinal. It follows that $x \to \aleph_x$ is strictly increasing.

```
Lemma aleph_pr7  x: infinite_c x -> (* 18 *)
```

```
  exists y, is_ordinal y & omega_fct y = x.
Lemma aleph_pr8 x: is_ordinal x -> is_cardinal (omega_fct x). (* 18 *)

Lemma aleph_pr6_le_lec x y: x <=o y -> \aleph x <=c \aleph y.
Lemma aleph_pr6_lt_ltc x y: x <o y -> \aleph x <c \aleph y.
Lemma aleph_pr6_lec_le x y: is_ordinal x -> is_ordinal y ->
  (\aleph x) <=c (\aleph y) -> x <=o y.
Lemma aleph_pr6_ltc_lt x y: is_ordinal x -> is_ordinal y ->
  (\aleph x) <c (\aleph y) -> x <o y.
Lemma aleph_pr5c x: is_ordinal x -> infinite_c (\aleph x).
Lemma aleph_nz x: is_ordinal x -> \aleph x <> \0c.
Lemma aleph_nz1 x: is_ordinal x -> \0c <c \aleph x.
```

Let x be any cardinal. We know that $x < 2^x$, so that the set of cardinals t with $x < t \leq 2^x$ is non-empty and has a least element. This is the least cardinal y such that $x < y$. It will be called the *cardinal successor* of x. This cardinal is sometimes denoted x^+. This is an abuse of notations, as x^+ denotes the ordinal successor of x (which is the cardinal successor only if x is finite). If x is infinite, it has the form $x = \aleph_n$ and its cardinal successor is $x = \aleph_{n^+}$. The Cantor theorem says $x^+ \leq 2^x$. The Generalized Continuum Hypothesis is the claim that $x^+ = 2^x$ for any infinite cardinal.

```
Lemma aleph_pr10a x y: is_ordinal x ->
   y <c (\aleph (succ_o x)) -> y <=c (\aleph x).
Lemma aleph_pr10b x y: is_ordinal x ->
  \aleph x <c y -> (\aleph (succ_o x)) <=c y.
Lemma aleph_pr10c x: is_ordinal x ->
   (\aleph x) <c (\aleph (succ_o x)).

Definition ord_index x :=
  intersection (Zo (succ_o x) (fun z => \aleph z =x)).
Definition succ_c x:= omega_fct (succ_o (ord_index x)).

Lemma ord_index_pr n: is_ordinal n ->
   ord_index (\aleph n) = n.
Lemma ord_index_pr1 x: infinite_c x ->
   (is_ordinal (ord_index x) & \aleph (ord_index x) = x).
Lemma succ_c_pr n: is_ordinal n ->
   succ_c (\aleph n) = \aleph (succ_o n).
Lemma succ_c_pr1 x (y:= succ_c x): infinite_c x ->
 (x <c y & (forall z, x <c z -> y <=c z) & (forall z, z <c y -> z <=c x)).
Lemma succ_c_pr2 x: infinite_c x -> x <c succ_c x.
Lemma succ_c_pr3 x: infinite_c x ->  succ_c x <=c \2c ^c x.

Definition ContHypothesis:= succ_c \omega = \2c ^c \omega.
Definition GenContHypothesis:= forall x, infinite_c x -> succ_c x = \2c ^c x.
```

We show that $x \to \omega_x$ is a normal OFS. Given a limit ordinal α, we must show $\omega_\alpha = \sup_{\beta<\alpha}(\omega_\beta)$. Let S be the sup. The relation $\omega_\alpha \geq S$ is obvious. Moreover, α is not zero, so that $\omega_\alpha = \sup_{x\in T} x$, and we have to show $x \leq S$ for all $x \in T(\alpha)$. By definition of T, there is β such that $\beta < \alpha$ and $\mathrm{Card}(x) \leq_{\mathrm{Card}} \mathrm{Card}(\omega_\beta)$. Let $\gamma = \beta^+$. Since α is a limit ordinal we have $\gamma < \alpha$ so that $\omega_\gamma \leq S$. The previous result says $\mathrm{Card}(\omega_\beta) <_{\mathrm{Card}} \mathrm{Card}(\omega_\gamma)$ so that $\mathrm{Card}(x) <_{\mathrm{Card}} \mathrm{Card}(\omega_\gamma)$. This is $x < \omega_\gamma$ and concludes the proof.

```
Lemma aleph_pr11: normal_ofs omega_fct. (* 26 *)
```

Since ω_x is strictly increasing, it follows that $\aleph_x$ is strictly increasing. It follows that $\aleph_{x+1}$ is the supremum of all ordinals y such that $\mathrm{Card}(y) \leq \aleph_x$. Note that $y < \aleph_{x+1}$, since if we had equality, then y would be a cardinal, hence equal to its cardinal.

```
Lemma aleph_limit x: is_ordinal x -> limit_ordinal (omega_fct x).
Lemma aleph_pr12a x: is_ordinal x-> (* 35 *)
  \aleph (succ_o x) = \osup (set_ordinal_card_le (\aleph x)).
Lemma aleph_pr12b x z :
  is_ordinal x -> is_ordinal z -> (\aleph x) <c (cardinal z) ->
  (\aleph (succ_o x)) <=o z.
Lemma aleph_pr12c x z:
  is_ordinal x-> is_ordinal z -> cardinal z <=c (\aleph x) ->
  z <o (\aleph (succ_o x)).
Lemma aleph_pr12d x z:
  is_ordinal x -> is_ordinal z -> z <o (\aleph (succ_o x)) ->
  cardinal z <=c (\aleph x).
```

8.15 Cardinal Cofinality

Let $x > 1$ be a cardinal. We say that $(x_i)_{i\in \mathrm{I}}$ is a small family with sum y if $x_i < x$ for all i and $y = \sum x_i$. An example is the constant family $x_i = 1$ with $\mathrm{I} = x$. The least $\mathrm{Card}(\mathrm{I})$ for which there is a small family with sum x is called the *cofinality* of x. Since I is equipotent to its cardinal, we may replace "the least $\mathrm{Card}(\mathrm{I})$" by "the least cardinal I", or even "the least ordinal I". An infinite cardinal equal to its cofinality is called *regular*.

```
Definition csum_of_small1 x f:=
    cardinal_fam f & (forall i, inc i (domain f) -> V i f <c x).
Definition csum_of_small2 x z f:=
   csum_of_small1 x f & domain f = z.
Definition csum_of_small x y z f:=
   csum_of_small2 x z f & card_sum f = y.
Definition cofinality_c x:=
  least_ordinal(fun z => exists f, csum_of_small x x z f) x.
Definition regular_cardinal x :=
  infinite_c x & cofinality_c x = x.
```

Let $(x_i)_{i\in \mathrm{I}}$ be a family of cardinals. Assume I equipotent to J; then there exists a family $(x'_j)_{j\in \mathrm{J}}$ such that $\sum x_i = \sum x'_j$. The two families have the same range; this means for instance that if x_i is never zero, then x'_j is never zero. It also says that, if $x_i < x$ then $x'_i < x$. Taking for J the cardinal of I yields: if we have a small family indexed by I, we deduce a small family, with the same sum and same range, indexed by $\mathrm{Card}(\mathrm{I})$.

Let $y = \sum x_i$ and $s = \sup x_i$. Then $y \leq \mathrm{I}s$. In particular, if $x_i \leq x$, we have $y \leq \mathrm{I}x$; a fortiori this holds if $x_i < x$. Assume s infinite and $\mathrm{I} \leq s$. Then $\mathrm{I}s = s$ (note that I cannot be empty) so that $y = s$. This holds if, for instance there exists an index i such that x_i is infinite and $\mathrm{I} \leq x_i$.

Assume now that x is a successor cardinal, say $x = z^+$. Assume $\mathrm{I} < x$, and $x_i < x$. Then $x_i \leq z$ and $\mathrm{I} \leq z$ so that $y \leq z$, and $y < x$. This will be restated as x is regular.

```
Lemma csum_commutative2 f x y z: (* 13 *)
 csum_of_small x y z f ->
 exists g, (csum_of_small x y (cardinal (domain f)) g
   & range g = range f).
```

```
Lemma csum_commutative1 f x y z:
  csum_of_small x y z f ->
  exists g, csum_of_small x y (cardinal (domain f)) g.
Lemma csum_of_small_b1 x y z f:
  cardinal_fam f -> (forall i, inc i (domain f) -> V i f <=c x) ->
  card_sum f = y -> domain f = z ->
  y <=c (x *c z).
Lemma csum_of_small_b2 x y z f:
  csum_of_small x y z f -> y <=c (x *c z).
Lemma csum_of_small_b3 x y z f:
  csum_of_small x y z f ->
  (exists n, is_ordinal n & x = \aleph (succ_o n)) ->
  (cardinal z) <c x ->  y <c x.
Lemma csum_of_small_b4 f (s:= \osup (range f)) :
   cardinal_fam f -> infinite_c s ->
   (cardinal (domain f) <=c s) -> card_sum f = s.
Lemma csum_of_small_b5 f (s:= \osup (range f)) :
   cardinal_fam f ->
   (exists i, inc i (range f) & infinite_c i & cardinal (domain f) <=c i) ->
   card_sum f = s.
```

We state: if y is the cofinality of x, then $y \leq x$, there is a small family with sum x indexed by y, and y is the least ordinal satisfying this property. We may assume $x_i \neq 0$. If x is infinite, we may even assume $x_i \geq 2$ (replace all x_i by $x_i + 2$; this increases the sum by $2y$, but $x + 2y = x$).

```
Lemma cofinality_c_rw x (y:= cofinality_c x): \2c <=c x -> (* 20 *)
  ( y <=c x &
   (exists f, csum_of_small x x y f)
   & (forall z, is_ordinal z ->
      (exists f, csum_of_small x x z f) -> y <=o z)).
Lemma cofinality_c_pr2 x: \2c <=c x ->
    (exists f, csum_of_small x x (cofinality_c x) f
       & (forall i, inc i (domain f) -> V i f <> \0c)).
Lemma cofinality_c_pr3 x: infinite_c x -> (* 25 *)
    (exists f, csum_of_small x x (cofinality_c x) f
       & (forall i, inc i (domain f) -> \2c <=c V i f)).
```

If x is finite, we have $x = 1 + (x - 1)$ so that its cofinality is 2. Thus 2 is the only integer equal to its cofinality (but is not considered as a regular cardinal). If x is infinite, its cofinality is infinite, for if x_i is a family indexed by a finite set I, there is a greatest x_i, say x_k, and the sum is at most $\mathrm{I}x_k$; since I is finite and x infinite, $x = \sum x_i$ implies $x_k = x$. We characterize regular cardinals by the fact that any small family for which the index set has a cardinal $< x$ has value $< x$. In particular $\aleph_0$ and $\aleph_{\alpha+1}$ are regular.

```
Lemma infinite_gt_one x: infinite_c x -> \1c <c x.
Lemma cofinality_c_finite x: \2c <=c x -> finite_c x -> (* 33 *)
   cofinality_c x = \2c.
Lemma cofinality_c_small x: \2c <=c x -> (cofinality_c x) <=c x.
Lemma cofinality_c_cardinal x: \2c <=c x ->
  is_cardinal (cofinality_c x).
Lemma cofinality_infinite x: infinite_c x -> (* 32 *)
  infinite_c (cofinality_c x).
Lemma infinite_regular_pr x: infinite_c x -> (* 20 *)
  (regular_cardinal x <->
   forall f y z, csum_of_small x y z f -> cardinal z <c x ->y <c x).
```

```
Lemma regular_cardinal_omega: regular_cardinal \omega.
Lemma regular_initial_successor x: is_ordinal x ->
   regular_cardinal (\aleph (succ_o x)).
```

Let f be a normal OFS. Let Z_i be the complement of $f(i)$ in $f(i+1)$. We know that each ordinal is either $< f(0)$ or in some Z_i. Thus, any $z = f(t)$, where t is limit, is the disjoint union of $f(0)$ and the sets $(\mathrm{Z}_i)_{i<t}$. Thus $\mathrm{Card}(z) = \mathrm{Card}(f(0)) + \sum_{i<t} \mathrm{Card}(\mathrm{Z}_i)$. It follows that the cofinality of $\mathrm{Card}(z)$ is no greater than t. We shall later on consider a function $f : t \to z$, where t is a limit ordinal, and assume that the extension of f to $t^+ \to z^+$, defined by $f(t) = z$, is normal. The same conclusion holds.

Consider $f(x) = \omega_x$. In this special case $f(x)$ is a cardinal, and $\mathrm{Card}(z) = f(t)$. Moreover, $f(i) <_{\mathrm{Card}} f(i+1)$ so that the cardinal of Z_i is $f(i+1)$. It follows

$$\aleph_\alpha = \aleph_0 + \sum_{\beta<\alpha} \aleph_{\beta+1}. \tag{20a}$$

If we replace $\aleph_{i+1}$ by $\aleph_i$ and omit $\aleph_0$, we get a quantity which is $\leq \aleph_\alpha$; so if we add $\aleph_\alpha$ we get $\aleph_\alpha$. Thus

$$\aleph_\alpha = \sum_{\beta\leq\alpha} \aleph_\beta. \tag{20b}$$

Assume that α is a limit ordinal, E is subset of α, whose supremum is α. We have

$$\aleph_\alpha = \sum_{\beta\in\mathrm{E}} \aleph_\beta, \qquad (\sup \mathrm{E} = \alpha, \text{ limit ordinal}). \tag{20c}$$

(note that $\mathrm{Card}(\mathrm{E}) \leq \aleph_\alpha = x$, and the supremum of the $\aleph_\beta$ which is some $\aleph_\gamma$ cannot be $< \aleph_\alpha$). In particular

$$\aleph_\alpha = \sum_{\beta<\alpha} \aleph_\beta \qquad (\alpha \text{ limit ordinal}). \tag{20d}$$

Note that, if α is a successor, say $\gamma + 1$, then the sum in (20d) is $\aleph_\gamma$ by (20b).

```
Definition aleph_succ_comp x :=
    complement  (\aleph (succ_o x)) (\aleph x).
Lemma aleph_succ_pr1 x: is_ordinal x ->
  (forall t, inc t  (aleph_succ_comp x) <->
     ((\aleph x) <=o t & t <o (\aleph (succ_o x)))).
Lemma aleph_succ_pr2 x: is_ordinal x ->
  cardinal (aleph_succ_comp x) = \aleph (succ_o x).
Lemma aleph_succ_pr3 x y: is_ordinal x -> is_ordinal y ->
  x = y \/ disjoint (aleph_succ_comp x) (aleph_succ_comp y).
Lemma aleph_sum_pr1 x: is_ordinal x ->
  inc x \omega \/ exists y, is_ordinal y &  inc x (aleph_succ_comp y).

Lemma aleph_sum_pr2 x: is_ordinal x -> (* 38 *)
  \aleph x = \omega +c (card_sum (L x (fun z => \aleph(succ_o z)))).
Lemma aleph_sum_pr3 x: is_ordinal x -> (* 25 *)
  \aleph x = (card_sum (L (succ_o x) (fun z => \aleph z))).
Lemma aleph_sum_pr4 x E: limit_ordinal x -> (* 33 *)
  sub E x -> \osup E = x ->
  \aleph x = (card_sum (L E (fun z => \aleph z))).
Lemma aleph_sum_pr5 x: limit_ordinal x ->
  \aleph x = (card_sum (L x (fun z => \aleph z))).
```

8.16 Ordinal Cofinality

We know that successor cardinals are regular. What about limit ordinals? In general, they are not regular, in the exceptional case they are called "inaccessible". We first extend the notion of cofinality to ordered sets.

The cofinality of an ordered set E is the least cardinal α such that there is a subset M of E, which is cofinal in E and whose cardinal is α. Note that M contains all maximal elements of E, so that the cofinality of a finite set is the number of its maximal elements. One may replace cardinal by ordinal in the previous definition. We then require that the ordinal of M (ordered by the ordering induced from E) is α. In particular M has to be well-ordered, and E is totally ordered. Note that if E is totally ordered, such a set M exists (exercise 6.16). Note also that if E is well-ordered, that any subset M is well-ordered.

We start with the Bourbaki definition.

```
Definition cofinality_aux r :=
   (Zo (powerset (substrate r))
      (fun z => cofinal_set r z & worder (induced_order r z))).
Definition cofinality' r := (fun_image (cofinality_aux r)
     (fun z => ordinal (induced_order r z))).
Definition cofinality_alt x :=
  intersection (cofinality' (ordinal_o x)).

Lemma cofinality'_pr0  x (r:= ordinal_o x): is_ordinal x -> (* 14 *)
   (nonempty (cofinality' r) & ordinal_set (cofinality' r)).
Lemma cofinality_pr1 x (y:= cofinality_alt x) (r:= ordinal_o x):
  is_ordinal x ->
  ( exists z, sub z x & cofinal_set r z & worder (induced_order r z) &
      y = ordinal (induced_order r z)).
Lemma cofinality_pr2 x  (r:= ordinal_o x) z:
  is_ordinal x -> sub z x -> cofinal_set r z ->
  (cofinality_alt x) <=o ordinal (induced_order r z).
```

We say that a function $f : y \to x$ is *cofinal* if its range is cofinal. This means that, for any $t \in x$, there is $z \in y$ such that $f(z) \geq t$.

We define the *cofinality* of x, denoted cf(x), to be the least ordinal y for which there exists a cofinal function f with source y. This agrees with the previous definition. It may be interesting to assume f strictly increasing and normal.

```
Definition cofinal_function f x y :=
  function_prop f y x &
  (forall t, inc t x -> exists z, inc z y & t <=o W z f).
Definition cofinal_function_si f x y:=
  cofinal_function f  x y &
   (forall a b, inc a y -> inc b y ->
      a <o b-> (W a f) <o (W b f)).
Definition cofinal_function_si_normal f x y:=
   cofinal_function_si f x y &
   (forall a, inc a y -> limit_ordinal a ->
      W a f = \osup (image_by_fun f a)).

Definition cofinal_function_ex x y:= exists f, cofinal_function f x y.
Definition cofinality x :=
    least_ordinal (cofinal_function_ex x) x.
```

Since the identity function is cofinal, the cofinality of x is well-defined and is an ordinal $\leq x$. If x is zero, its cofinality is zero, and conversely. The cofinality of x is one if and only if x is a successor. In all other cases, the cofinality is a limit ordinal.

```
Lemma cofinal_function_pr2 a: is_ordinal a -> cofinal_function_ex a a.
Lemma cofinality_rw a (b:= cofinality a) :
  is_ordinal a -> (is_ordinal b & cofinal_function_ex a b &
    forall z, is_ordinal z -> cofinal_function_ex a z ->  b <=o z).
Lemma OS_cofinality a: is_ordinal a -> is_ordinal (cofinality a).
Lemma cofinality_pr3 a: is_ordinal a -> (cofinality a) <=o a.
Lemma cofinality0: cofinality \0o = \0o.
Lemma cofinality_n0 x: is_ordinal x -> x <> \0o ->
   (cofinality x) <> \0o.
Lemma cofinality1: cofinality \1o = \1o.
Lemma cofinality_succ x: is_ordinal x -> cofinality (succ_o x) = \1o.
Lemma cofinality_limit1 n: is_ordinal n ->
  (cofinality n = \1o <-> exists m, is_ordinal m & n = succ_o m).
Lemma cofinality_limit2 x (y:= cofinality x): is_ordinal x ->  (* 44 *)
  y = \0o \/ y = \1o \/ limit_ordinal y.
Lemma cofinality_limit3 x: limit_ordinal x -> limit_ordinal (cofinality x).
```

If x is any ordinal, there exists a strictly increasing cofinal function $f : \mathrm{cf}(x) \to x$. We may even assume f normal. We start with a cofinal function g. This function satisfies the desired property if $\mathrm{cf}(x)$ has zero or one element, so that we may assume $\mathrm{cf}(x)$ and x to be limit ordinals. One can show that the function h defined by transfinite induction via $h(a) = \sup(g(a), \sup_{b<a}(h(b)+1))$ is strictly increasing. A variant will give us a normal function.

Let h be any set. Let a be the source of h, $b = a^-$, $v_1 = \sup(h(b)^+, g(b))$, $v_2 = \sup\{h(t)\}$, and v be v_1 if $a = b^+$, and v_2 otherwise. We define $p(h)$ to be v if $v \in x$, and 0 otherwise. Since $0 \in x$, the quantity $p(h)$ is always in x. Note that $p(h)$ is well-defined for all h, but if h is not a function, then its source or quantities like $h(b)$ have no specific meaning.

We define, by transfinite induction on the set $\mathrm{cf}(x)$, a function f via the property p. This means, that for any $a \in \mathrm{cf}(x)$, $f(a) = p(h)$, where h is the restriction of f to the set A of all elements of $\mathrm{cf}(x)$ that are $< a$ for ordering induced by $\leq_{\mathrm{ord}}$ on $\mathrm{cf}(x)$. Because $p(h) \in x$, it follows that f is a function $\mathrm{cf}(x) \to x$. Since a is an ordinal, the source of h is $\mathrm{A} = a$. Note that $f(0) = 0$, because in this case $v = v_2 = \sup\emptyset$. This condition will be useful later on. Assume a limit, so that $v = v_2$. This is the supremum of all $f(t)$ for $t < a$. It is $< x$, for otherwise h would be a cofinal function, and cofinality of x would be $\leq a$, contradicting $a < y$. Thus $f(a) = \sup_{t<a} f(t)$, so that f is a normal function. Assume finally that a is a successor, say $a = \mathrm{B}^+$. Then $b = a^- = \mathrm{B}$ and $a = b^+$ so that $v = v_1$. We have $h(b) = f(b) \in x$ and $h(b)^+ \in x$. Thus $f(a)$ is the greatest of $f(b)^+$ and $g(b)$. We deduce $f(b) < f(b^+)$ and f is strictly increasing, and $g(b) \leq f(b^+)$ and f is cofinal.

We deduce that the cofinality X of x (according to Bourbaki) is $\mathrm{cf}(x)$. In fact, if f is strictly increasing, and cofinal with range z, that f induces an order isomorphism between $\mathrm{cf}(x)$ and z. Conversely, X is the ordinal of some cofinal z, so that there exists an isomorphism $\mathrm{X} \to z$, that induces a function $\mathrm{X} \to x$ with range z and this function is obviously cofinal.

```
Lemma cofinality_pr4 x (y:= cofinality x): is_ordinal x ->
   exists f, (cofinal_function_si_normal f x y) &
        (inc \1o y -> W \0o f = \0o).  (* 203 *)
Lemma cofinality_sd a: is_ordinal a -> (* 89 *)
  (cofinality a) = (cofinality_alt a).
```

Let f be a normal OFS. Consider the supremum $g(x)$ of the sequence $a_{i+1} = f(a_i)$ with $a_0 = x$. We have seen that this is the least fixed point of f that is $\geq x$. Obviously $g(x) = x$ if $f(x) = x$. Otherwise, the sequence a_i is cofinal in $g(x)$ so that the cofinality of $g(x)$ is ω.

Assume now that $f : y \to x$ is a normal function. We can define $g(x)$ as above, for $x < y$. We have $g(x) \leq y$, so that either $g(x) < y$, and $f(g(x))$ is defined and $g(x)$ is a fixed-point of f, or $g(x) = y$, case where the cofinality of y is ω. We state: if the cofinality of y is not ω, then any normal function $f : y \to y$ has arbitrarily large fix-points.

```
Definition least_fixedpoint_ge f x y:=
  x <=o y & f y = y & (forall z, x <=o z -> z <o y -> f z  <> z).

Lemma least_fixed_point_exists x f:
  is_ordinal x -> normal_ofs f -> exists y,
   least_fixedpoint_ge f x y.
Lemma normal_function_fixpoints x f: (* 109 *)
  is_ordinal x ->
  function_prop f x x ->
  (forall a b, inc a x -> inc b x ->
    a <o b-> (W a f) <o (W b f)) ->
  (forall a, inc a x -> limit_ordinal a ->
    W a f = \osup (image_by_fun f a)) ->
  (cofinality x <> \omega) ->
  (forall a, inc a x -> exists b, inc b x & a <=o b & W b f = b).
Lemma cofinality_least_fp_normal x y f: ( 75 *)
  normal_ofs f -> f x <> x ->  least_fixedpoint_ge f x y ->
  cofinality y = \omega.
```

We say that an ordinal x is *regular* if $\mathrm{cf}(x) = x$, and *singular* otherwise.

We have: whenever $f : x \to y$ is cofinal and strictly increasing, then $\mathrm{cf}(x) = \mathrm{cf}(y)$. Proof. Let f_x and f_y be cofinal mappings $\mathrm{cf}(x) \to x$ and $\mathrm{cf}(y) \to y$ respectively. Composing f and f_x gives a cofinal mapping $\mathrm{cf}(x) \to y$, so that $\mathrm{cf}(y) \leq \mathrm{cf}(x)$. On the other hand, if $a \in \mathrm{cf}(y)$, there is $t \in x$ such $f(t) \geq f_y(a)$. Call it $g(x)$. If $a \in x$, there is $b \in \mathrm{cf}(y)$ such that $f(x) \leq f_y(b)$, thus $x \leq g(b)$. This shows that g is a cofinal function.

If we take $x = \mathrm{cf}(y)$ it follows $\mathrm{cf}(\mathrm{cf}(y)) = \mathrm{cf}(y)$. A cofinality is regular. Zero and one are regular: they are the only regular finite ordinals. The next regular ordinal is ω.

```
Definition regular_ordinal x := is_ordinal x & cofinality x = x.
Definition singular_ordinal x :=  is_ordinal x & not (regular_ordinal x).

Lemma regular_0: regular_ordinal \0o.
Lemma regular_1: regular_ordinal \1o.
Lemma regular_finite x:  (* 6 *)
   regular_ordinal x -> x = \0o  \/ x = \1o \/ \omega <=o x.
Lemma cofinality_pr5 a b:  (* 40 *)
  is_ordinal a -> is_ordinal b ->
  (exists f, cofinal_function_si f b a) ->
  cofinality a = cofinality b.
Lemma cofinality_proj x: is_ordinal x ->
   cofinality (cofinality x) = cofinality x.
Lemma cofinality_reg x: is_ordinal x ->
   regular_ordinal (cofinality x).
Lemma cofinality_limit4 x: limit_ordinal x -> \omega <=o cofinality x.
Lemma regular_omega: regular_ordinal \omega.
```

On can show that a and $b+a$ have same cofinality (if a is non-zero), that a and $b\cdot a$ have same cofinality (if b is non-zero and a is limit), that a regular ordinal is indecomposable, and moreover, is 0, 1, ω or ω^y, where y is a limit ordinal (see exercises). It happens that a cofinality is 0, 1, or a regular cardinal. For this reason we consider mostly cardinals in what follows.

The cofinality y of an ordinal x is a cardinal (since the cardinal of y is an ordinal equipotent to it and $\leq y$). It follows that a regular ordinal is a cardinal.

The cofinality of an infinite cardinal is the cardinal cofinality as introduced above. In fact, let $f : \mathrm{cf}(x) \to x$ be cofinal, normal, and $f(0) = 0$. In the proof of (20a) we have explained that this gives a small family, indexed by $\mathrm{cf}(x)$ with sum x. This shows that $\mathrm{cf}(x)$ is not less than the cardinal cofinality. On the other hand, consider a small family x_i indexed by I. We pretend $\mathrm{cf}(x) \leq \mathrm{I}$. This is obvious when $\mathrm{I} \geq x$, so assume $\mathrm{I} < x$. We have $t = x$, for otherwise we would have $t < x$ and $\sum x_i < x$. This says that the sequence x_i is cofinal and $y \leq \mathrm{I}$.

```
Lemma cofinality_cardinal x: is_ordinal x ->
  is_cardinal (cofinality x).
Lemma regular_is_cardinal x: regular_ordinal x -> is_cardinal x.
Lemma cofinality_infinite_limit x:
   limit_ordinal x -> infinite_c (cofinality x).
Lemma cofinality_infinite_cardinal x:
   infinite_c x -> infinite_c (cofinality x).
Lemma cofinality_card1 x : infinite_c x -> (* 88 *)
   exists f, csum_of_small2 x (cofinality x) f.
Lemma cofinality_card x: infinite_c x -> (* 60 *)
  cofinality_c x = cofinality x.
Lemma cofinality_regular x (y:= cofinality x): is_ordinal x ->
  y = \0c \/ y = \1c \/ regular_cardinal y.
```

Exercise 16c concludes with: if ω_β is the cofinality of ω_α, then $\beta \leq \alpha$ and ω_α is regular if and only if $\alpha = \beta$.

```
Lemma cofinality_index n: is_ordinal n ->
   ord_index (cofinality (omega_fct n)) <=o n.
Lemma cofinality_index_regular n: is_ordinal n ->
   (regular_ordinal  (omega_fct n) <->
    ord_index (cofinality (omega_fct n)) = n).
```

If α is a limit ordinal then $\mathrm{cf}(\aleph_\alpha) = \mathrm{cf}(\alpha)$. Proof: let $c = \mathrm{cf}(\aleph_\alpha)$; consider a cofinal function $f : \mathrm{cf}(\alpha) \to \alpha$, and let E be the range of f. Relation (20c) gives us a small sum; thus $c \leq \mathrm{cf}(\alpha)$.Conversely, assume $\aleph_\alpha = \sum_{i\in \mathrm{I}} x_i$, where $\mathrm{Card}(\mathrm{I}) = c$. Write $x_i = \aleph_{y_i}$ (if x_i is finite, we let $y_i = 0$). Let $z = \sup y_i$. If $z = \alpha$, we have a cofinal function, and $\mathrm{cf}(\alpha) \leq c$. Otherwise $x_i \leq \aleph_\alpha$ (even when x_i is finite), so that $\mathrm{card}(\mathrm{I}) = \aleph_\alpha$ and $\mathrm{cf}(\alpha) \leq c$ again.

Thus, if $\alpha < \aleph_\alpha$, then $\aleph_\alpha$ is singular. In particular, $\aleph_\omega$ is the least singular cardinal.

```
Lemma regular_initial_limit0 x: limit_ordinal x -> (* 33 *)
  cofinality_c (\aleph x) <=o cofinality x.
Lemma regular_initial_limit1 x: limit_ordinal x -> (* 74 *)
  cofinality_c (\aleph x) = cofinality x.
Lemma regular_initial_limit2 x: limit_ordinal x ->
  cofinality_c (\aleph x) <=o x.
Lemma regular_initial_limit3 x: limit_ordinal x ->
    x <o \aleph x -> singular_cardinal (\aleph x).
Lemma regular_initial_limit4: singular_cardinal (\aleph \omega).
```

```
Lemma regular_initial_limit5 x:
  singular_cardinal x -> (\aleph \omega) <=c x.
```

If α is a limit ordinal and $\aleph_\alpha$ is regular, it is called *weakly inaccessible*. A necessary condition is that $\alpha = \omega_\alpha$. The condition is not sufficient, for instance, the least fixed-point of ω_x is a singular cardinal (more generally, if α is weakly inaccessible, the least fixed-point of ω_x greater than α is singular).

```
Definition inaccessible_w x :=
  regular_cardinal x & (exists n, limit_ordinal n & x = \aleph n).

Lemma inaccessible_pr1 x:
    inaccessible_w x -> x = \aleph x.
Lemma cofinality_least_fp_normal2 x y f:
  normal_ofs f -> f x <> x ->  least_fixedpoint_ge f x y ->
  y = \omega  \/ singular_ordinal y.
Lemma cofinality_least_fp_omega_normal y:
  least_fixedpoint_ge omega_fct \0o y -> singular_cardinal y.
```

In Exercise 16c, Bourbaki says: "there exists only one regular ordinal which is cofinal in a given totally ordered set E; this ordinal is equal to the final character of E, and if E is not empty and has no greatest element, it is an initial ordinal."

This statement can be formalized as: there exists a unique regular ordinal x such that there is a cofinal set F in E, so that the ordinal of F (with the ordering of E) is x. This is the final character of E, and is an initial ordinal if E satisfies the stated properties. We shall only prove partly this statement. Note first that, if E satisfies the stated condition, then any cofinal set F has to be infinite; so that if x exists, it is an initial ordinal. Let x be the final character of E; we know that it exists; there is a cofinal set F, and an order isomorphism $x \to$ F. Let $y = \mathrm{cf}(x)$; there is a cofinal function $g : y \to x$. Composing g and f yields a cofinal function, so that $x \leq y$. Thus $x = y$. This shows that x is regular. This part is not formally proven.

We show here: assume that x and y are two regular ordinals, and f, g two strictly increasing cofinal functions with the same target. Then $x = y$. This is the uniqueness part of the theorem. For simplicity, we assume that the target is ordered by $\leq_{\mathrm{ord}}$ but the result is independent of the ordering (the property holds even when the target is not totally ordered). There exists $h : x \to y$ such that $g(h(t)) \geq f(t)$ for all t. Let $a \in y$; there is $b \in x$ with $f(b) \geq g(a)$. Thus $g(h(b)) \geq f(b) \geq g(a)$. Since g is strictly increasing we deduce $g(b) \geq a$. This shows that h is cofinal; thus $y \leq x$. Exchanging x and y gives $x = y$.

```
Lemma regular_cofinal_si_unique x y z:  (* 24 *)
  regular_ordinal x -> regular_ordinal y ->
  (exists f, cofinal_function_si f z x) ->
  (exists f, cofinal_function_si f z y) ->
  x = y.
Lemma cofinality_limit_increasing x y:
  is_ordinal x -> is_ordinal y ->
  cofinality (omega_fct x) = omega_fct y ->
  (y <=o x &  regular_ordinal (omega_fct x) <-> (x=y)).
```

8.17 Infinite Products

Exercise 3.3 (page 340) is known as König's Theorem: if $(x_i)_{i\in I}$ and $(y_i)_{i\in I}$ are two families of cardinals such that $x_i < y_i$, then $\sum x_i < \prod y_i$. We first state a similar result where $<$ is replaced by $\leq$; in this case, we need $y_i \geq 2$.

```
Section Exercise3_3.
Variables f g :Set.
Hypothesis  (fgf: fgraph f)(fgg: fgraph g).
Hypothesis sd: domain f = domain g.
Hypothesis hg:  (forall i, inc i (domain g) -> \2c <=c (V i g)).

Lemma compare_sum_prod1: (* 157 *)
   (card_sum g) <=c  (card_prod g).

Lemma compare_sum_prod2 :
  (forall i, inc i (domain f) -> (V i f) <=c (V i g)) ->
   (card_sum f) <=c (card_prod g).
Lemma compare_sum_prod4 : (* 60 *)
  (forall i, inc i (domain f) -> (V i f) <c (V i g)) ->
   (card_sum f) <c (card_prod g).
End Exercise3_3.

Lemma compare_sum_prod  f g: (* 29 *)
  fgraph f ->  fgraph g -> domain f = domain g ->
  (forall i, inc i (domain f) -> (V i f) <c (V i g)) ->
  (csum f) <c (cardinal_prod g).
```

Let's study the cardinal power a^b in the case where some arguments are infinite. Assume first b finite, $b > 0$. If a is infinite, then $a^b = a$. Assume b infinite large, we have:

$$2 \leq a \leq 2^b, \quad \omega \leq b \implies a^b = 2^b.$$

This follows trivially from monotonicity of the power function and $b^2 = b$. We have $a^b = 2^b$ if $a \leq b$ (by Cantor), if $a = b$ and if a is the cardinal successor of b.

A direct application of König's theorem shows $\kappa^{\mathrm{cf}(\kappa)} > \kappa$. We deduce $\mathrm{cf}(2^\kappa) > \kappa$ when κ is infinite. In particular, the cofinality of $2^{\aleph_0}$ is $> \aleph_0$. Note that $\mathrm{cf}(\lambda^\kappa) > \kappa$ when λ and κ are infinite.

```
Lemma infinite_power1 a b:  \2c <=c a -> a <=c (\2c ^c b) -> infinite_c b ->
   a ^c b = \2c ^c b.
Lemma infinite_power1_a a b:  \2c <=c a -> a <=c b -> infinite_c b ->
   a ^c b = \2c ^c b.
Lemma infinite_power1_b x: infinite_c x ->  x ^c x = \2c ^c x.
Lemma infinite_power1_c x: infinite_c x ->
  (succ_c x) ^c x = \2o ^c x.
Lemma power_cofinality x: \2c <=c x -> x <c  x ^c (cofinality_c x).
Lemma power_cofinality2 x: infinite_c x  -> x <c (cofinality_c (\2c ^c x)).
Lemma power_cofinality3: \omega <c (cofinality_c (\2c ^c \omega)).
Lemma power_cofinality4 a b: infinite_c a -> infinite_c b ->
   b <c cofinality (a ^c b).
Lemma power_cofinality5 x y: \2c <=c x -> infinite_c y ->
  y <c cofinality_c (x ^c y).
```

We have: let $(x_i)_{i\in \mathrm{I}}$ be a family of non-zero cardinals, whose supremum is s; assume that I is a cardinal; we may consider I as an ordinal, thus an ordered set; assume x_i increasing for this ordering. Then $\prod x_i = s^{\mathrm{I}}$.

Notice first that if $x_i \geq 2$, then $\sup x_i \leq \sum x_i \leq \prod x_i$. The same holds if merely $x_i \geq 1$. The relation $\prod x_i \leq s^{\mathrm{I}}$ is trivial. The converse inequality is non-trivial. We may always assume the sequence increasing (indexed by some ordinal I), but in general I is not a cardinal.

Let J be some ordinal whose cardinal is greater than the cardinal of I. Let b such that $b \notin \mathrm{I}$, and x_b some cardinal greater than all x_i. Define by transfinite induction on J a function f such that $f(i)$ is some index j such that x_j is the least among those x_k where k has not the form $f(t)$ for some $t < j$. Each element of I can be chosen at most once, while b can be used more than once. Note that, if b is unused, then f is injective, contradicting the assumption on the cardinal of J. Thus, there is a least j such that $f(j) = b$. By definition, the set of all $f(i)$ for $i < j$ is then I. Thus, f is a bijection $j \to \mathrm{I}$, and $i \to x_{f^{-1}(i)}$ is an ordering of the sequence x_i. This shows that the sequence can always be assumed increasing.

Let's consider an example: x_i is either α or β. We assume that a and b are the number of times x_i is α or β. Then the product is $\alpha^a\beta^b$. Assume $\alpha \leq \beta$ and $b \leq a$. The number of factors is then a, and if the results applies, the product would be β^a. This is obviously an upper bound for the product, but it can be greater. Example: let β be any singular cardinal; choose a such that $\mathrm{cf}(\beta) < a < \beta$ and b such that $b < \mathrm{cf}(\beta)$; take also $\alpha = b$. Assume for simplicity that GCH holds; it follows that $\alpha^a = 2^a \leq \beta$ and $\beta^b = \beta$ so that the product is β. But $\beta^a = 2^\beta > \beta$. Conversely, assume that we can order the sequence x_i, so that the index set I is a cardinal. Then $a \leq b$. Proof: there is some $j \in \mathrm{I}$ such that $x_i = \alpha$ for $i < j$ and $x_i = \beta$ otherwise. Thus a is the cardinal of j and b is the cardinal of the complement. If $b < a$ it follows $\mathrm{Card}(j) = \mathrm{card}(\mathrm{I})$, and j is equipotent to I. By definition of a cardinal, any ordinal j equipotent to I is $\geq j$; but $j \in \mathrm{I}$ says $j < i$, so that $b < a$ is absurd.

Since I is infinite, there is a bijection $f : \mathrm{I} \times \mathrm{I} \to \mathrm{I}$. Set $y_{jk} = x_{f(j,k)}$. By associativity $\prod x_i = \prod_j (\prod_k y_{jk})$, and the result holds as each inner product is $\geq s$. The non-trivial point is: for any i, j, there is k such that $x_i \leq y_{jk}$. Assume $i = f(n, m)$. Since x_i is increasing, we get: for any n, m, j, there is k such that $f(n, m) \leq f(j, k)$. Not any f satisfies this condition. We use the function associated to the canonical ordering of pairs of ordinals (see page 98).

```
Lemma compare_sum_prod5  x (y := domain x) :
  (fgraph x) ->
  (forall a, inc a y -> \0c <c V a x) ->
  (\csup (range x)) <=c (card_prod x).  (* 41 *)
Lemma infinite_increasing_power x (y := domain x):  (* 67 *)
   (fgraph x) -> (infinite_c y) ->
   (forall a, inc a y -> \0c <c V a x) ->
   (forall a b, inc a y -> inc b y -> a <=o b -> V a x <=c V b x) ->
   (card_prod x) = (\csup (range x)) ^c y.
```

TODO[1]

We state [Hausdorff, 1904]

$$\aleph_{\alpha+1}^{\mathfrak{m}} = \aleph_\alpha^{\mathfrak{m}} \cdot \aleph_{\alpha+1} \qquad (\mathfrak{m} \neq 0). \tag{21a}$$

We deduce (by induction when $\mathfrak{m}$ is infinite) [Bernstein]

$$\aleph_n^{\mathfrak{m}} = 2^{\mathfrak{m}} \cdot \aleph_n \qquad (n \in \mathbf{N}, \mathfrak{m} \neq 0). \tag{21b}$$

[1] Assume $x = \omega_\alpha$ regular; assume $\mathrm{I} < x$ and $x_i < x$, then $\sum_{i\in \mathrm{I}} x_i < x$, where I and x_i are ordinal, and the $\sum$ is the ordinal sum.

Let $z = \aleph_\alpha$, $x = z^+ = \aleph_{\alpha+1}$ and $y = \mathfrak{m}$. If $x \leq y$, then $z^y = x^y = 2^y$ and (21a) holds trivially. Otherwise, assume $y < x$. We count here the number of functions $f : y \to x$. We first notice that $\sup f <_{\mathrm{ord}} x$; this can also be written as $\sup f \in x$; it follows from $\mathrm{Card}(\sup(f)) \leq \sum \mathrm{Card}(f(i)) \leq yz < x$. The first inequality holds since the supremum is a union, the second follows from $f(i) <_{\mathrm{ord}} x$, so that $\mathrm{Card}(f(i)) < x$, and $\mathrm{Card}(f(i) \leq z$; the last inequality follows from $yz = z$.

Let T be the union of all sets of functions $y \to t$ for $t \in x$. We have $\mathrm{Card}(\mathrm{T}) \leq \sum_x \mathrm{Card}(t^y)$. Note that $\mathrm{Card}(t^y) = \mathrm{Card}(t)^y$, and $\mathrm{Card}(t) \leq z$. Thus $\mathrm{Card}(\mathrm{T}) \leq z^y x \leq x^y$. There is a converse: to any function $f : y \to x$, we associate the restriction of f to t, the ordinal successor of $\sup f$ (note that $t \in x$, since x is a limit ordinal). This mapping is obviously injective. Thus $x^y \leq \mathrm{Card}(\mathrm{T})$, and $x^y = z^y x$.

```
Lemma infinite_power2a n (x:= \aleph (succ_o n)) y f: (* 29 *)
   is_ordinal n -> y <c x -> function_prop f y x ->
   \osup (image_of_fun f) <o x.
Lemma infinite_power2b n (x:= \aleph (succ_o n)) y:  (* 47 *)
  is_ordinal n -> y <c x ->
  x ^c y <=c cardinal (union (fun_image x (set_of_functions y))).
Lemma infinite_power2c n (x:= \aleph (succ_o n)) y:  (* 26 *)
  is_ordinal n -> y <c x -> y <> \0c ->
  cardinal (union (fun_image x (set_of_functions y)))
  <=c ((\aleph n) ^c y) *c x.

Lemma infinite_power2 n m (x:=\aleph n) (y:= \aleph (succ_o n)):
   is_ordinal n -> m <> \0c ->
   y ^c m = (x ^c m) *c y.  (* 32 *)
Lemma infinite_power2_bis x (y:= succ_c x) m:
  infinite_c x ->  m <> \0c ->
   y ^c m = (x ^c m) *c y.
Lemma infinite_power3 n m (x:=\aleph n):
   inc n Bnat ->  m <> \0c ->
   x ^c m = (\2c ^c m) *c x.  (* 22 *)
```

We also deduce: if a is an infinite cardinal, then 2^a is the least infinite cardinal b such that $b^a = b$, and the least such that $b^a < (b^+)^a$. Note that $b^a = b$ holds if $b = 0$, $b = 1$, but for no other integer. On the other hand $b^a < (b+1)^a$ is true if and only if $b = 0$ or $b = 1$ (if $b \geq 2$, either b and $b+1$ are finite, so that $b^a < (b+1)^a = 2^a$, or b is infinite, and $b+1 = b$).

```
Lemma infinite_power4 a (b:= \2c ^c a): infinite_c a -> (* 30 *)
  ( b ^c a =  b & b ^c a <c (succ_c b) ^c a
    & (forall c, \2c <=c c -> c ^c a = c -> b <=c c)
    & (forall c, infinite_c c -> c ^c a <c (succ_c c) ^c a -> b <=c c)).
```

For each ordinal γ we have (Tarski, 1025, [10])

$$\aleph^{\mathfrak{m}}_{\alpha+\gamma} = \aleph^{\mathfrak{m}}_{\alpha} \cdot \aleph^{\mathrm{Card}(\gamma)}_{\alpha+\gamma} \qquad (\mathrm{Card}(\gamma) \leq \mathfrak{m}). \tag{21c}$$

If we replace α by zero and rename γ into α, then (21c) reads:

$$\aleph^{\mathfrak{m}}_{\alpha} = 2^{\mathfrak{m}} \cdot \aleph^{\mathrm{Card}(\alpha)}_{\alpha} \qquad (\mathrm{Card}(\alpha) \leq \mathfrak{m}) \tag{21d}$$

The proof is by transfinite induction on γ. The non-trivial case is when γ is limit. In this case we have $\aleph^{\mathfrak{m}}_{\alpha+\gamma} \leq \prod_{t<\gamma} \aleph^{\mathfrak{m}}_{\alpha+t}$. We apply the induction property, and write the product as

the product of two other factors; there is a trivial bound for the second factor. The LHS is then at most $\aleph_\alpha^{\mathfrak{m}\,\mathrm{Card}(\gamma)} \cdot \aleph_{\alpha+\gamma}^{\mathrm{Card}(\gamma)\,\mathrm{Card}(\gamma)}$. Note that $\mathrm{Card}(\gamma)$ is infinite, and the exponents simplify. We use $\aleph_{\alpha+n}^{\mathfrak{m}} = \aleph_\alpha^{\mathfrak{m}} \cdot \aleph_{\alpha+n}$, which holds for any non-zero integer n, when $\mathfrak{m}$ is infinite.

```
Lemma infinite_power5 n p m (x:=\aleph n) (y:= \aleph (n +o p)):
   is_ordinal n -> is_ordinal p -> is_cardinal m -> m <> \0o ->
   cardinal p <=c m ->
   y ^c m = (x ^c m) *c (y ^c (cardinal p)). (* 154 *)

Lemma infinite_power6  p m (y:= \aleph p): (* 26 *)
  is_ordinal p -> is_cardinal m -> m <> \0o ->  cardinal p <=c m ->
   (infinite_c m \/ p <> \0o) ->
   y ^c m = (\2c ^c m) *c (y ^c (cardinal p)).
Lemma infinite_power6_0  p m (y:= \aleph p):
  is_ordinal p -> infinite_c m -> cardinal p <=c m ->
   y ^c m = (\2c ^c m) *c (y ^c (cardinal p)).
```

Applications

(21e) $$\aleph_\omega^{\aleph_1} = \aleph_\omega^{\aleph_0} \cdot 2^{\aleph_1}, \qquad \aleph_\alpha^{\aleph_1} = \aleph_\alpha^{\aleph_0} \cdot 2^{\aleph_1}, \qquad \aleph_\beta^{\aleph_2} = \aleph_\beta^{\aleph_1} \cdot 2^{\aleph_2} \qquad (\alpha < \omega_1, \beta < \omega_2).$$

$$\aleph_\alpha^{\aleph_\beta} = \aleph_\alpha^{\aleph_0} \cdot 2^{\aleph_\beta} \qquad (\omega \le \alpha < \omega_1)$$

(note that $\mathrm{Card}(\alpha) \le \aleph_0$ and $\mathrm{Card}(\beta) \le \aleph_1$ so that (21c) applies. The result is clear when we have equality (in particular in the last case). Otherwise, we want to prove $a = bc$, and the formula gives $a = b'c$, with $b' \le b$. But this implies $a \le bc$, while $bc \le a$ is trivial.

```
Lemma infinite_power6_1 a: a <o omega1 ->
  (\aleph a) ^c aleph1 =
  (\aleph a) ^c aleph0 *c \2c  ^c aleph1.
Lemma infinite_power6_2 a: a <o omega2 ->
  (\aleph a) ^c aleph2 =
  (\aleph a) ^c aleph1 *c \2c  ^c aleph2.
Lemma infinite_power6_3:
  (\aleph \omega) ^c aleph1 =
  (\aleph \omega) ^c aleph0 *c \2c  ^c aleph1.
Lemma infinite_power6_4 a b:
  infinite_o a -> a <o omega1 -> is_ordinal b ->
  (\aleph a) ^c (\aleph b) =
  (\aleph a) ^c aleph0 *c \2c  ^c (\aleph b).
```

We have

(21f) $$\aleph_{\omega_\alpha} = \tau^{\aleph_\beta} \implies \beta < \alpha; \qquad 2^{\aleph_\omega} = \aleph_{\omega_n} \implies \omega < n.$$

The second relation is a special case of the first. Notice $\tau \ge 2$. Let's compute cofinalities; on one hand this $\mathrm{cf}(\omega_\alpha)$ thus $\le \aleph_\alpha$; on the other hand it is $> \aleph_\beta$, thus $\beta < \alpha$. We deduce that $2^{\aleph_\omega} \ne \aleph_{\omega_n}$, for any integer n. The case $n = 4$ is interesting since one has: $2^{\aleph_\omega} < \aleph_{\omega_4}$, provided that $\aleph_\omega$ is a strong limit cardinal.[2] We also deduce: if $\aleph_{\omega_1}$ has the form x^y, where y is infinite and $x \ge 2$, then y has to be $\aleph_0$.

[2] by a famous result o Shelah; we shall not prove this

```
Lemma infinite_power6_5 a b c :
  is_ordinal a -> is_ordinal b ->
  \aleph (omega_fct a) = c ^c (\aleph b)  ->
  b <o a.
Lemma infinite_power6_6 n : is_ordinal n ->
 \2c ^c (\aleph \omega) = \aleph (omega_fct n) -> \omega <o n.
```

Assume $2^{\aleph_1} = \aleph_2$. Then $\aleph_\omega^{\aleph_1} = \aleph_\omega^{\aleph_0}$. We deduce $\aleph_\omega^{\aleph_0} \neq \aleph_{\omega_1}$, for otherwise we would have $\aleph_\omega^{\aleph_1} = \aleph_{\omega_1}$, contradicting the previous result. Assume $\aleph_\omega^{\aleph_0} > \aleph_{\omega_1}$. Then $\aleph_{\omega_1}^{\aleph_1} = \aleph_\omega^{\aleph_0}$ holds.

Assume $2^{\aleph_0} > \aleph_\omega$ (note that these two quantities cannot be equal, since their cofinalities are respectively $> \omega$ and $= \omega$). Then $\aleph_\omega^{\aleph_0} = 2^{\aleph_0}$. This is obvious.

Assume $2^{\aleph_0} \geq \aleph_{\omega_1}$. Then $\beth(\aleph_\omega) = 2^{\aleph_0}$ and $\beth(\aleph_{\omega_1}) = 2^{\aleph_1}$, where $\beth x = x^{\mathrm{cf}(x)}$ will be introduced later. The formulas hold because the cofinalities are $\aleph_0$ and $\aleph_1$.

```
Lemma infinite_power6_7a: \2c ^c aleph1 = aleph2 ->
 \aleph \omega ^c aleph1 = \aleph \omega ^c aleph0.
Lemma infinite_power6_7b:
  \2c ^c aleph1 = aleph2 ->
  (\aleph \omega) ^c aleph0 <> \aleph omega1.
```

```
Lemma infinite_power6_7a: \2c ^c aleph1 = aleph2 ->
  omega_fct \omega ^c omega_fct \1o = omega_fct \omega ^c omega_fct \0o.
Lemma infinite_power6_7b:
  \2c ^c (omega_fct \1o) = omega_fct \2o ->
  (omega_fct \omega) ^c (omega_fct \0o) <> omega_fct (omega_fct \1o).
Lemma infinite_power6_7c:
  \2c ^c (omega_fct \1o) = omega_fct \2o ->
   omega_fct (omega_fct \1o) <c (omega_fct \omega) ^c (omega_fct \0o) ->
  omega_fct (omega_fct \1o) ^c (omega_fct \1o)
    = (omega_fct \omega) ^c (omega_fct \0o).
Lemma infinite_power6_7d:
  omega_fct \omega <c \2c ^c \omega ->
  (omega_fct \omega) ^c \omega = \2c ^c \omega.
Lemma infinite_power6_7e (gimel_fct := fun x => x ^c (cofinality_c x)):
   \aleph omega1  <=c \2c ^c \omega ->
   ( gimel (\aleph \omega) = \2c ^c aleph0
   & gimel (\aleph omega1) = \2c ^c aleph1).
```

Consider an increasing sequence of cardinals x_i, and their product. We have shown that the product is the supremum raised to the power of the index set I, provided that this set I is an initial ordinal. We consider here the case where all x_i are infinite, so are of the form $\aleph_{\sigma_x i}$. We let α be the supremum of the $\sigma_x i$. The sequence of cardinals is increasing, whenever σ is an increasing function. Assume $\sup \aleph_{\sigma_x i} = \aleph_\alpha$, this will be called (H). If the index set is ω_β we get:

$$\aleph_\alpha^{\aleph_\beta} = \prod_{\xi<\omega_\beta} \aleph_{\sigma_\xi}. \tag{21-0}$$

We first establish two small lemmas; the first says that (H) holds if α is not in the range of σ (and the index set is a non-zero ordinal). The second lemma says that (H) holds if the sequence is stricyly increasing and the index set is a limit ordinal).

Note that (H) holds if α is a limit ordinal, by normality of $\aleph_i$. Note that α is a limit ordinal if α is not in the range of σ. This condition holds if σ is strictly increasing, and the index set is a limit ordinal.

```
Definition ord_fam_increasing X :=
  fgraph X &
  (forall u v, inc u (domain X) -> inc v (domain X) -> u <=o v ->
     V u X <=o V v X).
Definition ord_fam_increasing_strict X :=
  fgraph X &
  (forall u v, inc u (domain X) -> inc v (domain X) -> u <o v -> V u X <o V v X).


Lemma increasing_sup_limit1 X (a:= \osup (range X)) (b := domain X):
  ord_fam_increasing X -> is_ordinal b -> b <> \0c ->
  (forall t, inc t (domain X) -> V t X <> a)
  -> limit_ordinal a.
Lemma increasing_strict_weak X (b := domain X):
  ord_fam_increasing_strict  X -> limit_ordinal b ->
  ord_fam_increasing  X.
Lemma increasing_sup_limit2 X (a:= \osup (range X)) (b := domain X):
  ord_fam_increasing_strict  X -> limit_ordinal b ->
  limit_ordinal a.
Lemma  increasing_sup_limit3 X b:
  ord_fam_increasing X ->
  is_ordinal b -> domain X = omega_fct b ->
  limit_ordinal  (\osup (range X)) ->
  ((forall u, inc u (domain X) -> is_ordinal (V u X)) &
    \osup (range (L (omega_fct b) (fun z : Set => \aleph (V z X)))) =
    \aleph  (\osup (range X))). (* 32 *)
```

We give two versions of the theorem, in the first case, the sequence is increasing, in the second case, it is strictly increasing.

```
Lemma infinite_increasing_power1 X b:
  ord_fam_increasing X ->
  is_ordinal b -> domain X = omega_fct b ->
  limit_ordinal  (\osup (range X)) ->
  card_prod (L (domain X) (fun z => \aleph (V z X))) =
  \aleph (\osup (range X)) ^c \aleph b.
Lemma infinite_increasing_power2 X b:
  ord_fam_increasing_strict X ->
  is_ordinal b -> domain X = omega_fct b ->
  card_prod (L (domain X) (fun z => \aleph (V z X))) =
  \aleph (\osup (range X)) ^c \aleph b.
```

We give here another proof. Let B be an infinite cardinal, X and Y two sets with cardinal < B. Then there is some x in B, which is neither in X nor in Y (because the cardinal of X ∪ Y is < B). Assume that x and y are some ordinals < B, f is a function with source y. Let Y be the image of f. Then $Card(Y) \leq Card(y)$. Now x and y have cardinal < B. Thus we deduce: there exists an element z of B, that is not in the range of f and $z \geq x$ (this last condition is not $z < x$, thus $z \notin x$). Let g be any function B → B. Definition by transfinite induction a function f, such that $f(x)$ is the least element z such that $z \geq f(x)$ and z is not in the range of the restriction of g to the interval $]\leftarrow, x[$.

Let σ_ξ and α be as above. Let f be any function $\omega_\beta \to \omega_\alpha$. Fix $t \in \omega_\beta$. Then $f(t) < \omega_\alpha$. This implies that there is some x such that $f(t) \leq \omega_u$ and $u < \alpha$. Now there is ξ such that $u \leq \sigma_x i$. Let g be the function $t \mapsto u$. This function satisfies $f(t) \leq \omega_{\sigma_{g(t)}}$. We may assume g injective by the previous argument.

The result follows.

```
Lemma infinite_union2 x y z:
    infinite_c z -> cardinal x <c z -> cardinal y <c z ->
    nonempty (complement z (union2 x y)).  (*  29 *)
Lemma cofinality_pr6 a f (b:= omega_fct a):
   is_ordinal a ->
   function_prop f b b -> exists g,
   function_prop g b b & injection g &
   (forall x, inc x b ->  W x f <=o W x g).  (* 123 *)
Lemma cofinality_pr7 X b f: (* 62 *)
  ord_fam_increasing_strict X ->
  is_ordinal b -> domain X = omega_fct b ->
  inc f (set_of_functions (omega_fct b) (omega_fct  (union (range X)))) ->
  exists g, inc g (set_of_functions (omega_fct b) (omega_fct b)) &
     injection g &
    (forall x, inc x (source f) ->  W x f <=o omega_fct (V (W x g) X)).
```

The Tarski conjecture is that

$$\prod_{\xi<\beta} \aleph_{\sigma_\xi} = \aleph_\alpha^{\mathrm{Card}\beta} \tag{C}$$

holds whenever σ_ξ is an increasing family of ordinals, indexed by a limit ordinal β such that $\sigma_\xi < \alpha$ and $\sup_{\xi<\beta} \sigma_\xi = \alpha$. The formula holds when β is countable, and when β has Card(β) disjoint cofinal subsets. It holds if $\beta < \omega_1 + \omega$. It holds if SCH holds (this is an assumption weaker than GCH, it will be studied later on). One can prove that if it fails for some β, then it fails for $\omega_1 + \omega$.

Example (Josh and Shelah). Consider

$$p = \prod_{\xi<\omega_1} \aleph_\xi \prod_{n<\omega} \aleph_{\gamma+n}$$

We have

$$p = \aleph_{\omega_1}^{\aleph_1} \cdot \aleph_{\gamma+\omega}^{\aleph_0} \leq \aleph_{\gamma+\omega}^{\mathrm{Card}(\omega_1+\omega)} = \aleph_{\gamma+\omega}^{\aleph_1}. \tag{C'}$$

The conjecture says that we have equality. We have a counter-example if

$$\aleph_{\omega_1}^{\aleph_1} < \aleph_{\gamma+\omega}^{\aleph_1}, \qquad \aleph_{\gamma+\omega}^{\aleph_0} < \aleph_{\gamma+\omega}^{\aleph_1}.$$

Let $Y = \aleph_{\gamma+\omega}$, $Z = \aleph_\gamma$, $a = \aleph_0$ and $b = \aleph_1$. The Tarski formula says $Y^b = Z^b Y^a$. The second condition can be restated as $Y^a < Z^b$. Let $X = \aleph_{\omega_1}$. The first condition is equivalent to $X^b < Z^b$. It implies $X^b < Z$. A non-trivial result is that, if (C) fails, one can chose γ so that Z is singular with cofinality $\aleph_1$.

If $\gamma = \omega_1$ then p is the product of all infinite cardinals $\aleph_\alpha$ with $\alpha < \omega_1 + \omega$, and there is equality in (C'). In order to pove this, we start with some helper lemmas. The first one says that the supremum of all $\aleph_{\gamma+n}$ is $\aleph_{\gamma+\omega}$. (the composition of two normal function is a normal

function). We also state that an infinite product remains unchanged if we replace one factor x_i by x'_i provided there is another infinite factor x_j such that x_i and x'_i are both $\leq x_j$. This lemma will be used as follows: when we compute the product of all $\aleph^m_{a+n}$, (for finite n) we can ignore the case $n=0$, and apply (21c). Then $\aleph^n_{a+n} = \aleph_{a+n}$.

```
Lemma cardinal_omega2: cardinal \omega = cardinal (\omega +o \omega).
Lemma ord_sup_aleph_sum x: is_ordinal x ->
  \csup (range (L \omega (fun z : Set => \aleph (x +o z)))) =
    \aleph (x +o \omega).
Lemma ord_sup_aleph x: limit_ordinal x ->
  \csup (range (L x \aleph)) = \aleph x.
Lemma cprod_inf_eq x y i:
   fgraph x -> fgraph y -> (domain x = domain y) ->
   inc i (domain x) ->  (V i x) <> \0c -> V i y <> \0c ->
   (exists j, inc j (domain x) & j <> i & infinite_c (V j x) &
     V i x <=c V j x & V i y <=c V j x) ->
   (forall j, inc j (domain x) -> j = i \/ V j x = V j y) ->
  card_prod x = card_prod y. (* 56 *)
```

We have

$$\text{(21g)} \qquad \prod_{0<n<\omega} n = 2^{\aleph_0}; \quad \prod_{n<\omega} \aleph_n = \aleph_\omega^{\aleph_0}; \qquad \prod_{\alpha<\omega+\omega} \aleph_\alpha = \aleph_{\omega+\omega}^{\aleph_0}; \qquad \prod_{\alpha<\omega_1+\omega} \aleph_\alpha = \aleph_{\omega_1+\omega}^{\aleph_1}.$$

A possible proof for the first formula is: we first notice that $0<n<\omega$ can be replaced by $2\leq n<\omega$ so hat each factor x satisfies $2\leq x\leq\omega$ so that $2^\omega\leq p\leq\omega^\omega=2^\omega$.

In the first two cases, the product is s^{I}, where s is the supremum and I the cardinal of the index set. For the first formula, we have $s^{\mathrm{I}}=2^{\mathrm{I}}$. In the two other cases, the product is over all $\alpha<a+b$; we split the product as q_1q_2, and get $q=\aleph_a^a\aleph_{a+b}^b$. The result is trivial if $a=b$; but non-trivial if $a=\omega_1$ and $b=\omega$.

We have $\aleph^a_{a+b}=q_2=q_2^a=\prod\aleph^a_{a+n}=\prod\aleph^a_a\aleph_{a+n}=(\aleph^a_a)^b\prod\aleph_{a+n}$ (we use (21c) for non-zero n, this trick works only for $b=\omega$). This is $q_1^bq_2=q_1q_2=q$.

```
Lemma infinite_prod_pA:  (* 26 *)
  card_prod (L Bnat succ) = card_prod (L Bnat (fun z => (succ (succ z)))).

Lemma infinite_prod_pB1: union (range (L Bnat succ)) = \omega.
Lemma cprod_An1 a b f: is_ordinal a -> is_ordinal b ->
  card_prod (L (a +o b) f) =
   card_prod (L a f) *c card_prod (L b (fun z => (f (a +o z)))). (* 55 *)


Lemma infinite_prod_pB2:  card_prod (L Bnat succ) = \2c ^c \omega.
Lemma infinite_prod_pB:  card_prod (L Bnat succ) = \2c ^c \omega.
Lemma infinite_prod_pC:
   card_prod (L Bnat \aleph) = (\aleph \omega) ^c \omega.
Lemma infinite_prod_pD (o2 := \omega +o \omega):
   card_prod (L o2 \aleph) = \aleph2 ^c aleph0. (* 32 *)
Lemma infinite_prod_pE (o2 := omega1 +o \omega): (* 102 *)
   card_prod (L o2 \aleph) = (omega_fct o2) ^c aleph1.
```

We consider here a strictly increasing ordinal family σ_ξ, indexed by a limit ordinal β. Let α be the supremum of the family. We have obviously $\sigma_\xi<\alpha$, so that α is a limit ordinal. Let

$\tau_\xi = \aleph_{\sigma_\xi}$ for simplicity. This is a strictly increasing cardinal family. Its supremum is $\aleph_\alpha$. A variant of (20c) shows that $\aleph_\alpha = \sum \tau_\xi$.

Assume now $\beta = \omega^\gamma$, with $\gamma > 0$. We have

$$\prod_{\xi<\omega^\gamma} \aleph_{\sigma_\xi} = \aleph_\alpha^{\mathrm{Card}(\omega^\gamma)} \tag{21h}$$

Let A be the product and B the power. Relation $A \le B$ is trivial. Let $C = \prod \tau_i^{\mathrm{Card}(\beta)}$. By application of (20c) we have $B \le C$. Let $f : (i, j) \to \tau_i$ be defined on $\beta \times \beta$. By associativity, $\prod f = C$. To each $x \in \beta$ we associate the set $G_x \subset \beta \times \beta$ of all (u, v) such that $u \sharp v = x$. We use the following three properties of the natural sum $u \sharp v$. First, if u and v are in β, so is $u \sharp \beta$, so that G_x is a partition of $\beta \times \beta$; secondly G_x is a finite nonempty set (these properties hold because β is a power of ω). We use again associativity and get $C = \prod_x p_x$. p_x is the product of all τ_u such that $(u, v) \in G_x$. The last property of the natural sum is $u \le x$, so that $\tau_u \le \tau_x$ and $p_x \le \tau_x^{\mathrm{Card}(G_x)}$, Since G_x is non-empty finite, one deduces $p_x \le \tau_x$, thus $C \le A$.

```
Definition ordinal_fam_si X b:=
  ordinal_fam X &  (domain X = b) &
  (forall u v, inc u (domain X) -> inc v (domain X) ->
    u <o v -> V u X <o V v X).

Lemma infinite_prod1 X  b (Y:= L (domain X) (fun z => \aleph (V z X)))
  (a := \osup (range X)):
  limit_ordinal b ->  ordinal_fam_si X b ->
  (limit_ordinal a &  sub (range X) a &\csup (range Y) = \aleph a).  (* 40 *)
Lemma infinite_prod2 X  b (Y:= L (domain X) (fun z => \aleph (V z X)))
  (a := \osup (range X)):
  limit_ordinal b ->  ordinal_fam_si X b ->
   \aleph a = card_sum Y.   (* 25 *)
Lemma infinite_prod3 X c  (Y:= L (domain X) (fun z => \aleph (V z X)))
   (b:= \omega ^o c):
   is_ordinal c -> c <> \0o ->  ordinal_fam_si X b ->
  (card_prod Y) = (\csup (range Y)) ^c (cardinal b). (* 101 *)
```

In [10, th. 7], Tarski deduces how to compute τ^λ. He assumes that $\tau = \aleph_\alpha$ for some limit ordinal α. If $\lambda < \mathrm{cf}(\alpha)$, he says $\tau^\lambda = \sum_{\xi<\alpha} \aleph_\xi^\lambda$. This is relation (22b) below, with a sum instead of a supremum (but, as noted, the sum is the supremum). In the other case, τ^λ is the product of some $\aleph_{\sigma_\xi}^\lambda$, the index set being $\mathrm{cf}(\alpha)$.

$$\prod_{\xi<\mathrm{cf}(\alpha)} \aleph_{\sigma_\xi}^{\aleph_\beta} = \aleph_\alpha^{\aleph_\beta} \qquad (\alpha = \sup \sigma_\xi, \quad \aleph_\beta \ge \mathrm{cf}(\aleph_\alpha)). \tag{21i}$$

Let I be the domain of σ_ξ. The trick is that this is $I = \omega^\gamma$, for some ordinal γ. We can restate it as: I is an indecomposable ordinal. The assumption is that I is a cofinality, thus is an infinite cardinal. Assume $a <_{\mathrm{ord}} I$ and $b <_{\mathrm{ord}} I$. We have $\mathrm{Card}(a) < \mathrm{Card} I$. Since $\mathrm{Card}(a +_{\mathrm{ord}} b) = \mathrm{Card}(a) +_{\mathrm{Card}} \mathrm{Card}(b) = \sup(\mathrm{Card}(a), \mathrm{Card}(b)) <_{\mathrm{Card}} I$, we get $a +_{\mathrm{ord}} b \ne I$ and I is indecomposable.

Write $\tau = \aleph_\alpha$, $\lambda = \aleph_\beta$, and let I be the domain of σ_ξ. By (21h), the product is $\tau^{I.\lambda}$, and $I.\lambda = \lambda$, since $\lambda \ge I$. Note that $I = \mathrm{cf}(\alpha)$ is the least possible domain of a sequence σ_ξ whose supremum is α.

```
Lemma cardinal_indecomposable x: infinite_c x ->
```

```
   ord_indecomposable x.
Lemma cardinal_indecomposable2 x: infinite_c x ->
  (exists n, is_ordinal n & n <> \0o & x = \omega ^o n).
Lemma infinite_prod4 X a b
  (Y:= L (domain X) (fun z => \aleph (V z X) ^c \aleph b))
  (c := cofinality a):
  limit_ordinal a -> c <=c \aleph b ->
  is_ordinal b ->  ordinal_fam_si X c -> (a = \osup (range X)) ->
  (card_prod Y) = \aleph a ^c (\aleph b).
```

Examples

$$\aleph_\omega^{\aleph_0} = \prod_{0<\xi<\omega} \xi, \qquad \aleph_\omega^{\aleph_\beta} = \prod_{\xi<\omega} \aleph_\xi^{\aleph_\beta}$$

(21j) $$\aleph_{\omega_1}^{\aleph_0} = \sum_{\xi<\omega_1} \aleph_\xi^{\aleph_0}, \qquad \aleph_{\omega_1}^{\aleph_1} = \prod_{\xi<\omega_1} \aleph_\xi, \qquad \aleph_{\omega_1}^{\aleph_\beta} = \prod_{\xi<\omega_1} \aleph_\xi^{\aleph_\beta} \quad (\beta \geq 1).$$

The first relation is the first of (21g). The third relation is a consequence of (22b) and will be proved later.

```
Lemma infinite_prod_o1 (X :=  L omega1 (fun z => z)):
   (exists n, is_ordinal n & n <> \0o & omega1 = \omega ^o n &
   ordinal_fam_si X omega1 & (\osup (range X) = omega1)).
Lemma infinite_prod_pF:
   card_prod (L omega1 \aleph) = (\aleph omega1) ^c aleph1.
Lemma infinite_prod_pG b: \1o <=o b ->
   card_prod (L omega1 (fun z => (\aleph z) ^c (\aleph b)))
   = (\aleph omega1) ^c (\aleph b).
Lemma infinite_prod_pH b: is_ordinal b ->
   card_prod (L \omega (fun z => (\aleph z) ^c (\aleph b)))
   = (\aleph \omega) ^c (\aleph b).
```

We have

(21k) $$\prod_{\xi\leq\beta} \aleph_\xi = \aleph_\beta^{\mathrm{Card}\,\beta}, \qquad \prod_{\xi<\alpha} \aleph_\xi = \aleph_\alpha^{\mathrm{Card}\,\alpha}, \qquad (\alpha \text{ limit}; \beta > 0).$$

Let's start with the second formula. We can write $\alpha = \beta + \omega^n$, The case $\beta = 0$ follows from (21h). Otherwise, we know that β is a limit ordinal, that has the same cardinal as α. We proceed as in the proof of (21g); the product is AB, where A is the product of all $\aleph_\xi$, with $\xi < \beta$, and B is the product of all as two products, the first over all $\xi < \beta$, $\aleph_{\beta_\xi}$ with $\xi < \omega^n$. We may assume by induction, that the relation holds for β. This allows us to evaluate A. The second factor can be evaluated by (21h), so that the product becomes $\aleph_\beta^{\mathrm{Card}(\alpha)} \aleph_\alpha^{\mathrm{Card}(\omega^n)}$. The conclusion follows by using rewrite (21c) with $\alpha = \beta + \omega^n$.

Denote by p_β the first product, and by q_α the second. We have $p_\beta = q_\beta \aleph_\beta$. If β is a limit ordinal, we have $q_\beta = \aleph_\beta^{\mathrm{Card}(\beta)}$, this is at least $\aleph_\beta$, so that $p_\beta = \aleph_\beta^{\mathrm{Card}(\beta)}$. This show the first formula in the case where β is limit. Note that $p_1 = \aleph_0 \aleph_1 = \aleph_1$. This shows the first formula in the case $\beta = 1$. By transfinite induction, it suffices to show that if the formula holds for β it holds for its successor. This follows by (21c). The non-trivial point is to show that $\aleph_\beta^{Card(\beta)} = \aleph_\beta^{Card(\beta+1)}$ which holds since β is non-zero and $\aleph_\beta$ is infinite.

```
Lemma infinite_prod5 a:
  (limit_ordinal a) ->
  card_prod (L a \aleph) = (\aleph a) ^c (cardinal a). (* 45 *)
Lemma infinite_prod6 a:
  is_ordinal a -> a <> \0o ->
  card_prod (L (succ_o a) \aleph) = (\aleph a) ^c (cardinal a). (* 56 *)
```

8.18 Infinite powers

We may consider the supremum of x^y when for $x \leq z$ or $y \leq z$. We start with the case where the exponent is fixed. The result depends on how the exponent compares to the cofinality of z.

Assume that x is an infinite cardinal $y < \mathrm{cf}(x)$. Then any function $f : y \to x$ is bounded (we have already shown this in the case where $x = \aleph_{n+1}$ and $y < x$). It follows that x^y can be identified to a subset of all functions $y \to t$, for $t \in x$. The cardinal of x^y is thus $\leq \sum \mathrm{Card}(t^y)$. Let $z = \mathrm{Card}(t)$ so that $\mathrm{Card}(t^y) = z^y$. For each cardinal z there are at most x ordinals t satisfying $z = \mathrm{Card}(t)$. It follows

$$\kappa^\lambda = \kappa \sum_{\tau<\kappa} \tau^\lambda \qquad (\omega < \kappa, 0 < \lambda < \mathrm{cf}(\kappa)). \tag{22a}$$

Assume $\kappa = \aleph_\alpha$ where α is a limit ordinal. In this case, the supremum of the τ^λ is at least the supremum of the τ, thus κ. This means that the sum is equal to the supremum, and we do not need the factor κ. Since moreover $\mathrm{cf}(\kappa) = \mathrm{cf}(\alpha)$, we have

$$\aleph_\alpha^\lambda = \sup_{\xi<\alpha} \aleph_\xi^\lambda \qquad (\alpha \text{ limit}, \lambda < \mathrm{cf}(\alpha)). \tag{22b}$$

```
Lemma infinite_power7a x y f: infinite_c x ->  (* 26 *)
    y <c cofinality_c x -> function_prop f y x ->
   \osup (image_of_fun f) <o x.
Lemma infinite_power7b x y:
  infinite_c x -> y <c cofinality_c x ->
  x ^c y <=c cardinal (union (fun_image x (set_of_functions y))). (* 47 *)
Lemma infinite_power7c x y:  (* 54 *)
  is_cardinal x -> is_cardinal y ->
  cardinal (union (fun_image x (set_of_functions y)))
  <=c (card_sum (L (set_of_cardinals_lt x) (fun z => z ^c y))) *c x.
Lemma infinite_power7d x (y := cardinal (set_of_cardinals_lt x)):
   is_cardinal x -> x <> \0c ->
   (y <=c x & y <> \0c).
Lemma infinite_power7 x y: (* 30 *)
  infinite_c x -> y <c cofinality_c x -> y <> \0c ->
  x ^c y =  (card_sum (L (set_of_cardinals_lt x) (fun z => z ^c y))) *c x.
Lemma infinite_power7e n y:
  limit_ordinal n -> is_cardinal y -> y <> \0c ->
  \aleph n <=c \osup (fun_image n (fun z => (\aleph z) ^c y)).
Lemma infinite_power7f n y:
  limit_ordinal n -> y <c cofinality n ->
  (\aleph n) ^c y = \osup (fun_image n (fun z => (\aleph z) ^c y)).
```

One has

$$\kappa^\lambda = \left[\sup_{\alpha<\kappa} \alpha^\lambda\right]^{\mathrm{cf}(\kappa)} \qquad (\lambda \geq \mathrm{cf}(\kappa)). \tag{22c}$$

Assume $\kappa = \sum_{i\in I} x_i$, with $x_i < \kappa$. One may assume $x_i \neq 0$ so that $\kappa \leq \prod x_i$ and $\kappa^\lambda \leq \prod x_i^\lambda$. Let $S = \sup \alpha^\lambda$ so that $x_i^\lambda \leq S$ and $\kappa^\lambda \leq S^I$. Note that, if $\lambda \geq \kappa$, then $\alpha^\lambda = 2^\lambda = S$, and the result is trivial. Thus, the formula is interesting when $\mathrm{cf}(\kappa) \leq \lambda < \kappa$ (case where κ is singular).

```
Lemma infinite_power8 n (x:= \aleph n) (z:= cofinality_c x) y:
  is_ordinal n -> z <=c y ->
  x ^c y = (\osup (fun_image (set_of_cardinals_lt x) (fun t => t ^c y))) ^c z.
```

Give two infinite cardinals, one has

$$x^y = \begin{cases} 2^y & \text{if } x \leq y \\ z^y & \text{if } z < x \text{ and } x \leq z^y \\ x & \text{if } (\forall z, z < x \implies z^y < x) \text{ and } y < \mathrm{cf}(x) \\ x^{\mathrm{cf}(x)} & \text{if } (\forall z, z < x \implies z^y < x) \text{ and } y \geq \mathrm{cf}(x) \end{cases} \tag{22d}$$

Proof. The first two cases are obvious. Taking $z = 2$ shows $y < x$. This excludes the case $x = \aleph_0$. Assume that $x = \aleph_{n+1}$ is a cardinal successor, thus is regular. Write $x = t^+$; so that (222b) reads $x^y = t^y x$. By assumption $t^y < x$ so that $x^y = x$. Consider finally the case where $x = \aleph_n$ where n is a limit ordinal. If $y < \mathrm{cf}(x)$, the result follows from (22b); otherwise from (22a).

```
Lemma infinite_power9 x y: infinite_c x -> infinite_c y -> (* 65 *)
  ( (x <=c y ->  x ^c y = \2c ^c y)
  & (forall  z, z <c x -> x <=c z ^c y -> x ^c y = z ^c y)
  & (forall z,  z <c x -> z ^c  y <c x) ->
     ( ( y <c cofinality_c  x  -> x ^c y = x)
       & (cofinality_c x <=c y -> x ^c y = x ^c (cofinality_c x)))).
```

Relation (22a) holds for any $\lambda \geq \kappa$, since all powers are 2^λ. We deduce: if κ is a regular cardinal, then (22a) holds for any non-zero λ;

```
Lemma infinite_power7g x y: (* 51 *)
  infinite_c x -> x <=c y ->
  x ^c y =  (card_sum (L (set_of_cardinals_lt x) (fun z => z ^c y))) *c x.
Lemma infinite_power7h x y:
  regular_cardinal x -> \0c <c y ->
  x ^c y =  (card_sum (L (set_of_cardinals_lt x) (fun z => z ^c y))) *c x.
```

Assume that x is infinite. There is y such that $x < y$ and $y^x = y$. It suffices to take $y = 2^x$. There is y such that $x < y$ and $y^x > y$. We may proceed as follows. Let z be the cardinal successor of x, say $x = \aleph_n$ and $z = \aleph_{n+1}$. Note that z is not a fixed point of ω. Let y be the least fixed point of ω that is $\geq z$. Its cofinality is $\omega \leq x$. We have $y < y^{\mathrm{cf}(y)} \leq y^x$.

```
Lemma infinite_power6w y: infinite_c y -> (* 31 *)
  ( (exists x, y <c x & x ^c y = x) &
    (exists x, y <c x & x <c x ^c y)).
```

Consider

$$x^{<y} = \sup_{z <_{\mathrm{card}} y} x^z = \sum_{z <_{\mathrm{card}} y} x^z.$$

The first equality here is the definition. The second holds if $x \geq 2$ and y infinite. (Let c be the cardinal of all z such that $z < y$, we have $c \leq y$; since $x^s > s$ by Cantor, we deduce $c \leq s$, where

s is the supremum, so that the sum is the supremum). Note that $x^{<0} = 0$, $x^{<1} = 1$, $1^{<y} = 1$, $0^{<y} = 1$ (if $y > 0$). We shall ignore these trivial cases. If x and y are finite then $x^{<y}$ is finite. If x is infinite and y finite non-zero, then $x^{<y} = x$. In the trivial case where y is the successor of z we have $x^{<y} = x^z$.

```
Definition cpow_less x y :=
  \csup (fun_image (set_of_cardinals_lt y) (fun t => x ^c t)).

Notation "x ^<c y" := (cpow_less x y) (at level 30).

Lemma cpow_less_alt x y :
  infinite_c y -> \2c <=c x ->
  x ^<c y = card_sum (L (set_of_cardinals_lt y) (fun t => x ^c t)).
Lemma cpow_less_pr0 x y:
  cardinal_set (fun_image (set_of_cardinals_lt y) (fun t => x ^c t)).
Lemma CS_cpow_less x y: is_cardinal (x ^<c y).
Lemma cpow_less_pr1 x y:  \0c <c x -> is_cardinal y -> x ^<c y <=c x ^c y.
Lemma cpow_less_pr2 x y z: z <c y -> x ^c z <=c (x ^<c y).
Lemma cpow_less_pr3 x y: \0c <c x -> inc y Bnat ->
  x ^<c (succ y) = x ^c y.
Lemma cpow_less_pr4 x y: \0c <c x -> infinite_c y ->
  x ^<c (succ_c y) = x ^c y.
```

We have

$$\kappa \leq 2^{<\kappa} \leq \kappa^{<\kappa} \leq \kappa^{\kappa} \qquad (\omega \leq \kappa).$$

$$2^{\kappa} = (2^{<\kappa})^{\mathrm{cf}(\kappa)} \qquad (\omega \leq \kappa).$$

In fact, if $\kappa = \sum x_i$, then $2^{\kappa} = \prod 2^{x_i} \leq \prod 2^{<\kappa} = (2^{<\kappa})^{\mathrm{cf}(\kappa)} \leq (2^{\kappa})^{\mathrm{cf}(\kappa)} = 2^{\kappa}$. If $x = \omega$, then $2^{<x}$ and $\omega^{<x}$ are both equal to x.

Let $s(x)$ be the sum of all x^z for $z \leq x$ and $s'(x)$ the sum for $z < x$. In the case where x is a successor, say $x = y^+$ we have $s'(x) = 2^y$. This is because $s'(x) = x^{<x} = x^y$; the result follows from $x \leq 2^y$. In any case $s(x) = 2^x$, since $s(x) = x^{<x} + x^x$.

```
Lemma cpow_less_compare x: infinite_c x ->
  (x <=c \2c ^<c x & \2c ^<c x <=c x ^<c x & x ^<c x <=c x ^c x).
Lemma cpow_less_pr5 x: infinite_c x ->
  \2c ^c x = (\2c ^<c x ) ^c (cofinality_c x).
Lemma cpow_less_pr7 : \2c ^<c  \omega = \omega.
Lemma cpow_less_pr7b (x := \omega): x = x ^<c x.
Lemma cpow_less_pr10 x: infinite_c x -> (* 32 *)
  card_sum (L (set_of_cardinals_le x) (fun t => x ^c t)) = \2c ^c x.
Lemma cpow_less_pr11 x (y := succ_c x) : infinite_c x ->
  card_sum (L (set_of_cardinals_lt y) (fun t => y ^c t)) = \2c ^c x.
```

We say that the power function x^y is is "eventually constant below z" if there exists t such that $t \leq y < z$ implies $x^y = x^t$. This implies $x^{<z} = x^t$. We say that the "continuum function is eventually constant below z" if there exists t such that $t \leq y < z$ implies $2^y = 2^t$. This implies $2^{<z} = 2^t$. If x is singular, and the continuum function is eventually constant below x, then $2^x = 2^{<x}$. We may assume $\mathrm{cf}(x) \leq t$, so that $2^{<x} = 2^t$; we have $2^x = (2^{<x})^{\mathrm{cf}(x)} = 2^{t.\mathrm{cf}(x)}$. Since x is singular, we may assume $\mathrm{cf}(x) \leq t$, so that $t.\mathrm{cf}(x) = t$.

```
Definition cpow_less_ecb x :=
```

```
  (exists a, a <c x & forall b, a <=c b -> b <c x -> \2c ^c a = \2c ^c b).
Lemma cpow_less_pr6 x:  singular_cardinal x -> cpow_less_ecb x ->
  -> \2c ^c x = \2c ^<c x.  (* 26 *)
```

Assume $2^{\aleph_\alpha} = \aleph_{\alpha+\beta}$ for any ordinal α. Note that this is GCH if $\beta = 1$. Then β is finite. In effect, consider the least α such that $\alpha + \beta < \beta$. Then $0 < \alpha \leq \beta$ and α is a limit ordinal. Let $x = \aleph_{\alpha+\beta}$. If $\xi \leq \alpha$, then $2^{\aleph_\xi} \leq \aleph_{\xi+\beta} \leq \aleph_{\alpha+\beta} = x$. Assume $\alpha \leq \xi \leq \alpha + \alpha$ so that $\xi = \alpha + \lambda$ with $0 \leq \lambda < \alpha$. By minimality of α, $\lambda + \beta = \beta$. By assumption $2^\xi = \aleph_{\alpha+\lambda+\beta} = \aleph_{\alpha+\beta} = x$. Thus $2^{<\alpha+\alpha} = x$, and the supremum is strict. Let $\kappa = \aleph_{\alpha+\alpha}$. Note that $\mathrm{cf}(\alpha+\alpha) = \mathrm{cf}(\alpha)$, so that $\mathrm{cf}(\kappa) < \kappa$ and κ is a singular cardinal. We may apply lemma cpow_less_pr6. It says $2^\kappa = 2^{<\kappa}$; and this quantity is x. But the assumption says that is is $\aleph_{\alpha+\alpha+\beta}$, which is greater than x, absurd.

```
Lemma genconthypothesis_alt b: is_ordinal b -> (* 80 *)
  (forall a, is_ordinal a -> \2c ^c (omega_fct a) = omega_fct (a +o b)) ->
  b <o \omega.
```

We denote by $\beth\kappa$ the quantity $\kappa^{\mathrm{cf}(\kappa)}$. The behavior of 2^κ is uniquely determined by $\beth\kappa$. Let $p(\kappa)$ be the property that $2^{<\kappa}$ is some 2^μ with $\mu < \kappa$. We have:

$$2^\kappa = \begin{cases} \beth\kappa & \text{if } \kappa \text{ is a successor} \\ 2^{<\kappa} \cdot \beth\kappa & \text{if } \kappa \text{ is limit and } p \text{ holds} \\ \beth 2^{<\kappa} & \text{otherwise} \end{cases}$$

Note that, if κ is regular (for instance if it is a successor) then $\beth\kappa = \kappa^\kappa = 2^\kappa$. If κ is limit, regular and p holds, then $2^{<\kappa} \leq 2^\kappa = \beth\kappa$; but if κ is singular and p holds, we have $2^{<\kappa} = 2^\kappa$; we conclude by $\beth\kappa \leq 2^\kappa$. In the last case, $2^\kappa = (2^{<\kappa})^{\mathrm{cf}(\kappa)}$ and all we need to show is that κ and $2^{<\kappa}$ have the same cofinality. Let $f : \mathrm{cf}(x) \to x$ be a cofinal function, define $g(t) = 2^{f(t)}$. By assumption $2^y < 2^{<x}$ whenever $y < x$, so that g is a function $\mathrm{cf}(x) \to 2^{<x}$. It is cofinal: let t be any ordinal such that $t <_{\mathrm{ord}} 2^{<x}$. We pretend that there are cardinals $\bar{t}$ and u such that $t \leq_{\mathrm{ord}} \bar{t}$, $u < x$ and $\bar{t} \leq 2^u$, from which $t \leq g(u)$ follows. If t is finite, we choose $\bar{t} = \omega$ and the result is clear. We pretend that $2^{<x}$ is not the successor of u: since 2^w is never $2^{<x}$, we would have $2^w \leq u$, thus $2^{<x} \leq u$, absurd. If t is a cardinal, we choose $\bar{t} = t$; so that $\bar{t} < 2^{<x}$. In the case where t is not a cardinal, we choose for $\bar{t}$ the cardinal successor of the cardinal of t. The same relation holds. Obviously $t \leq_{\mathrm{ord}} \bar{t}$.

Conversely, let $g : \mathrm{cf}(\lambda) \to \lambda$, where $\lambda = 2^{<x}$. Let $f(t)$ be such that $g(t) \leq 2^{f(t)}$ and $f(t) < x$. This exists, according to the previous discussion. Let u be the supremum of f. Then 2^u is a supremum of g.

```
Definition gimel_fct x := x ^c (cofinality_c x).

Lemma gimel_prop1 x: regular_cardinal x ->
  \2c ^c x = gimel_fct x.
Lemma gimel_prop2 x: infinite_c x ->
  gimel_fct x <=c \2c ^c x.
Lemma gimel_prop n (x:= omega_fct n): is_ordinal n ->
 ( (n = \0c  -> \2c ^c x = gimel_fct x)
   & ( (exists m, n = succ_o m) -> \2c ^c x = gimel_fct x)
   & (limit_ordinal n -> cpow_less_ecb x ->
       \2c ^c x = \2c ^<c x *c  gimel_fct x)
   & (limit_ordinal n -> not (cpow_less_ecb x) ->
       \2c ^c x = gimel_fct( \2c ^<c x))).  (* 162 *)
```

Comments. We explained above how x^y can be computed. It is either z^z for some infinite z less than x, or x, or 2^y or $\beth x$. Thus, if $\beth x$ is known for all x, then x^y can be computed. Note that $\beth x > x^+$, and if $2^{\mathrm{cf}(x)} \geq x$ then $\beth k = 2^{\mathrm{cf}(x)}$. One may thus assume (SCH, Singular Cardinal Hypothesis): whenever $2^{\mathrm{cf}(x)} < x$ one has $\beth x = x^+$. Then the knowledge of 2^t for all regular t determines x^y. [this needs to be explained]

8.19 Inaccessible cardinals

If x is a weakly inaccessible cardinal, then the number of regular cardinals $< x$ is x (note that $x = \aleph_x$, and all $\aleph_{i+1}$ are regular).

```
Lemma inaccessible_pr2 x:  inaccessible_w x ->
  cardinal (Zo x regular_cardinal) = x.
```

We say that x is *dominant* if x is an infinite cardinal such that $a < x$ and $b < x$ implies $a^b < x$. Note that $x \neq 0$ and $2^b < x$ for all b implies that x is dominant (note that x cannot be finite). If x is a cardinal, $x_0 = x$; $x_{n+1} = 2^{x_n}$, then $b = \sup x_i$ is the least dominant cardinal $> x$. Note that $b = \sum x_i$ so that $2^b = \prod x_i \leq b^\omega$. The reverse equality is trivial so that $2^b = b^\omega = \omega^b$. In Exercise 21c, Bourbaki deduces that if $\mathfrak{b}$ is the least dominant cardinal greater than ω, then $\mathfrak{b}^{\aleph_0} = (2^{\mathfrak{b}})^{\mathfrak{b}}$. This is an example of $a < b$ and $c < d$ but $a^c = b^d$.

```
Definition card_dominant x:=
   infinite_c x & forall a b, a <c x -> b <c x -> a ^c b <c x.
Definition next_dominant x :=
  \osup (image_of_fun (induction_defined (fun z =>\2c ^c z) x)).

Lemma card_dominant_pr1 x: is_cardinal x -> (* 28 *)
   x <> \0c -> (forall m, m <c x -> \2c ^c m <c x) -> card_dominant x.
Lemma card_dominant_pr2: card_dominant \omega.
Lemma next_dominant_pr x (y:= next_dominant x): is_cardinal x ->
   (card_dominant y &  x <c y &
   (forall z,  card_dominant z -> x <c z -> y <=c z)).
Lemma card_dominant_pr3 x (y := next_dominant x) :
  is_cardinal x -> \2c ^c y  =  y ^c \omega.
Lemma card_dominant_pr4 (b:=  least_non_trivial_dominant):
  (card_dominant b
   & \omega <c b
   & (forall z, card_dominant z -> \omega <c z -> b <=c z)
   & (b ^c \omega = \omega ^c b)
   & (b ^c \omega = \2c ^c b)
   & (b ^c \omega = (\2c ^c b) ^c b)).
```

Consider a cardinal x such that, whenever $(x_i)_{i\in \mathrm{I}}$ is a family indexed by a set I of cardinal $< x$, such that $x_i < x$, we have $\prod x_i < x$. This condition holds for zero (there is no such I), and for two (the result is clear when I is empty, otherwise, there is a single index i, and the product is x_i). This condition does not hold for $i = 1$ (consider $\mathrm{I} = \emptyset$). Assume $x > 2$. Take $x_i = 2$; it follows that x is dominant (in particular infinite). Assume that x is the successor of y. Take $x_i = \mathrm{I} = y$. We get $y^y < x$. But $y^y = 2^y > y$, thus $y^y \geq x$, absurd. Thus, $x = \omega$ or x is a limit cardinal. Let x_i be a small family. Let I' be the subset of I formed of all indices such that $x_i \geq 2$, and J the complementary. Then $\sum x_i \leq \mathrm{J} + \sum_{\mathrm{I}'} x_i$ and the second sum is $\leq \prod \sum_{\mathrm{I}'} x_i$. Thus $\sum x_i < x$. We deduce that x is regular, and either ω or limit regular, that is

inaccessible. Conversely, assume x inaccessible. Consider a family x_i and the supremum s. We have $\prod x_i \leq s^{\mathrm{I}}$. If $s < x$, we get $\prod x_i < x$ since x is dominant. Otherwise $s = x$ and $\sum x_i = x$. This implies $\mathrm{I} \geq \mathrm{cf}(x)$. But x is regular and $x = \mathrm{cf}(x)$, absurd.

```
Definition inaccessible x :=
   inaccessible_w x & (forall t, t <c x -> \2c ^c t <c x).

Definition cprod_of_small f x:=
  cardinal_fam f  &
  (forall i, inc i (domain f) -> V i f <c x) &
  domain f <c x.
Lemma inaccessible_pr3 x: \2c <c x ->
   (forall f, cprod_of_small f x -> card_prod f <c x) ->
   card_dominant x.
Lemma inaccessible_pr4 x: \2c <c x ->
   (forall f, cprod_of_small f x -> card_prod f <c x) ->
   (x = \omega \/ (exists n : Set, limit_ordinal n & x = \aleph n)).
Lemma inaccessible_pr5 x: \2c <c x ->
   (forall f, cprod_of_small f x -> card_prod f <c x) ->
   (x = \omega \/ inaccessible x).
Lemma inaccessible_pr6 x:  inaccessible x ->
   (forall f, cprod_of_small f x -> card_prod f <c x).
```

Assume κ inaccessible. Then $\kappa^\lambda = \kappa$ whenever $0 < \lambda < \kappa$. This holds, because $\lambda < \mathrm{cf}(\kappa)$, so that $\kappa^\lambda = \kappa \sum \tau^\lambda$. Since κ is dominant, we have $\tau^\lambda < \kappa$, so that the sum is $\leq \kappa$. It follows $k^{<k} = k$.

```
Lemma inaccessible_pr7 x y:  inaccessible x ->
  y <> \0c -> y <c x ->  x ^c y = x. (* 25 *)
Lemma inaccessible_pr8 x:  inaccessible x -> x = x ^<c x.
```

Let's state: the cofinality of $a + b$ is that of b, whenever b is a non-zero an ordinal (note that the set of all $a + x$ with $x < b$ is cofinal in the sum). We deduce $\mathrm{cf}(a + \omega) = \omega$. Thus $\mathrm{cf}(\aleph_{\alpha+\omega}) = \aleph_0$, whenever α is an ordinal. We deduce that there is no set containing all singular cardinals: if E is such a set, x its supremum, $\aleph_\alpha \geq x$, then $\aleph_{\alpha+\omega}$ is singular and not in the set.

```
Lemma cofinality_sum a b: is_ordinal a -> is_ordinal b -> b <> \0o ->
   cofinality (a +o b) = cofinality b.  (* 24 *)
Lemma cofinality_sum1 a: is_ordinal a ->
   cofinality_c (omega_fct(a +o \omega)) = \omega.
Lemma cofinality_sum2 a: is_ordinal a ->
   singular_cardinal (omega_fct(a +o \omega)).
Lemma singular_non_collectivizing:
  not (exists E, forall x, singular_cardinal x -> inc x E).
```

8.20 Consequences of GCH

We assume here GCH, that says that 2^x is the cardinal successor of x, whenever x is infinite.

```
Section GenContHypothesis_Props.
Hypothesis gch: GenContHypothesis.
```

The main result is

$$x^y = \begin{cases} 1 & \text{in case } y = 0 \\ x & \text{in case } 0 < y < \mathrm{cf}(x) \\ x^+ & \text{in case } \mathrm{cf}(x) \le y \le x \\ y^+ & \text{otherwise} \end{cases}$$

The nontrivial point in the proof is the third case. We use (21e) and pretend $\tau^\lambda \le x$. Let $w = \sup(\tau, \lambda)$. Then $\tau^\lambda \le w^w = 2^w = w^+$ (assuming w infinite, otherwise the result is trivial). Now $w < x$ says $w^+ \le x$.

```
Lemma infinite_power10 x y (z := x ^c y): infinite_c x ->
  ( (y = \0c -> z = \1c)
   & (y <> \0c -> y <c cofinality_c x -> z = x)
   & (cofinality_c x <=c  y -> y <=c x -> z = succ_c x)
   & (x <c y -> z = succ_c y)).  (* 73 *)
```

An easy consequence is $\kappa^{\mathrm{cf}(\kappa)} = \kappa^+$ whenever κ is infinite.

If $0 < \lambda < \mathrm{cf}(\kappa)$ then $\kappa^\lambda = \kappa$.

If κ is regular, then $\kappa^{<\kappa} = \kappa$.

We have $2^{<\kappa} = \kappa$ for any infinite κ. Assume $y < \kappa$; then $2^y \le \kappa$, as this is obvious if y is finite, and equivalent to $y^+ \le \kappa$ otherwise; so that $2^{<\kappa} \le \kappa$ holds. Let $y = 2^{<\kappa}$. If $y < \kappa$ then $2^y \le 2^{<\kappa} = y$, contradicting Cantor.

If x is weakly inaccessible, it is strongly inaccessible. All that needs to be done is to show that if $t < x$ implies $2^t < x$. This is obvious if t is finite; otherwise $x = \aleph_m$ and $2^t = \aleph_{n+1}$ where $n < m$. The relation relation $n+1 < m$ holds since m is a limit ordinal.

If x is inaccessible or ω, then $x^{<x} = x$. Assume $x = y^+$, so that $x^{<y} = x^y$. Now GCH says $x = 2^y$ so that $x^y = 2^{yy} = 2^y = x$. Conversely, assume $x^{<x} = x$. This relation holds if $x = 0$, $x = 1$ and $x = 2$; otherwise x is infinite; thus is a successor, $\aleph_0$ or limit. In the case it is regular, thus inaccessible if GCH holds.

Property (21e) $\kappa^\lambda = \kappa \sum_{\tau<\kappa} \tau^\lambda$ holds for every λ if and only if κ is regular. We have already seen that it holds for $\lambda < \mathrm{cf}(\kappa)$ and $\lambda \ge \kappa$. Thus, κ is regular, it holds for all λ. Assume that it holds for all λ, as well as GCH. An easy consideration shows that κ has to be zero or infinite. Let's exclude the first case, take $\lambda = \mathrm{cf}(\kappa)$. This is an infinite cardinal and the LHS is $> \kappa$; there there is some y, with $y < \kappa$ such that $y^\lambda > \kappa$. If $\lambda \le y$ then y is infinite and $y^\lambda \le y^y = y^+ \le \kappa$, absurd. Thus $y \le \lambda$ and $y^\lambda = 2^\lambda$ (note that $y \ge 2$).

Thus, it holds for all λ if κ is regular. Conversely, assume that it holds for any non-zero λ. Let's exclude the case $\kappa = 0$. If $\kappa = 1$, then the RHS is zero, absurd. If $\kappa = 2$, then the RHS is 2, absurd. Taking $\lambda = 1$, $\tau = 2$ shows that κ is infinite. Each term of the sum is at most 2^x, where x is the supremum of τ and λ. Assume $\lambda < \kappa$. If GCH holds, all terms are $\le \kappa$, and so is the RHS. Take $\lambda = \mathrm{cf}(\kappa)$, then the LHS is $> \kappa$. We deduce $\mathrm{cf}(\kappa) \ge \kappa$ hence κ is regular.

```
Lemma infinite_power10_a x: infinite_c x ->
   x ^c (cofinality_c x) = succ_c x.
Lemma infinite_power10_b x y:
  infinite_c x -> y <c  (cofinality_c x) -> y <> \0c ->
  x ^c y = x.
Lemma cpow_less_pr8 x: infinite_c x ->
  \2c ^<c  x = x.
Lemma cpow_less_pr9 x: regular_cardinal x ->
```

```
  x ^<c  x = x.
Lemma inaccessible_weak_strong x:
   inaccessible_w x -> inaccessible x.
Lemma inaccessible_pr8_gch x: \2c <c x ->  (* 40 *)
   ((x = \omega) \/
     (exists n, is_ordinal n & x = omega_fct (succ_o n)) \/ inaccessible x
   <-> x = x ^<c x).
Lemma infinite_power7h_rev x: \0c <c x ->  (* 79 *)
  (forall y, \0c <c y ->
  x ^c y =  (card_sum (L (set_of_cardinals_lt x) (fun z => z ^c y))) *c x)
  -> regular_cardinal x.
End GenContHypothesis_Props.
```

8.21 Other properties

Given a sequence of n ordinals x_i we consider all permutations σ of the interval $[1,n[$ and the sums $x_\sigma = \sum_i x_{\sigma(i)}$. We set $N(x)$ to be the number of distinct ordinals in the list (the cardinal of the set of all x_σ). This quantity is between 1 and $n!$. It is n in the trivial cases $n = 0$ and $n = 1$. It can be 1 or 2 in the case $n = 2$ (it depends on whether x_0 commutes with x_1).

```
Definition ord_sum_compose n X g :=
   ord_sum_expansion (gcompose X (graph g)) n.
Definition all_sums n X :=
  fun_image (set_of_permutations (interval_co_0a n))  (ord_sum_compose n X).
Definition all_sums_card n X := cardinal (all_sums n X).

Lemma ex10_a n X: inc n Bnat -> (* 13 *)
  all_sums_card n X <=c (factorial n).
Lemma ex10_b X n f:  (* 7 *)
   inc n Bnat -> expansion_ax X n ->
   inc f (set_of_permutations (interval_co_0a n)) ->
   (expansion_ax  (gcompose X (graph f)) n &
     is_ordinal (ord_sum_compose n X f)).
Lemma ex10_c n X: inc n Bnat -> \1c <=c all_sums_card n X. (* 5 *)
Lemma ex10_d X n: (n = \0c \/ n = \1c) -> (* 5 *)
  expansion_ax X n -> all_sums_card n X = \1c.
Lemma ex10_e X n: inc n Bnat -> expansion_ax X (succ n) ->  (* 8 *)
  expansion_ax (restr X (interval_co_0a n)) n.
Lemma ex10_f X: expansion_ax X \2c ->  (* 7 *)
  ord_sum_expansion X \2c = (V \1c X) +o (V \0c) X.
Lemma ex10_g X: expansion_ax X \3c -> (* 12 *)
  ord_sum_expansion X \3c = (V \2c X) +o  ((V \1c X) +o (V \0c) X).
Lemma ex10_n n X (f := identity (interval_co_0a n)) : (* 7 *)
  inc n Bnat ->  expansion_ax X n ->
  (inc f (set_of_permutations (interval_co_0a n))
  & (ord_sum_compose n X f = ord_sum_expansion X n)).
Lemma ex10_h: exists X, (* 44*)
   expansion_ax X \2c & all_sums_card \2c X = \2c.
Lemma ex10_i n X f: inc n Bnat -> expansion_ax X n -> (* 38 *)
  inc f (set_of_permutations (interval_co_0a n)) ->
   all_sums_card n X =  all_sums_card n (gcompose X (graph f)).
```

We consider now the supremum of the set of integers k such that there exists a sequence x_i of length n such that $N(x) = k$. This is the greatest of all $N(x)$. We will call it $f(n)$. We have

$f(0) = 1$, $f(1) = 1$ and $f(2) = 2$.

We have $f(3) \geq 5$. This follows from the following consideration: Consider first the three numbers $A = \omega^2$, $B = \omega$ and $C = \omega + 1$. We have $C + A = B + A = A$ so that $B + C + A = C + B + A = A$. Moreover $C + A + B = A + B$ and $B + A + C = A + C$. The five ordinals $A + B + C$, $A + C + B$, and the three previous sums are all different; since $A + x = A + y$ implies $x = y$, all we need to show is that the numbers B, C, $B + C$ and $C + B$ are non-zero and distinct. These numbers are B, $B + 1$, $B + B + 1$ and $B + B$.

```
Definition ex10_function n :=
 \osup (Zo Bnat (fun z => exists X, expansion_ax X n & all_sums_card n X = z)).

Lemma Bnat_sup_pr T (s:= \osup T) k: (* 19 *)
  sub T Bnat -> inc k Bnat -> (forall i, inc i T -> i <=c k) ->
  (inc s Bnat & s <=c k &
    (forall i, inc i T -> i <=c s)
    & (T = emptyset \/ inc s T)).
Lemma ex10_j n (f := ex10_function n): inc n Bnat ->  (* 18 *)
  (inc f Bnat &
   f <=c factorial n &
   (exists X, expansion_ax X n & all_sums_card n X = f) &
   (forall X, expansion_ax X n -> all_sums_card n X <=c f)).
Lemma ex10_k: (ex10_function \0c = \1c & ex10_function \1c = \1c). (* 8 *)
Lemma ex10_l: ex10_function \2c = \2c.  (* 4 *)
Lemma ex10_m (* 65 *)
  (A:= \omega ^o \2o) (B:= \omega) (C := \omega +o \1o)
  (s1 := A +o (B +o C)) (s2:= A +o (C +o B)) (s3 := B +o (A +o C))
  (s4 := B +o (C +o A)) (s5:= C +o (A +o B)) (s6 := C +o (B +o A)):
  cardinal (union2 (union2 (doubleton s1 s2) (doubleton s3 s5))
    (doubleton s4 s6)) = card_five.
```

We have $f(3) = 5$. The follows from considerations of degree in the Cantor Normal Form. We show here that any non-zero ordinal has a non-trivial CNF, and a degree $d(x)$. If $d(x) < d(y)$ then $x + y = y$.

```
Lemma the_cnf_pb x (X := the_CNF x) (n := the_CNF_len x):
   is_ordinal x -> x <> \0o ->
   (CNF_axn X n & x = CNFv X).
Lemma the_cnf_pc x y :
   is_ordinal x -> is_ordinal y -> x <> \0o -> y <> \0o ->
   the_CNF_degree x <o the_CNF_degree y ->
   x <<o y.
```

We say that a sequence x_i is of type k if, either $k = 0$ and one x_i or zero, or there are k elements in the list of degree m, all other elements are of degree $> m$. Each non-empty sequence has a unique type. We denote by $f_k(n)$ the supremum of all $N(x)$ where x is of length n and type k. This quantity is zero is no such x, and is the maximum otherwise. Moreover, $f(n)$ is the supremum of all $f_k(n)$.

```
Definition ex10_condition_0 X n :=
  exists i, inc i (interval_co_0a n) & V i X = \0o.
Definition ex10_type0 X n k :=
  (forall i, inc i (interval_co_0a n) ->  V i X <> \0o) &
  (exists m, is_ordinal m &
```

```
     (forall i, inc i (interval_co_0a n) ->
       m <=o the_CNF_degree (V i X)) &
     cardinal (Zo (interval_co_0a n) (fun i => the_CNF_degree (V i X) = m)) =k).
Definition ex10_type X n k:=
   ((k = \0c) & (ex10_condition_0 X n)) \/
   (k <> \0c &  (ex10_type0 X n k)).

Lemma ex10_type_p1 X n k1 k2: (* 16 *)
  (ex10_type X n k1) -> (ex10_type X n k2) -> k1 = k2.
Lemma ex10_type_p2 X n: (* 27 *)
  inc n Bnat -> n <> \0c -> expansion_ax X n ->
  exists k,  k <=c n & ex10_type X n k.
Lemma ex10_type_p3 n k (f := ex10_function_aux n k): (* 16 *)
  inc n Bnat -> n <> \0c -> k <=c n ->
  (inc f Bnat &
   f <=c  (ex10_function n) &
   (forall X,  expansion_ax X n -> ex10_type X n k -> all_sums_card n X <=c f)
   & ((forall X,  expansion_ax X n -> ~(ex10_type X n k))
      \/ (exists  X, expansion_ax X n & ex10_type X n k &
          all_sums_card n X= f))).
Lemma ex10_type_p4  n
  (g := ex10_function_aux n) (f:= ex10_function n): (* 29 *)
 ( inc n Bnat -> n <> \0c ->
 ( (forall k,  k <=c n -> (g k) <=c f)
   & (exists k,  k <=c n & (g k) = f)
   & f= \osup (fun_image (interval_cc_0a n) g)).
```

Let's notice that $f_0(n+1) = f(n)$. We first show that if X is of type zero, there is a permutation Y of X such that N(Y) = N(X) and $Y_n = 0$. Consider any permutation σ of $[0; n]$ and let Y_σ be the sum of all $Y_{\sigma(i)}$. Assume $\sigma(k) = n$. Define $\tau(i)$ be $\sigma(i)$ if $i < k$, and $\sigma(i+1)$ if $k < i < n$. This is a permutation of $[0, n-1]$. Let Z_τ be the sum of all $Z_{\tau(i)}$, where Z is the restriction of Y to $[0, n-1]$; We have $Z_\tau = a + b$ and $Y_\sigma = a + c + b$, where $c = Y_{\sigma(k)}$. By construction $c = 0$ and $Z_\tau = Y_\sigma$. Thus N(Z) = N(Y) = N(X).

```
Lemma ex10_type_p5 n X: inc n Bnat ->  expansion_ax X n ->
  ex10_condition_0 X n -> exists g,
  let Y := gcompose X (graph g) in
    inc g (set_of_permutations (interval_co_0a n)) &
    all_sums_card n X = all_sums_card n Y &  V (predc n) Y = \0o.
```

We have $f(3) = 5$. Proof. Consider three ordinal numbers, x, y and z. Assume $x = 0$, so that the sums are $y + z$ and $z + y$. The maximum number of different sums is then $f(2)$, obtained for instance for $x = 0$, $y = 1$ and $z = \omega$. Assume that all numbers are non-zero ordered by increasing degree. Assume all degrees different, so that $x \ll y \ll z$. If we add them together, there are only four possible results z, $z + x$, $z + y$ and $z + y + x$. If $x = 1$, $y = \omega$ and $z = \omega^2$, these quantities are all distinct. Assume that x and y have the same degree, which is less that the degree of z. There is then a fifth possibility, $z + x + y$. If $z = \omega^2$, $y = \omega + 1$ and $x = \omega$ these quantities are all distinct (previous lemma). Assume that y and z have the same degree, which is greater than the degree of x. We have four possibilities $y + z$, $z + y$, $y + z + x$, and $z + y + x$. If $x = 1$, $y = \omega^2$ and $z = \omega^2 + 1$ these quantities are all distinct. Assume now that x, y and z have the same degree, let's say, they have the form $\omega^n \cdot x_1 + x'$. Let $c = x_1 + y_1 + z_1$. Then $x + y + z = \omega^n \cdot c + z'$. The coefficient is independent of the ordering (as x_1, y_1 and z_1 are integers). Thus the sum depends only on the last term, and there are at most three possibilities.

General case. If $0 \le k \le n$, we define $f_k(n)$ to be the maximal value of $\mathrm{N}(x)$ for all sequences x sof length n, such that there exists an ordinal m, such that k elements of the list have degree m, all others have degree $> m$. If $k = 0$, the condition becomes: one of the x_i is zero.

We define $f(n)$ to be the maximum of all $\mathrm{N}(x_i)$. We say that the sequence $(x_i)_i$ satisfies (P) if $\mathrm{N}(x_i) = f(n)$. Let's compute $f(n)$. We shall show that there exists a sequence (x_i) satisfying (P) which is of finite degree (all x_i are of finite degree). By distributivity, $\omega^k \cdot (\sum_i x_{\sigma(i)}) = \sum_i \omega^k \cdot x_i$. Moreover $\omega^k \cdot x = \omega^k \cdot y$ implies $x = y$. Thus $(\omega^k \cdot x_i)_i$ satisfies (P). Define the valuation of $(x_i)_i$ to be the least exponent in the Cantor Normal Form of any x_i. The valuation of $(\omega^k \cdot x_i)_i$ is at least k.

We have $f(1) = 1$, $f(2) = 2$. We have shown $f(3) = 5$ by considering different cases. Introduce $\mathrm{C}_k = k.2^{k-1} + 1$. Note that $f(3) = \mathrm{C}_2$. We generalize the case $n = 3$ as follows.

Cantor notices that one can compare ordinal via their normal form. Consider two number in form (14b). Assume that for $i = 1$, $i = 2$, up to $i = p$ the coefficients and exponents are the same. Then the two number can be written as $\alpha = \mathrm{X} + \mathrm{Y}$ and $\alpha' = \mathrm{X} + \mathrm{Y}'$, where Y and Y′ are two sums in normal form, and p is maximal. The two quantities α and α' compare the same as Y and Y′. If one of the sums is empty. comparison is trivial. Otherwise, the sum with the greatest leading exponent is the greatest element (since $\mathrm{Y} \ll \mathrm{Y}'$ or $\mathrm{Y}' \ll \mathrm{Y}$. If the exponents are the same, it suffices to compare the coefficients. Note that the same method can be used to compare elements in the form (14a), the key relation being that $\alpha < \gamma^\lambda$ whenever $\lambda_1 < \lambda$.

Consider a sequence of n ordinals x_i, with degree n_i. Let m be the least degree and k the number of elements of degree k. Let $f_k(n)$ be the maximum different sums, subject to this condition; We have $f(n) = \sup_{0<k\le n} f_k(n)$

If $k = n$, we get $f_n(n) = n$, for if $x_i = \omega^m \cdot c_i + x'_i$, the sum is $\omega^m \cdot (\sum x_i) + x'_j$, where j is the index of the least term. Chose $x_i = \omega + i$. We get n distinct terms. Assume $k < n$. Write y_i instead of x_i for $k > i$. We have $x_i + y_j = y_j$. All sums have the form $s_y + s_x$, where s_y is the sum of a permutation of the y_i, while s_x is a sum of a permutation of a subset of the x_i. There are $f(n-k)$ possibilities for s_y. Let's count the number of possibilities of s_x. Assume $x_i = \omega^m c_i + r_i$. Consider $\sum_{i\in \mathrm{K}} x_i$ where K is a subset of $[1, k]$, with elements listed in some order. Let j be the last index. We have $\sum_{i\in \mathrm{K}} x_i = \omega^m (\sum_{i\in \mathrm{K}} c_i) + r_j$. Let $c_\mathrm{K} = \sum_{i\in \mathrm{K}} c_i$. This quantity depend only of the set of indices, not on the ordering. For fixed j, there are 2^{k-1} possibilities. Thus, we have at most C_k possibilities. Take $m = 1$, $c_i = 2^i$ and $c_i = i$. Then $s_x = \omega \cdot c_\mathrm{K} + j$ and we have $k.2^{k-1} + 1$ different values. Chose for (y_i) a sequence that satisfies (P) with valuation > 1. This means that $x_y + s_x$ uniquely defines s_x. We get $f_k(n) = \mathrm{C}_k f(n-k)$. Note that $f_{n-1}(n) \ge f_n(n)$. We have shown

$$f(n) = \sup_{0<k<n} \mathrm{C}_k f(n-k), \qquad \mathrm{C}_k = k.2^{k-1} + 1$$

.

The table below shows for each n the value $f(n)$, the indices k that realize the supremum, the ratio $f(n)/f(n-1)$ and $f(n)/f(n-5)$ as well as $f(n)/81^{\lfloor n/5 \rfloor}$. Here $a = 81/33 = \mathrm{C}_5/\mathrm{C}_4 = 2.45$, $b = 193/81 = \mathrm{C}_6/\mathrm{C}_5 = 2.38$, and $c = 81/ab^3 = 2.44$.

We pretend that $f(n) = \mathrm{C}_5 f(n-5)$ for $n \ge 20$. The proof is as follows. Consider the following statements

(C1) $a = \mathrm{C}_5/\mathrm{C}_4$, $b = \mathrm{C}_6/\mathrm{C}_5$, $c = 81/ab^3$
(C2) $b < c < a < \mathrm{C}_2/\mathrm{C}_1 < \mathrm{C}_4/\mathrm{C}_3 < \mathrm{C}_3/\mathrm{C}_2$
(C3) $\mathrm{C}_4\mathrm{C}_6 < \mathrm{C}_5^2$, ratio 1.030
(C4) $\mathrm{C}_6^4 < \mathrm{C}_5^4\mathrm{C}_4$, ratio 1.024

4	13	3	2.6	
5	33	4	2.54	
6	81	5	a	81
7	193	6	b	96.5
8	449	7	2.32	89.8
9	1089	4	2.42	83.8
10	2673	4,5	a	81
11	6561	5	a	81
12	15633	5,6	b	81
13	37249	6	b	82.9
14	88209	4,5	2.37	81
15	216513	4,5	a	81
16	531441	5	a	81
17	1266273	5,6	b	81
18	3017169	5,6	b	81
19	7189057	6	b	81.5
20	17537553	4,5	c	81

(C5) $C_4^2 C_5^2 < C_6^3$, ratio 1.006.

(C6) $C_3 C_6 < C_4 C_5$ (ratio 1.065) (C7) $C_3 C_6^2 < C_5^3$ (ratio 1.097) (C8) $C_3 C_4 C_5 < C_6^2$ (ratio 1.072) (C9) $C_3 < b^3$ (ratio 1.040)

(C3) says $b < a$. Relation (C4) says $b \le c$. Relation (C5) says that $a^2 b^3 \ge 81$.

We start with the following assertions
(H0) $f(5p) = 81^p / a$ for $p > 0$,
(H1) $f(5p+1) = 81^p$ for $p \ge 0$,
(H2) $f(5p+2) = b.81^p$ for $p \ge 1$,
(H3) $f(5p+3) = b^2.81^p$ for $p \ge 2$,
(H4) $f(5p+4) = b^3.81^p$ for $p \ge 3$,
(H5) $f(3) = 5$,
(H6) $f(4) = 13$,
(H7) $f(8) = 449$,
(H8) $f(9) = 1089$
(H9) $f(14) = 88209$.

It follows from the first relations that $f(n+1)/f(n)$ is a for $n \ge 5$ and $n = 0 \bmod 5$, it is c when $n + ge19$ and $n = 4 \bmod 5$. It is b otherwise (for $n \ge 6$, $n \ge 12$ and $n \ge 18$ when n is 1, 2 and 3 modulo 5, respectively. Since $C_2/C_1 = 5/2$, it follows that for $n \ge 7$, we have

$$C_1 f(n-1) \le C_2 f(n-2) \le C_3 f(n-3) \le C_4 f(n-3) \le C_5 f(n-5)$$

Set $p = k-5$ and $q = n-k$. All we need to show is (for q non-zero, $p+q \ge 15$)

$$C_{p+5}/C_5 \le f(p+q)/f(q)$$

Assume $p = 5i+k$. Let $T = 81^i$. Define $r(p)$ to be T, bT, b^2T, b^3T and $81/a$, if k is respectively 0, 1, 2, 3 and 4. Then $C_{p+5}/C_5 \le r(p)$.

Let's compute the right hand side r_{pq} of this equation. We write $p = 5i+k$, with $0 \le k < 5$. Let $T = 81^i$. In the case $k = 0$ we get $r = T$ (with the possible exceptions of 2, 3, 4, 8, 9, and 14). In the case $k = 1$, we get $r = a$T, $r = b$T or $r = c$T, depending on the cases; in particular

$r \geq b\mathrm{T}$. In the case $k = 2$, r/T is b^2, ab, cb or $81/b^3$; it is at least b^2. In the case $k = 3$, r/T is one of ab^2, b^3, cb^2, $81/b^2$. We deduce $r \geq b^3\mathrm{T}$. Finally, if $k = 4$ we get $81/a$, $81/b$ and $81/c$, which is at least $81/a$.

8.22 Order types

We assume the existence of a function $\mathrm{Ord}(x)$ that satisfies two criteria: First, if x is an ordering, then x is order-isomorphic to $\mathrm{Ord}(x)$. Second, if x and y are order-isomorphic, then $\mathrm{Ord}(x) = \mathrm{Ord}(y)$.

Objects of the form $\mathrm{Ord}(x)$ are called order-type; they can be compared by $x \prec_{\mathrm{Ord}} y$, meaning that there is an order-isomorphism of x onto some subset z of y.

```
Parameter order_type: Set -> Set.
Axiom order_type_exists:
  forall x, order x -> order_isomorphic x (order_type x).
Axiom order_type_unique:
  forall x y, order_isomorphic x y -> (order_type x = order_type y).
Definition is_order_type x := exists y, order y & x = order_type y.
Definition order_type_le x y:=
   is_order_type x & is_order_type y &
   exists f, exists z,
    sub z (substrate y) & order_isomorphism f x (induced_order y z).
Notation "x <=t y" := (order_type_le x y) (at level 60).
Notation "x <=O y" := (order_le x y) (at level 60).
```

We have $\mathrm{Ord}(r) = \mathrm{Ord}(o(\mathrm{ord}(r)))$ whenever r is well-ordered. This relation $x \prec_{\mathrm{Ord}} y$ is reflexive and transitive (but not antisymmetric).

```
Lemma OT_ordinal_compat x: worder x ->
  order_type x = order_type (ordinal_o (ordinal x)).
Lemma OT_prop0 x: order x -> (order_type x) \Is x.
Lemma OT_prop1 x: order x -> is_order_type (order_type x).
Lemma OT_prop2 x: is_order_type x -> order x.
Lemma OT_prop3 x: order x -> order (order_type x).
Lemma OT_prop4 x: order x ->
  order_type (order_type x) = order_type x.

Lemma order_le_alt2 r r': r <=O r' ->
   (exists f, order_morphism f r r').
Lemma order_le_alt3 r r':
  r <=O r' <-> (exists f, order_morphism f r r').
Lemma order_le_transitive: forall x y z,
  x <=O y -> y <=O z -> x <=O z. (* 28 *)
Lemma OTorder_le_alt r r':
  r <=t r' <-> (is_order_type r & is_order_type r' &
    exists f, order_morphism f r r').
Lemma OTorder_le_alt2 r r':
  r <=t r' <-> (is_order_type r & is_order_type r' & r <=O r').
Lemma OTorder_le_compat1 r: order r ->
  r <=O (order_type r).
Lemma OTorder_le_compat2 r: order r ->
  (order_type r) <=O r.
```

```
Lemma OTorder_le_compat r r': order r -> order r' ->
  (r <=O r' <-> (order_type r) <=O  (order_type r')).

Lemma OT_order_le_reflexive x: is_order_type x -> x <=t x.
Lemma OT_order_le_transitive x y z:
  x <=t y -> y <=t z -> x <=t z.
```

We define here the ordinal sum and ordinal product, according to Bourbaki.

```
Definition OT_sum r g :=
   order_type (order_sum r (L (domain g) (fun z => (order_type (V z g))))).
Definition OT_prod r g :=
   order_type (order_prod r (L (domain g) (fun z => (order_type (V z g))))).
Definition OT_sum2 a b :=
  order_type (order_sum2 (order_type a) (order_type b)).
Definition OT_prod2 a b :=
  order_type (order_prod2 (order_type a) (order_type b)).
Notation "x +t y" := (OT_sum2 x y) (at level 50).
Notation "x *t y" := (OT_prod2 x y) (at level 40).
```

We show some basic properties of the ordinal sum and product.

```
Lemma OT_sum2_pr a b:
  a +t b = OT_sum canonical_doubleton_order (variantLc a b).
Lemma OT_prod2_pr a b:
  a *t b = OT_prod canonical_doubleton_order (variantLc b a).

Lemma OT_sum_ordertype r g:
  order r -> substrate r = domain g -> order_fam g ->
  is_order_type (OT_sum r g).
Lemma OT_prod_ordertype r g:
  worder r -> substrate r = domain g -> order_fam g ->
  is_order_type (OT_prod r g).
Lemma OT_sum2_ordertype a b: is_order_type a -> is_order_type b ->
  is_order_type (a +t b).
Lemma OT_prod2_ordertype a b: is_order_type a -> is_order_type b ->
  is_order_type (a *t b).
```

We show what happens when an ordering is replaced by an isomorphic one.

```
Lemma OT_sum_invariant3 r g:
  order r -> substrate r = domain g -> order_fam g ->
  order_type (order_sum r g) =
    OT_sum r (L (substrate r) (fun i => order_type (V i g))).
Lemma OT_prod_invariant3 r g:
  worder r -> substrate r = domain g -> order_fam g ->
  order_type (order_prod r g) =
    OT_prod r (L (substrate r) (fun i => order_type (V i g))).
Lemma OT_sum_invariant5 a b c: order a -> order b -> order c ->
  (order_sum2 a b) \Is c ->
  (order_type a) +t (order_type b) = order_type c.
Lemma OT_prod_invariant5 a b c: order a -> order b -> order c ->
  (order_prod2 a b) \Is c ->
  (order_type a) *t (order_type b) = order_type c.
```

We show here associativity of the sum and product, and show that a product of two terms is a sum.

```
Definition order_type_fam g :=
  fgraph g &  (forall x, inc x (domain g) -> is_order_type (V x g)).

Lemma OT_sum_assoc1 r g r' g':
  order r -> substrate r = domain g  -> order_type_fam g ->
  order r' -> substrate r' = domain g' -> order_fam g' ->
  r = order_sum r' g' ->
  let order_sum_assoc_aux :=
    fun l =>
    OT_sum (V l g') (L (substrate (V l g')) (fun i => V (J i l) g)) in
  OT_sum r g = OT_sum r' (L (domain g') (order_sum_assoc_aux)).  (* 30 *)

Lemma OT_prod_assoc1 r g r' g':
  worder r -> substrate r = domain g  -> order_type_fam g ->
  worder r' -> substrate r' = domain g' -> worder_fam g' ->
  r = order_sum r' g' ->
  (forall i, inc i (domain g') ->  finite_set (substrate (V i g'))) ->
  let order_prod_assoc_aux :=
    fun l =>
    OT_prod (V l g') (L (substrate (V l g')) (fun i => V (J i l) g)) in
  OT_prod r g = OT_prod  r' (L (domain g') order_prod_assoc_aux). (* 31 *)
Lemma OT_sum_assoc3 a b c:
  is_order_type a -> is_order_type b -> is_order_type c ->
  a +t (b +t c) = (a +t b) +t c. (* 17 *)
Lemma OT_prod_assoc3 a b c:
  is_order_type a -> is_order_type b -> is_order_type c ->
  a *t (b *t c) = (a *t b) *t c. (* 17 *)
Lemma OT_sum_distributive3 a b c:
  is_order_type a -> is_order_type b -> is_order_type c ->
  c *t (a +t b) = (c *t a) +t (c *t b). (* 21 *)

Lemma OT_prod_pr1 a b c:
  is_order_type a -> is_order_type b -> worder c -> b \Is c ->
  a *t b = OT_sum c (cst_graph (substrate c) a).
```

We show compatibility of sum and product with the order.

```
Lemma OT_sum_increasing2 r f g: order r ->
  fgraph f -> fgraph g -> substrate r = domain f -> substrate r = domain g ->
  (forall x, inc x (domain f) -> (V x f) <=t (V x g)) ->
  (OT_sum r f) <=t (OT_sum r g). (* 20 *)
Lemma OT_prod_increasing2 r f g: worder r ->
  fgraph f -> fgraph g -> substrate r = domain f -> substrate r = domain g ->
  (forall x, inc x (domain f) -> (V x f) <=t (V x g)) ->
  (OT_prod r f) <=t (OT_prod r g).  (* 20 *)
Lemma OT_sum_increasing4 r f j:  order r ->  (* 33 *)
  substrate r = domain f ->
  sub j (domain f) -> order_type_fam f ->
  (OT_sum (induced_order r j) (restr f j)) <=t (OT_sum r f).
Lemma OT_prod_increasing4 r f j:  worder r ->  (* 41 *)
  substrate r = domain f ->
  sub j (domain f) -> order_type_fam f ->
  (forall x, inc x  (complement (domain f) j) ->
```

```
     substrate (V x f) <> emptyset) ->
  (forall j, inc j (domain f) -> is_order_type (V j f)) ->
  (OT_prod (induced_order r j) (restr f j)) <=t (OT_prod r f).
```

We study now the properties of opposite order-types.

```
Definition OT_opposite x := order_type (opposite_order x).

Lemma OT_opposite1 a b: a \Is b ->
  (opposite_order a) \Is (opposite_order b).
Lemma OT_opposite2 x: order x ->
   opposite_order (opposite_order x) = x.

Lemma OT_double_opposite x: is_order_type x ->
   OT_opposite (OT_opposite x) = x.
Lemma OT_opposite_sum r f: order r ->
  substrate r = domain f -> order_type_fam f ->
  OT_opposite (OT_sum r f) =
  OT_sum (opposite_order r) (L (substrate r) (fun z => OT_opposite (V z f))).
```

8.23 The cardinals, according to Zermelo, 1908

We implement here a part of the theory of Zermelo as described in his paper "Investigations in the foundations of set theory I" (see [12]). It has seven axioms: Axiom I is the axiom of extent, axiom II corresponds to the axiom of the pair, Axiom III is the axiom of separation SC, axiom IV is the axiom of the powerset, Axiom V is the axiom of the union, axiom VI is the axiom of choice and axiom VII the axiom of infinity. The bold face numbers are the paragraph numbers of the Zermelo paper.

Note that Zermelo uses $\mathfrak{S}$A for the union of a family of sets, $\mathfrak{D}$A for its intersection and $\mathfrak{U}$T for the powerset of T. He uses A + B and [A, B], for the union or intersection of two sets. There is no mention of ordered pairs, thus no graphs. The product of two or more sets is given by the following definition.

13. "Let T be a set whose elements M, N, R, ... are various (mutually disjoint) sets, and let S_1 be any subset of its union $\mathfrak{S}$T. Then it is definite for every element M of T whether the intersection [M, S_1] consists of a single element or not. Thus all those elements of T that have exactly one element in common with S_1 are the elements of a certain subset T_1 of T, and it is again definite whether T_1 = T or not. All subsets S_1 of $\mathfrak{S}$T that have exactly one element in common with each element of T then are, according to Axiom III, the elements of a set P = $\mathfrak{P}$T, which according to axioms III and IV is a subset of $\mathfrak{U}\mathfrak{S}$T and will be called the connection set associated with T, or the product of the sets M, N, R, ... If T = {M, N} we write $\mathfrak{P}$T = MN."

Note the product is independent of the ordering of the factors. An element of a product of *n* sets has exactly *n* elements, but if we drop the condition that the sets are disjoint, there may be less than *n* elements.

```
Definition zprod a := Zo (powerset (union a))
  (fun y => forall x, inc x a -> is_singleton (intersection2 y x)).
Definition zprod2 a b:= zprod (doubleton a b).
```

We have $X \in AB$ if and only if $X \subset A \cup B$ and the two sets $X \cap A$ and $X \cap B$ are singletons.

```
Lemma zprod2_pr a b y:
  inc y (zprod2 a b) <->
  (sub y (union2 a b) &
    is_singleton (intersection2 y a) &
    is_singleton (intersection2 y b)).
```

We denote by $s_X(A)$ the unique element of $X \cap A$. If $X \in AB$, then $X \cap A = \{s_X(A)\}$ and $X \cap B = \{s_X(B)\}$. From this we deduce that $s_X(A)$ is in A and X, and also that $s_X(B)$ is in B and X.

```
Definition zpr x a := choose (fun z => (intersection2 x a) = singleton z).
Lemma zpr_prop : forall x a,
  is_singleton (intersection2 x a) -> (intersection2 x a = singleton (zpr x a)).
Lemma zprod2_pr1 a b x:
  inc x (zprod2 a b) ->
  ((intersection2 x a = singleton (zpr x a)) &
  (intersection2 x b = singleton (zpr x b))).
Lemma zprod2_pr0 a b x:
  inc x (zprod2 a b) ->
  (inc (zpr x a) a & inc (zpr x b) b & inc (zpr x a) x & inc (zpr x b) x).

Lemma zprod2_pr0aa a b x:
  inc x (zprod2 a b) ->inc  (zpr x a) a.
Lemma zprod2_pr0ax a b x:
  inc x (zprod2 a b) ->inc  (zpr x a) x.
Lemma zprod2_pr0bb a b x:
  inc x (zprod2 a b) -> inc  (zpr x b) b.
Lemma zprod2_pr0bx a b x:
  inc x (zprod2 a b) ->inc  (zpr x b) x.
```

Conversely, an element in A and X must be $s_X(A)$. From this we deduce $X = \{s_X(A), s_X(B)\}$. If $x \in A$ and $x \in X$, then $s_X(A) = x$.

```
Lemma zprod2_pr1a a b x z:
  inc x (zprod2 a b) ->
  inc z a -> inc z x -> z = zpr x a.
Lemma zprod2_pr1b a b x z:
  inc x (zprod2 a b) ->
  inc z b -> inc z x -> z = zpr x b.
Lemma zprod2_pr2 a b y:
  inc y (zprod2 a b) ->
  y = doubleton (zpr y a) (zpr y b).
```

We characterize here the product AB when B is a singleton $\{b\}$, as the set of all $\{a, b\}$ for $a \in A$.

```
Lemma intersection_singleton a b c:
  (intersection2 a (singleton b) = singleton c) <->
  (inc c a & c = b).
Lemma is_singleton_int a b c:
  inc c a -> inc c b -> (forall u, inc u a-> inc u b -> u = c) ->
  is_singleton (intersection2 a b).
Lemma zprod_singleton M r: ~ inc r M ->
  let N := zprod2 M (singleton r) in
    ( (forall u, inc u M -> inc (doubleton u r) N) &
      (forall x, inc x N -> exists u, inc u M & x = doubleton u r)).
```

15. "A *mapping of* M *onto* N is a subset ϕ of the product MN such that each element of M + N occurs as an element in one and only one element $\{m, n\}$ of ϕ. Two elements m and n that occur together in one element of ϕ are said to be 'mapped onto each other'. Two sets are said to be *immediately equivalent* if there exists at least one such ϕ."

The definition is symmetric with respect to M and N. The union of these two sets is uniquely determined by ϕ as the union of ϕ. Zermelo assumes the two sets disjoint. In fact, if M = N, there is only one mapping, the set of all singletons. We add the disjointness condition to the definition of "equivalent" (but this really changes nothing).

```
Definition zmap f a b := sub f (zprod2 a b) &
  (forall x, inc x (union2 a b) -> exists_unique (fun z => inc z f & inc x z)).
Definition ziequivalent a b := disjoint a b & exists f, zmap f a b.

Lemma zmap_symm f a b:
  zmap f a b -> zmap f b a.
Lemma zequiv_symm a b: ziequivalent a b -> ziequivalent b a.
Lemma zmap_pr1 f a b: zmap f a b ->
  union f = union2 a b.
```

Examples: disjoint singletons are equivalent as well as disjoint doubletons.

```
Lemma zmap_example1 a b: a <> b ->
  let A := singleton a in
    let B := singleton b in
    zmap (singleton (doubleton a b)) A B.
Lemma zmap_example2 a b c d: let A := doubleton a b in
  let B := doubleton c d in
    disjoint A B -> a <> b -> c <> d ->
    zmap (doubleton (doubleton a c) (doubleton b d)) A B.
Lemma zequiv_example1 a b: a <> b ->
  ziequivalent (singleton a) (singleton b).
Lemma zequiv_example2  a b c d: let A := doubleton a b in
  let B := doubleton c d in
    disjoint A B -> a <> b -> c <> d ->
    ziequivalent A B.
```

Assume $f(a) \in$ B whenever $a \in$ A, where A and B are two disjoint sets. The set of all $z \in$ A $\cup$ B of the form $\{a, f(a)\}$ with $a \in$ A is an element of AB. Assume f bijective; this set is then a mapping.

```
Definition zbijective F a b :=
  (  (forall x, inc x a -> inc (F x) b)
    & (forall x x', inc x a -> inc x' a -> F x = F x' -> x = x')
    & (forall y, inc y b -> exists x, inc x a & F x = y)).

Lemma zmap_example3 F a b: disjoint a b -> zbijective F a b ->
  zequivalent a b.
```

Let's denote by $w_f(x)$ the element such that $w_f(x) \in f$ and $x \in w_f(x)$. If f is a mapping, $x \in$ A $\cup$ B, such an object exists and is unique. We can restate this as: if x and y are in f, if s_x(A) = s_y(A) then $x = y$; the same holds for B.

```
Definition zmap_aux f x := choose (fun z =>  inc z f &inc x z  ).
```

```
Lemma zmap_aux_pr1 f a b x:
  zmap f a b -> inc x (union2 a b) ->
  (inc (zmap_aux f x) f & inc x (zmap_aux f x)).
Lemma zmap_aux_pr2 f a b x y:
  zmap f a b -> inc x (union2 a b) -> inc y f -> inc x y
  -> y = (zmap_aux f x).
Lemma zmap_aux_pr3a f a b x y:
  zmap f a b -> inc x f -> inc y f ->  zpr x a = zpr y a
  -> x = y.
Lemma zmap_aux_pr3b f a b x y:
  zmap f a b -> inc x f -> inc y f ->  zpr x b = zpr y b
  -> x = y.
```

Consider now $s_A(w_f(x))$. This is the unique element of A in $w_f(x)$. This is obviously x if $x \in A$. It is also defined for $x \in B$. We shall sometimes denote this by $f_A(x)$. Similarly for $f_B(x)$. We have $f_A(f_B(x)) = x$ if $x \in A$ and $f_B(f_A(x)) = x$ if $x \in B$. This implies that f_B is bijective (in the sense given above).

```
Definition zmap_val f a x:= zpr (zmap_aux f x) a.

Lemma zmap_val_pr1a  f a b x: zmap f a b ->  inc x (union2 a b) ->
  inc (zmap_val f a x) a.
Lemma zmap_val_pr1b  f a b x: zmap f a b ->  inc x (union2 a b) ->
  inc (zmap_val f b x) b.

Lemma zmap_val_pr1 f a b x: zmap f a b ->  inc x (union2 a b) ->
  (inc (zmap_val f a x) a & inc (zmap_val f b x) b).
Lemma zmap_val_pr2a f a b x: zmap f a b ->  inc x a ->
  (zmap_val f a x) = x.
Lemma zmap_val_pr2b  f a b x: zmap f a b ->  inc x b ->
  (zmap_val f b x) = x.

Lemma zmap_val_pr3a  f a b x: zmap f a b ->  inc x a ->
  (zmap_val f a (zmap_val f b x)) = x.
Lemma zmap_val_pr3b f a b x: zmap f a b ->  inc x b ->
  (zmap_val f b (zmap_val f a x)) = x.

Lemma zmap_bijective  f a b: zmap f a b ->
  zbijective (zmap_val f b) a b.
```

16. Zermelo says: "It is definite for two disjoint sets whether they are equivalent or not", since this is the same checking whether a set Ω is empty or not. We just show here that the set of mappings $A \to B$ exists.

```
Lemma zmap_set a b: exists s,
  forall f, zmap f a b <-> inc f s.
```

17. If ϕ is a mapping $M \to N$, and M_1 is a subset of M, there exists a subset N_1 of N which is equivalent to M_1 via a subset of ϕ. We first note that if M and N are disjoint, if M_1 is any subset of M and N_1 any subset of N, then M_1 and N_1 are disjoint. Let ϕ_1 be the set of all $X \in \phi$ such that $s_X(M) \in M_1$, then N_1 is the set of all $z \in N$ of the form $z = s_X(N)$ for some $X \in \phi_1$.

```
Lemma sub_disjoint a b a' b':
```

```
  disjoint a b -> sub a' a -> sub b' b -> disjoint a' b'.
Lemma zmap_sub f a b a': zmap f a b -> sub a' a ->
  exists f', exists b',
    (sub f' f & sub b' b & zmap f' a' b').

Lemma zequiv_sub a b a': ziequivalent a b -> sub a' a ->
  exists b',  (sub b' b & ziequivalent a' b').
```

18. Assume that f maps A to B, and g maps B to C. For instance, we map $\{a,b\}$ to $\{c,d\}$ and then to $\{b,a\}$, where all four elements are distinct. Composition of these mappings yield the permutation on $\{a,b\}$. But this is not a mapping according to Zermelo. Thus, in the following lemma, we assume A and C disjoint. The set of all $z = \{s_x(\mathrm{A}), s_y(\mathrm{C})\}$ in $\mathfrak{P}(\mathrm{A}\cup\mathrm{C})$ such that there exist $x \in f$ and $y \in g$ such that $s_x(\mathrm{B}) = s_y(\mathrm{B})$ is a mapping $\mathrm{A}\to\mathrm{C}$. We give here a simpler proof: both f_{A} and g_{B} are bijective, hence the composition is also bijective. Thus we have a bijection $\mathrm{A}\to\mathrm{C}$, which gives a mapping, since A and C are disjoint.

```
Lemma zmap_transitive a b c: disjoint a c ->
  zequivalent a b -> zequivalent b c -> zequivalent a c.
```

19. In §10, Zermelo says that for every set M there is a set N such that $\mathrm{N}\subset\mathrm{M}$ and $\mathrm{N}\notin\mathrm{M}$. We can restate this without quantifiers as: the set $\{\mathrm{N}\in\mathfrak{P}(\mathrm{M}), \mathrm{N}\notin\mathrm{M}\}$ is non-empty. It suffices in fact to take the set of all elements of M that do not belong to themselves. This theorem can be used as: for every set M there is a set N such that $\mathrm{N}\notin\mathrm{M}$.

Thus, given M and N, there exists r not in M such that, if $\mathrm{R} = \{r\}$, the set MR is disjoint from M and N (an element x of MR has the form $\{y,r\}$, if it is in M or N, it is in $\mathrm{M}\cup\mathrm{N}$ and r is in the union of this set). The function $x \mapsto \{x,r\}$ is bijective, thus we get a mapping $\mathrm{M}\to\mathrm{MR}$.

It follows that, for any M and N, there exists M' disjoint from M and N, equivalent to M.

```
Lemma disjointness M: exists N,
  sub N M & ~ inc N M.
Lemma disjointness1  M N: exists r,
  let M1 := zprod2 M (singleton r) in
   ( ~ (inc r M)  & disjoint M M1 & disjoint N M1).
Lemma zmap_example4 M r:
  let N := zprod2 M (singleton r) in
    ~ (inc r M) -> disjoint M N ->
    ziequivalent M N.
Lemma zequiv_example4 M N: exists M',
  disjoint N M' & ziequivalent M M'.
```

19. Zermelo deduces that there is no set containing all sets equivalent to M, since if T is such a set, we have a set x equivalent to M disjoint from $\mathfrak{S}\mathrm{T}$, thus cannot be in T (note that this argument does not hold if x is empty, so that we start with a lemma: if M is empty, so is x).

```
Lemma zequiv_empty M: ziequivalent M emptyset -> M = emptyset.

Lemma zequiv_no_graph M: nonempty M ->
  ~ (exists S, forall M',  ziequivalent M M' -> inc M' S).
```

21. Zermelo says that is "definite" the existence of a set R disjoint form both sets M and N and equivalent to them. This justifies the following definition of "mediately equivalent". This is an equivalence relation.

```
Definition zequiv  M N := exists R, ziequivalent M R &  ziequivalent N R.

Lemma zequiv_reflexive M: zequiv M M.
Lemma zequiv_symmetric M N:
   zequiv  M N -> zequiv  N M.
Lemma zequiv_transitive M N P:
   zequiv  M N -> zequiv N  P -> zequiv M P.
```

One can define the cardinal of X to be some set equivalent to X, and the cardinal sum and product of two disjoint sets as the cardinal of the union or product. In order to show $A + (B + C) = (A + B) + C$ it suffices to show that there some mutually sets A', B' and C', equivalent to A, B and C, disjoint from $A \cup B \cup C$.

Chapter 9

The size of one

When I was young, I was intrigued by the following quote of Bourbaki:

Bien entendu il ne faut pas confondre le *terme* mathématique *désigné* (chap. I, § 1, n° 1) par le symbole « 1 » et le mot « *un* » du langage ordinaire. Le terme désigné par « 1 » est égal, en vertu de la définition donnée ci-dessus, au terme désigné par le symbole

$$\begin{aligned}(*)\qquad \tau_Z((\exists u)(\exists \mathrm{U})(u = (\mathrm{U}, \{\emptyset\}, \mathrm{Z}) \text{ and } \mathrm{U} \subset \{\emptyset\} \times \mathrm{Z}\\ \text{and } (\forall x)((x \in \{\emptyset\}) \implies (\exists y)((x, y) \in \mathrm{U}))\\ \text{and } (\forall x)(\forall y)(\forall y')(((x, y) \in \mathrm{U} \text{ and } (x, y') \in \mathrm{U}) \implies (y = y'))\\ \text{and } (\forall y)((y \in \mathrm{Z}) \implies (\exists x)((x, y) \in \mathrm{U})))).\end{aligned}$$

Une estimation grossière montre que le terme ainsi *désigné* est un assemblage de plusieurs dizaines de milliers de signes (chacun de ces signes étant l'un des signes τ, $\Box$, $\vee$, $\neg$, $=$, $\in$, $\supset$).

English translation is ([3, p. 158]): The mathematical *term denoted* (Chapter 1, § 1, no. 1) by the symbol "1" is of course not to be confused with the *word* "one" in ordinary language. The term denoted by "1" is equal, by virtue of the definition above, to the term denoted by the symbol (*). As a rough estimate, the term so *denoted* is an assembly of several tens of thousands of signs (each of which is one of τ, $\Box$, $\vee$, $\neg$, $=$, $\in$, $\supset$).

The same expression appears in the English version and in the French one (except that "and" is replaced by "et" in French). By definition, 1 is $\mathrm{Card}(\{\emptyset\}) = \tau_Z(\mathrm{Eq}(\{\emptyset\}, \mathrm{Z}))$.

If 1 is a big object, then how big is 2? or the set N of integers, or the set R of real numbers? If $\mathrm{P}(\mathrm{X}, n)$ is: $\mathrm{X} = (x, y, z)$ and $x^n + y^n = z^n$ and $x \neq 0$ and $y \neq 0$ and $z \neq 0$, what is the size of the formula $(\forall n)(n \in \mathrm{N} \implies (\exists \mathrm{X})(\mathrm{X} \in \mathrm{R}^3 \text{ and } \mathrm{P}(\mathrm{X}, n)))$? If this is an assembly with millions of signs, how big will be a proof of it? If billions of signs are needed, how can one check the proof? I am convinced that it is a bad idea to try to reduce the object under consideration to its basic components (a kind of normal form) and apply low-level theorems to it; Fermat's theorem cannot be proved by exhibiting an assembly that is a proof in the sense of Bourbaki. On the other hand, the estimation of Bourbaki shows that it should be possible to print the whole assembly denoting 1 on an A0-size poster. Alas, the estimation is wrong.

First comments. For simplicity, we shall write Y instead of $\{\emptyset\}$. The formula is hence

$$
\begin{aligned}
(**)\qquad \tau_Z((\exists u)(\exists U)(\quad & u = (U,Y,Z)\\
& \text{and } U \subset Y \times Z\\
& \text{and } (\forall x)((x \in Y) \implies (\exists y)((x,y) \in U))\\
& \text{and } (\forall x)(\forall y)(\forall y')(((x,y) \in U \text{ and } (x,y') \in U) \implies (y = y'))\\
& \text{and } (\forall y)((y \in Z) \implies (\exists x)((x,y) \in U))\\
& \text{and } (\forall x)(\forall x')(\forall y)(((x,y) \in U \text{ and } (x',y) \in U) \implies (x = x')))).
\end{aligned}
$$

This is of the form $\tau_Z(W_Z)$ where W_Z is $(\exists u)(\exists U)\,W$, and where W is of the form P1 and P2 and P3 and P4 and P5 and P6. Properties P1 and P2 say that u is a correspondence with source Y, target Z and graph U, properties P3 and P4 say that u is a function, property P5 that u is injective, and P6 that u is injective. In other words, W says that U is the graph of a bijection with source Y and target Z. Our formula is thus $\tau_Z(\mathrm{Eq}(Y,Z))$, as it should be. Note that P6 is missing in (*). In fact, if $(x,y) \in U$, then $x \in Y$, hence $x = \emptyset$, so that P6 is a consequence of other relations. Hence, the term designed by (*) is *equal* but not *identical* to 1. In what follows, we shall discuss the size of (**).

The case of Coq. The expression W has the form

```
exists u : Set,
   exists U : Set,
     u = J (J U emptyset) Z &
     (forall x : Set, inc x U -> inc x (record Y (fun _ : Set => Z))) &
     (forall x : Set, inc x Y -> exists y : Set, inc (J x y) U) &
     (forall x y y' : Set, inc (J x y) U -> inc (J x y') U -> y = y') &
     (forall y : Set, inc y Z -> exists x : Set, inc (J x y) U) &
     (forall x x' y : Set, inc (J x y) U -> inc (J x' y) U -> x = x')
```

Expanding everything but J gives:

```
exists u : Set,
  exists U : Set,
    u = J (J U emptyset) Z &
    (forall x : Set,
     (exists a : U, Ro a = x) ->
     exists  a : record (IM (fun _ : one_point => emptyset))
             (fun _ : Set => Z), Ro a = x) &
    (forall x : Set,
     (exists a : IM (fun _ : one_point => emptyset), Ro a = x) ->
     exists y : Set, exists a : U, Ro a = J x y) &
    (forall x y y' : Set,
     (exists a : U, Ro a = J x y) -> (exists a : U, Ro a = J x y') -> y = y') &
    (forall y : Set,(exists a : Z, Ro a = y) ->
     exists x : Set, exists a : U, Ro a = J x y) &
    (forall x x' y : Set, (exists a : U, Ro a = J x y) ->
     (exists a : U, Ro a = J x' y) -> x = x')
```

Because of our implementation of τ in Coq, the full expansion of 1 is three times as big as this quantity. This gives a total of 500 tokens (we count forall, exists, Set, etc, as a single token). In [4], the symbol that looks like ⊃ has been withdrawn, and a pair is no more a primitive. As explained in the first part of this document, we do not use the same implementation as

Bourbaki. In particular, we use a primitive object two_points which is a set with two distinct elements. The expansion of the last line of the previous code is then:

```
(exists a : U,
  Ro a = IM (fun t : two_points =>
        match t with
        | two_points_a => IM (fun _ : one_point => x')
        | two_points_b =>
           IM (fun t0 : two_points =>
              match t0 with
              | two_points_a => emptyset
              | two_points_b => IM (fun _ : one_point => y)
               end)
        end)) -> x = x')
```

If we estimate this as 50 tokens, this gives a total size of one that is less than 2000 tokens. For Bourbaki, a correspondence is an object that satisfies P1 and P2 (namely $u = (U, Y, Z)$ and $U \subset Y \times Z$), and for us, a correspondence is a data structure, that has three fields (source, target and graph) and some properties; the quantity U should be replaced by graph u, whose normal form is let (_, _, graph) := u in graph. Denoting by u the triple formed of these three objects, by U, Y and Y the components of the triples, then U is a graph, whose domain is a subset of Y and whose range is a subset of Z. This is equivalent to $U \subset Y \times Z$. This is also equivalent to say that the graph of the correspondence is a subset of the product of the source and the range, but becomes much larger if we expand everything that can be expanded. We estimate that the number of tokens, after full expansion, is roughly 16000.

In Version 4, we change out mind and decided that pairs are defined by an axiom. This means that J is unexpandable. This reduces the size of almost everything. For instance, this is the normal form of P y.

```
chooseT
   (fun x : Set =>
    (exists x0 : Set, exists y0 : Set, y = J x0 y0) ->
    exists y0 : Set, y = J x y0 &
    ((exists x0 : Set, exists y0 : Set, y = J x0 y0) -> False) ->
    x = emptyset) (nonemptyT_intro emptyset)
```

In this new version, 1 becomes small enough in order to study its structure.

It contains 197 exists, 184 Set, 117 Ro, 96 J, 90 graph, 64 fun, 75 primitives like IM, 59 emptyset, 42 keywords (like return), 28 chooseT, 28 False, 21 forall, 724 identifiers (like x, y), 967 punctuation signs (like ->), 255 operators (like =) (parentheses and commas are not counted). Total: 3100 tokens.

Later on, we removed the axiom of the pair, and the call to the axiom of choice in the definition of the projectors. For instance $P = \bigcup\bigcap$. We show here the normal form of P. The normal form of one with this definition is huge.

```
P= union (intersection x)
= IM (fun i : Union_integral (IM
     (fun z : Zorec (fun a : IM
        (fun i : Union_integral x => Ro
           (let (Union_param, Union_elt) as u return
              (Ro (let (Union_param, _) := u in Union_param)) := i in
                   Union_elt)) =>
```

```
            forall z : Set,
              (exists a0 : x, Ro a0 = z) ->
              exists a0 : z, Ro a0 = Ro a) =>
      let (a, _) := z in Ro a)) =>
Ro (let (Union_param, Union_elt) as u
       return (Ro (let (Union_param, _) := u in Union_param)) :=
       i in
    Union_elt))
```

Preliminary computations. Let's try to count exactly the numbers of signs of one in Bourbaki. Remember that the empty set is the following list of signs:

$$\tau\neg\neg\neg\in\tau\neg\neg\in\square\square\square$$

It is easy to count them: there are 12 signs. The definition of the empty set is $\tau_x((\forall y)(y\notin x))$. If P is a formula we denote by L[P] its length; assume that P depends on z and has α signs and β other signs. Remember that $(\exists z)P(z)$ is $(\tau_z P|z)P$, this is the formula obtained by replacing all z by $\tau_z P$. Thus we have

$$L[(\exists z)P(z))] = (\alpha+1)(\alpha+\beta).$$

Since $(\forall z)P(z)$ is $\neg(\exists z)(\neg(Pz))$, we get

$$L[(\forall z)P(z)] = 1+(\alpha+1)(\alpha+\beta+1).$$

In the case of $(\forall y)(y\notin x)$, we have $\alpha=1$ and $\beta=4$, the length is 11, hence $L[\emptyset]=12$. The following formulas are obvious:

$$L[A \text{ or } B] = 1+L[A]+L[B]$$

$$L[A \text{ and } B] = 4+L[A]+L[B]$$

$$L[A \implies B] = 2+L[A]+L[B]$$

$$L[A \iff B] = 8+2L[A]+2L[B]$$

The set of all x such that $P(z)$ is $\tau_y((\forall z)(z\in y) \iff P)$ hence

$$L[\{z,P(z)\}] = 4\alpha^2+36\alpha+47+(4\alpha+6)\beta.$$

The length of $z=x$ or $z=y$ is $3+x+y+2z$, and since $\{x,y\}$ is the set of all z such that $z=x$ or $z=y$ we get

$$L[\{x,y\}] = 177+14x+14y.$$

Since $\{\emptyset\}=\{\emptyset,\emptyset\}$ we get

$$L[\{\emptyset\}] = 513.$$

Assume that W has length $\alpha+\beta U+\gamma Z+u$. The size of $(\exists U)W$ is

$$(\beta+1)(\alpha+\beta+\gamma Z+u);$$

the size of $(\exists u)(\exists U)W$ is

$$(\beta+1)(\beta+2)(\alpha+\beta+\gamma Z+1);$$

hence the size of one is

$$1+(\beta+1)(\beta+2)(\alpha+\beta+\gamma+1).$$

Size of a triple. We estimate here the length of the expression $u = (\mathrm{U},\mathrm{Y},\mathrm{Z})$, assuming that the triple is the pair of pairs $((\mathrm{U},\mathrm{Y}),\mathrm{Z})$. Since the pair (x,y) is $\{\{x\},\{x,y\}\}$ we have

$$\mathrm{L}[(x,y)] = 5133 + 588x + 196y.$$

$$\mathrm{L}[((x,y),z)] = 3023337 + 345744x + 115248y + 196z$$

$$\mathrm{L}[u = (\mathrm{U},\mathrm{Y},\mathrm{Z})] = 62\,145\,562 + 345\,744\mathrm{U} + 196\mathrm{Z} + u$$

Let α be the constant that appears here, and β the coefficient of U. Since the size of one is at least $\alpha\beta^2$, this formula shows that the size is a bit larger than what Bourbaki claims. Instead of "several tens of thousands", it is at least 10^{18}.

The English version of Bourbaki (as well as the 1956 French edition) has the special symbol that looks like $\supset$ and creates a pair. We use here the notation L', when we count the size of the English one; we have

$$\mathrm{L}'[u = (\mathrm{U},\mathrm{Y},\mathrm{Z})] = u + \mathrm{U} + \mathrm{Z} + 516.$$

Size of a graph. We try to find the size of expression P2, namely $\mathrm{U} \subset \mathrm{Y} \times \mathrm{Z}$. If B is $\mathrm{Y} \times \mathrm{Z}$, this expression is $(\forall z)(z \in \mathrm{A} \implies z \in \mathrm{B})$, its size is $22 + 3\mathrm{A} + 3\mathrm{B}$. The size of "$z = (x,y)$ and $x \in \mathrm{Y}$ and $y \in \mathrm{Z}$" is $z+2x+2y+\mathrm{Y}+\mathrm{Z}+12$. If this is P, then $\mathrm{Y} \times \mathrm{Z}$ is the set of all z so that there exists x such that there exists y such that P. Quantifying $\exists y$ gives $42 + 6x + 3\mathrm{Y} + 3\mathrm{Z} + 3z$, quantifying $\exists x$ gives $336 + 21(\mathrm{Y} + \mathrm{Z} + z)$. Taking the set of all z gives $32807 + 1890(\mathrm{Y} + \mathrm{Z})$.

$$\mathrm{L}'[\mathrm{U} \subset \mathrm{Y} \times \mathrm{Z}] = 98443 + 3\mathrm{U} + 5670(\mathrm{Y} + \mathrm{Z});$$

$$\mathrm{L}'[\mathrm{P2}] = 3\,007\,153 + 3\mathrm{U} + 5670\mathrm{Z}.$$

For the French version, the length of P is

$$589x + 5144 + 197y + z + \mathrm{Y} + \mathrm{Z}.$$

Quantifying over y gives

$$1057518 + 198\mathrm{Y} + 198\mathrm{Z} + 116622x + 198z.$$

Quantifying over x gives

$$136931729220 + 23091354(\mathrm{Y} + \mathrm{Z} + z).$$

Taking the set of all z gives

$$\mathrm{L}[\mathrm{Y} \times \mathrm{Z}] = 12649889797944532895 + 2132842656761388(\mathrm{Y} + \mathrm{Z});$$

$$\mathrm{L}[\mathrm{U} \subset \mathrm{Y} \times \mathrm{Z}] = 37949669393833598707 + 3\mathrm{U} + 6398527970284164(\mathrm{Y} + \mathrm{Z});$$

$$\mathrm{L}[\mathrm{P2}] = 41232114242589374839 + 3\mathrm{U} + 6398527970284164\mathrm{Z}.$$

This number is so big that it is impossible to put in a computer the full expansion of $\mathrm{U} \subset \{\emptyset\} \times \mathrm{Z}$.

Size of a bijection. Let's try to find the size of the expressions P2, P3, P4, P5 and P6. They are similar. We start with $(x,y) \in \mathrm{U}$, the size is

$$5134 + 588x196y + \mathrm{U};\ \text{or}\ 2 + x + y + \mathrm{U}.$$

The size is $(\exists y)((x,y) \in \mathrm{U})$ is

$$1050010 + 197\mathrm{U} + 115836x \text{ or } 6 + 2\mathrm{U} + 2x.$$

The size of $x \in \mathrm{Y} \implies (\exists y)((x,y) \in \mathrm{U})$ is

$$1050013 + 197\mathrm{U} + 115837x + \mathrm{Y} \text{ or } 9 + 2\mathrm{U} + 3x + \mathrm{Y}.$$

The size of $(\forall x)(x \in \mathrm{Y} \implies (\exists y)((x,y) \in \mathrm{U}))$ is

$$135049848139 + 22820086\mathrm{U} + 115838\mathrm{Y} \text{ or } 53 + 8\mathrm{U} + 4\mathrm{Y}.$$

The size of P3 is

$$135109273033 + 22820086\mathrm{U} \text{ or } 2105 + 8\mathrm{U}.$$

The size of $(x,y) \in \mathrm{U}$ and $(x,y') \in \mathrm{U}$ is

$$10272 + 1176x + 196y + 196y' + 2\mathrm{U} \text{ or } 8 + 2x + y + y' + 2\mathrm{U};$$

The size of $(x,y) \in \mathrm{U}$ and $(x,y') \in \mathrm{U} \implies y = y'$ is

$$10275 + 1176x + 197y + 197y' + 2\mathrm{U} \text{ or } 11 + 2x + 2y + 2y' + 2\mathrm{U};$$

The size of $(\forall y')((x,y) \in \mathrm{U}$ and $(x,y') \in \mathrm{U} \implies y = y')$ is

$$2073655 + 396\mathrm{U} + 34848x + 39006y \text{ or } 43 + 6\mathrm{U} + 6x + 6y.$$

The size of $(\forall y)(\forall y')((x,y) \in \mathrm{U}$ and $(x,y') \in \mathrm{U} \implies y = y')$ is

$$82408606635 + 15446772\mathrm{U} + 9082701936x \text{ or } 351 + 42\mathrm{U} + 42x.$$

Finally the size of P4 is

$$830988285585569103965 + 14029842596479736\mathrm{U} \text{ or } 16943 + 1806\mathrm{U}.$$

The size of $(\exists x)((x,y) \in \mathrm{U})$ is

$$3370258 + 589\mathrm{U} + 115444y \text{ or } 6 + 2\mathrm{U} + 2y.$$

The size of $y \in \mathrm{Z} \implies (\exists x)((x,y) \in \mathrm{U})$ is

$$3370261 + 589\mathrm{U} + 115445y + \mathrm{Z} \text{ or } 9 + 2\mathrm{U} + 3y + \mathrm{Z}.$$

Hence the size of P5 is

$$402410930323 + 67997694\mathrm{U} + 115446\mathrm{Z} \text{ or } 53 + 8\mathrm{U} + 4\mathrm{Z}.$$

The size of $(x,y) \in \mathrm{U}$ and $(x',y) \in \mathrm{U} \implies x = x'$ is

$$10275 + 589x + 589x' + 392y + 2\mathrm{U} \text{ or } 11 + 2x + 2x' + 2y + 2\mathrm{U}.$$

The size of $(\forall y)((x,y) \in \mathrm{U}$ and $(x',y) \in \mathrm{U} \implies x = x')$ is

$$4192525 + 786\mathrm{U} + 231477(x + x') \text{ or } 43 + 6\mathrm{U} + 6x + 6x'.$$

The size of $(\forall x')(\forall y)((x,y) \in \mathrm{U}$ and $(x',y) \in \mathrm{U} \implies x = x')$ is

$$1024059366435 + 181941708\mathrm{U} + 53581833006x \text{ or } 351 + 42\mathrm{U} + 42x.$$

Hence the size of P6 is

$$57741990789964425582095 + 9748770215064355956\mathrm{U} \text{ or } 16943 + 1806\mathrm{U}.$$

The size of the expressions P3 to P6 is hence

$$58572979076087514889416 + 9889068641119971100\mathrm{U} + 115446\mathrm{Z} \text{ or } 36044 + 3628\mathrm{U} + 4\mathrm{Z}.$$

This gives, for the French version $\alpha = 58614211190330166409837$ $\beta = 9889068641120316847$ $\gamma = 6398527970399806$, and for the English version: This gives $\alpha = 3033733$, $\beta = 36732$ and $\gamma = 5675$.

Conclusion. The statement, at the start of the chapter, quoted from the 1956 edition of Bourbaki, translated in English in 1968, is grossly wrong since the size is not several thousands, but in fact

$$4\,150\,763\,939\,024\,663.$$

Moreover, the 1970 decision to remove the axiom of the ordered pair had a dramatic effect on the size of one, that increases to

$$573306704401798033758237640367224116154353941968147665929668 9.$$

Chapter 10

Exercises

There are 111 exercises for this chapter (plus 9 concerning direct and inverse limits). We give here the number of lines of the code of all those that are solved.

Section 1. 1 (12), 2 (431), 3 (954), 4 (373), 5 (137), 6 (975), 7 (137), 8 (37), 9 (183), 10 (128), 11 (482), 12 (14), 13 (107), 14 (79), 15 (348), 16 (307), 17 (326), 18 (626), 19 (191), 20 (497), 21 (627), 22 (628), 23 (274*), 24(532*). Ex 23 is incomplete, as well as ex 24.

Section 2. 1 (278), 2 (137), 3 (143), 4 (136), 5 (14), 6 (294), 7 (775) , 9 (38), 10 (39), 11 (193), 12 (420), 13 (3650), 14(400), 15(1200), 16(190), 17 (1460), 18(550), 19(*), 20(220).

Section 3. 1 (78), 2 (59), 3 (276), 4 (?), 5 (85), 6 (18).

Section 4. 1 (100), 2 (51), 3 (21), 4 (193), 5 (774), 6 (642*), 7 (270), 8 (735).

Section 6. 1 (134), 2 (49), 3 (55), 4 (73), 5 (299), 6 (108), 7 (131), 8 (94), 9 (99), 10 (650), 11(105), 12 (900*), 13(340), 14(500), 15 (308).

We present here a simple (160 lines) proof of Zermelo's theorem. It says that, for any set E, there is a well-ordering whose support is E; it assumes existence of some function r such that, whenever $A \subset E$ and A is nonempty, then $r(A) \in A$. The ordering we construct will be such that $r(A)$ is the least element of A. This ordering is unique. Assume the problem solved. Let Ω be the set of all intervals $[x, \rightarrow [$, together with the empty set (note that Ω is the set of complements of the set of segments of E). Let $p(A) = A - \{r(A)\}$, where $r(A)$ is the least element of A, and $p(\emptyset) = \emptyset$. A chain for p is a set F satisfying (a) $F \subset \mathfrak{P}(E)$, (b) $E \in F$, (c) F is stable by p and (d) F is stable by intersection. The set Ω is the smallest chain. Condition (b) holds as E has a least element; condition (c) holds since $p([x, \rightarrow [) =]x, \rightarrow [$; if x is the greatest element of E, this set is empty, otherwise has a least element y, thus is $[y, \rightarrow [$. Finally, consider a set whose elements are of the form $[x_i, \rightarrow [$. If the set of x_i is unbounded, the intersection of the intervals is empty, otherwise, the set of upper bounds has a least element y, and the intersection of the intervals is $[y, \rightarrow [$. We leave it to the reader to show that Ω is contained in every chain. If A is a subset of E we define $d(A)$ to be the intersection of all elements of Ω that contain A. Assume A non-empty; then $d(A)$ is a non-empty element of Ω, thus of the form $[x, \rightarrow [$, and x is the least element of A. In particular, if we consider $R(x) = d(\{x\})$, then $R(x) = [x, \rightarrow [$. This implies that $x \leq y$ if and only if $R(y) \subset R(y)$. This shows that the ordering on E is uniquely defined by Ω, thus by p, thus by r.

We start with defining the ordering.

```
Definition worder_of (E:Set): Set.
move=> E.
set p:=  fun a => complement a (singleton (rep a)).
set (chain:= fun F => sub F (powerset E) & inc E F &
    (forall A, inc A F -> inc (p A) F)
    & (forall A, sub A F -> nonempty A -> inc (intersection A) F)).
set om :=  intersection (Zo (powerset (powerset E)) chain).
set d:= fun p => intersection (Zo om (fun x => sub p x)).
set R := fun x => d (singleton x).
set r:= graph_on (fun x y => (sub (R y) (R x))) E.
exact r.
Defined.
```

Let's prove the theorem. We have first to introduce local variables such as m, p, R, etc.

```
Lemma Zermelo_ter: forall E,   (* 166 *)
  worder (worder_of E) & substrate (worder_of E) = E.
Proof.
move=> E.
rewrite /worder_of.
set p:=  fun a => complement a (singleton (rep a)).
set (chain:= fun F => sub F (powerset E) & inc E F &
    (forall A, inc A F -> inc (p A) F)
    & (forall A, sub A F -> nonempty A -> inc (intersection A) F)).
set om := intersection _.
set (R:= fun p => intersection (Zo om (fun x => sub (singleton p) x))).
have omv: om =  intersection (Zo (powerset (powerset E)) chain) by done.
set res := graph_on _ _.
set (m:= fun a => forall x, inc x om -> sub x a \/ sub a x).
```

We have $p(\mathrm{A}) \subset \mathrm{A}$, and if $\mathrm{A} = \emptyset$, then $p(\mathrm{A}) = \mathrm{A}$.

```
have pe: p emptyset = emptyset.
  by empty_tac x xe; move: xe; rewrite /p; srw; case; case; case.
have sp: forall a, sub (p a) a by move=> t; rewrite /p; apply sub_complement.
```

We show here that the powerset of E is a chain. This will have as a consequence that Ω is well-defined.

```
have cp:chain (powerset E).
  split; fprops; split; first  by apply powerset_inc; fprops.
  split; first by  move=> A; aw; move=>  AE; apply sub_trans with A.
  move=> A AP [x xA]; move: (AP _ xA); aw => xE.
  move=> t ti;  apply xE; apply (intersection_forall ti xA).
```

We show here that Ω is a chain.

```
have co :chain om.
 rewrite omv.
  have aux: (nonempty (Zo (powerset (powerset E)) chain)).
    by exists (powerset E); apply Z_inc; aw; fprops.
  rewrite /om; red; ee.
        by rewrite /om; apply intersection_sub; apply Z_inc; aw; fprops.
      apply intersection_inc=>//; move=>y; rewrite Z_rw/chain; aw; intuition.
    move=> A Ai; apply intersection_inc=>//.
```

```
    move=> y yi; move: (yi); rewrite Z_rw; move=> [_ [_[_ [q _]]]]; apply q.
    apply (intersection_forall Ai yi).
  move=> A sAi neA;  apply intersection_inc=> //.
  move=> y yi; move: (yi); rewrite Z_rw; move=> [_ [_[_ [_ q]]]]; apply q=>//.
  move=>t tA; move: (sAi _ tA) => ti.
  by apply (intersection_forall ti yi).
```

We show here that if C is any chain, then $\Omega \subset C$.

```
move: (co)=> [sop [Eo [po io]]].
have cio: forall x, chain x -> sub om x.
  move=> x xc;  rewrite / om; apply intersection_sub; apply Z_inc =>//.
  by aw; move:xc; rewrite /chain; intuition.
```

Now comes a big part of the proof. Let $m(A)$ be the property that for all $X \in \Omega$, we have either $X \subset A$ or $A \subset X$. We pretend that $m(A)$ holds for all elements of Ω. In fact, the set of elements that satisfy m is a chain, thus has to be Ω. The non-obvious point is to show that $m(A)$ implies $m(A')$ where A' is short for $p(A)$. In fact, let T be the set of elements B of Ω such that $B \subset A'$ or $A \subset B$. It contains E and is stable by intersection (if for all i, $A \subset T_i$, then $A \subset \bigcap T_i$, otherwise there is i such that $T_i \subset A'$ and $\bigcap T_i \subset A'$). Assume $B \in T$. If $B \subset A'$, then $B' \subset A'$. Since B' is in Ω, we have either $A \subset B'$ or $B' \subset A$. In the first case, $A' \subset B'$. Now we have $B' \subset A \subset B$. Let b' be $r(B')$. If $b' \in A$ then $B \subset A$, then $A = B$, hence $B' \subset A'$. Otherwise $A \subset B'$. In summary, T is a chain, thus must be Ω.

```
have am: om = Zo om m.
  apply extensionality; last by apply Z_sub.
  apply cio; red; ee.
        apply sub_trans with om; [by apply Z_sub| apply sop].
      by apply Z_inc=>//; move=> x xom; left;move: (sop  _ xom); aw.
    move=> A; rewrite Z_rw; move=> [Aom mA].
    apply Z_inc; first by apply (po _ Aom).
    have aux: (sub om (Zo om (fun x=> sub x (p A) \/ sub A x))).
      apply cio; red; ee.
            by apply sub_trans with om; first by apply Z_sub.
          by apply Z_inc=>//; right; move: (sop _ Aom); aw.
        move => B; rewrite Z_rw; move => [Bom ors].
        apply Z_inc; first by apply (po _ Bom).
        case ors => orsi.
          left;  apply sub_trans with B =>//; apply sp.
        case (mA _ (po _ Bom)) => aux; last by auto.
        case (inc_or_not (rep B) A)=> aux2.
          have BA: (sub B A).
            rewrite /p in aux2; move=> t tB.
            case (equal_or_not t (rep B)).
              by move=> ->.
            by move=> trB; apply aux;  rewrite /p; srw;  aw.
          have: (B = A) by apply extensionality.
          by move=> ->; intuition.
        right; move=> t tA; rewrite /p; srw; aw.
        split; [ by apply orsi| dneg trB; ue].
      move=> B sB neB; apply Z_inc.
        apply io =>//;  apply: sub_trans; [eexact sB| apply Z_sub].
      case (p_or_not_p (exists x, inc x B & sub x (p A))) => aux.
        move: aux=> [x [xB xp]]; left; move=> t ti; apply xp.
        apply (intersection_forall ti xB).
```

```
      right; move=> t tA;  apply intersection_inc=>//.
      move=> y yB; move: (sB _ yB); rewrite  Z_rw; move=> [yom ors].
      case ors; last by apply.
      by move => sy; elim aux; ex_tac.
    move=> x xom; case (mA _ xom) => hyp.
      move: (aux _ xom);  rewrite Z_rw; move=> [_ xpA].
      case xpA=>xpB; first by auto.
      have Ax: (A = x) by apply extensionality.
      rewrite Ax; right; apply sp.
    by right;  apply sub_trans with A=>//; apply sp.
  move=> A sAZ neA; apply Z_inc.
    apply io =>//; apply sub_trans with (Zo om m) =>//; apply Z_sub.
  move=> x xom.
  case (p_or_not_p (exists y, inc y A & sub y x)) => hyp.
    move: hyp=> [y [yA yx]]; right; move => t ti.
    apply yx; apply (intersection_forall ti yA).
  left; move=> t tx; apply intersection_inc=>//.
  move=> y yA; move: (sAZ _ yA); rewrite Z_rw; move=> [yom my].
  case (my _ xom);[ by apply | move=> yx; elim hyp; ex_tac; apply Z_sub].
```

Consequence: if A and B are in Ω, then A ⊂ B or B ⊂ A.

```
have st: forall a b, inc a om -> inc b om -> sub a b \/ sub b a.
  move=> a b; rewrite {2} am Z_rw; move=> aom [bom ba]; apply (ba _ aom).
set  d:= fun p => intersection (Zo om (fun x => sub p x)).
```

We pretend here that $d(p)$ is the least element of Ω that contains p; this amounts to the three following conditions: if $p \subset \mathrm{E}$, then $d(p) \in \Omega$, $p \subset d(p)$, and if $p \subset q$, where $q \in \Omega$, then $d(p) \subset q$.

```
have dpo: forall q, sub q E -> inc (d q) om.
  by move=> q qE; rewrite /d; apply io;[ apply Z_sub| exists E;apply Z_inc].
have pdp: forall q, sub q E -> sub q (d q).
  rewrite /d=> q qE t tq; apply intersection_inc.
    by exists E; apply Z_inc.
  by move => y; rewrite Z_rw; move=>[_]; apply.
have dpq: forall q r, inc r om -> sub q r -> sub q E-> sub (d q) r.
  by rewrite /d=> q r0 rom qr qE; apply intersection_sub; apply Z_inc.
```

Fix a set $\mathrm{P} \subset \mathrm{E}$ and consider $q = r(d(\mathrm{P}))$. We pretend that if $q \in d(\mathrm{P})$, then $q \in \mathrm{P}$. Recall that $d(\mathrm{P})' = d(\mathrm{P}) - \{q\}$. If q is not in P, we get $\mathrm{P} \subset d(\mathrm{P})'$, and by minimality, $d(\mathrm{P}) = d(\mathrm{P})'$, absurd. *We assume from now on that r is a choice function,* i.e., $r(\mathrm{X}) \in \mathrm{X}$, whenever X is a nonempty subset of E. Thus $q \notin \mathrm{P}$ implies that $d(\mathrm{P}) = \emptyset$, hence that P is empty (since $\mathrm{P} \subset d(\mathrm{P})$).

```
have rdq: forall q, sub q E -> nonempty q -> inc (rep (d q)) q.
  move=> q qE neq; case (inc_or_not (rep (d q)) q)=>// ni.
  have aux: (sub q (p (d q))).
    by rewrite /p=> t tq; srw; aw; split; [ apply (pdp _ qE)| dneg tr; ue].
  move: (dpq _ _ (po _ (dpo _ qE)) aux qE).
  rewrite /p; case  (emptyset_dichot (d q)).
    move=> dqe; rewrite dqe pe in aux.
    by move: neq=> [t tq]; elim (emptyset_pr (x:= t)); apply aux.
  move=> ned; move: (nonempty_rep ned) => rd dc; move: (dc _ rd); srw.
  aw; intuition.
```

We have seen that if $Q = d(P)$ and P is non-empty, then $r(Q) \in P$. Conversely, if $Q \in \Omega$ and $q = r(Q) \in P$, then $Q = d(P)$. In fact, if $d(P) \subset Q'$, one gets $q \in P \subset d(P) \subset Q'$, absurd; so that $Q' \subset d(P) \subset Q$. The conclusion follows since $q \in d(P)$.

```
have qdp: forall q r, inc r om -> sub q r -> inc (rep r) q -> r = d q.
  move => q r rom qr rq.
  have spE: sub q E.
    by apply sub_trans with r=>//;  apply powerset_sub; apply sop.
  move: (dpq _ _ rom qr spE) => sdqr.
  case (st _ _  (dpo _ spE) (po _ rom)) => ch.
    move: (pdp _ spE) => qdp.
    have:(inc (rep r) (p r)) by apply ch; apply qdp.
    rewrite /p; srw;  aw; intuition.
  apply extensionality =>// t tr.
  case (equal_or_not t (rep r)).
    by move=> ->; apply (pdp _ spE).
  by move=> tnr; apply ch; rewrite /p; srw; aw; auto.
```

Denote by $R(a)$ the quantity $d(\{x\})$. For $x \in E$, this is in Ω and contains x. The choice function has no choice here: $r(R(x)) = x$. As a consequence R is injective.

```
have Rp: forall x, inc x E ->
    (inc (R x) om & inc x (R x) &  rep (R x) = x).
  rewrite /R=> x xE.
  have p1: sub (singleton x) E by apply sub_singleton.
  move: (pdp _ p1) => p2; ee; first by  apply dpo.
  have nesi: (nonempty (singleton x)) by fprops.
  by move: (rdq _ p1 nesi); aw.
have Ri:forall x y, inc x E -> inc y E -> R x = R y -> x = y.
  move=> x y xE yE; move: (Rp _ xE)(Rp _ yE).
  by move=> [_ [_ p1]][_[_ p2]] p3; rewrite -p3 in p2; rewrite -p1 -p2.
```

We have $R(r(Q)) = Q$ for $Q \in \Omega$ by uniqueness.

```
have Rrq: forall q, inc q om -> nonempty q -> R (rep q) = q.
  move=> a qom neq; rewrite /R; symmetry; apply qdp =>//.
  by move=> t; aw; move=> ->;  apply nonempty_rep.
  fprops.
```

Fix two elements x and y and define $D = \{x, y\}$. We have $r(R(y)) \in D$ since $r(R(y)) = y$. Assume $D \subset R(y)$. This implies $R(y) = d(D)$. Thus $D \subset R(y)$ and $D \subset R(x)$ implies $R(x) = R(y)$. Lemma: if $x \in R(y)$, then $R(x) \subset R(y)$. The assumption says $D \subset R(y)$. Since Ω is totally ordered by inclusion, we have either our conclusion or $R(y) \subset R(x)$. But this implies $D \subset R(x)$ hence $R(x) = R(y)$.

```
have sRR: forall x y, inc x E -> inc y E -> inc x (R y) -> (sub (R x) (R y)).
  move=> x y xE yE xRy.
  move: (Rp _ xE)(Rp _ yE) => [Rom [xRx rR]] [Rom' [yRy rR']].
 case (st _ _ Rom Rom') =>// hyp.
  have p1:(sub (doubleton x y) (R y)).
    by move=> t td; case (doubleton_or td)=>->.
  have p2: (sub (doubleton x y) (R x)) by apply sub_trans with (R y).
  have p3: (inc (rep (R x)) (doubleton x y)) by rewrite rR; fprops.
  have p4: (inc (rep (R y)) (doubleton x y)) by rewrite rR'; fprops.
  move: (qdp _ _ Rom p2 p3) (qdp _ _ Rom' p1 p4).
  move=> -> ->;  fprops.
```

Since R is injective, the relation $x \leq y$ whenever $R(y) \subset R(x)$ is an ordering on E. The previous remarks says: if $x \in R(y)$ then $y \leq x$.

```
have or:order res.
  rewrite /res; apply order_from_rel1.
      by move=> x y z /= xy yz; apply sub_trans with (R y).
    by move=> u v uE vE vu uv; apply Ri=>//; apply extensionality.
  move => u ue; fprops.
have sr: substrate res = E.
  rewrite /res substrate_graph_on //;  move => u ue; fprops.
```

Given any nonempty subset A of E, the quantity $x = r(d(A))$ is in A. Let $y \in A$, we have $x \leq y$ since $y \in A \subset d(A) = R(x)$.

```
split=>//;split=>//.
move=> x xsr nex.
rewrite sr in xsr.
move: (rdq _ xsr nex) => rdx.
exists (rep (d x)); split.
  by aw;rewrite  sr.
aw; move=> a ax; aw; last by ue.
rewrite /res /gle graph_on_rw1;ee.
apply sRR; try apply xsr=>//.
move: ((pdp _ xsr) _ ax)=> adx.
have ne: (nonempty (d x)) by exists a.
rewrite Rrq //.
by apply dpo.
Qed.
```

We show here that $\mathfrak{P}(A \cap B) = \mathfrak{P}(A) \cap \mathfrak{P}(B)$. We first note that any subset of both A and B is a subset of $A \cap B$, then that, if $A \subset B$ then $\bigcup A \subset \bigcup B$ and $\mathfrak{P}(A) \subset \mathfrak{P}(B)$.

```
Lemma union_monotone3 A B: (* 2 *)
  sub A B -> sub (union A) (union B).
Lemma powerset_mono A B:  (* 2 *)
  sub A B -> sub (powerset A)(powerset B).
Lemma intersection_greatest A B x:  (* 1 *)
   sub x A -> sub x B -> sub x (intersection2 A B).
Lemma powerset_inter A B:  (* 6 *)
   powerset (intersection2 A B) = intersection2 (powerset A)(powerset B).
```

This is example 1 page 148: *Let* $E = \{\alpha, \beta\}$ *be a set whose elements are distinct. It is easily verified that the subset* $\{(\alpha,\alpha),(\beta,\beta),(\alpha,\beta)\}$ *of* $E \times E$ *is the graph of a well-ordering on* E.

The example is now part of the main text, with TPa and TPb instead of α and β. Note that, if $\alpha = \beta$, the set E becomes a singleton, but it is still a well-ordering.

```
Definition example_worder a b:=
  union2(doubleton (J a a) (J b b)) (singleton (J a b)).
Lemma example_worder_related a b x y: (* 8 *)
    related  (example_worder a b) x y <->
    ( (x = a & y = a) \/  (x = b & y = b) \/  (x = a & y = b)).
```

```
Lemma substrate_example_worder a b: (* 12 *)
  substrate (example_worder a b) = doubleton a b.
Lemma example_is_worder a b:  (* 32 *)
  worder (example_worder a b).
```

Let $\mathfrak{F}$ be a set of subsets of A, ordered by inclusion, and such that for every totally ordered subset $\mathfrak{G}$ of $\mathfrak{F}$, the union of the sets of $\mathfrak{G}$ belongs to $\mathfrak{F}$. Then $\mathfrak{F}$ is inductive with respect to the relation $\subset$.

```
Lemma inductive_example1 A F: (* 7 *)
  sub A (powerset F) ->
  (forall S, (forall x y, inc x S -> inc y S -> sub x y \/ sub y x) ->
    inc (union S) A) ->
  inductive_set (inclusion_suborder A).
Lemma inductive_graphs a b:  (* 30 *)
  inductive_set (opposite_order (extension_order a b)).
```

The set $\Phi(E,F)$ of mappings of subsets of E into subsets of F is inductive, with respect to the order "v extends u" between u and v (i.e., the opposite of the extension ordering), because there is a common extension on a totally ordered subset.

We give two variants of the Exercise, showing the advantages of a function to be a set. Here is the old version.

```
Lemma inductive_graphs: forall a b,
  inductive_set (opposite_order (extension_order a b)).
Proof. ir. cp (extension_is_order a b). red. ir. nin H1. ee. awii H2. awii H0.
  assert (Hd: forall i j, inc i X -> inc j X ->
   agrees_on (intersection2 (source i) (source j)) i j).
  ir. cp (H0 _ H3). cp (H0 _ H4).  bwi H5; bwi H6. ee.
  cp (H2 _ _ H3 H4). ufi gge H11. red. ee.
  app intersection2sub_first. app intersection2sub_second.
  awii H11. nin H11; ee. ir. app W_extends. inter2tac.
  ir. sy. app W_extends. inter2tac.
  assert (He:forall i, inc i X -> function_prop i (source i) b).
  ir. red.  cp (H0 _ H3). bwi H4. eee.
  cp (extension_covering _ _ He Hd). nin H3. clear H4. nin H3. ee.
  red in H3. ee. assert (sub (source x) a). rw H5.
  red. ir. nin (unionf_exists H7). nin H8. cp (H0 _ H8). awi H10. ee.
  bwi H10. ee. app H11.
  assert (inc x (set_of_sub_functions a b)). bw. eee.
  exists x. red. ee. aw. ir.  aw. eee. red. cp (H0 _ H9). bwi H10. ee. am.
  am.  cp (H4 _ H9). red in H13. ee.
  red. ir. cp (in_graph_W H10 H16). rwi H17 H16.
  cp (inc_pr1graph_source H10 H16). rw H17. wr (H15 _ H18).  app W_pr3.
  app H13. rw H6. rw H12. fprops.
  app opposite_is_order.
Qed.
```

Here is the new version (ssreflect style).

```
Lemma inductive_graphs: forall a b,
  inductive_set (opposite_order (extension_order a b)).
```

```
Proof.
move => a b.
move: (extension_is_order a b)=> or.
move => X;aw; move=> Xsr [oX ].
have Xs:sub X (substrate (extension_order a b)) by aw.
have oo: order (opposite_order (extension_order a b))
   by  apply opposite_is_order =>//.
aw => aux.
have aag: forall i j, inc i X -> inc j X ->
   agrees_on (intersection2 (source i) (source j)) i j.
  move=> i j iX jX; move: (Xsr _ iX) (Xsr _ jX); bw.
  move=> [fi [si ti]][fj [sj tj]].
  red; ee; [ apply intersection2sub_first | apply intersection2sub_second | ].
  move=> c; aw; move=> [csi csj].
  move: (aux _ _ iX jX); rewrite /gge; aw.
  case;move=> [_ [_ ext]]; [ | symmetry]; apply W_extends =>//.
have afp:forall i, inc i X -> function_prop i (source i) b.
  move=> i iX; move: (Xsr _ iX); bw; red; ee.
move: (extension_covering afp aag) => [[g [[fg [sg tg]]] gp] _].
have ssg: (sub (source g) a).
  rewrite sg => t; aw; srw; move =>[y [yX tsy]].
  by move: (Xsr _ yX); bw; move=> [_ [ssy _]]; apply ssy.
have gsab: (inc g (set_of_sub_functions a b)) by bw; ee.
exists g; red; ee; aw.
move=> y yX; move: (Xsr _ yX); bw; move=> [fy [ssy ty]].
aw; ee.
red; ee; last by rewrite tg ty.
move=> w wg; move: (in_graph_W fy wg) => pg.
rewrite pg in wg; move: (inc_pr1graph_source fy  wg) => ps.
move: (gp _ yX); rewrite /agrees_on; move => [w1 [_ w3]].
by rewrite pg -(w3 _ ps); apply W_pr3 =>//; apply w1.
Qed.
```

The objective here is to give a proof of the Cantor-Bernstein theorem that says that if there is an injection from A into B and an injection from B into A, there is a bijection. (In the main text, we proved this theorem, using an alternate method; it has 55 + 24 lines of proof).

We start with a lemma. Assume that $g : \mathfrak{P}(\mathrm{E}) \to \mathfrak{P}(\mathrm{E})$ is an increasing function. Then g has a fixed-point. Indeed, let A the the set of all x such that $x \subset g(x)$, and U the union of A. If $x \in \mathrm{A}$ then $x \subset \mathrm{U} \subset g(\mathrm{U})$. This gives $\mathrm{U} \subset g(\mathrm{U})$. It implies $g(\mathrm{U}) \in \mathrm{A}$. But this is equivalent to $g(\mathrm{U}) \subset \bigcup \mathrm{S}$, so that $\mathrm{U} = g(\mathrm{U})$.

Consider two injections $f : \mathrm{E} \to \mathrm{F}$ and $g : \mathrm{F} \to \mathrm{E}$. For $\mathrm{X} \subset \mathrm{E}$, consider $h(\mathrm{X}) = \mathrm{E} - g\langle \mathrm{F} - f\langle \mathrm{X} \rangle \rangle$. Since taking the complementary of a set by a function is decreasing and the composition of two decreasing functions is increasing, this function is increasing. If has a fixed point M. We have $\mathrm{E} - \mathrm{M} = g\langle \mathrm{F} - f\langle \mathrm{M} \rangle \rangle$. Let $\mathrm{T} = g\langle \mathrm{F} - f\langle \mathrm{M} \rangle \rangle$. This is the complementary of M. Every element in T is of the form $g(y)$ for some y not of the form $f(x)$ for $x \in \mathrm{T}$. We use the choice function in order to get a function f_2. Note that the choice is limited, this y is unique by injectivity of g. We consider the function f_1 whose value is f on M and f_2 on T.

This function is injective. Assume $f_1(x) = f_1(y)$. If one of x, y is in M and the other one is in T, one image is in $f(\mathrm{M})$, and the other is in the complement, absurd. If both elements are in M, we use injectivity of f. Otherwise $f_1(x) = x'$ where $x = g(x')$ and $f_1(y) = y'$ where

$y = g(y')$. We use injectivity of g. This function is clearly surjective, thus is a bijection $\mathrm{E} \to \mathrm{F}$.

```
Lemma Cantor_Bernstein_aux E (g: Set -> Set): (* 15 *)
  (forall x, sub x E -> sub (g x) E) ->
  (forall x y, sub x y -> sub y E -> sub (g x) (g y)) ->
  (exists m, sub m E & g m = m).
Lemma Cantor_Bernstein2 f g: (* 4* *)
  injection f -> injection g -> source f = target g -> source g = target f ->
  equipotent (source f)(source g).
```

In the main text, we proved the principle of transfinite induction by applying transfinite_principle2. We give here a direct proof. We assume that E is well-ordered, and whenever $x \in \mathrm{E}$, the relation $p(x)$ is a consequence of $\mathrm{H}(x)$ that says that all $y < x$ satisfy p. Assume that there is some x not satisfying p. Then there is a least y not satisfying p. This implies $\mathrm{H}(y)$, thus $p(y)$, absurd.

```
Theorem transfinite_principle_bis r (p:Set-> Prop):
  worder r ->
  (forall x, inc x (substrate r) ->
    (forall y, inc y (substrate r) -> glt r y x -> p y) -> p x) ->
  forall x, inc x (substrate r) -> p x.
Proof.
move => [or wor] hyp x xsr; ex_middle npx.
set (X:=Zo (substrate r) (fun x => ~ p x)).
have neX: (nonempty X) by exists x; rewrite /X; apply: Z_inc.
have Xsr: sub X (substrate r) by rewrite/X; apply: Z_sub.
move:(wor _ Xsr neX)=> [y []]; aw => yX yle.
move: (yX); rewrite /X Z_rw; move=> [ysr npy].
elim npy; apply: hyp =>//.
move=> t tsr ty; ex_middle npt.
have tX: inc t X by rewrite /X; apply: Z_inc.
move: (yle _ tX); aw =>//; move=> nty; order_tac.
Qed.
```

10.1 Section 1

1. *Let* E *be an ordered set in which there exists at least one pair of distinct comparable elements. Show that if* R$\{x, y\}$ *denotes the relation "*$x \in \mathrm{E}$ *and* $y \in \mathrm{E}$ *and* $x < y$*", then* R *satisfies the first two conditions of no. 1 but not the third.*

```
Lemma Exercise1_1 r (E:= substrate r) (* 6 *)
     (s := fun x y => (inc x E & inc y E & glt r x y)) :
    order r ->  (exists x, exists y, x <> y & related r x y)
    -> (transitive_r s &  antisymmetric_r s & ~(reflexive_rr s)).
```

---

2. *(a) Let* E *be a preordered set and let* S$\{x, y\}$ *be an equivalence relation on* E*. Let* R$\{\mathrm{X}, \mathrm{Y}\}$ *denote the relation "*$\mathrm{X} \in \mathrm{E}/\mathrm{S}$ *and* $\mathrm{Y} \in \mathrm{E}/\mathrm{S}$ *and for each each* $x \in \mathrm{X}$ *there exists* $y \in \mathrm{Y}$ *such that* $x \leq y$*". Show that* R *is a preorder relation on* E/S*, called the* quotient *by* S *of the relation*

$x \leq y$. The quotient E/S*, endowed with this preorder relation, is called (by abuse of language; cf Chapter IV, § 2, no. 6) the quotient by* S *of the preordered set* E.

In what follows, we shall denote by $\leq_E$ the given preorder, and by $x \leq_E y$ the associated preorder relation; in the same way, $\leq_Q$ is the quotient preorder and $x \leq_Q y$ its associated relation. In other words, $x \leq_Q y$ is the relation $R\{x, y\}$ defined in the text and $\leq_Q$ is the graph of this relation on E/S.

```
Definition quotient_order_r r s X Y :=
  inc X (quotient s) & inc Y (quotient s) &
  forall x, inc x X -> exists y, inc y Y & gle r x y.
Definition quotient_order r s := graph_on (quotient_order_r r s) (quotient s).
```

First part: we show that $x \leq_Q y$ is a preorder relation.

```
Lemma Exercise1_2a r s: (* 7 *)
  is_equivalence s -> preorder r -> substrate s = substrate r ->
  preorder_r (quotient_order_r r s).
```

We now show that $\leq_Q$ is a preorder on E/S, associated to the relation $x \leq_Q y$.

```
Lemma quotient_order_pr r s x y: (* 2 *)
  related (quotient_order r s) x y <-> quotient_order_r r s x y.
Lemma quotient_is_preorder r s: (* 2 *)
  is_equivalence s -> preorder r -> substrate s = substrate r ->
  preorder (quotient_order r s).
Lemma substrate_quotient_order r s: (* 7 *)
  is_equivalence s -> preorder r -> substrate s = substrate r ->
  substrate (quotient_order r s) = quotient s.
```

(b) Let ϕ be the canonical mapping of E *onto* E/S*. Show that if g is a mapping of the preordered quotient set* E/S *into a preordered set* F *such that $g \circ \phi$ is an increasing mapping, then g is an increasing mapping. The mapping ϕ is increasing if and only if* S *satisfies the following condition*

> *(C) the relations $x \leq y$ and $x \equiv x'$* (mod S) *in* E *imply that there exists $y' \in$ E such that $y \equiv y'$* (mod S) *and $x' \leq y'$.*

Every equivalence relation S *which is* compatible *(in x) with the preorder relation $x \leq y$ (Chapter II, § 6, no. 3) is a fortiori weakly compatible (in x and y) with this relation.*

We consider now a function g such that $g \circ \phi$ is an increasing mapping, and show that g is increasing. Assume $X \leq_Q Y$, and that we want to show $g(X) \leq_F g(Y)$. Since X is a class, there is $x \in X$ hence there is $y \in Y$ such that $x \leq_E y$. Thus $g(\phi(x)) \leq_F g(\phi(y))$. It remains to show that $X = \phi(x)$ and $Y = \phi(y)$.

```
Definition increasing_pre f r r':=
  is_function f & preorder r & preorder r' & substrate r = source f
  & substrate r' = target f  &
  forall x y,  gle r x y  -> gle r' (W x f) (W y f).

Lemma Exercise1_2b1 r s g r': (* 15 *)
  is_equivalence s -> substrate s = substrate r ->
  is_function g -> quotient s = source g ->
  increasing_pre (compose g (canon_proj s)) r r' ->
  increasing_pre g (quotient_order r s) r'.
```

The converse is true if ϕ is increasing, this is equivalent to the following condition.

```
Definition weak_order_compatibility r s:=
  preorder r & is_equivalence s &  substrate s = substrate r &
  forall x y x', gle r x y -> related s x x' -> exists y',
    (related s y y'  & gle r x' y').
```

We show that strong compatibility implies weak compatibility (assume $x \leq y$ and $x \equiv x'$ (mod S) in E, we can take $y' = y$ in $y \equiv y'$ (mod S) and $x' \leq y'$ since y is in the substrate of S).

```
Lemma strong_order_compatibility r s: (* 4 *)
  preorder r -> is_equivalence s -> substrate s = substrate r ->
  (forall x x' y,  gle r x y -> related s x x' -> gle r x' y) ->
  weak_order_compatibility r s.
```

Let's show that the canonical projection is increasing.

```
Lemma compatibility_proj_increasing r s: (* 20 *)
  is_equivalence s -> preorder r -> substrate s = substrate r ->
  (weak_order_compatibility r s <->
   increasing_pre (canon_proj s) r  (quotient_order r s)).
```

(c) Let E_1 *and* E_2 *be two preordered sets. Show that if* S_1 *is the equivalence relation* $\mathrm{pr}_1 z = \mathrm{pr}_1 z'$ *on* $E_1 \times E_2$*, then* S_1 *is weakly compatible in* z *and* t *with the product preorder relation* $z \leq t$ *on* $E_1 \times E_2$ *(but is not usually compatible with this relation in* z *or* t *separately); moreover if* ϕ_1 *is the canonical mapping of* $E_1 \times E_2$ *onto* $(E_1 \times E_2)/S_1$*, and if* $\mathrm{pr}_1 = f_1 \circ \phi_1$ *is the canonical decomposition of* pr_1 *with respect to the equivalence relation* S_1*, then* f_1 *is an isomorphism of* $(E_1 \times E_2)/S_1$ *into* E_1*.*

We show here weak compatibility of the product of two preorder relations on E_1 and E_2 and the equivalence S_1 in $E_1 \times E_2$ defined by $\mathrm{pr}_1 z = \mathrm{pr}_1 z'$.

```
Lemma exercise1_2c1 r1 r2:  (* 11 *)
  preorder r1 -> preorder r2 ->
  weak_order_compatibility (order_product2 r1 r2)
  (first_proj_eq (substrate r1) (substrate r2)).
```

If S_1 is compatible with $\leq$, then if $x \leq x$ and if x and y are related by S_1, we have $x \leq y$ or $y \leq x$ (depending on whether S_1 is compatible with the first or second argument). Consider the case of the product order and the first projection. If we take $x = (a, b)$ and $y = (a, c)$, then $x \leq y$ or $y \leq x$, hence $b \leq c$ or $c \leq b$. This means that all elements in E_2 are related. Note that if E_1 is empty, the product is empty and the condition is vacuous.

```
Lemma exercise1_2c2 r1 r2: (* 18 *)
  let compatibility r s :=
   (forall x x' y,  gle r x y -> related s x x' -> gle r x' y) in
  preorder r1 -> preorder r2 -> nonempty (substrate r1) ->
  compatibility (order_product2 r1 r2)
  (first_proj_eq (substrate r1) (substrate r2)) ->
  r2 = coarse (substrate r2).
```

Same proof, with definitions of x and y exchanged.

```
Lemma exercise1_2c3 r1 r2: (* 18 *)
  let compatibility r s :=
   (forall x y y',  gle r x y -> related s y y' -> gle r x y') in
  preorder r1 -> preorder r2 -> nonempty (substrate r1) ->
  compatibility (order_product2 r1 r2)
  (first_proj_eq (substrate r1) (substrate r2)) ->
  r2 = coarse (substrate r2).
```

We show that $\mathrm{pr}_1 = f_1 \circ \phi_1$ implies that f_1 is an isomorphism; we first introduce the definition of a preorder isomorphism. If E_2 is empty, then pr_1 is in general not surjective, thus f_1 is not surjective; this explains why we add the condition $\mathrm{E}_2 \neq \emptyset$. If $f_1(x) = f_1(y)$, where x and y are the classes of x' and y', then $\mathrm{pr}_1 x' = \mathrm{pr}_1 y'$, hence $x' \equiv y'$ and $x = y$, so that f_1 is injective. With the same notations, $f_1(x) \leq f_1(y)$ if and only if $\mathrm{pr}_1 x' \leq \mathrm{pr}_1 y'$.

```
Definition preorder_isomorphism f r r' :=
  (order r) & (order r') &
  (bijection f) & (substrate r = source f) &  (substrate r' = target f) &
  (forall x y, inc x (source f) -> inc y (source f) ->
    (gle r x y <-> gle r' (W x f) (W y f))).

Lemma exercise1_2c4 r1 r2 f:  (* 73 *)
  let s := (first_proj_eq (substrate r1) (substrate r2)) in
    let r:= order_product2 r1 r2 in
  is_function f -> source f = quotient s -> target f = (substrate r1) ->
  preorder r1 -> preorder r2 -> nonempty (substrate r2) ->
  f \co (canon_proj s)=(first_proj (product (substrate r1) (substrate r2)))
  ->  preorder_isomorphism f (quotient_order r s) r1.
```

(d) With the hypothesis of (a), suppose that E *is an* ordered *set and that the following condition is satisfied:*

(C') The relations $x \leq y \leq z$ *and* $x \equiv z \pmod{\mathrm{S}}$ *in* E *imply* $x \equiv y \pmod{\mathrm{S}}$.

Show that $\mathrm{R}\{x, y\}$ *is then an* order *relation between* X *and* Y *in* E/S.

```
Definition quotient_order_axiom r s:=
  forall x y z, gle r x y -> gle r y z -> related s x z -> related s x y.

Lemma Exercise1_2d r s: (* 16 *)
  is_equivalence s -> order r -> substrate s = substrate r ->
  quotient_order_axiom r s ->
  order (quotient_order r s).
```

(e) Give an example of a totally ordered set E *with four elements and an equivalence relation* S *and* E *such that neither of the conditions (C) and (C') is satisfied, but such that* E/S *is an ordered set.*

Assume that E is a totally ordered finite set, and consider two classes X and Y; they have a greatest element x and y. The condition $x \leq y$ is equivalent to $\mathrm{X} \leq \mathrm{Y}$, so that the quotient is totally ordered. Assume $a < b < c$, S is such that a and c are related by S, but no other pair of distinct elements are related. Then (C′) is false. In (C), take a, b and c for x, y and y'. Since $y \equiv y'$ implies $y = y'$, condition (C) is false. This gives an example with three elements. But a fourth one may be added without harm.

(f) Let E *be an ordered set, let* f *be an increasing mapping of* E *into an ordered set* F, *and let* $\mathrm{S}\{x, y\}$ *be the equivalence relation* $f(x) = f(y)$ *on* E. *Then the condition (C') is satisfied.*

Moreover the condition (C) is satisfied if and only if the relations $x \leq y$ and $f(x) = f(x')$ imply that there exists $y' \in \mathrm{E}$ such that $x' \leq y'$ and $f(y) = f(y')$. Let $f = g \circ \phi$ be the canonical decomposition of f. Then g is an isomorphism of E/S *onto* $f(\mathrm{E})$ *if and only if this condition is satisfied and, in addition, the relation $f(x) \leq f(y)$ implies that there exists x', y' such that $f(x) = f(x')$, $f(y) = f(y')$, and $x' \leq y'$.*

```
Lemma Exercise1_2f1 r r' f: (* 5 *)
  increasing_fun f r r' ->
  quotient_order_axiom r (equivalence_associated f).

Lemma Exercise1_2f2 r r' f: (* 16 *)
  increasing_fun f r r' ->
  (weak_order_compatibility r (equivalence_associated f) <->
  (forall x y x', gle r x y -> inc x' (source f) -> W x f = W x' f ->
    exists y', inc y' (source f) & gle r x' y' &  W y f = W y' f)).
```

The next point is straightforward, but a bit longish. Let CC and DD be the two conditions. Assume $f = g \circ \phi$, and f increasing. We must show that g is an isomorphism if and only if CC and DD hold. If $\mathrm{X} \in \mathrm{E/S}$ we denote by x the representative of X. We have then $g(\mathrm{X}) = f(x)$. The equivalence relation is defined by: $a \in \mathrm{X}$ if and only if $f(x) = f(a)$, $a \in \mathrm{E}$ and $x \in \mathrm{E}$.

If x and y in E, X and Y are their equivalence classes, then $f(x) \leq f(y)$ implies $g(\mathrm{X}) \leq g(\mathrm{Y})$. Assume that g is a morphism; we deduce $\mathrm{X} \leq \mathrm{Y}$. Expanding this, we get: if x' is in the class of x, there exists y' such that $f(y) = f(y')$ and $x' \leq y'$. The conclusion follows easily.

```
Lemma Exercise1_2f3 r r' f g: (* 76 *)
  let CC:= forall x y x', gle r x y -> inc x' (source f) -> W x f = W x' f ->
    exists y', inc y' (source f) & gle r x' y' &  W y f = W y' f in
  let DD:= forall x y, inc x (source f) -> inc y (source f) ->
    gle r' (W x f) (W y f) -> exists x', exists y',
      W x f = W x' f & W y f = W y' f & gle r x' y' in
  increasing_fun f r r' ->
  composable g (canon_proj (equivalence_associated f)) ->
  f = compose g (canon_proj (equivalence_associated f)) ->
  (order_morphism g (quotient_order r (equivalence_associated f)) r'
      <-> (CC & DD)).
```

---

3. *Let* I *be an ordered set and let* $(\mathrm{E}_\iota)_{\iota \in \mathrm{I}}$ *be a family of non-empty ordered sets indexed by* I.

(a) Let F *be the* sum *(Chapter II, § 4, no. 8) of the family* $(\mathrm{E}_\iota)_{\iota \in \mathrm{I}}$*; for each $x \in \mathrm{F}$, let $\lambda(x)$ be the index ι such that $x \in \mathrm{E}_\iota$; and let* G *be the graph consisting of all the pairs $(x, y) \in \mathrm{F} \times \mathrm{F}$ such that either $\lambda(x) < \lambda(y)$ or else $\lambda(x) = \lambda(y)$ and $x \leq y$ in $\mathrm{E}_{\lambda(x)}$. Show that* G *is the graph of an ordering on* F*. The set* F *endowed with this ordering is called the* ordinal sum *of the family* $(\mathrm{E}_\iota)_{\iota \in \mathrm{I}}$ *(relative to the ordering on* I*) and is denoted* $\sum_{\iota \in \mathrm{I}} \mathrm{E}_\iota$*. Show that the equivalence relation corresponding to the partition* $(\mathrm{E}_\iota)_{\iota \in \mathrm{I}}$ *of* F *satisfies conditions (C) and (C') of Exercise 2, and that the quotient ordered set (Exercise 2) is canonically isomorphic to* I.

We have defined the ordinal sum in section 8.1. We dropped the condition of E_ι being *non-empty*, as this will allow us to prove theorems of the form $0 + x = x$. Lemma orsum_order says that G is an ordering.

We prove here the last proposition. Let's consider two variables, r and g. We define some quantities associated to them (prefixed by E13), in order, the ordinal sum F, its substrate $\bar{F}$, the second projection λ, the equivalence S associated to λ. Assumption H1 says that r is an ordering, g a functional graph on the substrate of r such that each g_ι is an ordering. Assumption H2 says that the sets E_ι are non-empty.

```
Section Exercise1_3a.
Variables r g: Set.
Definition E13_F:= order_sum r g.
Definition E13_sF:= sum_of_substrates g.
Definition E13_lam := second_proj E13_sF.
Definition E13_S:= equivalence_associated (second_proj E13_sF).
Definition E13_H1:= orsum_ax r g.
Definition E13_H2:= is_graph g &
  (forall i, inc i (domain g) -> nonempty (substrate (V i g))).
```

We show, successively, that $\bar{F}$ is a graph, that pr_2 can be considered as a surjective function on F, that its target is I if H2 is true, and that it is an increasing function.

```
Lemma Exercise1_3a0: is_function E13_lam. (* 1 *)
Lemma Exercise1_3a1: is_graph E13_sF. (* 3 *)
Lemma Exercise1_3a2: surjection E13_lam. (* 4 *)
Lemma Exercise1_3a3: E13_H2 -> domain g = target E13_lam. (* 6 *)
Lemma Exercise1_3a3': substrate E13_S = E13_sF. (* 2 *)
Lemma Exercise1_3a3'': substrate E13_S = source E13_lam. (* 1 *)
Lemma Exercise1_3a4: E13_H1 -> E13_H2 -> (* 8 *)
  increasing_fun E13_lam E13_F r.
```

Consider now the "equivalence relation corresponding to the partition $(E_\iota)_{\iota\in I}$ of F". The set $\bar{F}$ is the union of f', the graph of $\iota \mapsto E_\iota \times \{\iota\}$, where f is the graph of $\iota \mapsto E_\iota$. Consider now a function f'' whose graph is f'. This function defines a partition. By abuse of notations, F is identified with $\bar{F}$, $E_\iota \times \{\iota\}$ with E_ι and f'' with f'. Let S$'$ be the equivalence associated to f''. We show here that x and y are related by S$'$ (resp. S) if and only if x and y are in the disjoint union and $\mathrm{pr}_2 x = \mathrm{pr}_2 y$. This shows S$' =$ S.

```
Definition disjoint_union_function f :=
  corresp (domain f)(range (disjoint_union_fam f))(disjoint_union_fam f).

Lemma disjoint_union_function_pr f: (* 2 *)
  is_function (disjoint_union_function f) &
  graph (disjoint_union_function f) =  (disjoint_union_fam f).
Lemma Exercise1_3a5 x y (f :=  (fam_of_substrates g)):  (* 15 *)
  related (partition_relation (disjoint_union_function f) (disjoint_union f))
  x y <-> (inc x E13_sF & inc y E13_sF & Q x = Q y).
Lemma Exercise1_3a6 x y:  (* 4 *)
  related E13_S x y <-> (inc x E13_sF & inc y E13_sF & Q x = Q y).
```

We show that the classes of S are the sets $E_\iota \times \{\iota\}$. If ϕ is the canonical projection of $\bar{F}$ on $\bar{F}$/S, there exists a function g defined by $\lambda = g \circ \phi$ (canonical decomposition) because of the surjectivity of λ, it is unique. It is the function induced by λ by passing on the quotient. We can restate it as: an element X of the quotient is of the form $\iota \mapsto E_\iota \times \{\iota\}$, and $g(X) = \iota$. We show that g is a function, that $g \circ \phi$ exists and that g is bijective.

```
Lemma Exercise1_3a7: is_equivalence E13_S. (* 1 *)
Lemma Exercise1_3a8 a: E13_H2 ->  (* 32 *)
  (inc a (quotient E13_S) <-> exists i,
    inc i (domain g) & a = product (V i (fam_of_substrates g)) (singleton i)).
Lemma Exercise1_3a9: is_function (fun_on_quotient E13_S E13_lam). (* 2 *)
Lemma Exercise1_3a10:  (* 3 *)
  (fun_on_quotient E13_S E13_lam) \coP (canon_proj E13_S).
Lemma Exercise1_3a11: (* 1 *)
  E13_lam = (fun_on_quotient E13_S E13_lam) \co (canon_proj E13_S).
Lemma Exercise1_3a12 x:  E13_H2 -> (* 10 *)
  inc x (quotient E13_S) -> exists i,
    inc i (domain g) & x = product (V i (fam_of_substrates g)) (singleton i) &
    W x (fun_on_quotient E13_S E13_lam) = i.
Lemma Exercise1_3a13: E13_H2 -> (* 19 *)
   bijection (fun_on_quotient E13_S E13_lam).
```

We apply Exercise 2f; the two conditions are easily satisfied. We deduce that g is an isomorphism on its image. Since we know that g is bijective, it is an order isomorphism between F/S and I.

```
Lemma Exercise1_3a14: E13_H1 -> quotient_order_axiom E13_F E13_S.  (* 4 *)
Lemma Exercise1_3a15: E13_H1 ->  E13_H2 -> (* 25 *)
  order_isomorphism  (fun_on_quotient E13_S E13_lam)
  (quotient_order E13_F E13_S) r.
End Exercise1_3a.
```

(b) If the set I *is the ordinal sum of a family* $(J_\lambda)_{\lambda\in L}$ *of ordered sets, where* L *is an ordered set, show that the ordered set* $\sum_{\iota\in I} E_\iota$ *is canonically isomorphic to the ordinal sum* $\sum_{\lambda\in L} F_\lambda$ *where* $F_\lambda = \sum_{\iota\in I_\lambda} E_\iota$ *("associativity of the ordinal sum"). If* I *is the linearly ordered set* $\{1,2\}$*, we write* $E_1 + E_2$ *for the ordinal sum of* E_1 *and* E_2*. Show that* $E_2 + E_1$ *and* $E_1 + E_2$ *are not necessarily isomorphic.*

The associativity formula has been shown in the main text as orsum_assoc_iso. Consider two ordered sets E_1 and E_2 that we identify as subsets of the sum $E_1 + E_2$. Let x be the greatest element of the sum $E_1 + E_2$. Assume E_2 not empty; thus $y \in E_2$. Since $y \le x$, we deduce $x \in E_2$. Obviously, for all $z \in E_2$ we have $z \le x$. Thus, we have shown: if the sum $E_1 + E_2$ has a greatest element, then E_2 has a greatest element. This shows that $E_1 + E_2$ is not always isomorphic to $E_2 + E_1$.

```
Lemma orsum2_greatest r r' x: (* 15 *)
  order r -> order r' -> nonempty (substrate r') ->
  greatest_element (order_sum2 r r') x -> greatest_element r' (P x).
```

(c) An ordinal sum $\sum_{\iota\in I} E_\iota$ *is right directed if and only if* I *is right directed and* E_ω *is right directed for each* ω *of* I*.*

(d) An ordinal sum $\sum_{\iota\in I} E_\iota$ *is totally ordered if and only if* I *and each* E_ι *is totally ordered.*

(e) An ordinal sum $\sum_{\iota\in I} E_\iota$ *is a lattice if and only if the following conditions are satisfied:*

> *(I) The set* I *is a lattice, and for each pair* (λ,μ) *of non-comparable indices in* I*,* $E_{\sup(\lambda,\mu)}$ *(resp.* $E_{\inf(\lambda,\mu)}$*) has a least (resp. greatest) element.*

> *(II) For each* $\alpha \in I$ *and each pair* (x, y) *of elements of* E_α *such that the set* $\{x, y\}$ *is bounded above (resp. bounded below) in* E_α*, the set* $\{x, y\}$ *has a least upper bound (resp. greatest lower bound) in* E_α*.*
>
> *(III) For each* $\alpha \in I$ *such that* E_α *contains a set of two elements which has no upper bound (resp. no lower bound) in* E_α*, the set of indices* $\lambda \in I$ *such that* $\lambda > \alpha$ *(resp.* $\lambda < \alpha$*) has a least element (resp. a greatest element)* β*, and* E_β *has a least element (resp. greatest element).*

The French edition corrects point (c) as: "il faut et il suffit que I soit filtrant à droite et que, pour tout élément maximal ω de E, E_ω soit filtrant à droite." This can be translated as: An ordinal sum $\sum_{\iota\in I} E_\iota$ is right directed if and only if I is right directed and E_ω is right directed for each maximal element ω of I.

The proof is as follows. We identify E_k with a subset of the disjoint union. Assume the sum directed. For every pair of indices i and j, there is $x \in E_i$ and $y \in E_j$, and if $z \in E_k$ is an upper bound for x and y, then k is an upper bound for i and j. Let k be a maximal index, x and y in E_k. Every upper bound of x and y in the sum must be in E_k. Conversely, let $x \in E_i$ and $y \in E_j$ be in the disjoint union; assume I right directed, so that there is $k \in I$ with $i \le k$ and $j \le k$. If k is one of i and j, but not both, then $x < y$ or $y < x$. If $i < k$ and $j < k$, there is $z \in E_k$ with $x < z$ and $y < z$. If k is not maximal, there is k' with $k < k'$, and the same conclusion holds. Finally, if $i = j = k$ and k is maximal, we have an upper bound since E_k is right directed.

```
Section Exercise13b.
Variables r g: Set.
Hypothesis oa: orsum_ax r g.
Definition orsum_ax2:=
 (forall i, inc i (domain g) -> nonempty (substrate (V i g))).

Lemma orsum_total1: orsum_ax2  -> (* 13 *)
  total_order (order_sum r g) -> (total_order r &
    forall i, inc i (domain g) -> total_order (V i g)).
Lemma orsum_total2:  (* 11 *)
  total_order r ->
  (forall i, inc i (domain g) -> total_order (V i g)) ->
  total_order (order_sum r g).

Lemma orsum_pr0:  orsum_ax2 -> (* 5 *)
  forall i, inc i (substrate r) ->
  exists y, inc y (V i (fam_of_substrates g)) &
    inc (J y i) (sum_of_substrates g).
Lemma orsum_pr1: orsum_ax2 ->   (* 2 *)
  forall i, inc i (domain g) ->
   exists y, inc y (V i (fam_of_substrates g)) &
    inc (J y i) (substrate (order_sum r g)).
Lemma orsum_g1 i x i' x', (* 2 *)
  inc (J i x) (sum_of_substrates g)  -> inc (J i' x') (sum_of_substrates g)->
  gle r x x' -> x <> x' ->
  gle (order_sum r g) (J i x) (J i' x').

Lemma orsum_directed:  orsum_ax2 ->  (* 53 *)
  (right_directed (order_sum r g) <-> (right_directed r &
    forall i, maximal_element r i -> right_directed (V i g))).
```

We show that if an ordinal sum is a lattice, so is the index set I. The idea is the following.

Consider two indices a and b. If they are comparable, there is a supremum. Otherwise, let $x \in \mathrm{E}_a$, $y \in \mathrm{E}_b$ and $z = \sup(x, y)$. We have $z \in \mathrm{E}_c$, and c is an upper bound of a and b. Let d be another upper bound. Since a and b are incomparable, we have $a < d$ and $b < d$. For every $t \in \mathrm{E}_d$ we have $x < t$ and $y < t$ hence $z \leq t$ hence $c \leq d$.

```
Lemma orsum_lattice1: orsum_ax2 ->  (* 51 *)
  lattice (order_sum r g) -> lattice r.
```

With the same notations as before, assume that a and b cannot be compared, let $x \in \mathrm{E}_a$, $y \in \mathrm{E}_b$ and $z \in \mathrm{E}_c = \sup(x, y)$. Then $c = \sup(a, b)$. Every z' in E_c is an upper bound of x and y. Hence $\sup(x, y)$ is the smallest element of E_c.

```
Lemma orsum_lattice2:  orsum_ax2 -> (* 74 *)
  lattice (order_sum r g) ->
  (forall i j, inc i (domain g) -> inc j (domain g) ->
    (gle r i j \/ gle r j i \/ (exists u, exists v,
      least_element (V (sup r i j) g) u &
```

Assume $a = b$ and t is an upper bound for x and y in E_a. Then $z \leq t$, thus $c \leq a$, hence $a = c$. This means $z \in \mathrm{E}_a$, so that x and y have a least upper bound in E_a.

```
Lemma orsum_lattice3 i x y: (* 36 *)
  lattice (order_sum r g) ->
  inc i (domain g) -> gle (V i g) x t -> gle (V i g) y t ->
    has_supremum (V i g) (doubleton x y).
Lemma orsum_lattice4 i x y t: (* 36 *)
  lattice (order_sum r g) ->
  inc i (domain g) -> gle (V i g) t x -> gle (V i g) t y ->
    has_infimum (V i g) (doubleton x y).
```

Assume again $a = b$, but now, suppose no upper bound for x and y exists in E_a. Then the interval $]a, c[$ is empty and z is the least element of E_c. The first condition can be restated as: c is the least element of $]a, \rightarrow[$.

```
Lemma orsum_lattice5  i x y: orsum_ax2 ->   (* 41 *)
  lattice (order_sum r g) ->
  inc i (domain g) ->  inc x (V i  (fam_of_substrates g))
    ->  inc y (V i  (fam_of_substrates g)) ->
  (forall t, inc t (V i (fam_of_substrates g))
     -> ~ (gle (V i g) x t & gle (V i g) y t)) ->
  exists j, inc j (domain g) &
    least_element (induced_order r (Zo (domain g) (fun k=> glt r i k))) j &
    exists z, least_element  (V j g) z.
Lemma orsum_lattice6 i x y:  orsum_ax2 ->  (* 41 *)
  lattice (order_sum r g) ->
  inc i (domain g) ->  inc x (V i  (fam_of_substrates g))
    ->  inc y (V i  (fam_of_substrates g)) ->
  (forall t, inc t (V i (fam_of_substrates g))
     -> ~ (gle (V i g) t x & gle (V i g) t y)) ->
  exists j, inc j (domain g) &
    greatest_element (induced_order r (Zo (domain g) (fun k=> glt r k i))) j &
    exists z, greatest_element  (V j g) z.
```

We are now ready for the main result.

```
Lemma orsum_lattice: orsum_ax2 ->  (* 184 *)
  (lattice (order_sum r g) <->
  ((lattice r) &
    (forall i j, inc i (domain g) -> inc j (domain g) ->
      (gle r i j \/ gle r j i \/ (exists u, exists v,
        least_element (V (sup r i j) g) u &
        greatest_element (V (inf r i j) g) v))) &
    (forall i x y t, inc i (domain g) -> gle (V i g) x t -> gle (V i g) y t ->
      has_supremum (V i g) (doubleton x y)) &
    (forall i x y t, inc i (domain g) -> gle (V i g) t x -> gle (V i g) t y ->
      has_infimum (V i g) (doubleton x y)) &
    (forall i x y,
      inc i (domain g) ->  inc x (V i (fam_of_substrates g))
         ->  inc y (V i (fam_of_substrates g)) ->
      (forall t, inc t (V i (fam_of_substrates g))
          -> ~ (gle (V i g) x t & gle (V i g) y t)) ->
      exists j, inc j (domain g) &
        least_element (induced_order r (Zo (domain g) (fun k=> glt r i k))) j &
        exists z, least_element  (V j g) z) &
    (forall i x y, inc i (domain g) ->  inc x (V i (fam_of_substrates g))
         ->  inc y (V i (fam_of_substrates g)) ->
      (forall t, inc t (V i (fam_of_substrates g))
           -> ~ (gle (V i g) t x & gle (V i g) t y)) ->
      exists j, inc j (domain g) &
        greatest_element (induced_order r (Zo (domain g) (fun k=> glt r k i)))
        j &
        exists z, greatest_element  (V j g) z))).
```

---

4. * *Let* E *be an ordered set, and let* $(E_\iota)_{\iota\in I}$ *be the partition of* E *formed by the connected components of* E *(Chapter II, § 6, Exercise 10) with respect to the reflexive and symmetric relation "either* $x = y$ *or* x *and* y *are not comparable".*

(a) Show that if $\iota \neq \kappa$ *and if* $x \in E_\iota$ *and* $y \in E_\kappa$*, then* x*,* y *are comparable; and that if, for example,* $x \leq y$*,* $y' \in E_\kappa$*, and if* $y' \neq y$*, then also* $x \leq y'$ *(use the fact that there exists no partition of* E_κ *into two sets* A *and* B *such that every element of* A *is comparable with every element of* B*).*

(b) Deduce from (a) that the equivalence relation S *corresponding to the partition* (E_ι) *of* E *is compatible (in* x *and* y*) with the order relation* $x \leq y$ *on* E*, and that the quotient ordered set* E/S *(Exercise 2) is totally ordered.*

(c) What are the connected components of an ordered set $E = F \times G$ *which is the product of two totally ordered sets?* *

We first recall some facts about connected components. We shall denote by R the relation "either $x = y$ or x and y are not comparable", and by S the equivalence relation associated; the sets E_ι are the equivalence classes of S.

```
Definition not_comp_rel r := fun x y =>
  inc x (substrate r) & inc y (substrate r) &
  x = y \/ ~ ((gle r x y) \/ (gle r y x)).

Definition ncr_equiv r :=
```

```
  Exercice1.Sgraph (not_comp_rel r) (substrate r).
Definition ncr_component r :=
  Exercice1.connected_comp (not_comp_rel r) (substrate r).

Lemma ncr_properties r: order r ->  (* 14 *)
  (is_equivalence (ncr_equiv r) &
  substrate  (ncr_equiv r) =  substrate r &
  (forall x, inc x (substrate r) -> class (ncr_equiv r) x = ncr_component r x)&
  (forall x y, not_comp_rel r x y -> related (ncr_equiv r) x y)).
```

We restate "$\iota \neq \kappa$ and if $x \in E_\iota$ and $y \in E_\kappa$, then x, y are comparable" as: either the classes (mod S) of x and y are equal, or x, y are comparable. This can also be restated as $R\{x, y\}$ implies $S\{x, y\}$.

```
Lemma Exercice1_4a1 r x y: order r -> (* 5 *)
  inc x (substrate r) -> inc y (substrate r) ->
  gle r x y \/ gle r y x \/ class (ncr_equiv r) x = class (ncr_equiv r) y.
```

Consider the hint of the second claim. Consider a class C modulo S that is the union of two sets A and B such that each element of A is comparable with each element of B. Assume neither set empty, let $a \in A$ and $b \in B$. Then a and b are related by S. This means that there is a chain $(x_i)_i$, relating a and b, thus a chain $(x_i)_i$ with head in A and tail in B; by induction, there exist elements $a' \in A$ and $b' \in B$ related by R (if a chain is non-trivial, its head a' is related to another chain with head b'; these elements are in the same equivalence class (namely C) because they are related by R; if $b' \in A$, the result follows by induction, otherwise $b' \in B$ and we have a solution). Since a' and b' are comparable and related by R, they are equal. Thus the intersection of A and B is nonempty.

```
Lemma Exercice1_4a2 r y: (* 28 *)
  order r -> inc y (substrate r) ->
  forall a b, union2 a b = class (ncr_equiv r) y ->
    (forall u v, inc u a -> inc v b -> gle r u v \/ gle r v u) ->
    a = emptyset \/ b = emptyset \/ nonempty (intersection2 a b).
```

Let C be a class for S, $x \in E - C$, A the set of all $y \in C$ such that $x \leq y$, B the set of $y' \in C$ such that $y' \leq x$. We have shown $C = A \cup B$. Obviously, if $y \in A$ and $y' \in B$ then $y' \leq y$, and $y \neq y'$ (this would imply $y = x$ hence $x \in C$). As a consequence one of A and B is empty.

```
Lemma Exercice1_4a3 r x y y': order r -> (* 32 *)
  inc x (substrate r) -> related (ncr_equiv r) y y' -> gle r x y ->
  related (ncr_equiv r) x y \/ gle r x y'.
```

Let's show the compatibility of the equivalence and the order. We must show that if x and x' are in the same class, if y and y' are in the same class, then $x \leq y$ is the same as $x' \leq y'$. The assumption is that the classes are distinct. We know that $x' \leq y'$, $y' \leq x'$ or $R\{x', y'\}$. The first possibility is the desired result, the last one says that the classes are the same. We have shown that $y' \leq x'$ implies $y' \leq x$ and $x \leq y$ implies $x \leq y'$. Thus, the second possibility says $x = y'$, absurd.

```
Lemma Exercice1_4b1 r x y x' y': order r ->  (* 20 *)
  related (ncr_equiv r) x x' -> related (ncr_equiv r) y y' ->
  class (ncr_equiv r) x <> class (ncr_equiv r) y ->
  gle r x y -> gle r x' y'.
```

The quotient order is an order, according to condition (C'). To show it, assume $x \le y \le z$, x and z in the same class. If x and y are not in the same class, then $x \le y$ implies $z \le y$. This says $y = z$, absurd. The quotient order is total, since, given two distinct classes, the representatives are comparable. Now, if $x \in \mathrm{X}$ and $y \in \mathrm{Y}$, the compatibility condition says $\mathrm{X} \le \mathrm{Y}$ in the quotient if and only if $x \le y$ in the substrate.

```
Lemma Exercice1_4b r: order r -> (* 35 *)
  total_order(quotient_order r (ncr_equiv r)).
```

Consider a product of two totally ordered sets $\mathrm{E} = \mathrm{F} \times \mathrm{G}$. Let's determine the set of components. We have $x \le y$ if and only if $\mathrm{pr}_1 x \le \mathrm{pr}_1 y$ and $\mathrm{pr}_2 x \le \mathrm{pr}_2 y$. We exclude the case where the sets are empty. If G is a singleton, then $\mathrm{pr}_2 x \le \mathrm{pr}_2 y$ is always true, and the product is totally ordered. The set of components is then isomorphic to E. If a and a' are the least elements of F and G, $l = (a, a')$, then l is the least element of E, and $\{l\}$ is a component. In the same fashion, if g is the greatest element of E then $\{g\}$ is a component.

We assume now that E has at least two elements b and c, F has at least two elements b' and c'. We may assume $b < c$ and $b' < c'$. We pretend that the class C of (b, c') contains all elements of E, with the possible exception of l and g. It contains obviously (c, b'). Let $y = (b, b')$, and assume that $y \neq l$. Then $y \in \mathrm{C}$. In fact, if $a < b$, then $\mathrm{R}\{(a, c'), (b, b')\}$ and $\mathrm{R}\{(a, c'), (c, b')\}$, so that (b, b') and (c, b') are in the same class. On the other hand, if $a' < b'$, then $\mathrm{R}\{(c, a'), (b, b')\}$ and $\mathrm{R}\{(c, a'), (b, c')\}$, so that (b, b') and (b, c') are in the same class. In the same way, (c, c') is the class if it is not the greatest element. Assume $c < d$. Then (c, b') and (d, b') are related to (b, c'). Then (c, b') and (d, b') are related to (b, c'), and (b, c') and (c, c') are related to (d, b'). This shows that the six elements of the product $\{b, c, d\} \times \{b', c'\}$, minus the greatest and least elements are in the same class. By symmetry, if $d' \in \mathrm{F}$, the six elements of the product $\{b, c, d\} \times \{b', c', d'\}$, minus the greatest and least elements are in the same class.

Thus, if E has a least and a greatest element l and g, there are three classes, $\mathrm{E} - \{l, g\}$, $\{l\}$, and $\{g\}$. If E has a least element l and no greatest element, there are two classes, $\mathrm{E} - \{l\}$, and $\{l\}$. If E has a greatest element g and no least element, there are two classes, $\mathrm{E} - \{g\}$, and $\{g\}$. Otherwise there is a single class.

```
Lemma Exercice1_4c1 r x: (* 13 *)
   order r -> greatest_element r x ->
   ncr_component r x = singleton x.
Lemma Exercice1_4c2 r x:  (* 13 *)
   order r -> least_element r x ->
   ncr_component r x = singleton x.
Lemma Exercice1_4c3 r r' x y: (* 19 *)
  order r -> total_order r' ->
  substrate r = singleton x -> inc y (substrate r') ->
  ncr_component (order_product2 r r') (J x y) = singleton (J x y).
Lemma Exercice1_4c4 r r' b c b' c' u: (* 119 *)
  total_order r -> total_order r' ->
  glt r b c -> glt r' b' c' ->
  inc u (substrate (order_product2 r r')) ->
   (least_element (order_product2 r r') u
     \/ greatest_element (order_product2 r r') u
     \/ inc u (ncr_component (order_product2 r r') (J b c'))).
```

5. *Let* E *be an ordered set. A subset* X *of* E *is said to be* free *if no two distinct elements of* X *are comparable. Let* $\mathfrak{I}$ *be the set of free subsets of* E*. Show that, on* $\mathfrak{I}$*, the relation "given any* $x \in X$*, there exists* $y \in Y$ *such that* $x \leq y$*" is an order relation between* X *and* Y*, written* $X \leq Y$*. The mapping* $x \to \{x\}$ *is an isomorphism of* E *onto a subset of the ordered set* $\mathfrak{I}$*. If* $X \subset Y$ *where* $X \in \mathfrak{I}$ *and* $Y \in \mathfrak{I}$*, show that* $X \leq Y$*. The ordered set* $\mathfrak{I}$ *is totally ordered if and only if* E *is totally ordered, and then* $\mathfrak{I}$ *is canonically isomorphic to* E*.*

We restate the definition as: if $x \in X$ and $y \in X$ and $x \leq y$ then $x = y$.

```
Definition free_subset r X := forall x y, inc x X -> inc y X ->
  gle r x y -> x = y.
Definition set_of_free_subsets r:=
  Zo (powerset (substrate r)) (fun X=> free_subset r X).
Definition free_subset_compare r X Y:=
  inc X (set_of_free_subsets r) & inc Y (set_of_free_subsets r) &
  forall x, inc x X -> exists y, inc y Y & gle r x y.
Definition free_subset_order r:=
  graph_on (free_subset_compare r) (set_of_free_subsets r).
```

The relation $\forall x \in X, \exists y \in Y, x \leq y$ is clearly a preorder relation. Antisymmetry follows from the fact that if $x \in X$, $y \in Y$, $x \leq y$ and $x' \in X$, $y \leq x'$ we have $x \leq x'$ by transitivity, then $x = x'$ when X is free, and $x = y$ by antisymmetry.

```
Lemma Exercise1_5w r x a: order r -> (* 2 *)
 inc x (set_of_free_subsets r) -> inc a x ->
 gle r a a.
Lemma Exercise1_5a r: order r -> (* 19 *)
  order_r (free_subset_compare r).
Lemma fs_order_gle r x y: (* 3 *)
  gle (free_subset_order r) x y <-> free_subset_compare r x y.
Lemma fs_order_or r: (* 2 *)
  order r -> order (free_subset_order r).
Lemma fs_order_sr  r: (* 3 *)
  order r -> substrate (free_subset_order r) = set_of_free_subsets r.
```

A singleton is free; the mapping $x \mapsto \{x\}$ is an isomorphism onto its range. If E is a totally ordered set, then the only nonempty-free subsets are the singletons, so that the range is $\mathfrak{I} - \{\emptyset\}$.

```
Lemma Exercise1_5b r x: order r -> (* 2*)
  inc x (substrate r) -> inc (singleton x) (set_of_free_subsets r).
Lemma Exercise1_5c r x y: order r -> (* 6 *)
  inc x (substrate r) ->inc y (substrate r) ->
  (gle r x y <-> gle (free_subset_order r) (singleton x) (singleton y)).
Lemma Exercise1_5d r: order r -> (* 8 *)
  order_morphism (BL singleton (substrate r)  (set_of_free_subsets r))
  r (free_subset_order r).
Lemma Exercise1_5e r X: total_order r -> (* 4 *)
  inc X (set_of_free_subsets r)-> small_set X.
```

If $X \subset Y$ then $X \leq Y$ (this is trivial). If E is totally ordered so is $\mathfrak{I}$. In fact, the empty set is the least element of $\mathfrak{I}$; two non-empty sets are singletons, and singletons are compared according to their elements. The converse is trivial. Bourbaki says that $\mathfrak{I}$ is canonically isomorphic to E in this case. This is obviously wrong: as noted above $x \mapsto \{x\}$ is an isomorphism between E and $\mathfrak{I} - \{\emptyset\}$.

```
Lemma Exercise1_5f r X Y: order r -> (* 2 *)
  inc X (set_of_free_subsets r) -> inc Y (set_of_free_subsets r) ->
  sub X Y ->  gle (free_subset_order r) X Y.
Lemma Exercise1_5g r: total_order r -> (* 21 *)
  total_order (free_subset_order r).
Lemma Exercise1_5h r: order r -> (* 4 *)
  total_order (free_subset_order r) -> total_order r.
```

---

6. *Let* E *and* F *be two ordered sets and let* $\mathscr{A}$(E,F) *be the subset of the product ordered set* F^E *consisting of the* increasing *mappings of* E *into* F.

We change the definition of $\mathscr{A}$(E,F): it will be a subset of $\mathscr{F}$(E;F), the set of mappings from E to F, that is canonically isomorphic to F^E. It will be ordered by the ordering on functions.

```
Definition set_of_increasing_mappings r r' :=
  Zo (set_of_functions (substrate r) (substrate r'))
  (fun z=> increasing_fun z r r').
Definition increasing_mappings_order r r' :=
  induced_order (order_function (substrate r) (substrate r') r')
  (set_of_increasing_mappings r r').
Definition first_projection f a b:= BL (fun z=> P (W z f)) a b.
Definition secnd_projection f a c:= BL (fun z=> Q (W z f)) a c.
Definition two_projections a b c :=
  BL (fun z => (J (first_projection z a b)
    (secnd_projection z a c)))
  (set_of_functions a (product b c))
  (product (set_of_functions a b) (set_of_functions a c)).
Definition two_projections_increasing r r' r'' :=
  restriction2 (two_projections (substrate r) (substrate r')(substrate r''))
  (set_of_increasing_mappings r (order_product2 r' r''))
  (product (set_of_increasing_mappings r r')
    (set_of_increasing_mappings r r'')).
Definition second_partial_map2 r r' r'':=
  BL (fun f=> restriction2
    (second_partial_function f)
    (substrate r) (set_of_increasing_mappings r' r''))
  (set_of_increasing_mappings (order_product2 r r') r'')
  (set_of_increasing_mappings r (increasing_mappings_order r' r'')).
```

Given a function $f : E \to F \times G$, we can consider the two projections $f_1 : E \to F$ and $f_2 : E \to G$. We first show that $f \mapsto (f_1, f_2)$ is a bijection $\mathscr{F}(E;F \times G)$ onto $\mathscr{F}(E;F) \times \mathscr{F}(E;G)$.

```
Lemma Exercice1_6a f a b c: (* 10 *)
  is_function f -> source f = a ->
  target f = product b c ->
  (bl_axioms (fun z=> P (W z f)) a b &
    bl_axioms (fun z=> Q (W z f)) a c &
    is_function (first_projection f a b) &
    is_function (secnd_projection f a c) &
    (forall x, inc x a -> W x (first_projection f a b) = P (W x f)) &
```

```
    (forall x, inc x a -> W x (secnd_projection f a c) = Q (W x f))).
Lemma Exercice1_6b a b c: (* 3 *)
  bl_axioms
    (fun z => (J (first_projection z a b)
      (secnd_projection z a c)))
    (set_of_functions a (product b c))
    (product (set_of_functions a b) (set_of_functions a c)).
Lemma Exercice1_6c a b c: (* 27 *)
  bijection  (two_projections a b c).
```

In what follows; r, r' and r'' are some orderings. We give here some trivial lemmas making the definitions explicit.

```
Section Exercise1_6a.
Variables r r': Set.
Hypotheses (or: order r)(or': order r').

Lemma set_of_increasing_mappings_pr f:  (* 1 *)
  inc f (set_of_increasing_mappings r r') <->
    (is_function f & source f = (substrate r) & target f = substrate r'
    & increasing_fun f r r').
Lemma imo_or:  (* 4 *)
  order (increasing_mappings_order r r').
Lemma imo_sr: (* 3 *)
  substrate (increasing_mappings_order r r') = set_of_increasing_mappings r r'.
Lemma imo_gle f g: (* 8 *)
  gle (increasing_mappings_order r r') f g <->
  (inc f (set_of_increasing_mappings r r') &
    inc g (set_of_increasing_mappings r r') &
    function_order_r (substrate r) (substrate r') r' f g).
```

(a) Show that if E, F, G *are three ordered sets, then the ordered set* $\mathscr{A}(E, F \times G)$ *is isomorphic to the product ordered set* $\mathscr{A}(E, F) \times \mathscr{A}(E, G)$.

Now we show that the projections are compatible with the order: if f is increasing, both projections are increasing; the converse is equally true. This means that $f \mapsto (f_1, f_2)$ induces a bijection $\mathscr{A}(E, F \times G)$ onto $\mathscr{A}(E, F) \times \mathscr{A}(E, G)$. Finally, we show that it is an order isomorphism.

```
Section Exercise1_6.
Variables r r' r'': Set.
Hypotheses (or: order r)(or': order r')(or'': order r'').

Lemma Exercice1_6d f: (* 14 *)
  increasing_fun f r (order_product2 r' r'') ->
  (increasing_fun (first_projection f (substrate r) (substrate r')) r r' &
    increasing_fun (secnd_projection f (substrate r) (substrate r'')) r r'').


Lemma Exercice1_6e: (* 20 *)
  (restriction2_axioms
    (two_projections (substrate r) (substrate r')(substrate r''))
    (set_of_increasing_mappings r (order_product2 r' r''))
    (product (set_of_increasing_mappings r r')
      (set_of_increasing_mappings r r''))).
Lemma Exercice1_6f: (* 36 *)
  bijection (two_projections_increasing r r' r'').
```

```
Lemma Exercice1_6g: (* 63 *)
  order_isomorphism (two_projections_increasing r r' r'')
  (increasing_mappings_order r (order_product2 r' r''))
  (order_product2 (increasing_mappings_order r r')
    (increasing_mappings_order r r'')).
```

(b) Show that if E, F, G *are three ordered sets, then the ordered set* $\mathscr{A}(E \times F, G)$ *is isomorphic to the ordered set* $\mathscr{A}(E, \mathscr{A}(F, G))$.

Let's show that $S = \mathscr{A}(E \times F, G)$ is isomorphic to the ordered set $T = \mathscr{A}(E, \mathscr{A}(F, G))$. We shall assume E and F non-empty. For otherwise the product $E \times F$ is empty, case where there is a single element in S. If F is empty, $\mathscr{A}(F, G)$ has a single element (the empty function) and $\mathscr{A}(E, \mathscr{A}(F, G))$ has a single element (that maps everything to the empty function). Assume E empty. Then $\mathscr{A}(E, \mathscr{A}(F, G))$ contains only the empty function. Whatever the orderings on the sets S and T, since the sets are singletons, the orders are trivial and any function between S and T is an isomorphism.

In all our lemmas we consider orderings r, r' and r'', with substrate E, F and G. An increasing function $f : r \times r' \to r''$ if a function with source $E \times F$ and target G; the first lemma says, that if r and r' are orders, the substrate of $r \times r'$ is indeed the product of the substrates, namely $E \times F$, and if these two sets are non-empty, one can recover the sets from the product. The second lemma says that f_x is increasing for all $x \in E$. The third lemma says that the range of $x \mapsto f_x$ is a subset of the set of increasing functions.

```
Lemma Exercice1_6h f: (* 3 *)
  nonempty (substrate r) -> nonempty (substrate r') ->
  increasing_fun f (order_product2 r r') r'' ->
 ( (domain (source f)) = substrate r & (range (source f)) = substrate r').
Lemma Exercice1_6i f x: (* 18 *)
  nonempty (substrate r) -> nonempty (substrate r') ->
  increasing_fun f (order_product2 r r') r'' ->
  inc x (substrate r) -> increasing_fun (second_partial_fun f x) r' r''.
Lemma Exercice1_6j f: (* 20 *)
  nonempty (substrate r) -> nonempty (substrate r') ->
  increasing_fun f (order_product2 r r') r'' ->
  (restriction2_axioms (second_partial_function f) (substrate r)
    (set_of_increasing_mappings r' r'')).
```

We consider now Φ =second_partial_map (this is the bijection from $\mathscr{F}(E \times F; G)$ onto $\mathscr{F}(E, \mathscr{F}(F; G))$. For each $a \in E$, $\Phi(f)(a)$ is in $\mathscr{F}(F; G))$; if f is increasing this is in $\mathscr{A}(F, G)$. Thus we can restrict $\Phi(f)$ as a function $\overline{\Phi(f)}$ from E into $\mathscr{A}(F, G))$. The mapping $f \mapsto \overline{\Phi(f)}$ will be our isomorphism. It is clearly injective; proving surjectivity is a bit longer.

If f_a and $a \mapsto f_a$ are increasing, then $a \le a'$ and $b \le b'$ implies $f_a(b) \le f_a(b') \le f_{a'}(b')$ so that f is increasing. Conversely, if f is increasing, then f_a is increasing since if $b \le b'$ then $f_a(b) \le f_a(b')$, using $a \le a$; and $a \mapsto f_a$ is increasing since if $a \le a'$ then $f_a(b) \le f_{a'}(b)$, using $b \le b$. These are the only properties of the order that are used here.

```
Lemma Exercice1_6k: (* 30 *)
  nonempty (substrate r) -> nonempty (substrate r') ->
  bl_axioms (fun f=> restriction2
    (second_partial_function f)
    (substrate r) (set_of_increasing_mappings r' r''))
  (set_of_increasing_mappings (order_product2 r r') r'')
```

```
  (set_of_increasing_mappings r (increasing_mappings_order r' r'')).
Lemma Exercice1_6l: (* 84 *)
  nonempty (substrate r) -> nonempty (substrate r') ->
  let f:= second_partial_map2 r r' r'' in
    (is_function f &
      source f = (set_of_increasing_mappings (order_product2 r r') r'') &
      target f = (set_of_increasing_mappings r
        (increasing_mappings_order r' r'')) &
    bijection f).
Lemma Exercice1_6m:  (* 59 *)
  nonempty (substrate r) -> nonempty (substrate r') ->
  order_isomorphism (second_partial_map2 r r' r'')
  (increasing_mappings_order (order_product2 r r') r'')
  (increasing_mappings_order r (increasing_mappings_order r' r'')).
```

(c) If $\mathrm{E} \neq \emptyset$*, then* $\mathscr{A}(\mathrm{E},\mathrm{F})$ *is a lattice if and only if* F *is a lattice.*

Assume $t \in \mathrm{E}$. For $a \in \mathrm{F}$, let C_a be the constant function with value a. Then $\mathrm{C}_a \in \mathscr{A}(\mathrm{E},\mathrm{F})$. Moreover evaluation at t shows that $\mathrm{C}_a \leq \mathrm{C}_b$ is equivalent to $a \leq b$.

```
Lemma constant_increasing (* 5 *)
   y (Hy: inc y (substrate r')):
    (inc (constant_function (substrate r) (substrate r') y Hy)
      (set_of_increasing_mappings r r')).
Lemma constant_increasing1: (* 7 *)
  nonempty (substrate r) ->
  forall y (Hy: inc y (substrate r'))  y' (Hy': inc y' (substrate r')),
    gle r' y y' <->
    gle (increasing_mappings_order r r')
    (constant_function (substrate r) (substrate r') y Hy)
    (constant_function (substrate r) (substrate r') y' Hy').
```

Assume that F is a lattice; given two functions f and g we can consider the function $x \mapsto \sup(f(x), g(x))$. It is the least upper bound of f and g. Conversely, assume that $\mathscr{A}(\mathrm{E},\mathrm{F})$ is a lattice. Given two values a and b in F, the constant functions C_a and C_b with values a, and b are in $\mathscr{A}(\mathrm{E},\mathrm{F})$. Let $t \in \mathrm{E}$ and f be the supremum of C_a and C_b, and $c = f(t)$. This is an upper bound of a and b. Let d be an other upper bound. Then C_d is an upper bound of C_a and C_b, hence $f \leq \mathrm{C}_d$. Evaluating at t gives $c \leq d$. The proof is long but presents no difficulty.

```
Lemma Exercice1_6n:   (* 175 *)
  nonempty (substrate r) ->
  (lattice r' <-> lattice (increasing_mappings_order r r')).
```

(d) Suppose that E *and* F *are both non-empty. Then* $\mathscr{A}(\mathrm{E},\mathrm{F})$ *is totally ordered if and only if one of the following conditions is satisfied:*
(α*)* F *consists in a single element;*
(β*)* E *consists in a single element and* F *is totally ordered;*
(γ*)* E *and* F *are both totally ordered and* F *has two elements.*

We first show that if $\mathscr{A}(\mathrm{E},\mathrm{F})$ is totally ordered and E non-empty, then F is totally ordered, using constant functions as above. If F is a singleton, then $\mathscr{A}(\mathrm{E},\mathrm{F})$ has a single element, hence is totally ordered. If E is a singleton, all functions are constant and $\mathscr{A}(\mathrm{E},\mathrm{F})$ is isomorphic to F. Finally, assume that E and F are two totally ordered sets, and F has two elements. We may assume that these elements are distinct and satisfy $a < b$. Assume $f(u) < g(u)$. Then

$f(u) = a$ and $g(u) = b$. If no such u exists, then $f \geq g$. Otherwise, consider v; if $f(v) > g(v)$ we get $f(v) = b$ and $g(v) = a$. Since u and v can be compared, this implies that one of f and g is non-increasing.

```
Lemma Exercice1_6o: (* 7 *)
  nonempty (substrate r)->
  total_order (increasing_mappings_order r r') ->
  total_order r'.
Lemma Exercice1_6p:  (* 10 *)
  is_singleton (substrate r') ->
  total_order (increasing_mappings_order r r').
Lemma Exercice1_6q: (* 10 *)
  is_singleton (substrate r) -> total_order r' ->
  total_order (increasing_mappings_order r r').
Lemma Exercice1_6r:  (* 41 *)
  total_order r -> total_order r' ->
  (exists a, exists b, substrate r' = doubleton a b) ->
  total_order (increasing_mappings_order r r').
```

We have shown that each of the three conditions imply that $\mathscr{A}(\mathrm{E},\mathrm{F})$ is totally ordered. Let's consider the converse. We know that F is totally ordered. If the first two conditions are false, then E and F have at least two elements. Since F is totally ordered, we may assume $a < b$ in F.

For $u \in \mathrm{E}$ we consider the mapping f_u that associates a if $x \leq u$ and b otherwise. This is an increasing mapping thus an element of $\mathscr{A}(\mathrm{E},\mathrm{F})$. Consider two incomparable elements u and v of E. We have $a = f_u(u) = f_v(v)$ while $b = f_u(v) = f_v(u)$, so that f_u and f_v are incomparable. Thus $\mathscr{A}(\mathrm{E},\mathrm{F})$ totally ordered implies that E is totally ordered.

Assume now that F has more than two elements. We can find three elements such that $a < c < b$. Consider $u < v$ in E. We have $a = f_u(u) < \mathrm{C}_c(u) = c = \mathrm{C}_c(v) < f_u(v) = b$. Thus the two functions f_u and C_c are incomparable.

```
Lemma Exercice1_6s:  (* 97 *)
  nonempty(substrate r) -> nonempty (substrate r') ->
  ( total_order (increasing_mappings_order r r') <->
   (is_singleton (substrate r') \/
    (is_singleton (substrate r) & total_order r') \/
    (total_order r' & total_order r
      & exists u, exists v, substrate r' = doubleton u v))).

End Exercise1_6.
```

---

7. *In order that every mapping of an ordered set* E *into an ordered set* F *with at least two elements, which is both an increasing and a decreasing mapping, should be constant on* E*, it is necessary and sufficient that* E *should be connected with respect to the reflexive and symmetric relation "x and y are comparable" (Chapter II, § 6, Exercise 10). This condition is satisfied if* E *is either left or right directed.*

If F is empty or has a single element, then all functions with values in F are constant. This explains why F is assumed to have at least two elements. Denote by $x \equiv y$ the relation x and y

are comparable, and by $x \sim y$ the equivalence relation associated to it. We start, as in Exercise 4, with some properties of the connected components of this relation.

```
Definition ocomparable r:= fun x y => (gle r x y \/ gle r y x).

Definition cr_equiv r :=
  Exercice1.Sgraph (ocomparable r) (substrate r).
Definition cr_component r :=
  Exercice1.connected_comp (ocomparable r) (substrate r).

Lemma cr_properties r: order r -> (* 17 *)
  (is_equivalence (cr_equiv r) &
   (forall x y, ocomparable r x y -> (inc x (substrate r)& inc y (substrate r))) &
   substrate  (cr_equiv r) =  substrate r &
   (forall x, inc x (substrate r) -> class (cr_equiv r) x = cr_component r x) &
   (forall x y, ocomparable r x y -> related (cr_equiv r) x y)).
```

Given two elements of E, if they have an upper bound or lower bound, this bound is related to both elements. Thus if E is directed, it has a single component.

```
Lemma Exercice1_7a r x: right_directed r -> (* 8 *)
  inc x (substrate r) -> cr_component r x = substrate r.
Lemma Exercice1_7b r x: left_directed r -> (* 8 *)
  inc x (substrate r) -> cr_component r x = substrate r.
```

If f is increasing and decreasing, the relation $x \equiv y$ implies $f(x) = f(y)$. Thus f is constant on chains. If E is the class of x for $\equiv$ then f must be constant.

```
Lemma Exercice1_7c r r' f x y: (* 2 *)
  increasing_fun f r r' -> decreasing_fun f r r' -> ocomparable r x y ->
  W x f = W y f.
Lemma Exercice1_7d r r' f: (* 14 *)
  increasing_fun f r r' ->  decreasing_fun f r r' ->
  (exists x, inc x (substrate r) & cr_component r x = substrate r)
  -> (is_constant_function f).
```

Converse. We assume that F is a set with at least two elements a and b, and that E is not connected; more precisely we assume that there is $c \in$ E so that the component C of c is not E. We consider the function g that maps x to a if $x \in$ C, and to b otherwise. This is a non-constant function. Assume $x \leq y$. Then $x \in$ C and $y \in$ C are equivalent, so that $g(x) = g(y)$. As a consequence g is increasing and decreasing.

```
Lemma Exercice1_7e r r': (* 41 *)
  order r -> order r' ->
  (exists u, exists v, inc u (substrate r') & inc v (substrate r')  & u <>v)
  -> (exists x, inc x (substrate r) & cr_component r x <> substrate r)
  -> exists f, increasing_fun f r r' &  decreasing_fun f r r' &
    ~(is_constant_function f).
```

8. *Let* E *and* F *be two ordered sets, let f be an increasing mapping of* E *into* F, *and g an increasing mapping of* F *into* E. *Let* A *(resp.* B*) be the set of all $x \in$ E *(resp. $y \in$ F) such that $g(f(x)) = x$ (resp. $f(g(y)) = y$). Show that the two ordered sets* A *and* B *are canonically isomorphic.*

The restriction of the function f is a bijection from A onto B. It is clearly increasing.

```
Lemma Exercice1_8 r r' f g: (* 37 *)
  let A := Zo (substrate r) (fun z => W (W z f) g = z) in
    let B := Zo (substrate r') (fun z => W (W z g) f = z) in
      increasing_fun f r r' ->  increasing_fun g r' r ->
      (induced_order r A) \Is (induced_order r' B).
```

---

9. * *If* E *is a lattice, prove that*

$$\sup_j(\inf_i x_{ij}) \leq \inf_i(\sup_j x_{ij})$$

for every finite "double" family (x_{ij}). $_*$

Assume that E is a lattice. Let F be a finite subset of E. If F has a supremum, and $x \in$ E, then $F \cup \{x\}$ has a supremum. By induction on the cardinal of F, every finite subset F of E has a supremum. We have $x \geq \sup F$ if and only if $x \geq y$ for all $y \in$ F. Thus, if f is a family, with finite domain I and with values in E, then $x \geq \sup_{i\in I} f(i)$ if and only if $x \geq f(i)$ for all $i \in$ I.

```
Section Exercise1_9.
Variable r: Set.
Hypothesis lr: lattice r.

Lemma lattice_finite_sup1 X x a: (* 9 *)
  sub X (substrate r) -> least_upper_bound r X x -> inc a (substrate r) ->
  least_upper_bound r (tack_on X a) (sup r x a).
Lemma lattice_finite_inf1 X x a: (* 9 *)
  sub X (substrate r) -> greatest_lower_bound r X x -> inc a (substrate r) ->
  greatest_lower_bound r (tack_on X a) (inf r x a).
Lemma lattice_finite_sup2 x: (* 9 *)
  finite_set x -> sub x (substrate r) -> nonempty x ->
  has_supremum r x.
Lemma lattice_finite_inf2 x:  (* 9 *)
  finite_set x -> sub x (substrate r) -> nonempty x ->
  has_infimum r x.
Lemma lattice_finite_sup3 x y: (* 7 *)
  finite_set x -> nonempty x -> sub x (substrate r) ->
  (gle r (supremum r x) y <-> (forall z, inc z x -> gle r z y)).
Lemma lattice_finite_inf3 x y: (* 7 *)
  finite_set x -> nonempty x -> sub x (substrate r) ->
  (gle r y (infimum r x) <-> (forall z, inc z x -> gle r y z)).
Lemma lattice_finite_sup4 f y: (* 5 *)
  fgraph f -> finite_set (domain f) -> nonempty (domain f) ->
  sub (range f) (substrate r) ->
  (gle r (sup_graph r f) y <-> (forall z, inc z (domain f) -> gle r (V z f) y)).
Lemma lattice_finite_inf4 f y:  (* 5 *)
  fgraph f -> finite_set (domain f) -> nonempty (domain f) ->
```

```
  sub (range f) (substrate r) ->
  (gle r y (inf_graph r f) <-> (forall z, inc z (domain f) -> gle r y (V z f))).
Lemma lattice_finite_sup5 f: (* 7 *)
  fgraph f -> finite_set (domain f) -> nonempty (domain f) ->
  sub (range f) (substrate r) ->
  inc (sup_graph r f) (substrate r).
Lemma lattice_finite_inf5 f: lattice r -> (* 7 *)
  fgraph f -> finite_set (domain f) -> nonempty (domain f) ->
  sub (range f) (substrate r) ->
  inc (inf_graph r f) (substrate r).
```

It suffices to apply the previous lemmas.

```
Lemma Exercice1_9 I1 I2 f: (* 35 *)
   fgraph f -> domain f = product I1 I2 ->
  finite_set I1 -> finite_set I2 -> nonempty I1 -> nonempty I2 ->
  sub (range f) (substrate r) ->
  gle r
  (sup_graph r (L I2 (fun j => inf_graph r (L I1 (fun i => V (J i j) f)))))
  (inf_graph r (L I1 (fun i => sup_graph  r(L I2 (fun j => V (J i j) f))))).
End Exercise1_9.
```

---

10. *Let* E *and* F *be two lattices. Then a mapping* f *of* E *into* F *is increasing if and only if*

$$f(\inf(x,y)) \leq \inf(f(x),f(y))$$

for all $x \in \mathrm{E}$ *and* $y \in \mathrm{E}$.

* *Give an example of an increasing mapping* f *of the product ordered set* $\mathbf{N}\times\mathbf{N}$ *into the orders set* $\mathbf{N}$ *such that the relation*

$$f(\inf(x,y)) = \inf(f(x),f(y))$$

is false for at least one pair $(x,y) \in \mathbf{N}\times\mathbf{N}$.*

The first part is easy. We also have: f is increasing if and only if

$$\sup(f(x),f(y)) \leq f(\sup(x,y)).$$

A counter-example is addition on $\mathbf{N}$. If we take $x=(1,0)$ and $y=(0,1)$, then $f(x)=f(y)=1$, hence $\inf(f(x),f(y))=1$. On the other hand, $\inf(x,y)=(0,0)$ and $f(\inf(x,y))=0$.

```
Section Exercise1_10.
Variable r r': Set.
Hypothesis (lr: lattice r) (lr': lattice r').

Lemma Exercise1_10 f: (* 14 *)
  is_function f -> substrate r = source f ->
  substrate r' = target f ->
  ((increasing_fun f r r') <->
   (forall x y, inc x (substrate r) -> inc y (substrate r) ->
```

```
    gle r' (W (inf r x y) f) (inf r' (W x f) (W y f)))).

Lemma Exercise1_10b f: (* 14 *)
  is_function f -> substrate r = source f ->
  substrate r' = target f ->
  ((increasing_fun f r r') <->
   (forall x y, inc x (substrate r) -> inc y (substrate r) ->
    gle r' (sup r' (W x f) (W y f)) (W (sup r x y) f) )).
Lemma product2_lattice: (* 25 *)
  lattice (order_product2 r r').

End Exercise1_10.

Lemma lattice_bnat: lattice Bnat_order. (* 1 *)
Lemma lattice_bnat2: lattice (order_product2 Bnat_order Bnat_order). (* 1 *)

Lemma  Exercise1_10_bis: (* 37 *)
  let r :=  (order_product2 Bnat_order Bnat_order) in let r' := Bnat_order in
  let f := BL (fun z => (P z) +c (Q z)) (product Bnat Bnat) Bnat in
    (lattice r & lattice r' & is_function f & substrate r = source f &
  substrate r' = target f &   (increasing_fun f r r') &
  exists x, exists y, inc x (substrate r) & inc y (substrate r) &
    (W (inf r x y) f)  <> (inf r' (W x f) (W y f))).
```

---

11. *A lattice* E *is said to be* complete *if every subset of* E *has a least upper bound and a greatest lower bound in* E*; this means in particular that* E *has a greatest and a least element.*

(a) Show that if an ordered set E *is such that every subset of* E *has a least upper bound in* E*, then* E *is a complete lattice.*

The first claim is obvious: it suffices to take the supremum of infimum of the empty set. The second claim is easy: Let X be a set and X' be the set of lower bounds. If X' has a supremum, this is the infimum of X.

```
Definition complete_lattice r := order r &
  forall X, sub X (substrate r) -> (has_supremum r X & has_infimum r X).

Lemma exercise1_11a r: (* 6 *)
  complete_lattice r ->
  ((exists a, greatest_element r a) & (exists b, least_element r b)).
Lemma exercise1_11b r: order r ->  (* 12 *)
  (forall X, sub X (substrate r) -> has_supremum r X) ->
  complete_lattice r.
```

(b) A product of ordered sets is a complete lattice if only if each of the factors is a complete lattice.

Let X be a subset of $\prod \mathrm{E}_i$ and $\mathrm{X}_i = \mathrm{pr}_i \mathrm{X}$. If each E_i is a complete lattice, then $\sup \mathrm{X}_i$ is the supremum of the set X_i, and $i \mapsto \sup \mathrm{X}_i$ is the supremum of X. Conversely, if the product is a complete lattice, it is non-empty since it has a smallest element. If $\mathrm{X}_i \subset \mathrm{E}_i$ we can find a set X with $\mathrm{X}_i = \mathrm{pr}_i \mathrm{X}$. If x is the supremum of X then x_i is the supremum of X_i.

```
Lemma exercise1_11c g: (* 49 *)
```

```
  order_fam g ->
  (forall i, inc i (domain g) -> complete_lattice (V i g)) ->
  complete_lattice (order_product g).
Lemma exercise1_11d g:  (* 45 *)
  order_fam g -> complete_lattice (order_product g) ->
  (forall i, inc i (domain g) -> complete_lattice (V i g)).
```

(c) An ordinal sum (Exercise 3) $\sum_{\iota\in I} E_\iota$ *is a complete lattice if and only if the following conditions are satisfied:*
(I) I *is a complete lattice.*
(II) If J *is a subset of* I *which has no greatest element, and if* $\sigma = \sup J$*, then* E_σ *has a least element.*
(III) For each $\iota \in I$ *every subset of* E_ι *which has an upper bound in* E_ι *has a least upper bound in* E_ι*.*
(IV) For each $\iota \in I$ *such that* E_ι *has no greatest element, the set of all* $\kappa > \iota$ *has a least element* α *and* E_α *has a least element.*

Condition (III) has to be replaced by: "every non-empty subset of E_ι which has an upper bound has a supremum". Example. Consider the sum of two sets, a singleton and the opposite order of **N**. The empty set is bounded in **N** but has no greatest lower bound.

We recall that the ordinal sum is the set of all (i, x_i) where $i \in I$ and $x_i \in E_i$, and $(i, x_i) < (j, x_j)$ if either $i < j$ (in I) or $i = j$ and $x_i < x_j$ (in the ordered set E_i). We assume E_i non-empty, so that there is a function k defined on I such that $k(i) \in E_i$.

Assume first that the ordinal sum is a complete lattice. Let J be a subset of I, and consider the least upper bound x of $k(J)$. This element has the form x_j for $j \in J$, so that j is the supremum of J, and condition (I) holds. Assume that J has no greatest element, so that $j \notin J$. Every element in E_j is an upper bound of $k(J)$. Thus x_j must be the least element of E_j. This shows condition (II).

Consider now a subset X_i of E_i. This can be identified to a subset X of the sum that has a least upper bound x. Assume that $u \in X_i$ and X_i is bounded above by v. We have $(i, u) \le x \le (i, v)$ so that $x \in E_i$ and x has the form (i, y). Then y is the least upper bound of X_i; this shows point (III).

Consider finally point (IV). Let $i \in E$. Consider J, the subset of I formed of indices $j > i$; consider E_i as a subset X of the sum. Let x_k be its supremum. Since E_i is nonempty, there is y_i with $y_i \le x_k$, hence $i \le k$. If $i = k$, then y_k is the greatest element of E_i. Let's assume that E_i has no greatest element. This implies $k \in J$. For every $j \in J$, any element of E_j is an upper bound of X, thus $k \le j$. This means that k is the least element of J. Take $j = k$. Then x_k is the least element of E_k. This proves (IV).

Conversely, assume the four assumptions true. Take a subset X of the sum. Let J be the set of all i such that at least one element of X is in E_i. This set has a least upper bound, say i. Assume first that J has no greatest element. By assumption (II), the set E_i has a least element x. We pretend that the pair $x' = (i, x)$ is the least upper bound of X. It is a strict upper bound of X since i is a strict upper bound of J. Consider another upper bound, say $z' = (j, z)$. We have $i \le j$. If $i < j$ it follows $x' < z'$. If $i = j$, we have $x \le z$ in E_i hence $x' \le z'$.

Assume now that J has a greatest element j. Let X_j be the (nonempty) set of all x_k of X such that $k = j$. Consider first the case where X_j has an upper bound and apply (III). There a least upper bound x of X_j. Let $x_j = (j, x)$ in the ordinal sum. This is the supremum of X.

Assume that X_j has no upper bound in E_j. In particular E_j has no greatest element; by (IV) there is an index k, the least index such that $k > j$ and a least element x in E_k. This is the

supremum of X.

```
Definition greatest_induced r X x := greatest_element (induced_order r X) x.
Definition least_induced r X x := least_element (induced_order r X) x.

Lemma exercise1_11e r g: (* 234 *)
  orsum_ax r g -> orsum_ax2 g ->
  (complete_lattice (order_sum r g) <->
  (complete_lattice r
    & (forall j, sub j (substrate r) ->
      ~ (exists u, greatest_induced r j u) ->
      exists v, least_element (V (supremum r j) g) v)
    & (forall i x, inc i (substrate r) -> sub  x (substrate (V i g)) ->
      (exists u, upper_bound (V i g) x u) ->  nonempty x ->
      (exists u, least_upper_bound  (V i g) x u))
    & (forall i, inc i (substrate r) ->
      ~ (exists u, greatest_element (V i g) u ) ->
      exists v, least_induced r (Zo (substrate r) (fun j =>
        glt r i j)) v
      & exists w, least_element (V v g) w))).
```

(d) The ordered set $\mathscr{A}(E,F)$ *of increasing maps of an ordered set* E *into an ordered set* F *(Exercise 6) is a complete lattice if and only if* F *is a complete lattice.*

Assume that $\mathscr{A}(E,F)$ is a complete lattice. If E is non-empty, then F is a complete lattice, see Exercise 6 (c).

Let's show the converse. We consider a subset X of $\mathscr{A}(E,F)$, and for each $x \in E$ the set G_x of all $f(x)$ for $f \in X$. If F is a complete lattice, this set has a least upper bound, say f_x. This gives us a function $f : x \mapsto f_x$, which is the least upper bound. The function is increasing by the following argument: assume $a \leq b$. We have $g(b) \leq \sup_{h\in X} h(b)$ if $g \in X$. Thus $g(a) \leq g(b) \leq f(b)$. Taking the supremum over g gives $f(a) \leq f(b)$.

```
Lemma exercise1_11f r r': (* 36 *)
  order r -> order r' ->  nonempty (substrate r) ->
  complete_lattice  (increasing_mappings_order r r') -> complete_lattice r'.
Lemma exercise1_11g r r': (* 50 *)
  order r -> order r' ->
  complete_lattice r' -> complete_lattice  (increasing_mappings_order r r').
```

---

12. *Let* Φ *be a mapping of a set* A *into itself. Let* $\mathfrak{F}$ *be the subset of* $\mathfrak{P}(A)$ *consisting of all* $X \subset A$ *such that* $f(X) \subset X$ *for each* $f \in \Phi$. *Show that* $\mathfrak{F}$ *is a complete lattice with respect to the relation of inclusion.*

Apply lemma union_is_sup1 show that the union $\bigcup X \in \mathfrak{F}$ of a family of sets is the least upper bound (The greatest lower bound is the intersection if non-empty, the set E otherwise).

```
Lemma Exercise1_12 E f: (* 14 *)
  is_function f -> source f = E -> target f = E ->
  complete_lattice (inclusion_suborder (Zo (powerset E) (fun X =>
    sub (image_by_fun f X) X))).
```

---

13. *Let* E *be an ordered set. A mapping f of* E *into itself is said to be a* closure *if it satisfies the following conditions: (1) f is increasing, (2) for each $x \in \mathrm{E}$, $f(x) \geq x$, (3) for each $x \in \mathrm{E}$, $f(f(x)) = f(x)$. Let* F *be the set of elements of* E *which are invariant under f.*

(a) Show that for each $x \in \mathrm{E}$ the set F_x of elements $y \in \mathrm{F}$ such that $x \leq y$ is not empty and has a least element, namely $f(x)$. Conversely, if G *is a subset of* E *such that, for each $x \in \mathrm{E}$, the set of all $y \in \mathrm{G}$ such that $x \leq y$ has a least element $g(x)$, then g is a closure and* G *is the set of elements of* E *that are invariant under g.*

(b) Suppose that E *is a complete lattice. Show that the greatest lower bound in* E *of any non-empty subset of* F *belongs to* F.

(c) Show that if E *is a lattice, then $f(\sup(x, y)) = f(\sup(f(x), f(y)))$ for each pair of elements x, y of* E.

Let's start with the easy points. Assume that E is a complete lattice, X a (possibly empty) subset of F, and y its greatest lower bound. For $x \in \mathrm{X}$ we have $y \leq x$, thus $f(y) \leq f(x) = x$, so that $f(y)$ is a lower bound and $f(y) \leq y$. Since $f(y) \leq y$, this shows that y is a fixed-point. Assume now that E is merely a lattice. Consider two elements x, y of E. Let $z = \sup(x, y)$ and $\mathrm{T} = \sup(f(x), f(y))$. By Exercise 10 we have $\mathrm{T} \leq f(z)$. Since $x \leq f(x)$ and $y \leq f(y)$ we deduce $z \leq \mathrm{T}$. We deduce $f(z) \leq f(\mathrm{T}) \leq f(f(z)) = f(z)$. Thus $f(z) = f(\mathrm{T})$.

First part of (a) is trivial. Second part is a bit more complicated. Consider a set G. Let G_x be the set of upper bounds of x that are in G. Assume that this set has a least element. We deduce a function g such that $g(x)$ is the least element of G_x. The key relation is: if $x \in \mathrm{E}$ and $y \in \mathrm{G}$, if $x \leq y$ then $g(x) \leq y$. In particular, if $x \in \mathrm{G}$ then $g(x) = x$.

Let $\mathrm{P}_\mathrm{G}(x, y)$ be the property that y is the least element of G_x. We have an assumption that for each x there is a y satisfying $\mathrm{P}(x, y)$. We use the axiom of choice and define a function $y = g_\mathrm{G}(x)$. This assumption says that $g_\mathrm{G}(x) \in \mathrm{G}_x$ and is the smallest element of this set. The first assumption says that, if $y = g_\mathrm{G}(x)$, then y belongs to E (so that there is a function $g : \mathrm{E} \to \mathrm{E}$), it belongs to G and $x \leq y$. If $x \in \mathrm{G}$, then $x \in \mathrm{G}_x$, the second assumption says $y \leq x$, hence $y = x$. Conversely, if $x = y$ then $x \in \mathrm{G}$ (since $y \in \mathrm{G}$). Assume now $a \leq b$. Since $b \leq g(b)$ we have $g(b) \in \mathrm{G}_a$, hence $g(a) \leq g(b)$. This shows that the function is increasing, thus is a closure.

```
Definition is_closure  f r :=
  increasing_fun f r r  &
  (forall x, inc x (substrate r) -> gle r x (W x f)) &
  (forall x, inc x (substrate r) -> W (W x f) f = W x f).

Definition set_of_invariants f := Zo (source f) (fun x => W x f = x).
Definition set_of_upper_bounds F r x := Zo F (fun y => gle x y).


Section Exercise1_13.
Variables r f: Set.
Hypothesis cf: is_closure f r.

Lemma Exercise1_13c E: (* 10 *)
  complete_lattice r ->
  let F := set_of_invariants f in
    sub E F -> inc (infimum r E) F.
Lemma Exercise1_13d x y: (* 14 *)
  lattice r ->
  inc x (substrate r) -> inc y (substrate r) ->
  W (sup r x y) f = W (sup r (W x f) (W y f)) f.
```

```
Lemma Exercise1_13a x:  (* 10 *)
  let F := set_of_invariants f in
    inc x (source f) ->
    least_element (induced_order r (set_of_upper_bounds F r x)) (W x f).
End Exercise1_13.

Lemma Exercise1_13b r G: (* 46 *)
  order r -> sub G (substrate r) ->
  let g:= fun x => the_least_element (induced_order r
    (set_of_upper_bounds G r x))  in
  (forall x, inc x (substrate r) -> exists y,
    least_element (induced_order r (set_of_upper_bounds G r x)) y) ->
  (is_closure (BL g (substrate r) (substrate r)) r &
    (G = set_of_invariants (BL g (substrate r) (substrate r)))).
```

---

14. *Let* A *and* B *be two sets, and let* R *be any subset of* $A \times B$. *For each subset* X *of* A *(resp. each subset* Y *of* B*) let* $\rho(X)$ *(resp.* $\sigma(Y)$*) denote the set of all* $y \in B$ *(resp.* $x \in A$*) such that* $(x, y) \in R$ *for all* $x \in A$ *(resp.* $(x, y) \in R$ *for all* $y \in B$*). Show that* ρ *and* σ *are decreasing mappings and that the mapping* $X \to \sigma(\rho(X))$ *and* $Y \to \rho(\sigma(Y))$ *are closures (Exercise 13) in* $\mathfrak{P}(A)$ *and* $\mathfrak{P}(B)$ *respectively (ordered by inclusion).*

The definition has to be corrected as: "For each subset X of A (resp. each subset Y of B) let $\rho(X)$ (resp. $\sigma(Y)$) denote the set of all $y \in B$ (resp. $x \in A$) such that $(x, y) \in R$ for all $x \in X$ (resp. $(x, y) \in R$ for all $y \in Y$)."

The functions σ and ρ are trivially decreasing, so that there composition is increasing. We have $x \subset \sigma\rho x$ and $y \subset \rho\sigma x$. The same argument as in Proposition 2 of § 1, no. 5, shows $\rho\sigma\rho = \rho$ and $\sigma\rho\sigma = \sigma$.

```
Lemma Exercice1_14 A B R: (* 70 *)
  let rho := fun X => Zo B (fun y => forall x,inc x X -> inc (J x y) R) in
    let sigma := fun Y => Zo A (fun x => forall y, inc y Y -> inc (J x y) R) in
      let fr:=BL rho (powerset A) (powerset B) in
        let fs:= BL sigma (powerset B) (powerset A) in
          let iA := inclusion_order A in
            let iB := inclusion_order B in
              sub R (product A B) ->
              ( decreasing_fun fr iA iB & decreasing_fun fs iB iA &
                is_closure (compose fs fr) iA & is_closure (compose fr fs) iB).
```

---

15. *(a) Let* E *be an ordered set, and for each subset* X *of* E *let* $\rho(X)$ *(resp.* $\sigma(X)$*) denote the set of upper (resp. lower) bounds of* X *in* E. *Show that, in* $\mathfrak{P}(E)$, *the set* $\tilde{E}$ *of subsets* X *such that* $X = \sigma(\rho(X))$ *is a complete lattice, and that the mapping* $i : x \to \sigma(\{x\})$ *is an isomorphism (called* canonical*) of* E *onto an ordered subset* E' *of* $\tilde{E}$ *such that, if a family* (x_ι) *of elements of* E *has a least upper bound (resp. greatest lower bound) in* E, *the image of this least upper bound (resp. greatest lower bound) is the least upper bound (resp. greatest lower bound) in* $\tilde{E}$ *of the family of images of the* x_ι. $\tilde{E}$ *is called the* completion *of the ordered set* E.

(b) Show that, for every subset X *of* E, $\sigma(\rho(X))$ *is the least upper bound in* $\tilde{E}$ *of the subset* $i(X)$ *of* $\tilde{E}$. *If* f *is any increasing mapping of* E *into a complete lattice* F, *there exists a unique increasing mapping* $\bar{f}$ *of* $\tilde{E}$ *into* F *such that* $f = \bar{f} \circ i$ *and* $\bar{f}(\sup Z) = \sup(\bar{f}(Z))$ *for every subset* Z *of* $\tilde{E}$.

(c) If E *is totally ordered, show that* $\tilde{E}$ *is totally ordered.*

```
Definition set_of_up_bounds r X :=
  Zo (substrate r)(fun z => upper_bound r X z).
Definition set_of_lo_bounds r X :=
  Zo (substrate r)(fun z => lower_bound r X z).
Definition set_of_uplo_bounds r X := set_of_lo_bounds r (set_of_up_bounds r X).
Definition completion r:=
  Zo (powerset (substrate r)) (fun z => z = set_of_uplo_bounds r z).
Definition completion_order r := inclusion_suborder (completion r).
```

Let $f = \sigma \circ \rho$. This is a closure, according to the previous exercise, but we give a direct proof. The function f is increasing, $x \subset f(x)$, $\sigma\rho\sigma = \sigma$, and $f(f(x)) = f(x)$. We deduce that, if $X \subset E$, then $\sigma(X)$ and $f(X)$ are in $\tilde{E}$.

```
Lemma Exercise1_15a1 r A B: sub A B -> (* 2 *)
  sub (set_of_up_bounds r B) (set_of_up_bounds r A).
Lemma Exercise1_15a2 r A B: sub A B -> (* 2 *)
  sub (set_of_lo_bounds r B) (set_of_lo_bounds r A).
Lemma Exercise1_15a3 r A B: sub A B -> (* 1 *)
  sub (set_of_uplo_bounds r A) (set_of_uplo_bounds r B).
Lemma Exercise1_15a4 r A:  (* 2 *)
  sub A (substrate r) -> sub A (set_of_uplo_bounds r A).
Lemma Exercise1_15a5 r A: sub A (substrate r) -> (* 4 *)
  set_of_lo_bounds r (set_of_up_bounds r (set_of_lo_bounds r A)) =
  (set_of_lo_bounds r A).
Lemma Exercise1_15a6 r A: sub A (substrate r) -> (* 2 *)
  set_of_uplo_bounds r (set_of_uplo_bounds r A) =
  (set_of_uplo_bounds r A).
Lemma Exercise1_15a7 r A: sub A (substrate r) -> (* 1 *)
  inc (set_of_uplo_bounds r A) (completion r).
Lemma Exercise1_15a8 r A: sub A (substrate r) -> (* 2 *)
  inc (set_of_lo_bounds r A) (completion r).
```

Assume now that E is an ordered set, and assume that A has a least upper bound α. Then $\sigma(A)$ is the set of all x such that $x \leq \alpha$. If B has a least upper bound β, then $\sigma(A) = \sigma(B)$ is $\alpha = \beta$. In particular, if A and B are singletons we get A = B.

If E has a least element e, then $\{e\}$ is the least element of $\tilde{E}$, otherwise $\emptyset$ is the least element of $\tilde{E}$. In any case E is the greatest element.

```
Section Exercise1_15.
Variable r:Set.
Hypothesis or: order r.

Lemma Exercise1_15a9  x y: (* 7 *)
  inc x (substrate r) -> inc y (substrate r) ->
  (set_of_lo_bounds r (singleton x) = set_of_lo_bounds r (singleton y))
    -> x = y.
Lemma Exercise1_15a10 e: (* 14 *)
```

```
  least_element r e ->
  least_element (completion_order r) (singleton e).
Lemma Exercise1_15a11: (* 12 *)
  ~ (exists e, least_element r e) ->
  least_element (completion_order r) emptyset.
Lemma Exercise1_15a12: (* 3*)
  exists x, least_element (completion_order r) x.
Lemma Exercise1_15a13: (* 6 *)
  greatest_element (completion_order r) (substrate r).
```

We show that the completion is a complete lattice. Given a family X_i, the least upper bound is $f(\bigcup X_i)$, the greatest lower bound is $f(\bigcap X_i)$. This is because f is increasing and $f(X_i) = X_i$. In the case of intersection, we have to consider the case where the family is empty.

```
Lemma Exercise1_15a14 X: (* 16 *)
  sub X (completion r) ->
  least_upper_bound (completion_order r) X (set_of_uplo_bounds r (union X)).
Lemma Exercise1_15a15 X: (* 17 *)
  sub X (completion r) -> nonempty X ->
  greatest_lower_bound (completion_order r) X
  (set_of_uplo_bounds r (intersection X)).
Lemma Exercise1_15a16:  (* 3 *)
  complete_lattice (completion_order r).
```

Let's study the property of $i(x) = \sigma(\{x\})$. We know that it is injective from E into $\tilde{E}$. It is an order isomorphism on its image.

```
Lemma Exercise1_15a17: (* 2 *)
  bl_axioms (fun z => set_of_lo_bounds r (singleton z))
  (substrate r) (substrate (completion_order r)).
Lemma Exercise1_15a18a x: (* 1 *)
  inc x (substrate r) -> (inc x (set_of_lo_bounds r (singleton x))).
Lemma Exercise1_15a18: (* 11 *)
  order_morphism (BL  (fun z => set_of_lo_bounds r (singleton z))
      (substrate r) (substrate (completion_order r)))
     r (completion_order r).
```

Let's study the links between i and bounds. In the case of the greatest lower bound, the empty family is an exception.

```
Lemma Exercise1_15a19 X x: (* 26 *)
  sub X (substrate r) -> least_upper_bound r X x ->
  least_upper_bound  (completion_order r)
  (fun_image X (fun z => set_of_lo_bounds r (singleton z)))
  (set_of_lo_bounds r (singleton x)).
Lemma Exercise1_15a20  X x: (* 38 *)
  sub X (substrate r) -> greatest_lower_bound r X x ->
  greatest_lower_bound  (completion_order r)
  (fun_image X (fun z => set_of_lo_bounds r (singleton z)))
  (set_of_lo_bounds r (singleton x)).
```

Consider now point (b): $\sigma(\rho(X)) = \sup i\langle X\rangle$ is easy. The statement "every increasing function f can be extended to $\bar{f}$ such that $f = \bar{f} \circ i$ and $\bar{f}(\sup Z) = \sup(\bar{f}(Z))$" is wrong.

Assume that $E = \{a, b, c\}$ with, $a \leq c$ and $b \leq c$. Then $\tilde{E}$ has four elements, the singletons $\alpha = \{a\} = i(a)$ and $\beta = \{b\} = i(b)$, the least element $\emptyset$ and the greatest element E. Note that

$\mathrm{E} = i(c)$. The first assumption on $\bar{f}$ gives $\bar{f}(\alpha) = f(a)$, $\bar{f}(\beta) = f(b)$ and $\bar{f}(\mathrm{E}) = f(c)$. The second assumption, for $\mathrm{Z} = \{\alpha,\beta\}$, reads $f(\sup(a,b)) = \sup(f(a),f(b))$. This is not necessarily the case, so that $\bar{f}$ does not always exist.

```
Lemma Exercise1_15b1 X: (* 18 *)
  sub X (substrate r) ->
  least_upper_bound (completion_order r)
  (fun_image X (fun z => set_of_lo_bounds r (singleton z)))
  (set_of_uplo_bounds r X).

End Exercise1_15.
```

Consider now (c): the set $\tilde{\mathrm{E}}$ if totally ordered whenever E is totally ordered. This follows from that fact that, if $\mathrm{X} \in \tilde{\mathrm{E}}$, $a \in \mathrm{X}$ and $b \leq a$ then $b \in \mathrm{X}$.

```
Lemma Exercise1_15c r: total_order r -> (* 19 *)
  total_order (completion_order r).
```

¶ 16. *A lattice* E *is said to be* distributive *if it satisfies the following two conditions*

(D') $$\sup(x,\inf(y,z)) = \inf(\sup(x,y),\sup(x,z))$$

(D") $$\inf(x,\sup(y,z)) = \sup(\inf(x,y),\inf(x,z))$$

for all x, y, z in E. *A totally ordered set is a distributive lattice.*

(a) Show that each of the conditions (D'), (D") separately implies the condition

(D) $$\sup(\inf(x,y),\inf(y,z),\inf(z,x)) = \inf(\sup(x,y),\sup(y,z),\sup(z,x))$$

for all x, y, z in E.

(b) Show that the condition (D) implies the condition

(M) $$\textit{If } x \geq z, \textit{ then } \sup(z,\inf(x,y)) = \inf(x,\sup(y,z))).$$

Deduce that (D) implies each of (D') and (D"), and hence that the three axioms (D), (D') and (D") are equivalent (to show, for example, that D *implies* D′*, take the least upper bound of x and each side of (D) and use (M)).*

(c) Show that each of the two conditions

(T') $$\inf(z,\sup(x,y)) \leq \sup(x,\inf(y,z)),$$

(T") $$\inf(\sup(x,y),\sup(z,\inf(x,y))) = \sup(\inf(x,y),\inf(y,z),\inf(z,x)),$$

for all x, y, z in E *is necessary and sufficient for* E *to be distributive. (To show that (T') implies (D"), consider the element*

$$\inf(z,\sup(x,\inf(y,z))).)$$

Let's introduce the following definitions.

```
Definition distributive_lattice1 r :=
  forall x y z, inc x (substrate r) ->inc y (substrate r) ->
    inc z (substrate r) ->
    sup r x (inf r y z) = inf r (sup r x y) (sup r x z).
Definition distributive_lattice2 r :=
  forall x y z, inc x (substrate r) ->inc y (substrate r) ->
    inc z (substrate r) ->
    inf r x (sup r y z) = sup r (inf r x y) (inf r x z).
Definition distributive_lattice3 r :=
  forall x y z, inc x (substrate r) ->inc y (substrate r) ->
    inc z (substrate r) ->
    sup r (inf r x y) (sup r (inf r y z) (inf r z x)) =
    inf r (sup r x y) (inf r (sup r y z) (sup r z x)).
Definition distributive_lattice4 r :=
  forall x y z, inc x (substrate r) ->inc y (substrate r) ->
    inc z (substrate r) ->
    gle r z x -> sup r z (inf r x y) = inf r x (sup r y z).
Definition distributive_lattice5 r:=
  forall x y z, inc x (substrate r) ->inc y (substrate r) ->
    inc z (substrate r) ->
    gle r (inf r z (sup r x y)) (sup r x (inf r y z)).
Definition distributive_lattice6 r :=
  forall x y z, inc x (substrate r) ->inc y (substrate r) ->
    inc z (substrate r) ->
    inf r (sup r x y) (sup r z (inf r x y))
    = sup r (inf r x y) (sup r (inf r y z) (inf r z x)).
```

Let's show the following trivial facts. In particular, we state associativity of sup and inf, which are implicit in the formulation (D). In a totally ordered set, formula (D') is true. It suffices to consider all possibilities of a triple (x, y, z).

```
Lemma inf_comm r x y: (* 1 *)
  inf r x y = inf r y x.
Lemma sup_comm r x y: (* 1 *)
  sup r x y = sup r y x.
Lemma total_order_dlattice r: (* 18 *)
  total_order r -> distributive_lattice1 r.

Section Exercise1_16.
Variable r: Set.
Hypothesis lr: lattice r.

Lemma lattice_props: (* 39 *)
  let E := substrate r in
    ( (forall x y, inc x E->  inc y E -> inc (sup r x y) E)
      & (forall x y, inc x E->  inc y E -> inc (inf r x y) E)
      & (forall x y, inc x E->  inc y E -> sup r (inf r x y) y = y)
      & (forall x y, inc x E->  inc y E -> inf r (sup r x y) y = y)
      & (forall x y z, inc x E->  inc y E -> inc z E ->
        sup r x (sup r y z) = sup r (sup r x y) z)
      & (forall x y z, inc x E->  inc y E -> inc z E ->
        inf r x (inf r y z) = inf r (inf r x y) z)
      & (forall x, inc x E -> sup r x x = x)
      & (forall x, inc x E -> inf r x x = x)
      & (forall x y, inc x E->  inc y E -> sup r (sup r x y) x = sup r x y)
      & (forall x y, inc x E->  inc y E -> inf r (inf r x y) x = inf r x y)).
```

Both conditions (D') and (D") imply (D). This is easy: we rewrite (D') five times.

```
Lemma Exercise1_16a: (* 31 *)
  ( (distributive_lattice1 r -> distributive_lattice3 r) &
    (distributive_lattice2 r -> distributive_lattice3 r)).
```

Let's show that (D) implies (M), (D) and (D'). Notice first that if $z \leq x$, then $\sup(x,z) = x$ and $\inf(x,z) = x$. Injecting these relations in (M) and simplifying further yields (D), so that (D) implies (M).

Let's show that that (D) implies (D'). Write (D) as $\alpha = \beta$. Set $a = \sup(x,\alpha)$. This the supremum of four terms, two of them being smaller than x. After simplification, we see that a is the LHS of (D'). Thus, we have to show $\sup(x,\beta) = \inf(\sup(x,y),\sup(x,z))$. Since β is a infimum we can apply (M) and get $\sup(x,\beta) = \inf(\sup(x,y),d)$ Applying (M) to d gives the result.

Exchanging inf and sup and applying (M) in the other way shows that (D) implies (D").

```
Lemma Exercise1_16b: (* 54 *)
  ( (distributive_lattice3 r -> distributive_lattice4 r) &
    (distributive_lattice3 r -> distributive_lattice1 r) &
    (distributive_lattice3 r -> distributive_lattice2 r)).
```

We prove here that (D") implies (T') and (T') implies (D'). The first claim is obvious. We first notice that in (D') the sup is smaller than the inf. Write (D') as $a = b$. Let $c = \inf(z,\sup(x,y))$. Applying (T') gives $a \leq \sup(x,c)$ and $c \leq b$, from which $a \leq b$ follows. We also show that (D") is equivalent to (T"). One implication is easy. Conversely (T") (of the form $a = b$) implies (T') (of the form $a' \leq b'$), since $a' \leq a$ and $b \leq b'$ are clear.

```
Lemma Exercise1_16c:  (* 35 *)
  (distributive_lattice3 r <-> distributive_lattice5 r).
Lemma Exercise1_16d:  (* 35 *)
  (distributive_lattice3 r <-> distributive_lattice6 r).
End Exercise1_16.
```

¶ 17. *A lattice* E *which has a least element* α *is said to be* relatively complemented *if, for each pair of elements x, y of* E *such that* $x \leq y$*, there exists an element* x' *such that* $\sup(x,x') = y$ *and* $\inf(x,x,') = \alpha$*. Such an element* x' *is called a* relative complement *of x with respect to y.*

We define now a relatively complemented set, a Boolean lattice, the complement and the standard complement (see below), and show that in a relatively complemented set, a complement does exists. Assume that α is the least element and ω the greatest element. We have $\sup(x,\alpha) = x$, $\inf(x,\omega) = x$ $\inf(x,\alpha) = \alpha$ and $\sup(x,\omega) = \omega$.

```
Definition complement_pr r x y x' :=
 (inc x' (substrate r) &sup r x x' = y & inf r x x' =  the_least_element r).

Definition relatively_complemented r:=
  lattice r & (exists u, least_element r u) &
  (forall x y, gle r x y -> exists x', complement_pr r x y x').
```

```
Definition boolean_lattice r:=
  relatively_complemented r &  (exists u, greatest_element r u) &
  distributive_lattice3 r.

Definition the_complement r x y:=
   choose (fun x' => complement_pr r x y x').

Definition standard_completion r x :=
  the_complement r x (the_greatest_element r).

Lemma sup_monotone r a b c: (* 5 *)
  lattice r -> inc a (substrate r) -> gle r b c->
  gle r (sup r a b)  (sup r a c).
Lemma inf_monotone r a b c:  (* 5 *)
  lattice r -> inc a (substrate r) -> gle r b c->
  gle r (inf r a b)  (inf r a c).

Lemma the_complement_pr r x y: (* 2 *)
  relatively_complemented r ->  gle r x y ->
  complement_pr r x y (the_complement r x y).
Lemma least_greatest_pr r: order r -> (* 9 *)
  ((forall a, inc a (substrate r) -> (exists u, least_element r u) ->
    sup r (the_least_element r) a = a) &
    (forall a, inc a (substrate r) ->  (exists u, greatest_element r u) ->
      inf r a (the_greatest_element r) = a) &
    (forall a, inc a (substrate r) -> (exists u, least_element r u) ->
      inf r (the_least_element r) a = (the_least_element r)) &
    (forall a, inc a (substrate r) ->  (exists u, greatest_element r u)->
      sup r a (the_greatest_element r) = (the_greatest_element r))).
Lemma least_greatest_pr1 r a: boolean_lattice r -> (* 2 *)
  inc a (substrate r) ->
  ( sup r (the_least_element r) a = a &
    inf r a (the_greatest_element r) = a &
    inf r (the_least_element r) a = (the_least_element r) &
    sup r a (the_greatest_element r) = (the_greatest_element r)).
```

[*] *(a) Show that the set* E *of vector subspaces of a vector space of dimension* ≥ 2*, ordered by inclusion, is a relatively complemented lattice, but that if* x*,* y *are two elements of* E *such that* $x \leq y$*, there exists in general several distinct complements of* x *with respect to* y*.*[*]

Assume that X is a subspace of Y; consider a basis $(x_i)_{i\in \mathrm{I}}$ of Y, such that for some $\mathrm{J} \subset \mathrm{I}$, $(x_i)_{i\in \mathrm{J}}$ is a basis of X. Let X' be the space spanned by $(x_i)_{i\in \mathrm{I}-\mathrm{J}}$. This is a relative complement. Fix $i \in \mathrm{I}-\mathrm{J}$. We may replace x_i by any element of Y not in $\mathrm{X}\cup\mathrm{X}'$; this changes X', but it will remain a relative complement. An example of such an element is $x_i + x_j$ where $j \in \mathrm{J}$. It exists if X is neither of dimension 0 nor equal to Y.

(b) If E *is distributive and relatively complemented, show that if* $x \leq y$ *in* E*, there exists a unique relative complement of* x *with respect to* y*.* E *is said to be a* Boolean lattice *if it is distributive and relatively complemented and if, moreover, it has a greatest element* ω*. For each* $x \in$ E*, let* x^* *be the complement of* x *with respect to* ω*. The mapping* $x \to x^*$ *is an isomorphism of* E *onto the ordered set obtained by endowing* E *with the opposite ordering, and we have* $(x^*)^* = x$*. If* A *is any set, then the set* $\mathfrak{P}(\mathrm{A})$ *of all subsets* A*, ordered by inclusion, is a Boolean lattice.*

Let's show that in a distributive and relatively complemented set, there exists a unique complement. Consider x, complemented by x' and x'', and apply relation (D) to these three

quantities. We get $\sup(\alpha, \sup(\alpha, \inf(x', x'))) = \inf(y, \inf(y, \sup(x', x')))$. This simplifies to $\inf(x', x'') = \sup(x', x'')$ and to $x' = x''$.

Let's call "standard completion" and denote by x^* the completion with the greatest element of E. By uniqueness of completion and commutativity of supremum and infimum, we have $(x^*)^* = x$.

```
Lemma Exercice1_17a r x y:    (* 18 *)
  relatively_complemented r ->
  distributive_lattice3 r -> gle r x y ->
  exists_unique (fun x' => complement_pr r x y x').
Lemma standard_completion_pr r x: (* 2 *)
  boolean_lattice r -> inc x (substrate r) ->
  complement_pr r x (the_greatest_element r) (standard_completion r x).
Lemma standard_completion_unique r x y: (* 6 *)
  boolean_lattice r -> inc x (substrate r) ->
  complement_pr r x (the_greatest_element r) y ->
  y = standard_completion r x.
Lemma standard_completion_involutive r x: (* 4 *)
  boolean_lattice r -> inc x (substrate r) ->
  standard_completion r (standard_completion r x) = x.
```

Consider two elements x and y, their standard completion a and b. Let $c = \inf(a, b)$. We have $\inf(y, c) = \alpha$. We have $\sup(y, c) = \sup(y, a)$ (we use (D′)). Assume $x \leq y$ so that $\sup(x, a) \leq \sup(y, a)$. Since $\sup(x, a) = \omega$ we deduce $\sup(y, c) = \omega$. As a consequence, c is the standard completion of y, hence $b = c = \inf(a, b)$. This shows $b \leq a$.

```
Lemma standard_completion_monotone r x y: (* 25 *)
  boolean_lattice r -> gle r x y ->
  gle r (standard_completion r y) (standard_completion r x).
```

We show that $x \mapsto x^*$ is an isomorphism. This is obvious since it is increasing (for the order on E and its reverse) and involutive.

```
Lemma Exercise1_17b r: boolean_lattice r -> (* 16 *)
  order_isomorphism (BL (standard_completion r) (substrate r)(substrate r))
  r (opposite_order r).
```

We know that $\mathfrak{P}(A)$ is a lattice, where inf and sup are intersection and union. It is a boolean lattice, where x^* is the complement of x in A. We first show that $\emptyset$ is the least element, and A the greatest. We show distributivity via (T′).

```
Lemma Exercise1_17c A: (* 61 *)
  (boolean_lattice (inclusion_order A) &
    (forall x, inc x (powerset A) ->
      standard_completion (inclusion_order A) x = complement A x)).
```

(c) If E *is a* complete *Boolean lattice (Exercise 11), show that for each family* (x_λ) *of elements of* E *and each* $y \in$ E *we have*

$$\inf(y, \sup_\lambda(x_\lambda)) = \sup_\lambda(\inf(y, x_\lambda)).$$

(Reduce to the case $y = \alpha$*, and use the fact that if* $\inf(z, x_\lambda) = \alpha$ *for every index* λ*, then* $z^* \geq x_\lambda$ *for every* λ*).*

The hint "Reduce to the case $y = \alpha$" is strange.

We start with four formulas: $\inf(y, \sup(y^*, x)) = \inf(y, x)$, $\sup(y, \inf(y^*, x)) = \sup(y, x)$, and the same with y and y^* exchanged.

Consider a family x_λ, let $u = \sup_\lambda(\inf(y, x_\lambda))$ and $v = \inf(y, \sup_\lambda(x_\lambda)$. We have to show $v = u$. Let X be the range of the family x_λ and Y the set of all $\inf(y, x_\lambda)$. Then $u = \sup Y$ and $v = \inf(y, \sup X)$. Since E is a complete lattice, the two quantities $\sup X$ and $\sup Y$ are defined and are in E. The relation $u \leq v$ is clear, and $v \leq y$ is obvious. It follows $u \leq y$.

Define $z = \sup(y^*, u)$. We have $\inf(y, z) = \inf(y, \sup(y^*, u)) = \inf(y, u) = u$. Similarly, if $z' = \sup(y^*, \sup X)$ we have $\inf(y, z') = v$. It suffices to show $z = z'$. The relation $z \leq z'$ is clear.

For each λ, $\inf(y, x_\lambda) \in Y$, thus $\inf(y, x_\lambda) \leq \sup Y$. Taking the supremum with y^* and simplifying gives $\sup(y^*, x_\lambda) \leq z$, thus $x_\lambda \leq z$ and $z' \leq z$.

```
Lemma Exercise1_17d r x y: boolean_lattice r -> (* 12 *)
  inc x (substrate r) ->  inc y (substrate r) ->
  let ys := (standard_completion r y) in
  (inf r y (sup r ys x) = inf r y x &
  sup r y (inf r ys x) = sup r y x &
  inf r ys (sup r y x) = inf r ys x &
  sup r ys  (inf r y x) = sup r ys x).
Lemma Exercise1_17e r x y: (* 42 *)
  boolean_lattice r -> complete_lattice r ->
  inc y (substrate r) -> sub x (substrate r)->
  inf r y (supremum r x)
   = supremum r (fun_image x (fun z => inf r y z)).
```

¶ 18. * *Let* A *be a set with at least three elements, let $\mathscr{P}$ be the set of all partitions of* A*, ordered by the relation "$\mathfrak{w}$ is finer than $\mathfrak{w}'$" between $\mathfrak{w}$ and $\mathfrak{w}'$ (no 1, Example 4). Show that $\mathscr{P}$ is a complete lattice (Exercise 11), is not distributive (Exercise 17), but is relatively complemented (To prove the last assertion, well-order the sets belonging to a partition.)*$_*$

In what follows $a \leq b$ will denote the relation "a is coarser than b"; this is the opposite relation of "a is finer than b". In all three statements of the Exercise, $\leq$ can be replaced by $\geq$.

By Exercise 1.11b, the set $(\mathscr{P}, \geq)$ is a complete lattice if every family has a supremum for $\geq$, or an infimum for $\leq$. Sketch of the proof. Given a partition A, we can consider the equivalence $\sim_A$ associated to A ($x \sim_A y$ if and only if x and y are in some z with $z \in A$). We have $A \leq B$ if $x \sim_A y \implies x \sim_B y$. Then $\sup(A, B)$ is the partition associated to "$x \sim_A y$ and $x \sim_B y$", and $\inf(A, B)$ is the set of connected components of the relation "$x \sim_A y$ or $x \sim_B y$" (see Part I, Exercise 6.10). From this, one could characterize the infimum of a family of partitions. In what follows, we shall not consider the associated equivalence.

We add an extension to Exercise 1.11; the set $(\mathscr{P}, \geq)$ is a complete lattice if every family has a infimum for $\geq$, or a supremum for $\leq$. It is a complete lattice if $(\mathscr{P}, \leq)$ is a complete lattice.

```
Lemma exercise1_11h r: order r ->   (* 12 *)
  (forall X, sub X (substrate r) -> has_infimum r X) ->
  complete_lattice r.
```

```
Lemma exercise1_11i r: (* 5 *)
  complete_lattice r -> complete_lattice (opposite_order r).
```

Consider the set of all $A \cap B$ for $A \in \varpi$ and $B \in \varpi'$. We have shown (in Part I of this report) that this is the least upper bound (for the "coarser" ordering for coverings). When we remove the empty set, we get the least upper bound of two partitions. We first recall the property of the covering intersection. We show here that this defines the supremum of two elements.

```
(*
  Lemma intersection_covering2_pr: forall x y z,
    inc z (intersection_covering2 x y) =
    exists a, exists b, inc a x & inc b y & intersection2 a b = z.
*)

Definition intersection_partition2 u v :=
  Zo (intersection_covering2 u v) (fun z => nonempty z).

Lemma disjoint_pr1 a b: (* 2 *)
  (forall x, inc x a -> inc x b -> a = b) ->
  (a=b \/ disjoint a b).
Lemma intersection_is_partition2 u v x: (* 25 *)
  partition u x -> partition v  x ->
  partition (intersection_partition2 u v) x.
Lemma intersection_p2_comm:  u v:  (* 3 *)
  (intersection_partition2 u v) = (intersection_partition2 v u).
Lemma intersection_is_sup2_a u v x: (* 4 *)
  partition u x -> partition v  x ->
  gle (coarser x) u (intersection_partition2 u v).
Lemma intersection_is_sup2 u v x: (* 12 *)
  partition u x -> partition v  x ->
  least_upper_bound (coarser x)(doubleton u v)(intersection_partition2 u v).
Lemma intersection_sup2 E u v:
   partition u E -> partition v E ->
    sup (coarser E) u v = (intersection_partition2 u v).
Lemma intersection_is_sup2_b E u v: (* 8 *)
   partition u E -> partition v E ->
    sup (coarser E) u v = (intersection_partition2 u v).
```

Consider now a nonempty family F_i of partitions of X. For any element $x = (x_i)_i$ of $\prod F_i$, we can consider the subset $\bar{x} = \bigcap x_i$ of E. The set of all $\bar{x}$ (minus the emptyset) is a partition of E. It is the least upper bound of the family. If follows that any non-empty set of partitions has a least upper bound. It follows that E is a complete lattice (it has a least element, which is empty if E is empty, {E} otherwise).

```
Definition intersection_partition f :=
  complement (fun_image (productb f) intersectionb) (singleton emptyset).

Lemma intersection_is_partition f x: (* 34 *)
  fgraph f -> (forall u, inc u (domain f) -> partition (V u f) x) ->
  nonempty (domain f) ->
  partition (intersection_partition f) x.
Lemma intersection_is_sup_a f x y: (* 6 *)
  fgraph f -> (forall u, inc u (domain f) -> partition (V u f) x) ->
  inc y (range f) ->
  gle (coarser x) y (intersection_partition f).
```

```
Lemma intersection_is_sup f x: (* 24 *)
  fgraph f -> (forall u, inc u (domain f) -> partition (V u f) x) ->
  nonempty (domain f) ->
  least_upper_bound (coarser x) (range f) (intersection_partition f).
Lemma exercise1_18a E:  complete_lattice (coarser E). (* 34 *)
```

Let's now show that $\mathscr{P}$ is not distributive. It is clear that we can use the coarser or finer order indifferently. If E is empty, or is a singleton, there is a unique partition. If E has two elements, there are two partitions, the least and the greatest ones. In these cases, the set is distributive.

Assume that E has at least three elements, x, y and z, and let $T = \{x, y, z\}$ and $F = E - T$. If U is a set, we denote by $\bar{U}$ the quantity $U \cup \{F\} - \{\emptyset\}$. The empty set is not an element of $\bar{U}$, and if the empty set is not an element of U we have $\bar{U} = U$ if F is empty and $\bar{U} = U \cup \{F\}$ otherwise. If U is a partition of T, then $\bar{U}$ is a partition of E. Let p_x be the partition of T formed of $\{x\}$ and $\{y, z\}$ and $P_x = \bar{p}_x$. Define similarly P_y and P_z. Let $\alpha = \bar{a}$ and $\omega = \bar{o}$ where a is the greatest partition of T and o the least (a is a set containing three singletons and $\emptyset = \{T\}$). This gives five partitions of E. We show that $\mathscr{P}$ is not distributive by considering the application of (D') to P_x, P_y and P_z. It says

$$\sup(P_x, \inf(P_y, P_z)) = \inf(\sup(P_x, P_y), \sup(P_x, P_z)).$$

Since o is the greatest partition of T we get that that $\omega \leq \bar{P}$ whenever P is a partition of T. Assume now a is a partition such that $a \leq P_y$ and $a \leq P_z$; then a contains two sets b and b' such that $\{x, y\} \subset b$ and $\{x, z\} \subset b'$. These sets are equal, since they cannot be disjoint. We deduce $a \leq \omega$ so that $\inf(P_y, P_z) = \omega$. Since $\omega \leq P_x$ we get

$$P_x = \inf(\sup(P_x, P_y), \sup(P_x, P_z)).$$

Assume $\{a, b\} = \{x, y\}$. Let p_t be the set containing $\{a\}$ and $\{b, z\}$, and let $P_t = \bar{p}_t$. Then P_t is one of P_y and P_z, thus is a partition. Since P_x and P_t have three elements each the intersection has nine elements, some being empty. A tedious study shows that the intersection is α. We deduce $\sup(P_x, P_y) = \sup(P_x, P_z) = \alpha$ hence $P_x = \alpha$. This is absurd.

```
Definition big_set3 E :=
 (exists x, exists y, exists z, inc x E & inc y E & inc z E & x <> y & x <> z
   & y <> z).

Lemma exercise1_18b E: big_set3 E -> (* 204 *)
  ~ (distributive_lattice2 (coarser E)).
```

Let's characterize the finest partition (it is α such that for all x, x is coarser than α, or α is finer than x).

```
Lemma exercise1_18c E: (* 9 *)
  largest_partition E =  the_greatest_element (coarser E).
```

Let's show that the set is relatively complemented for the "finer" ordering. If $\leq$ denotes "coarser" we have to show: for all X and Y such that $Y \leq X$, there exists X' such that $\inf(X, X') = Y$ and $\sup(X, X') = \alpha$.

Bourbaki hints to well-order the elements; this is not needed; however, we use the axiom of choice that asserts that for any $a \in X$ there is an element e_a such that $e_a \in a$ (recall that a

in non-empty). Define P_v to be the set of all e_a for $a \in X$ and $a \subset v$. We consider the set Z formed of all P_v for $v \in y$ and all singletons $S_b = \{b\}$, with $b \in E$, that are not of the form e_a for $a \in X$.

Obviously, $S_b \subset E$ and $P_v \subset E$. Consider $x \in E$. There is a such that $x \in a \in X$, and $b \in Y$ such that $a \subset b$. Assume $x = e_c$ for some $c \in X$. We deduce $x \in c \in X$, hence $a = c$ and $x \in P_b$. Otherwise $x \in S_x$, and $S_x \in Z$. Thus $\bigcup Z = E$.

Assume $v \in Y$. There is x such that $x \in v \in Y$, and some u such that $x \in u \in X$. There is also some $w \in Y$ such that $u \subset w$. This gives $x \in w \in Y$ hence $v = w$, thus $u \subset v$. We have $e_u \in P_v$, so that P_v is non-empty.

The elements of Z are mutually disjoint. This is obvious if they have the form S_b, since S_b is a singleton. By construction if $S_b \in Z$ then b is in no P_v. Assume $x \in P_u \cap P_v$. Then $x = e_a = e_b$, with $a \subset u$ and $b \subset v$. The two sets u and v contain x, thus cannot be disjoint, thus are equal and $P_u = P_v$.

We have $Y \leq Z$. Given a singleton $S_b \in Z$, there is u such that $b \in u \subset Y$, thus $S_b \subset u$. On the other hand, $P_v \subset v$. Since $Y \leq X$, we deduce $Y \leq W$, where $W = \inf(X, Z)$.

Conversely, we have $W \leq Y$. Consider an element a of Y. We have $P_a \in Z$. Since $W \subset Z$ there exists $b \in W$ such that $P_a \subset b$. Consider $t \in a$. Since $t \in E$, there is c such that $t \in c \in X$, and d_1 such that $c \subset d_1 \in Y$. It follows $d_1 = a$, $c \subset a$. We deduce $e_c \in P_a$, thus $e_c \in b$. Since $c \in X$ and $W \leq X$ there exists $d \in W$ such that $c \subset d$. Obviously, $e_c \in d$, so that b and d are not disjoint. It follows $b = d$, thus $c \subset b$, and $t \in b$.

Let's show $\sup(X, Z) = \alpha$. We have to show that the intersection of X and Z contains all singletons and nothing else. Consider the intersection of an element a of X and an element b of Z. The result is obvious if b is a singleton. Assume $b = P_v$. If $t \in a \cap b$ then $t = e_c$ for some $c \in X$ such that $c \subset v$. Since t is in a and c, these two sets cannot be disjoint, thus are equal, and $t = e_a$. Thus $a \cap b$ is empty or is a singleton. Conversely, consider an element x in E. There is a such that $x \in a \in X$, and b such that $a \subset b \in Y$. If $x = e_a$, the previous argument says that $a \cap P_b = \{x\}$. Otherwise we know $S_x \in Z$, $x \cap S_x = \{x\}$.

```
Lemma exercise1_18d E x y: (* 146 *)
  let r := coarser E in
    gle r y x -> exists x',
      inc x' (substrate r) & inf r x x' = y & sup r x x' = largest_partition E.
```

---

19. *An ordered set* E *is said to be* without gaps *if it contains two distinct comparable elements and if, for each pair of elements x, y such that $x < y$, the open interval $]x, y[$ is not empty. Show that the ordinal sum $\sum_{\iota \in I} E_\iota$ (Exercise 3) is without gaps if and only if the following conditions are satisfied:*

(I) Either I *contains two distinct comparable elements, or else there exists $\iota \in I$ such that E_ι contains two distinct comparable elements.*

(II) Each E_ι which contains at least two distinct comparable elements is without gaps.

(III) If α, β are two elements of I *such that $\alpha < \beta$ and if the interval $]\alpha, \beta[$ in* I *is empty, then either E_α has no maximal element or else E_β has no minimal element.*

In particular, every ordinal sum $\sum_{\iota \in I} E_\iota$ of sets without gaps is itself without gaps, provided that no E_ι has a maximal element (or provided that no E_ι has a minimal element). If I *is*

without gaps, and if each E_ι *is either without gaps or contains no two distinct comparable elements, then* $\sum_{\iota\in I} E_\iota$ *is without gaps.*

Assume the sum without gaps. Condition (I) says that the sum has two distinct comparable elements. Condition (II) is immediate. Consider now condition (III). Consider a maximal element x of E_α, a minimal element y of E_β, where $\alpha < \beta$. We have $x < y$ (considered as elements of the sum); hence there is z such that $x < y < z$. It index cannot be α not β, hence there is a γ such that $\alpha < \gamma < \beta$.

Converse. We use part (I) to show that there are at least two comparable elements in the sum. Consider two elements x and y such that $x < y$. Assume $x \in E_\alpha$ and $y \in E_\beta$. There are two cases to consider: if $\alpha < \beta$, we conclude by (III), otherwise, $\alpha = \beta$ and $x < y$ in E_α and we conclude by (II).

```
Definition without_gaps r :=
  order r & (exists x, exists y, glt r x y) &
  (forall x y, glt r x y -> exists z, glt r x z & glt r z y).

Section Exercise1_19.
Variables (r g: Set).
Hypotheses  (ax:orsum_ax r g) (ax2: orsum_ax2 g).
Hypothesis nesr: nonempty (substrate r).

Lemma Exercise1_19a: (* 125 *)
  (without_gaps (order_sum r  g) <->
    (((exists i, exists j, glt r i j) \/
      (exists i, exists x, exists  y, inc i (substrate r) & glt (V i g) x y))
    & (forall i x y, inc i (substrate r) ->  glt (V i g) x y ->
      without_gaps (V i g))
    & (forall i j, glt r i j ->
      (exists k, glt r i k & glt r k j)
      \/ (forall u, ~ (maximal_element (V i g) u))
      \/ (forall u, ~ (minimal_element (V j g) u))))).
```

Second claim. We assume the index set non-empty. Last claim. The condition "Each E_ι is either without gaps or contains no two distinct comparable elements", is nothing else than (II).

```
Lemma Exercise1_19b: (* 5 *)
  (forall i u, ~ (maximal_element (V i g) u)) ->
  (forall i, inc i (substrate r) -> without_gaps (V i g)) ->
  without_gaps (order_sum r g).
Lemma Exercise1_19c: (* 5 *)
  (forall i u, ~ (minimal_element (V i g) u)) ->
  (forall i, inc i (substrate r) -> without_gaps (V i g)) ->
  without_gaps (order_sum r g).
Lemma Exercise1_19d: (* 6 *)
  without_gaps r ->
    (forall i, inc i (substrate r) ->
      (without_gaps (V i g) \/
        (forall x y, inc x (substrate (V i g)) -> inc y (substrate (V i g))
            -> ~ (glt (V i g) x y))))
    ->  without_gaps (order_sum r g).
End  Exercise1_19.
```

¶ 20. *An ordered set* E *is said to be* scattered *of no ordered subset of* E *is without gaps (Exercise 19). Every subset of a scattered set is scattered.* * *Every well-ordered set of more than one element is scattered.*∗

The first claim is obvious. Every well-ordered set is scattered. In fact, if F is a subset of E, that has at least two elements we can consider the least element x of F and the least element y of $\mathrm{F} - \{x\}$. Then $]x, y[$ is empty.

```
Definition scattered r := order r &
  (forall x, sub x (substrate r) -> ~ (without_gaps (induced_order r x))).

Lemma Exercise1_20a r x: (* 3 *)
  sub x (substrate r) -> scattered r -> scattered (induced_order r x).
Lemma Exercise1_20b : forall r, worder r -> scattered r. (* 22 *)
```

(a) Suppose that E *is scattered. Then if x, y are two elements of* E *such that $x < y$, there exists two elements x', y' of* E *such that $x \leq x' < y' \leq y$, and such that the interval $]x', y'[$ is empty.* * *Give an example of a totally ordered set which satisfies this condition and is not scattered (consider Cantor's triadic set).*∗

We prove that $\exists x, \exists y, \mathrm{P}$ is the negation of $\forall x, \forall y, \neg \mathrm{P}$.

```
Lemma exists_proof2 : (* 2 *)
  forall p :  Set ->Set ->Prop, ~ (forall x y, ~ p x y)
  -> (exists x, exists y, p x y).
```

We just say that $[x, y]$ is not without gaps.

```
Definition Exercise1_20_prop r:=
forall x y, glt r x y ->
  exists x', exists y',
   gle r x x' & glt r x' y'&  gle r y' y &
    (forall z, ~ (glt r x' z & glt r z y')).

Lemma Exercise1_20c  r: (* 19 *)
  scattered r -> Exercise1_20_prop r.
```

Let's consider the set of all functions $\mathbf{N} \to \{a, b\}$ ordered lexicographically, where $a < b$. The Cantor set is defined by $a = 0$ and $b = 2$, and to each function f, one associates the value $\sum_i f(i)/3^i$, considered as a real number. The lexicographic ordering is compatible with the ordering of the real numbers. We do not need real numbers here. We first show that the set is totally ordered.

```
Definition cantor_tri_order:=
  order_product Bnat_order
  (L Bnat (fun _: Set => canonical_doubleton_order)).
Definition cantor_tri_sub:= productb  (L Bnat (fun _: Set => two_points)).

Lemma cantor_tri_order_axioms: (* 4 *)
 orprod_ax Bnat_order
  (L Bnat (fun _: Set => canonical_doubleton_order)).
Lemma cantor_tri_order_total : total_order cantor_tri_order. (* 3 *)
```

We pretend that $f < g$ if and only if there is an index i such that $f(i) = a$, $g(i) = b$, and $f(j) = g(j)$ for $i < j$.

```
Lemma cantor_tri_order_sr1 : (* 3 *)
  prod_of_substrates (L Bnat (fun _ : Set => canonical_doubleton_order)) =
   cantor_tri_sub.
Lemma cantor_tri_order_sr : (* 3 *)
 substrate cantor_tri_order = cantor_tri_sub.
Lemma cantor_tri_order_glt x x': (* 24 *)
  glt cantor_tri_order x x' <->
  (inc x cantor_tri_sub   & inc x' cantor_tri_sub
    & exists j, inc j Bnat &
      (forall i, inc i Bnat -> i <c j -> V i x = V i x')
      & (V j x = TPa & V j x' = TPb)).
```

Consider two functions $f < g$. There is an index i such that $f(i) = a$, $g(i) = b$, and $f(j) = g(j)$ for $i < j$. Define f_a to be the function h such that $h(j) = f(j)$ for $j \leq i$ and $h(j) = a$ for $j > i$. Define g_b similarly. Then $f \leq f_a < g_b \leq g$, and the interval $]f_a, g_b[$ is empty. Thus the Cantor set satisfies the condition. It is however not scattered.

In effect, let G be the set of all sequences such that, for some index i_0, we have $f_i = a$ for $i \geq i_0$. Let F be the complement. We pretend that this subset is without gaps. Let A be the constant function a, and B the function map maps 0 to b and other values to a. These are two elements of F that satisfy B < A.

Assume now $f < g$ and $g \in$ F. There is an index i such that $f(i) = a$ and $g(i) = b$, and $f(i) = g(j)$ for $i < j$. Since $g \in$ F there is an index k such that $i < k$ and $g(j) = b$. Let h be like g with $h(k) = a$. Then $f < g < h$.

```
Lemma  Exercise1_20d: Exercise1_20_prop cantor_tri_order. (* 92 *)
Lemma  Exercise1_20e: ~ (scattered cantor_tri_order).  (* 71 *)
```

(b) An ordinal sum $\sum_{\iota\in I} E_\iota$ *(where neither* I *nor any* E_ι *is empty) is scattered if and only if* I *and each* E_ι *is scattered. (Note that* E *contains a subset isomorphic to* I *and that every subset* F *of* E *is the ordinal sum of those sets* $F \cap E_\iota$ *which are non-empty; finally use Exercise 19.)*

The condition "I nonempty" is not needed here. The condition $E_\iota \neq \emptyset$ was implicit in Exercise 19. We assume $e_i \in E_i$, so that $f_i = (e_i, i)$ is in the sum $S = \sum E_i$. Assume S scattered, and let's show that I is scattered. We consider a subset X of I, and W the set of all e_i for $i \in X$. We have to show that, if X is without gaps, then W is without gaps. Since there are elements $i < j$ in X, there are elements $f_i < f_j$ in W. Assume $x < y$ in W, say $x = f_a$ and $y = f_b$. We have $a < b$ so that there is c such that $a < c < d$. Then $x < f_c < y$ in W. Consider now an index i. For $x \in E_i$, the element $f_x = (x, i)$ is in S, and the same argument as above show that E_i is scattered.

Converse. Let S' be a subset of S; Let I' be the set of $i \in I$ such that $S' \cap E_i$ is nonempty, ordered by the ordering of I. Let E'_i be the set of $x \in E_i$ such that $(x, i) \in S'$. It is non-empty if $i \in I'$. We order it by the ordering induced from E_i. Consider now the ordinal sum $\sum_{I'} E'_i$. Its substrate is S', and its ordering is that of S. Assume S' without gaps, so that conditions (I), (II) and (III) of exercise 19 hold. If E'_i has two comparable elements, it is without gaps, according to condition (II); this contradicts the fact that E_i is scattered. This implies that each element of E'_i is maximal and minimal, so that condition (III) becomes: if $\alpha < \beta$ in I', then the interval $]\alpha, \beta[$ is non-empty. Moreover condition (I) says that I' has at least two elements. Thus I' is without gaps and I is not scattered.

```
Lemma  Exercise1_20f r g: (* 156 *)
  orsum_ax r g ->  orsum_ax2  g ->
```

```
  (forall i, inc i (domain g) -> nonempty (substrate (V i g))) ->
  (scattered (order_sum r g) <->
    (scattered r & forall i, inc i (domain g) -> scattered (V i g))).
```

---

21. *Let* E *be a non-empty totally ordered set, and let* S$\{x, y\}$ *be the relation "the closed interval with endpoints x, y is scattered" (Exercise 20). Show that* S *is an equivalence relation which is weakly compatible (Exercise 2) in x and y with the order relation on* E*, that the equivalence classes with respect to* S *are scattered sets, and that the quotient ordered set* E/S *is either without gaps or else consists of a single element. Deduce that* E *is isomorphic to an ordinal sum of scattered sets whose index set is either without gaps or else consists of a single element.*

There is no need to assume E non-empty.

We start with a complement to Exercise 1.2. Assume that both conditions (C) and (C') are satisfied and that $\leq$ is a total order on E. Then the quotient order E/S is totally ordered, and $X \leq Y$ in the quotient is equivalent to $x \leq y$ whenever $x \in X$ and $y \in Y$, and we may take the representatives of X and Y for x and y.

```
Lemma Exercise1_2g r s: (* 22 *)
  weak_order_compatibility r s->
  quotient_order_axiom r s -> total_order r ->
  let r' := (quotient_order r s) in
    forall x y, gle r' x y <-> (inc x (quotient s) & inc y (quotient s)
      & gle r (rep x) (rep y)).
Lemma Exercise1_2h r s: (* 9*)
  weak_order_compatibility r s->
  quotient_order_axiom r s -> total_order r ->
  total_order (quotient_order r s).
```

The condition "without gaps", simplified to WG is the conjunction of three conditions; $(E, \leq)$ is an ordered set, WG1 that says that there are x and y such that $x < y$ and WG2 that says that if $x < y$, then there is z such that $x < z < y$. The relation S is either $x \leq y$ and $[x, y]$ is scattered, or the same with x and y exchanged. Note that V scattered means that, if $U \subset V$, it is not without gaps for the ordering of E (which is the ordering of V).

In a totally ordered set, condition WG1 is: "there are two distinct elements in U", and its negation as "there is at most one element in U". We can now say: U is not without gaps if and only if either U is a small set, or it contains $a < b$ such that $]a, b[$ is empty.

```
Definition scattered_rel r x y :=
  (gle r  x y &  scattered (induced_order r (interval_cc r x y)))
  \/ (gle r  y x &  scattered (induced_order r (interval_cc r y x))).

Definition scattered_equiv r := graph_on (scattered_rel r) (substrate r).

Lemma Exercise1_21a r v: order r -> (* 3 *)
  sub v (substrate r) ->
  (scattered (induced_order r v) <->
    (forall u, sub u v ->  ~ without_gaps (induced_order r u))).
Lemma Exercise1_21b r u: total_order r -> (* 5 *)
```

```
  sub u  (substrate r) ->
  ((exists x, exists y, glt (induced_order r u) x y) <->
   (exists x, exists y, inc x u & inc y u & x<> y).
Lemma Exercise1_21c r u: total_order r -> (* 3*)
  sub u  (substrate r) ->
  ((forall a b, inc a u -> inc b u -> a = b) <->
    ~ (exists x, exists y, glt (induced_order r u) x y)).
Lemma Exercise1_21d r u: total_order r -> (* 21 *)
  sub u (substrate r) ->
  ((~ without_gaps (induced_order r u)) <->
   ((forall a b, inc a u -> inc b u -> a = b)
      \/ (exists a, exists b, inc a u & inc b u & glt r a b &
        (forall z, inc z u -> gle r z a \/ gle r b z)))).
```

Assume that U is without gaps, and consider two elements a, b of U such that $a < b$. The intersection $U \cap [a,b]$ is without gaps.

```
Lemma Exercise1_21e r u a b: total_order r -> (* 19 *)
  let v:= intersection2 u (interval_cc r a b) in
  sub u (substrate r) -> without_gaps (induced_order r u) ->
  (exists x, exists y,  inc x v & inc y v & glt r x y) ->
  without_gaps (induced_order r v).
Lemma Exercise1_21f r a b u: total_order r -> (* 3 *)
  sub u (substrate r) ->
  inc a u -> inc b u -> glt r a b -> without_gaps (induced_order r u) ->
  without_gaps (induced_order r (intersection2 u (interval_cc r a b))).
```

The union of two scattered intervals $[x,y]$ and $[y,z]$ is scattered. Proof by contradiction. Let u be a subset without gaps of the union, consider the intersection u_1 and u_2 with the two intervals. They are not without gaps.

Let's show that S is an equivalence. Symmetry is true by definition; reflexivity is a consequence of the fact that singletons are scattered. Transitivity is a consequence of the previous result if $x \leq y \leq z$, and if $x \leq z \leq y$ it is a consequence of 20a.

```
Lemma Exercise1_21g r x y z: total_order r -> (* 39 *)
  gle r  x y ->  gle r y z->
  scattered (induced_order r (interval_cc r x y)) ->
  scattered (induced_order r (interval_cc r y z)) ->
  scattered (induced_order r (interval_cc r x z)).
Lemma Exercise1_21h r: total_order r -> (* 41 *)
  equivalence_re (scattered_rel r) (substrate r).
Lemma Exercise1_21i r: total_order r -> (* 6 *)
  is_equivalence (scattered_equiv r).
Lemma Exercise1_21j r: total_order r -> (* 2 *)
  substrate (scattered_equiv r) = substrate r.
```

We simplify a bit the definition of being related by this relation. We first consider a mix of 21a and 21d.

```
Definition scattered_aux r x y :=
  gle r x y &
  (forall u, sub u (interval_cc r x y) ->
    ((forall a b, inc a u -> inc b u -> a = b)
```

```
      \/ (exists a, exists b, inc a u & inc b u & glt r a b &
        (forall z, inc z u -> gle r z a \/ gle r b z))).

Lemma Exercise1_21k r x y: total_order r -> (* 10 *)
  gle r  x y ->
  (scattered (induced_order r (interval_cc r x y)) <->
  (forall u, sub u (interval_cc r x y) ->
    ((forall a b, inc a u -> inc b u -> a = b)
      \/ (exists a, exists b, inc a u & inc b u & glt r a b &
        (forall z, inc z u -> gle r z a \/ gle r b z))))).

Lemma Exercise1_21l r x y: total_order r -> (* 7 *)
  (related (scattered_equiv r) x y <->
    (scattered_aux r x y \/ scattered_aux r y x)).
```

Let's show weak compatibility. Assume that x and x' and related, and $x \leq y$. We want to find y such that $x' \leq y'$ and y is related to y'. If $x' \leq y$ we may choose $y = y'$. Otherwise we have $x \leq y \leq x'$, and we chose $y = x'$. The interval $[y, x']$ is scattered, as a subset of a scattered set.

```
Lemma Exercise1_21m r: total_order r -> (* 15 *)
  weak_order_compatibility r (scattered_equiv r).
```

Let's show that equivalence classes are scattered. Let X be a subset of an equivalence class, assumed without gaps. It has two elements a and b such that $a < b$; and for every $c < d$, there is e such that $c < e < d$. These elements a and b are related so that the interval $[a, b]$ is scattered. Let Y be the intersection of X and the interval $[a, b]$. It is not without gaps, but has two comparable elements, namely a and b, hence there exists c, d such that $]c, d[\cap$Y is empty. But there is an $e \in]c, d[\cap$X. This element is clearly in $[a, b]$ hence in Y, absurd.

```
Lemma Exercise1_21n r x: (* 37 *)
  total_order r -> inc x (substrate r) ->
  scattered (induced_order r (class (scattered_equiv r) x)).
```

Since the equivalence S is weakly compatible with the order, it induces a preorder on the quotient. We show here that it is an order (cf. Exercise 1.2).

```
Lemma Exercise1_21o r: total_order r -> (* 10 *)
  quotient_order_axiom r (scattered_equiv r).
Lemma Exercise1_21p r: total_order r ->  (* 3 *)
  order (quotient_order r (scattered_equiv r)).
```

Another extension to Exercise 1.2.

```
Lemma Exercise1_2j r s: (* 12 *)
  weak_order_compatibility r s->
  quotient_order_axiom r s -> total_order r ->
  let r' := (quotient_order r s) in
    forall x y, inc x (substrate r ) -> inc y (substrate r) ->
      ((gle r x y -> gle r' (class s x) (class s y))
      & (glt r' (class s x) (class s y)) ->glt r x y).
```

Let's show that the quotient ordered set is either without gaps or a small set (empty or containing a single element). All we need to show is that if X < Y in the quotient, there is Z

such that X < Z < Y. Let x and y be the representatives of X and Y. They are not related by the equivalence S, hence $[x, y]$ is not scattered. We use contradiction, assume there is a set U with at least two elements, such that no interval is empty. Let $a \in U$. We have $x \leq a \leq y$, so that classes are in the same order. Let A be the class of a, so that $X \leq A \leq Y$. By assumption, one $\leq$ is equality. This implies $a \in X$ or $a \in Y$.

Let X_1 and X_2 be the intersections of U with X and Y. These sets have at most one element, for otherwise, we could find an empty interval $]a, b[$. This interval has a point c in U, which is in X or in Y. If we consider X_1, $c \leq b$, $b \in X$ and $c \in Y$ would imply $Y \leq X$, absurd.

Now U has three points, and is the union of two sets with at most one point, absurd.

```
Lemma Exercise1_21q r: total_order r -> (* 100 *)
  let r' := quotient_order r  (scattered_equiv r) in
    small_set (substrate r') \/ without_gaps r'.
```

We prove now the last part of the exercise. We assume that S is an equivalence on E that satisfies the assumptions of the previous lemma. Let I be the quotient set E/S. Consider the identity function on I, and write it $i \mapsto E_i$. This gives an ordinal sum $\sum_{i\in I} E_i$. The mapping $x \mapsto (x, \bar{x})$ is an order isomorphism, where $\bar{x}$ denotes the class of x for S. The ordinal sum is a disjoint union, and there is a natural equivalence S′; two elements x and y of E are related by S′ if and only if $f(x)$ and $f(y)$ are related by S′. Let Q be the quotient $\sum E_i/S'$; exercise 1-3 shows that this quotient is isomorphic to I (ordered by the quotient orders wrt S′ and S).

```
Lemma Exercise1_2i r s (* 64 *)
  (q := quotient s)
  (r' := quotient_order r s)
  (f' := identity_g q)
  (g' := L q (fun z => induced_order r z))
  (du := disjoint_union f')
  (f :=  BL (fun x => J x (class s x)) (substrate r) du):
  weak_order_compatibility r s->
  quotient_order_axiom r s -> total_order r ->
  (orsum_ax r' g'
    &(forall i, inc i (domain g') -> nonempty (substrate (V i g')))
    & substrate (order_sum r'  g') = du
    & (forall x y, inc x (substrate r) -> inc y (substrate r) ->
      (related s x y <->
        related (equivalence_associated (second_proj du)) (W x f) (W y f))
    & bijection f
    & order_isomorphism f r (order_sum r'  g')).
Lemma Exercise1_21r r: total_order r -> (* 18 *)
  exists r', exists g',
  (orsum_ax r' g'
    &(forall i, inc i (domain g') -> nonempty (substrate (V i g')))
    & r \Is (order_sum r' g')
    & (small_set (substrate r') \/ without_gaps r')
    & (forall i, inc i (domain g') -> scattered (V i g'))).
```

¶ 22. *(a) Let* E *be an ordered set; A subset* U *of* E *is said to be* open *if for each* $x \in U$, U *contains the interval* $[x, \rightarrow [$. *An open set* U *is said to be* regular *if there exists no open set*

$V \supset U$, *distinct from* U *such that* U *is cofinal in* V. *Show that every open set* U *is cofinal in exactly one regular open set* $\bar{U}$. *The mapping* $U \to \bar{U}$ *is increasing. If* U, V *are two open sets such that* $U \cap V = \emptyset$, *then also* $\bar{U} \cap \bar{V} = \emptyset$.

We start by showing the the union and intersection of open sets is open (note that the empty intersection is empty, hence open).

```
Definition open_o r u:=
  sub u (substrate r) & forall x y, inc x u -> gle r x y -> inc y u.
Definition open_r r u:=
  open_o r u & forall v, open_o r v -> sub u v ->
    cofinal_set (induced_order r v) u
    -> u = v.

Definition bar1_22 r u :=
  union (Zo (powerset(substrate r))
    (fun z => open_o r z &  cofinal_set (induced_order r z) u)).

Definition set_of_reg_open r := Zo (powerset (substrate r))
  (fun z => open_r r z).
Definition reg_open_order r :=
  inclusion_suborder (set_of_reg_open r).

Lemma inf_pr2 r x y z: (* 3 *)
  orcer r -> gle r z x  -> gle r z y ->
  (forall t, gle r t x -> gle r t y -> gle r t z) ->
  inf r x y = z.


Section Exercise1_22.
Variable r:Set.
Hypothesis or: order r.

Lemma Exercise1_22a u1 u2: (* 3 *)
  open_o r u1 -> open_o r u2 -> open_o r (union2 u1 u2).
Lemma Exercise1_22b u:  (* 12 *)
  (forall x, inc x u -> open_o r x) ->
  open_o r (intersection u).
Lemma Exercise1_22c u: (* 4 *)
  (forall x, inc x u -> open_o r x) ->
  open_o r (union u).
```

We show uniqueness of $\bar{U}$. If U is cofinal in U_1 and U_2, if $U_3 = U_1 \cup U_2$; regularity of U_1 shows $U_3 = U_1$. Similarly $U_3 = U_2$.

```
Lemma cofinal_induced v u:
  sub u (substrate r) ->
  ( cofinal_set (induced_order r u) v <->
    (sub v u & (forall x, inc x u -> exists y, inc y v & gle r x y))).
Lemma Exercise1_22d x u1 u2: (* 19 *)
  open_o r x -> open_r r u1 -> open_r r u2 ->
  sub x u1 -> sub x u2 ->
  cofinal_set (induced_order r u1) x -> cofinal_set (induced_order r u2) x
  -> u1 = u2.
```

Let $\bar{U}$ be the union of all open sets V containing U in which U is cofinal (since V = U is possible, we have $U \subset \bar{U}$). U is cofinal in $\bar{U}$.

```
Lemma Exercise1_22e u: (* 3 *)
  open_o r u ->  sub u (bar1_22 r u).
Lemma Exercise1_22f u: (* 2 *)
  open_o r u ->  sub (bar1_22 r u) (substrate r).
Lemma Exercise1_22g u: (* 4 *)
  open_o r u ->
  cofinal_set (induced_order r (bar1_22 r u)) u.
```

Consider $x \in E$, such that whenever $x \leq y$, y is bounded by an element of U. Then $x \in \bar{U}$ (consider $U \cup [x, \rightarrow [$). This criterion will be used a lot.

Assume $\bar{U}$ cofinal in V. If $x \in V$ and $x \leq y$, then $y \in V$ and is bounded by $z \in \bar{U}$, which is bounded by an element of U, hence $x \in \bar{U}$, and $\bar{U}$ is a regular open set.

```
Exercise1_22h u x: (* 16 *)
  open_o r u -> inc x (substrate r) ->
  (forall y, gle r x y -> exists z, inc z u &  gle r y z)
  -> inc x (bar1_22 r u).
Lemma Exercise1_22i u: (* 9 *)
  open_o r u ->
  open_r r (bar1_22 r u).
```

Assume $U \subset V$, and $x \in \bar{U}$, and $x \leq y$. Then $y \in \bar{U}$, hence is bounded by an element of U (thus V), and x is in $\bar{V}$. Hence $\bar{U} \subset \bar{V}$.

Assume that U and V are open sets. Consider an element $a \in \bar{U} \cap \bar{V}$, say $a \in K_1$ and $a \in K_2$. There is $x \in U$, with $a \leq x$. Since K_2 is open, we have $x \in K_2$. Thus, there is $y \in V$ such that $x \leq y$. Since U is open, we have $y \in U \cap V$. Thus, if $U \cap V = \emptyset$ then $\bar{U} \cap \bar{V} = \emptyset$.

```
Lemma Exercise1_22j u v: (* 6 *)
  open_o r u -> open_o r v -> sub u v ->
  sub (bar1_22 r u)  (bar1_22 r v).
Lemma Exercise1_22k  u v: (* 9 *)
  open_o r u -> open_o r v ->  intersection2 u v = emptyset ->
  intersection2 (bar1_22 r u)  (bar1_22 r v) = emptyset.
```

(b) Show that the set R(E) *of regular open sets of* E*, ordered by inclusion, is a complete Boolean lattice (Exercise 17). For* R(E) *to consist of two elements, it is necessary and sufficient that* E *should be non-empty and* right directed*.*

Let's start with the last point. The set R(E) has a least and a greatest element, namely $\emptyset$ and E. We have $\bar{X} = X$ whenever X is regular.

```
Lemma Exercise1_22m:  open_r r emptyset. (* 5 *)
Lemma Exercise1_22n: open_r r (substrate r). (* 2 *)
Lemma Exercise1_22p x:  -> (* 2 *)
  open_r r x -> x = (bar1_22 r x) .
```

Let I_x denote the interval $[x, \rightarrow [$. Assume that our set is not right directed. This means that there exists x and y such that $I_x \cap I_y = \emptyset$. Let $U_x = \bar{I}_x$. We have $U_x \cap U_y = \emptyset$. These two sets are regular and nonempty (they contain x and y). Thus $U_x \neq U_y$.

```
Lemma Exercise1_22o: (* 25 *)
  ~ (right_directed r) ->
  exists a, exists b, open_r r a & open_r r b & nonempty a & nonempty b& a <> b.
```

Assume first that R(E) is a doubleton, say {A,B}, with $A \neq B$. Since $A \subset E$ and $B \subset E$, one of A and B is non-empty, hence U is non-empty. Assume E not right directed. The previous lemma says that there are two non-empty elements in R(E). These elements are hence A and B. Thus both A and B are empty, contradicting $\emptyset \in R(E)$.

Converse. Assume E non-empty and right directed. The two elements E and $\emptyset$ are distinct and in R(E). Let's show that any U in R(E) is empty or E. We consider $x \in U$, $y \in E$. For any z with $y \leq z$ there is an upper bound for x and z; this bound is in U. By 22h, it follows $y \in \bar{U} = U$.

```
Lemma Exercise1_22q: (* 33 *)
  (exists a, exists b, a <> b &  set_of_reg_open r = doubleton a b) <->
  (nonempty (substrate r) & (right_directed r)).
```

Consider now the first part. We consider some properties of the ordering induced by inclusion on R(E). We first show that E and $\emptyset$ are the greatest and least elements. This is a trivial consequence of the fact that these sets are in R(E)

```
Lemma Exercise1_22r u v: (* 4 *)
  gle (reg_open_order r) u v <->
  (open_r r u & open_r r v& sub u v).
Lemma Exercise1_22s1 x: (* 2 *)
  inc x (substrate (reg_open_order r)) <-> open_r r x.
Lemma Exercise1_22s:  (* 4 *)
  greatest_element (reg_open_order r) (substrate r).
Lemma Exercise1_22t:  (* 4 *)
  least_element (reg_open_order r) (emptyset).
```

The function $U \mapsto \bar{U}$ is a closure. As a consequence, the bar of the union of a family of elements of R(E) is the least upper bound. This shows that the set is a complete lattice, thus is a lattice. The sup and inf of two elements are $\overline{U \cup V}$ and $\overline{U \cap V}$.

```
Lemma Exercise1_22u u v: (* 16 *)
  open_r r u -> open_r r v ->
  inf (reg_open_order r) u v = bar1_22 r (intersection2 u v).
Lemma Exercise1_22v X: (* 16 *)
  sub X (substrate (reg_open_order r)) ->
  least_upper_bound (reg_open_order r) X (bar1_22 r (union X)).
Lemma Exercise1_22w u v:  (* 5 *)
  open_r r u -> open_r r v ->
  sup (reg_open_order r) u v = bar1_22 r (union2 u v).
Lemma Exercise1_22x: (* 4 *)
  complete_lattice (reg_open_order r).
Lemma Exercise1_22y:  (* 3 *)
  lattice (reg_open_order r).
```

Assume $X \subset Y$, where X and Y are regular open sets. Let Z be the set of all elements of Y not bounded by an element of X. This is an open set. Every element of Y is bounded by an element of X or Z. This implies that Y is a subset of $\bar{Z} \cup X$, hence $Y = \sup(\bar{Z}, X)$. Obviously, $Z \cap X = \emptyset$, thus $\bar{Z} \cap \bar{X} = \emptyset$. Since $X = \bar{X}$, we get $\inf(\bar{Z}, X) = \emptyset$. As a consequence, our set is relatively complemented.

```
Lemma Exercise1_22z: (* 42 *)
  relatively_complemented (reg_open_order r).
```

We have to show that our set is distributive. Condition (T') reads $\overline{Z \cap \overline{X \cup Y}} \subset \overline{X \cup \overline{Y \cap Z}}$. Write this as $\bar{A} \subset \bar{B}$. By 1.22h, we have to show that for any $x \in \bar{A}$, and $x \leq x'$ there is $y \in B$ such that $x' \leq y$. We have $x' \in \bar{A}$, hence there is $x'' \in A$ such that $x' \leq x''$. Since A has the form $Z \cap \overline{X \cup Y}$ there is $y \in X \cup Y$ such that $x'' \leq y$. We have also $y \in Z$. We have $Z \cap (X \cup Y) \subset X \cup (Y \cap Z)$ (the relation we want to prove, without the bars). Hence $y \in X \cup (Y \cap Z) \subset \bar{A}$.

```
Lemma Exercise1_22A: (* 41 *)
  boolean_lattice (reg_open_order r).
```

(c) If F *is a cofinal subset of* E*, show that the mapping* $U \to U \cap F$ *is an isomorphism of* R(E) *onto* R(F).

Assume U regular in E. Then $U \cap F$ is regular. In fact, assume $U \cap F$ cofinal in an open set V of F. Let $x \in V$. We must show $x \in U \cap F$. We have obviously $x \in F$. Assume $x \leq y$. Since F is cofinal, there is $z \in F$ such that $y \leq z$. Since V is open, we have $z \in V$, so there is $t \in U \cap F$ with $z \leq t$. Thus, there is $t \in U$ such that $y \leq t$. By 1.22h, this says $x \in \bar{U}$, hence $x \in U$.

```
Lemma Exercise1_22B F x: (* 24 *)
  cofinal_set r F -> open_r r x ->
  open_r (induced_order r F) (intersection2 x F).
```

Thus $U \mapsto U \cap F$ maps R(E) into R(F). Assume $U \cap F \subset U' \cap F$. Let $x \in U$, and $x \leq y$. There exists $z \in F$ such that $y \leq z$. We have $z \in U'$, and by 1.22h, this says $x \in \bar{U}'$, hence $U \subset U'$.

```
Lemma Exercise1_22C F U U': (* 11 *)
  cofinal_set r F -> open_r r U -> open_r r U' ->
    sub (intersection2 U F) (intersection2 U' F) -> sub U U'.
End Exercise1_22.
```

Let g be the mapping $X \mapsto X \cap F$. We have shown that if $g(x) \subset g(y)$ then $x \subset y$. The converse is obvious. As a consequence, g is increasing and injective. It is also surjective. We use the same argument as before. If X is a regular open subset of F, Y be the elements $y \in E$ such that there is $x \in X$ with $x \leq y$, then $g(\bar{Y}) = X$.

```
Lemma Exercise1_22D r F: order r -> (* 55 *)
  cofinal_set r F ->
  order_isomorphism (BL (fun z => intersection2 z F) (set_of_reg_open r)
    (set_of_reg_open (induced_order r F)))
  (reg_open_order r)(reg_open_order (induced_order r F)).
```

(d) If E_1, E_2 *are two ordered sets, then every open set in* $E_1 \times E_2$ *is of the form* $U_1 \times U_2$*, where* U_i *is open in* E_i *(*$i = 1,2$*). The set* $R(E_1 \times E_2)$ *is isomorphic to* $R(E_1) \times R(E_2)$.

Let E be a set, ordered by the diagonal order ($x \leq y$ if and only if $x = y$). Then every subset is a regular open set. The product $E \times E$ is ordered by the diagonal order. As a consequence, there are open sets that are not products. The second part of the claim is wrong as well. The argument is the following. Assume that $E_1 = E_2$ has a single element. Then $R(E_1) = R(E_2)$ has two elements. The product $E_3 = E_1 \times E_2$ has a single element, and $R(E_3)$ has two elements. It cannot be isomorphic to $R(E_1 \times E_2)$ that has four elements.

```
Lemma Exercise1_22E r r' X X': (* 8 *)
```

```
  order r -> order r' ->
  open_o r X -> open_o r' X' -> open_o  (order_product2 r r') (product X X').
Lemma Exercise1_22F E X: (* 11 *)
  sub X E -> open_r (identity_g E) X.
Lemma Exercise1_22G E: (* 12 *)
  let r := identity_g E in
    (order_product2 r r = diagonal (product E E)).
```

¶ 23. *Let* E *be an ordered set and let* $R_0(E) = R(E) - \{\emptyset\}$ *(Exercise 22). For each* $x \in E$*, let* $r(x)$ *denote the unique regular open set in which the interval* $[x, \rightarrow [$ *(which is an open set) is cofinal. The mapping* r *so defined is called the canonical mapping of* E *into* $R_0(E)$*. Endow* $R_0(E)$ *with the order relation* opposite *to the relation of inclusion.*

We start with the definition of E_0 and its ordering.

```
Definition set_of_nreg_open r :=
  complement (set_of_reg_open r) (singleton emptyset).
Definition set_of_nreg_order r :=
  opposite_order (inclusion_suborder (set_of_nreg_open r)).
Definition canonical_reg_open r x :=
  bar1_22 r (Zo (substrate r) (fun z => gle r x z)).

Lemma Exercise1_23a r X: (* 4 *)
  inc X (set_of_nreg_open r) <-> (open_r r X & nonempty X).

Lemma Exercise1_23b r X Y: order r -> (* 2 *)
  (gle  (set_of_nreg_order r) X Y <->
  (nonempty X & nonempty Y & open_r r X & open_r r Y & sub Y X)).
```

(a) Show that the mapping r *is increasing and that* $r(E)$ *is cofinal in* $R_0(E)$*.*

We have the following interesting property: $y \in r(x)$ if and only if, whenever $y \leq z$, there is a common upper bound to x and z. The mapping r is increasing, as the composition of two increasing functions. In general, it is not strictly increasing (if E is right directed, it is constant).

```
Lemma Exercise1_23c r x: order r -> (* 2 *)
  open_o r  (Zo (substrate r) (fun z => gle r x z)).
Lemma Exercise1_23d1 r x: (* 2 *)
   order r -> inc x (substrate r) ->
  inc x (canonical_reg_open r x).
Lemma  Exercise1_23d2  r x:  (* 3 *)
  order r -> inc x (substrate r) ->
  inc (canonical_reg_open r x) (set_of_nreg_open r).
Lemma Exercise1_23e r x y: order r -> (* 10 *)
  inc x (substrate r) -> inc y (substrate r) ->
  (inc y (canonical_reg_open r x) <->
    forall z, gle r y z -> exists t, gle r z t & gle r x t).
Lemma  Exercise1_23f r x y: order r -> (* 9 *)
  gle r x y -> gle  (set_of_nreg_order r)
  (canonical_reg_open r x)  (canonical_reg_open r y).
```

If X is regular, then $X = \bar{X}$. If $x \in X$, then $[x, \to [\subset X$, thus $r(x) \subset X$. As a consequence, the image of r is cofinal in R_0.

```
Lemma  Exercise1_23g r: order r -> (* 14 *)
  cofinal_set (set_of_nreg_order r)
   (fun_image (substrate r) (canonical_reg_open r)).
```

(b) An ordered set E *is said to be* antidirected *if the canonical mapping* $r : E \to R_0(E)$ *is injective. For this to be so it is necessary and sufficient that the following two conditions should be satisfied.*

(I) If x *and* y *are two elements of* E *such that* $x < y$*, there exists* $z \in E$ *such that* $x < z$ *and such that the intervals* $[y, \to [$ *and* $[z, \to [$ *do not intersect.*

(II) If x *and* y *are two non-comparable elements of* E *then either there exists* $x' \geq x$ *such that the intervals* $[x', \to [$ *and* $[y, \to [$ *do not intersect, or else there exists* $y' \geq y$ *such that the intervals* $[x, \to [$ *and* $[y', \to [$ *do not intersect.*

Let $A(x, y)$ be the property that the intervals $[x, \to [$ and $[y, \to [$ do not intersect. We might replace $x < z$ in (I) by $x \leq z$, for $A(y, x)$ is false (since y is in the intersection). Our criterion 1.23e says: $a \in r(x)$ if and only if, for all b such that $a \leq b$, $A(b, x)$ is false.

Thus (I) can be written as: if $x < y$ then $x \notin r(y)$, and (II) as if x and y are non-comparable, then $x \notin r(y)$ or $y \notin r(x)$. Write this as "not ($x \in r(y)$ and $y \in r(x)$)". Since $x \in r(x)$ and $y \in r(y)$ this is $r(x) \neq r(y)$. Condition (II) becomes: if $r(x) = r(y)$, then x and y are comparable. But (I) excludes $x < y$ and $y < x$, so that (I) and (II) imply injectivity of r. Conversely, injectivity implies (II). Assume now $x < y$. There is z such that $x \leq z$ and $A(y, z)$, since otherwise we would have $r(x) \subset r(y)$. But $x \leq y$ implies $r(x) \subset r(y)$, thus $r(x) = r(y)$. If the set if antidirected, we get $x = y$, absurd.

```
Lemma  Exercise1_23h r: order r -> (* 58 *)
  let aux := (fun x y => forall z, gle r x z -> gle r y z -> False) in
  (anti_directed r) <->
  ((forall x y, glt r x y -> exists z, (glt r x z & aux y z))
    & forall x y, inc x (substrate r) -> inc y (substrate r) ->
      (gle r x y \/ gle r y x \/ (exists x', gle r x x' & aux x' y) \/
        (exists y', gle r y y' & aux x y'))).
```

(c) Show that, for every ordered set E, $R_0(E)$ *is antidirected and that the canonical mapping of* $R_0(E)$ *into* $R_0(R_0(E))$ *is bijective (use Exercise 22(a)).*

The condition A(X,Y) in $R_0(E)$ says that there is no upper bound for X and Y. We know that R(E) is a complete lattice, so that the condition becomes: there is only one upper bound, namely the empty set (that is not in R_0). Since $x \in X$ implies $X \leq r(x)$, the condition becomes X and Y are disjoint.

```
Lemma Exercise1_23i r x y: order r -> (* 10 *)
  inc x y -> inc y  (set_of_nreg_open r) ->
    gle (set_of_nreg_order r) y (canonical_reg_open r x).
Lemma Exercise1_23j r: order r -> (* 11 *)
  let r':= set_of_nreg_order r in
    (forall x y, inc x (substrate r') -> inc y (substrate r') ->
      ((forall z, gle r' x z ->  gle r' y z -> False) <->
       (disjoint  x y))).
```

We know that R(E) is a Boolean lattice. Given X and Y we construct a regular open set Z, a subset of X that is disjoint from Y. (This is the complement (for the lattice) of Y in X whenever $Y \subset X$). If Z is empty, then $X \subset Y$. This can be restated as: if $X \subset Y$ is false, then $Z \in R_0(E)$. It follows that $R_0(E)$ is antidirected. The canonical mapping of $R_0(E)$ into $R_0(R_0(E))$ is hence injective.

```
Lemma Exercise1_23k r: order r -> (* 63 *)
  anti_directed (set_of_nreg_order r).
```

The mapping $r : R_0(E) \to R_0(R_0(E))$ is injective by the previous lemma. We can rewrite 1.23i as: $y \in r(x)$ if and only if for all t, $y \leq t$ implies $t \cap x$ is non-empty. Proof: Let $a \in y$. We have $y \leq r(a)$, hence $r(a) \cap x$ is non-empty. This implies that there is $b \in x$, such that $a \leq b$. Conversely, assume every $a \in y$ bounded by an element of x; assume $y \leq t$. There is $a \in t \subset y$. If $a \leq b$ then $b \in t$. If moreover $a \in x$ the set $t \cap x$ is non-empty.

Hence $y \in r(x)$ if and only if every element of y is bounded by an element of x. It follows that if x is non-empty and open, then $y \in r(\bar{x})$ if and only if every element of y is bounded by an element of x.

Let Y be in $R_0(R_0(E))$. Proving surjectivity of r means showing existence of a set X such that every element of the union of the elements of Y is bounded by an element of X. Take $X = \cup Y$. The set $r(\bar{X})$ is the set of all $z \in E$, such that each element of z is bounded by an element of $\bigcup Y$. It follows $Y \subset r(\bar{X})$. Assume Y cofinal. Since Y is regular, this will imply $Y = r(\bar{X})$. Let X be a regular open set, such that each element of X is bounded by an element of the union of Y. We must show: There is $Z \in Y$ such that $Z \subset X$. Is this true?

```
Lemma Exercise1_23l r y: order r ->
  let r' := set_of_nreg_order r in
    inc y (set_of_nreg_open r') ->
    exists_unique (fun x => inc x (set_of_nreg_open r) &
      y =  canonical_reg_open r' x).
Proof. Abort.
```

24. * (a) An ordered set E is said to be *branched* (on the right) if for each $x \in E$ there exist y, z in E such that $x \leq y$, $x \leq z$ and the intervals $[y, \to [$ and $[z, \to [$ do not intersect. An antidirected set with no maximal element (Exercise 23) is branched.

(b) Let E be the set of intervals in **R** of the form $[k.2^{-n}, (k+1).2^{-n}]$ $(0 \leq k < 2^n)$, ordered by the relation $\supset$. Show that E is antidirected and has no maximal elements.

(c) Give an example of a branched set in which there exists no antidirected cofinal subset (Take the product of the set E defined in (b) with a well-ordered set which contains no countable cofinal subset, and use Exercise 22.)

(d) Give an example of an ordered set E which is not antidirected, but which has an antidirected cofinal subset (Note that an ordinal sum $\sum_{\xi \in E} F_\xi$ contains a cofinal subset isomorphic to E).*

```
Definition branched r :=
  order r & (forall x, inc x (substrate r) ->
    exists y, exists z, gle r x y & gle r x z &
      (forall t, gle r y t -> gle r z t -> False)).

Lemma Exercise1_24a r: (* 8 *)
```

```
  order r -> anti_directed r ->
  (forall x, inc x (substrate r) -> ~ maximal_element r x)
  -> branched r.
```

We consider the product $\mathbf{N}\times\mathbf{N}^*$, the set of pairs (p,q) with $q\neq 0$, with the relation $(p_1,q_1)\prec(p_2,q_2)$ if $p_1q_2\leq p_2q_1$. Let $x\sim y$ if $x\prec y$ and $y\prec x$. This is an equivalence relation and $\prec/\sim$ is an order. We denote by Q^+ the quotient, and by $\leq$ its ordering. The class of (a,b) will be denoted by a/b.

```
Definition Nstar := compl_singl Bnat \0c.
Definition Qplus1 := product Bnat Nstar.
Definition Qplus_eq_r x y := (P x) *c (Q y) = (P y) *c (Q x).
Definition Qplus1_le_r x y:= (P x) *c (Q y) <=c (P y) *c (Q x).

Definition Qplus_eq := graph_on Qplus_eq_r Qplus1.
Definition Qplus := quotient Qplus_eq.
Definition Qplus_or:= graph_on (fun x y => Qplus1_le_r (rep x) (rep y)) Qplus.


Lemma Qplus_eq_sr : substrate Qplus_eq = Qplus1. (* 1 *)
Lemma Qplus_eq_related x y: (* 1 *)
  related Qplus_eq x y <-> (inc x Qplus1 & inc y Qplus1 &  Qplus_eq_r x y).
Lemma Qplus1_inc1 x: inc x Qplus1 <->   (* 1 *)
  (is_pair x & inc (P x) Bnat & inc (Q x) Bnat & (Q x) <> \0c).

Lemma Qplus_equiv: is_equivalence Qplus_eq.  (* 15 *)
Lemma Qplus_inc1 x: (* 3 *)
   inc x Qplus -> (inc (rep x) Qplus1 & x = class Qplus_eq (rep x)).
Lemma Qplus_inc2 x: (* 1 *)
  inc x Qplus1 -> inc (class Qplus_eq x) Qplus.

Lemma Qplus_or_sr : substrate Qplus_or = Qplus. (* 2 *)
Lemma Qplus_leq_compat a b a' b':  (* 24 *)
  related Qplus_eq a a' ->  related Qplus_eq b b' ->
  (Qplus1_le_r a b <->  Qplus1_le_r a' b').
Lemma Qplus_or_gle1 x y:  (* 7 *)
  inc x Qplus1 -> inc y Qplus1 ->
  (Qplus1_le_r x y <-> gle Qplus_or (class Qplus_eq x) (class Qplus_eq y)).
Lemma Qplus_or_gle2 x y:  gle Qplus_or x y ->
  Qplus1_le_r (rep x) (rep y).
Lemma Qplus_or_or: order  Qplus_or. (* 26 *)
Lemma Qplus_or_tor: total_order  Qplus_or. (* 5 *)
```

Consider the elements of the form $a/2^n$, denoted a_n. They are in Q^+, and $a_n\leq b_n$ if and only if $a\leq b$.

```
Definition Qpair k n := class Qplus_eq (J k (\2c ^c  n)).
Lemma Qpair_q1 k n: inc k Bnat -> inc n Bnat ->  (* 3 *)
   inc (J k (\2c ^c n)) Qplus1.
Lemma Qpair_q k n:
   inc k Bnat -> inc n Bnat -> inc (Qpair k n) Qplus. (* 2 *)
Lemma Qpair_eq k n m: inc k Bnat -> inc n Bnat -> inc m Bnat ->
  Qpair k n = Qpair (k *c (\2c ^c m)) (m +c n).
Lemma  Qpair_le0 a b c d:  (* 3 *)
  inc a Bnat -> inc b Bnat -> inc c Bnat -> inc d Bnat ->
```

```
  let f:= fun k n =>  k *c (\2c ^c n) in
  (gle Qplus_or (Qpair a b) (Qpair c d) <-> (f a d) <=c (f c b)).

Lemma Qpair_le k k' n: (* 9 *)
 inc k Bnat -> inc k' Bnat -> inc n Bnat ->
  (k <=c k' <-> gle Qplus_or (Qpair k n) (Qpair k' n)).
```

Let E be the set of all intervals of the form $A_{kn} = [k_n, (k+1)_n]$, ordered by $\supset$. Note that k_n and $(k+1)_n$ are in the interval. Since, for any intervals I_1 and I_2, we have $I_1 \subset I_2$ if the endpoints of I_1 are in I_2 we get that $A_{kn} \leq A_{lm}$ is equivalent to

$$k.2^m \leq l.2^n \qquad (l+1).2^n \leq (k+1)2^m.$$

This relation implies $n \leq m$. If $m = n + p$ it reduces to

$$k2^p \leq l \qquad (l+1) \leq (k+1)2^p$$

In particular, if $p = 0$, it implies $k = l$, so that k and l are uniquely defined from A_{kn}.

```
Definition Qpairi k n := interval_cc Qplus_or (Qpair k n) (Qpair (succ k) n).
Definition Qpairis :=
  fun_image (product Bnat Bnat) (fun z => Qpairi (P z) (Q z)).
Definition Qpairi_o := opposite_order (inclusion_suborder Qpairis).

Lemma Qpairis_pr x: (* 4 *)
  inc x Qpairis <-> exists k, exists n, inc k Bnat & inc n Bnat& x =  Qpairi k n.

Lemma Qpairio_or: order Qpairi_o. (* 1 *)

Lemma Qpairio_sr: substrate Qpairi_o = Qpairis. (* 1 *)
Lemma Qpairio_gle x y:  (* 1 *)
   gle Qpairi_o x y <-> (inc x  Qpairis & inc y  Qpairis & sub y x).
Lemma Qpairis_pr1 n k x: inc x (Qpairi k n) (* 2 *)
  <-> (gle Qplus_or (Qpair k n) x &gle Qplus_or x (Qpair (succ k) n)).
Lemma Qpairis_pr2 k n: inc k Bnat -> inc n Bnat -> (* 9 *)
  (inc (Qpair k n)  (Qpairi k n) & inc  (Qpair (succ k) n) (Qpairi k n)).
Lemma Qpairio_gle1 k n l m: (* 12 *)
  inc k Bnat -> inc n Bnat ->
  inc l Bnat -> inc m Bnat ->
  let f:= fun k n => (k *c (\2c ^c n)) in
  gle Qpairi_o (Qpairi k n) (Qpairi l m) <->
    ((f k m) <=c  (f l n)  & (f (succ l) n) <=c (f (succ k) m)).
Lemma Qpairio_gle2 k n l m: (* 37 *)
  inc k Bnat -> inc n Bnat -> inc l Bnat -> inc m Bnat ->
  let f:= fun k n => (k *c (\2c ^c n)) in
  gle Qpairi_o (Qpairi k n) (Qpairi l m) <->
  (exists p, inc p Bnat & m = n +c p &
    (f k p) <=c l & (succ l) <=c (f (succ k) p)).
Lemma Qpairio_eq k n l m: (* 17 *)
  inc k Bnat -> inc n Bnat -> inc l Bnat -> inc m Bnat ->
   (Qpairi k n) = (Qpairi l m) -> (k = l & n = m).
```

The set E has no maximal element, since if $x = A_{kn}$, then $y = A_{2k,n+1}$ satisfies $x < y$. Let $x = A_{kn}$ and $y = A_{lm}$. Assume $x \leq z$ and $y \leq z$. Evaluating the conditions above shows that if $n \leq m$ then $x \leq y$. In particular, x and y are comparable. Thus condition (II) of the previous exercise is obviously satisfied. Condition (I) is also true: Assume first $x < y$. Write $x = A_{kn}$

and $y = A_{k.2^p+l,n+p}$. We have $p > 0$ and $0 \leq l < 2^p$. Take $z = A_{k.2^p+m,n+p}$, where $m = 1$ (if $l = 0$) and $m = 0$ (otherwise). Then y and z are non-comparable.

As a consequence, E is antidirected and branched.

```
Lemma Qpairio_gle3 k n l m z: (* 31 *)
  inc k Bnat -> inc n Bnat -> inc l Bnat -> inc m Bnat ->
  n <=c m  ->
  gle Qpairi_o (Qpairi k n) z -> gle Qpairi_o(Qpairi l m) z ->
  gle Qpairi_o (Qpairi k n) (Qpairi l m).
Lemma Qpairio_gle4 x y:  (* 9 *)
   inc x (substrate Qpairi_o) -> inc y (substrate Qpairi_o) ->
   gle Qpairi_o x y \/
   gle Qpairi_o y x \/
   (forall z : Set, gle Qpairi_o x z -> gle Qpairi_o y z -> False).

Lemma Exercise1_24b x: (* 22 *)
  inc x (substrate Qpairi_o) -> ~ (maximal_element Qpairi_o x).
Lemma Exercise1_24c: anti_directed Qpairi_o.  (* 78 *)
Lemma Exercise1_24d: branched Qpairi_o. (* 1 *)
```

Consider now points (c). A product is branched if one factor is branched. If the product is E × F, where E is the set studied above, if F is totally ordered, if X is a cofinal subset of the product, then $[x, \rightarrow [\cap [y, \rightarrow [$ is non-empty if and only if x and y are comparable.

Consider an ordinal sum $\sum E_i$. It is wrong that the sum has a cofinal set isomorphic to the index set. Assume for instance that the index set has a single element α. Then the sum has a cofinal set reduced to a single element, thus has a greatest element. But the sum is isomorphic to E_α. However, if for any maximal index α the set E_α has a greatest element x_α we can proceed as follows. Consider the element y_i of the sum, which is x_α in the previous case, any element of E_i otherwise. The set Y of all y_i is isomorphic to the index set. Consider x_i in the sum. If i is maximal, we have $x_i \leq x_\alpha = y_i$. Otherwise there is j such that $i < j$ and $x_i < y_j$. Thus Y is cofinal.

```
Lemma Exercise1_24e r r': (* 14 *)
  branched r -> order r' ->
  branched (order_product2 r r').
```

Points (c) and (d) remain to do.

10.2 Section 2

Consider an ordered set E, such that, whenever a and b are in E, $a \leq b$ is equivalent to $a \subset b$. Consider a family A_i that has an upper bound S. Then S contains the union U of the family. If this union is in E, it is the least upper bound of the family.

```
Lemma induced_sub_pr1 r X x: (* 2 *)
  (forall a b, gle r a b -> sub a b) ->
  upper_bound r X x -> sub (union X) x.
Lemma induced_sub_pr2 r X:  (* 5 *)
  order r ->
  (forall a b, inc a (substrate r) -> inc b (substrate r) ->
    (gle r a b <-> sub a b)) ->
```

```
  sub X (substrate r) -> inc (union X) (substrate r) ->
  least_upper_bound r X (union X).
Lemma inc_coarse a b E: (* 1 *)
  inc a E -> inc b E -> inc (J a b) (coarse E).
Lemma order_exten r r': order r -> order r' -> (* 1 *)
  (r = r') = (forall x y, gle r x y = gle r' x y).
```

1. *Show that, in the set of orderings on a set* E*, the minimal elements (with respect to the ordered relation "*Γ *is coarser than* Γ'*" between* Γ *and* Γ'*) are the total orderings on* E *and that if* Γ *is any ordering on* E*, the graph of* Γ *is the intersection of the graphs of the total orderings on* E *which are coarser than* Γ *(apply Theorem 2 of no. 4). Deduce that every ordered set is isomorphic to a subset of a product of totally ordered sets.*

We define here the set of orderings on E, and define the *opposite* of the coarser ordering (remember that coarse E is $E \times E$).

```
Definition set_of_orders x :=
  Zo (powerset (coarse x))(fun z => substrate z = x & order z).
Definition finer_order x :=
  inclusion_suborder (set_of_orders x).

Lemma set_of_orders_rw x z: (* 3 *)
  inc z (set_of_orders x) <-> (substrate z = x & order z).
Lemma fo_or x: (* 1 *)
  order (finer_order x).
Lemma fo_sr x: (* 1 *)
  substrate (finer_order x) = set_of_orders x.
Lemma fo_gle x u v: (* 2 *)
  gle (finer_order x) u v <->
  (order u & order v & substrate u = x & substrate v = x & sub u v).
Lemma fo_gle1 x u v: (* 12 *)
  gle (finer_order x) u v <->
  (order u & order v & substrate u = x & substrate v = x &
  forall a b, inc a x -> inc b x -> gle u a b -> gle v a b).
```

Let G be the graph of an ordering on E. Let G' be the union of G and the set of pair (a, b) such that $a \leq x$ and $y \leq b$. This is an ordering if $y \leq x$ is false. Assume G maximal. We deduce $G = G'$, hence $x \leq y$, since $(x, y) \in G'$. Thus G' is totally ordered. The converse is immediate.

```
Lemma Exercise2_1a r x y:  (* 37 *)
  order r -> inc x (substrate r) -> inc y (substrate r) ->
  ~ gle r y x ->
  let E :=  substrate r in
  let r' := union2 r (Zo (coarse E)(fun z=>  gle r (P z) x & gle r y (Q z)))
  in gle (finer_order E) r r' & inc (J x y) r'.
Lemma Exercise2_1b E a: (* 15 *)
  maximal_element (finer_order_order E) a <-> (total_order a & substrate a = E).
```

We show here that the set of orderings is inductive (for the "finer" ordering). Consider a totally ordered family X (each element being an ordering on E). If X is empty, is has an upper bound as there is the least one ordering on E. Otherwise, let U be the union of the family. This is an ordering on E, thus is the least upper bound of X.

First Corollary to Zorn's lemma says that for any ordering Γ there is a maximal ordering Γ' such that $\Gamma \leq \Gamma'$. This ordering is total, and extends Γ.

```
Lemma fo_inductive: (* 47 *)
  forall E, inductive_set (finer_order E).
Lemma order_total_extension r: order r -> (* 6 *)
  exists r', total_order r' & substrate r' = substrate r & sub r r'.
```

Let Γ_0 be an ordering, S the set of all total orders Γ' such that $\Gamma_0 \leq \Gamma'$ (where $\leq$ is "finer"). The previous lemma says that there is some Γ_1 in S. The relation $\Gamma \subset \bigcap S$ is obvious. Conversely, consider an element of $\bigcap S$. It is in Γ_1, thus is a pair (a, b) such that $a \leq_1 b$ (where $\leq_1$ is the ordering of Γ_1). In particular, a and b are in the substrate of Γ_0. Assume that $a \leq_0 b$ is false (where $\leq_0$ is the ordering of Γ_0). We can extend Γ_0 to an ordering such that $b \leq_2 a$, and this ordering can be extended to a total ordering, an element of S; for which we have $a \leq_2 b$, absurd.

```
Lemma Exercise2_1c r: (* 28 *)
 order r -> r = intersection (Zo (set_of_orders (substrate r))
   (fun r' => total_order r' & sub r r')).
```

Consider an ordering on E, and let B be the set of total orderings on E that extend it. Consider the set of functions $B \to E$. If f and g are two such functions, we say $f \leq g$ if for all $i \in B$, $f(i) \leq_i g(i)$, where $\leq_i$ is i considered as an ordering. This gives an ordered set, the product of the total orders of B. For each $x \in E$, we consider the constant function f_x with value x. The previous lemma says $f_x \leq f_y$ if and only if $x \leq y$. Thus $x \mapsto f_x$ is the desired isomorphism (the range is the set of constant functions).

```
Lemma Exercise2_1d r:   (* 28 *)
  order r -> exists g, exists h,
  order_fam g &
  (forall i, inc i (domain g) -> total_order (V i g)) &
  order_morphism h r (order_product g).
```

---

2. *Let* E *be an ordered set and let* $\mathfrak{B}$ *be the set of subsets of* E *which are well-ordered by the inducing ordering. Show that the relation "*X *is a segment of* Y*" on* $\mathfrak{B}$ *is an order relation between* X *and* Y *and that* $\mathfrak{B}$ *is inductive with respect to this order relation. Deduce that there exist well-ordered subsets of* E *which have no strict upper bound in* E.

Let's show that the set $\mathfrak{B}$ is inductive. We consider a totally ordered family X_i. We pretend that its union X is an upper bound. The non trivial point is to show that the union is well-ordered. Consider Y a nonempty subset of X. Let $a \in Y$, say $a \in X_i$. Let Z be the intersection of Y and X_i and b its least element. Let c be in Y, say $c \in X_j$. We have to show $b \leq c$. This is clear if $c \in X_i$, in particular when $X_j \subset X_i$. If this relation is false, then $X_i \subset X_j$, so that b and c are in X_j, thus are comparable. If $c \leq b$ we get $c \in X_i$, because X_i is a segment of X_j.

```
Lemma Exercise2_2a r: (* 79 *)
  let B:= Zo (powerset (substrate r)) (fun z=> worder (induced_order r z)) in
    let ss_order := Zo (coarse B)
      (fun z=> is_segment (induced_order r (Q z)) (P z)) in
      order r ->
      (order ss_order & substrate ss_order = B & inductive_set ss_order).
```

Let E be an ordered set. We apply Zorn's lemma to the ordering defined above. It says that we have a maximal element, i.e., a well-ordered subset X of E. Assume that X has an upper bound x and $Y = X \cup \{x\}$. This is a well-ordered set (adding of a greatest element to a well-ordered set), hence $Y \in \mathfrak{B}$, thus $x \in X$.

```
Lemma Exercise2_2b r: order r -> (* 44 *)
  exists x, sub x (substrate r) &  worder (induced_order r x) &
    forall z, upper_bound r x z -> inc z x.
```

---

3. *Let* E *be an ordered set. Show that there exist two subsets* A, B, *of* E *such that* $A \cup B = E$ *and* $A \cap B = \emptyset$ *and such that* A *is well-ordered and* B *has no least element (for example, take* B *to be the union of those subsets of* E *which have no least element).* * *Give an example in which there are several partitions of* E *into two subsets having these properties.* *

We first rewrite the condition "$A \cup B = E$ and $A \cap B = \emptyset$" as A is the complementary of a subset B of E. Since the set of integers is well-ordered, every bounded set has a least upper bound. If the set is non-empty, this is the greatest element. Fort the example, consider **N** with the opposite of its natural ordering, n an integer and $A = [0, n[$.

```
Lemma complement_p1 C A B: (* 7 *)
  union2 A B = C -> intersection2 A B = emptyset ->
  (sub A C & A = complement C B).
Lemma complement_p2 C A B: (* 1 *)
  union2 A B = C -> intersection2 A B = emptyset ->
  (sub B C & B = complement C A).
Lemma complement_p3 C B: (* 5 *)
 let A:=  complement C B in sub B C ->
  (union2 A B = C & intersection2 A B = emptyset).
Lemma complement_p4 C A:  (* 1*)
 let B:=  complement C A in sub A C ->
  (union2 A B = C & intersection2 A B = emptyset).
Lemma Bnat_greatest A: (* 33 *)
  sub A Bnat -> nonempty A ->
  (exists x, upper_bound Bnat_order A x) ->
  (exists x, greatest_element (induced_order Bnat_order A) x).


Definition ex23_prop r A B:=
  union2 A B = substrate r & intersection2 A B = emptyset &
  worder (induced_order r A) &
  (forall y, ~ (least_element (induced_order r B) y)).

Lemma Exercice2_3a r: (* 22 *)
  order r -> exists A, exists B,  ex23_prop r A B.
Lemma  Exercice2_3b n (r := opposite_order Bnat_order)   (* 34 *)
  (A := interval_co_0a n) (B:= complement Bnat A) :
   inc n Bnat -> (order r & ex23_prop r A B).
```

¶ 4. *An ordered set* E *is said to be* partially well-ordered *if every totally ordered subset of* E *is well-ordered. Show that in every ordered set* E *there exists a partially well-ordered subset which is cofinal in* E *(Consider the set* $\mathfrak{F}$ *of partially well-ordered subsets of* E*, and the order relation "*$X \subset Y$ *and no element of* $Y - X$ *is bounded above by any element of* X*" between* X *and* Y *of* $\mathfrak{F}$*. Show that* $\mathfrak{F}$ *is inductive with respect to this order relation).*

Consider an ordered set $(E, \leq)$. We denote by $\leq_X$ the ordering induced by $\leq$ on X, and by $p(F)$ the property that, for any subset X of F, if $\leq_X$ is total, then $\leq_X$ is a well-ordering. This is equivalent to: every nonempty subset X of F such that $\leq_X$ is total has a least element (for $\leq_X$).

Assume $F \in \mathfrak{F}$ and $t \in E$. Then $F \cup \{t\} \in \mathfrak{F}$. In fact, consider a non-empty totally ordered subset X of $F \cup \{t\}$. This set has a least element if it does not contain t, or contains only t. Consider otherwise the set $X - \{t\}$. It has a least element y, and $\inf(y, t)$ is the least element of X.

Let $f(x, y)$ be the property that, for any a in x and b in $y - x$, the relation $b \leq a$ is false. Consider the relation $x \prec y$ if x and y are two elements of $\mathfrak{F}$ such that $x \subset y$ and $f(x; y)$. The first condition says that the relation is antisymmetric. It is in fact an ordering.

Consider a family X_i of elements of $\mathfrak{F}$, its union X, and a non-empty totally ordered subset Y of X. We have $x \in Y \cap X_i$ for some x and i. Let $K = Y \cap X_i$. This is a non-empty totally ordered subset of X_i, thus has a least element y. We pretend that this is the least element of Y, provided that the family X_i is totally ordered by $\prec$. Consider $y' \in Y$. It is in some X_j. If $X_j \subset X_j$, then $y' \in K$ and the conclusion follows. Otherwise we have $y \in X_i$, $y' \in X_j - X_i$, this implies that $y' \leq y$ is false; but since the ordering on Y is total, it implies $y \leq y'$. This argument shows $X \in \mathfrak{F}$. From this, it is easy to deduce that X is an upper bound for X_i.

This show that $\mathfrak{F}$ is inductive, thus has a maximal element F. For no $t \notin F$ we have $F \leq F \cup \{t\}$. This says that F is cofinal in E.

```
Lemma Exercice2_4 r: order r -> (* 131 *)
  let  pworder :=  fun F => forall X,
     sub X F -> total_order (induced_order r X) -> worder (induced_order r X)
    in
    exists F, (pworder F & cofinal_set r F).
```

---

5. *Let* E *be an ordered set and let* $\mathfrak{I}$ *be the set of* free *subsets of* E*, ordered by the relation defined in § 1, Exercise 5. Show that, if* E *is inductive, then* $\mathfrak{I}$ *has a greatest element.*

This is a direct consequence of the first corollary to Zorn's lemma.

```
Lemma Exercise2_5 r: order r -> inductive_set r -> (* 11 *)
  exists x, greatest_element (free_subset_order r) x.
```

¶ 6. *Let* E *be an ordered set and let* f *be a mapping from* E *into* E *such that* $f(x) \geq x$ *for all* $x \in E$.

(a) Let $\mathfrak{S}$ *be the set of subsets* M *of* E *with the following properties: (1) the relation* $x \in M$ *implies* $f(x) \in M$*; (2) if a non-empty subset of* M *has a least upper bound in* E*, then this*

least upper bound belongs to M. *For each* $a \in$ E, *show that the intersection* C_a *of the sets of* $\mathfrak{S}$ *which contain* a *also belongs to* $\mathfrak{S}$*; that* C_a *is well-ordered; and that if* C_a *has an upper bound* b *in* E, *then* $b \in C_a$ *and* $f(b) = b$. C_a *is said to be the* chain *of* a *(with respect to the function* f*). (Consider the set* $\mathfrak{M}$ *whose elements are the empty set and the subsets* X *of* E *which contain* a *and have a least upper bound* m *in* E *such that* $m \notin X$ *or* $f(m) > m$*, and apply Lemma 3 of no. 3 to the set* $\mathfrak{M}$*.)*

(b) Deduce from (a) that if E *is inductive, then there exists* $b \in$ E *such that* $f(b) = b$.

Consider a nonempty well-ordered set E. If e is the least element, the segment with end-point e is empty. If the interval $]\leftarrow, a]$ is not the whole set, it is a segment with end-point b.

```
Lemma Exercise2_6f r: worder r -> (* 7 *)
  nonempty (substrate r) -> exists x,
    (inc x (substrate r) & segment r x = emptyset).

Lemma Exercise2_6g r a: (* 11 *)
   worder r -> inc a (substrate r) ->
  let m := Zo (substrate r) (fun z => glt r a z) in
    nonempty m -> exists b,
      inc b (substrate r) & (segment_c r a = segment r b).
```

We start with some definitions and show part (b), which is trivial, and independent of (a).

```
Section Exercise2_6.
Variables r f : Set.
Hypothesis or: order r.
Hypothesis ff: is_function f.
Hypothesis sf: substrate r = source f.
Hypothesis tf: substrate r = target f.
Hypothesis fxx: forall x, inc x (substrate r) -> gle r x (W x f).

Definition bigS :=
  Zo (powerset (substrate r))
  (fun M => (forall x, inc x M -> inc (W x f) M) &
    (forall N x, sub N M -> nonempty N -> least_upper_bound r N x -> inc x M)).

Definition chain a :=
  intersection (Zo bigS (fun z => inc a z)).

Lemma Exercise2_6i: inductive_set r -> (* 2 *)
  exists a, inc a (source f) & W a f = a.
```

We start with some trivial lemmas. We deduce $C_a \in \mathfrak{S}$. If we correct[1] the third claim as "if C_a has a least upper bound b, then $b \in C_a$ and $f(b) = b$" it becomes trivial.

```
Lemma Exercise2_6a: inc (substrate r) bigS.  (* 3 *)
Lemma Exercise2_6b a:  (* 1 *)
  inc a (source f) -> nonempty (Zo bigS (fun z => inc a z)).
Lemma Exercise2_6c a: (* 2 *)
  inc a (source f) -> inc a (chain a).

Lemma Exercise2_6d a:  (* 13 *)
  inc a (source f) -> inc (chain  a) bigS.
```

[1] as in the French edition of Bourbaki

```
Lemma Exercise2_6e a b: (* 7 *)
  inc a (source f) -> least_upper_bound r  (chain a) b ->
  (inc b  (chain a) & W b f = b).
```

Consider now the second claim: C_a is well-ordered. For this purpose we consider some sets. $\mathfrak{M}_0$ is the set all subsets of E that contain a and have a least upper bound. If x is such a set, we have $\sup x \in E$ and $f(\sup x) \in E$. Denote these two quantities by $p_1(x)$ and $p_2(x)$. The set $\mathfrak{M}_1$ contains those x for which $\sup x \notin x$, and the set $\mathfrak{M}_2$ contains those x for which $\sup x < f(\sup x)$. We define $p(x)$ as $p_1(x)$ and $p_2(x)$, whichever applies (we use p_1 for the intersection). The set $\mathfrak{M}$ will be the union of $\mathfrak{M}_1$ and $\mathfrak{M}_2$, to which we adjoin the empty set, with $p(\emptyset) = a$. By construction $p(x) \in E - x$, whenever $x \in \mathfrak{M}$.

We now apply Lemma 3.3. It asserts the existence of a set M, well-ordered by $\leq_M$, such that, (3) for any $x \in M$, if S_x is the segment (for $\leq_M$) with endpoint x, then $S_x \in \mathfrak{M}$ and $p(S_x) = x$. Moreover (4) $M \notin \mathfrak{M}$,

Since M is non-empty by (4), it has a least element, say x. We have $S_x = \emptyset$. Applying $p(S_x) = x$ gives $x = a$. Thus $a \in M$. Assume that M has least upper bound m. Thus $M \in \mathfrak{M}_0$. Now (4) says $m \in M$ and $m = f(m)$.

We notice that $\leq_M$ and $\leq$ coincide on M (so that M will be totally ordered by $\leq$). In fact, assume $y <_M x$ so that $y \in S_x$. The segment S_x, being non-empty in $\mathfrak{M}$, has a least upper bound c (hence $y \leq c$). We have $c \leq p(S(x))$ by construction, hence $c \leq x$, thus $y \leq x$.

Assume that M has a greatest element x (for $\leq_M$). This is a greatest element for $\leq$, hence is $\sup M$. We deduce $x = m$, thus $x = f(x) \in M$.

Let's show that M satisfies (1). Fix $y \in M$. Consider the set of all x such that $y <_M x$. If it is empty then y is the greatest element of M for $\leq_M$, thus is the greatest element for $\leq$, thus is m, hence $y = f(y) \in M$. Otherwise, there is $z \in M$ such that S_z is the set of all t such that $t \leq_M y$. The same argument as above shows that $y = \sup S_z$. Since S_z is non-empty and not in $\mathfrak{M}_1$, but in $\mathfrak{M}$, it is in $\mathfrak{M}_2$, so that $z = p(S_z) = f(\sup S_z) = f(y)$. This shows $f(y) \in M$.

Let's show that M satisfies (2). Take a non-empty subset N of M that has a least upper bound y, and let Q be the set of elements $z \in M$ such that $y \leq z$. Assume first Q empty. Take $t \in N$. If for some $u \in N$ we have $t \leq u$ then $t \leq u \leq y$. Otherwise, since M is totally ordered, t is an upper bound of N and $t \leq y$. Thus $y = \sup M$, hence $y \in M$. Assume Q non-empty; so that it has a least element z. In particular $y \leq z$. If $z \in N$, then $z \leq y$, and $y = z \in M$. Otherwise $N \subset S_z$. The segment has a least upper bound α and $y \leq \alpha \leq z$. Note that $\alpha \notin S_z$ (since $\alpha \in M$ implies $z \leq \alpha$). Thus $S_z \in \mathfrak{M}_1$. Evaluating p yields $\alpha = z$. It suffices to show $\alpha \leq y$ since it implies $y = \alpha = z \in M$. We must show that y is an upper bound of S_y. Thus assume $t <_M z$. Assume that for some $v \in N$ we have $t < v$. Then $v \leq y$, thus $t \leq y$. Otherwise, t is an upper bound for N and $y \leq t$. This implies $y \leq z$, absurd.

The properties shown above can be summarized as: M contains the chain of a. The chain is well-ordered as a subset of a well-ordered sets, with compatible orderings.

```
Lemma Exercise2_6ha: (* 174 *)
  inc a (source f) -> worder (induced_order r  (chain a)).
End Exercise2_6.
```

Second part of the Exercise. We must show that f has a fixed point. Let P(E) be the property: there is a chain C_a that has a least upper bound. We have shown that this least upper is a fixed point. Conversely, if f has a fixed point b, then the chain of b has a least

upper bound b. Thus P(E) is equivalent to the existence of a fixed point of f. Note that P is a consequence of Q(E) that says: every non-empty well-ordered subset of E has a least upper bound. Are we assumed to prove Q? this is unclear.

¶ 7. *Let* E *be an ordered set and let* F *be the set of all closures (§ 1, Exercise 13) in* E. *Order* F *by putting* $u \leq v$ *whenever* $u(x) \leq v(x)$ *for all* $x \in$ E. *Then* F *has a least element* e, *the identity mapping of* E *onto itself. For each* $u \in$ F, *let* I(u) *denote the set of elements of* E *which are invariant under* u.

(a) Show that $u \leq v$ *in* F *if and only if* I(v) $\subset$ I(u).

(b) Show that if every pair of elements of E *has a greatest lower bound in* E, *then every pair of elements of* F *has a greatest lower bound in* F. *If* E *is a complete lattice, then so is* F *(§ 1, Exercise 11).*

(c) Show that if E *is inductive (with respect to the relation* $\leq$*), then every pair* u, v *of elements of* F *has a least upper bound in* F *(Show that if* $f(x) = v(u(x))$ *and if* $w(x)$ *denotes the greatest element of the chain of* x, *relative to* f *(Exercise 6), then* w *is a closure in* E *and is the least upper bound of* u *and* v*.)*

Let's start with the definitions.

```
Section Exercise27.
Variable r: Set.
Hypothesis or: order r.

Definition set_of_closures  :=
  let E:=substrate r in
  Zo (set_of_functions E E) (fun z=> is_closure z r).

Definition closure_ordering  :=
  let E:=substrate r in
    induced_order (order_function E E r) (set_of_closures).
```

Let's show some trivial properties.

```
Lemma Exercise2_7a f g: (* 10 *)
  let E:=substrate r in
  gle (closure_ordering) f g <->
  (inc f (set_of_closures) & inc g (set_of_closures) &
    forall i, inc i (substrate r) -> gle r (W i f) (W i g)).
Lemma Exercise2_7b: (* 7 *)
  order closure_ordering &
  substrate closure_ordering = set_of_closures.
Lemma Exercise2_7c: (* 12 *)
  least_element (closure_ordering) (identity (substrate r)).
```

Assume I(g) $\subset$ I(f). Fix a; let $b = g(a)$. We have $a \leq b$, thus $f(a) \leq f(b)$. But $b \in$ I(g) so that $f(b) = b$, thus $f(a) \leq g(a)$. This shows $f \leq g$. Conversely, assume $a \in$ I(g). If $f \leq g$, we have $a \leq f(a) \leq g(a) = a$. This shows $a \in$ I(f).

```
Lemma Exercise2_7d f g: (* 16 *)
  inc f set_of_closures -> inc g set_of_closures ->
```

```
  (gle (closure_ordering) f g <->
      sub (set_of_invariants g) (set_of_invariants f)).
```

Consider now (b). The infimum of a family of closures, if it exists, is a closure. We start with a family of two elements.

```
Lemma Exercise2_7e f g: (* 57 *)
  (forall x y, inc x (substrate r) -> inc y (substrate r) ->
    has_infimum r (doubleton x y))
  -> inc f (set_of_closures)
  -> inc g (set_of_closures)
  -> has_infimum (closure_ordering) (doubleton f g).
Lemma Exercise2_7f: complete_lattice r -> (* 64 *)
  complete_lattice (closure_ordering).
```

We define $\mathrm{I}_x(f)$ the set of all elements y such that $x \leq y$ and $f(y) = y$. Let $\mathrm{J}(f)$ be the property that I_x has a least element, and denote this by $g(x)$. Then g is a closure and $\mathrm{I}(f) = \mathrm{I}(g)$. Assume now that u and v are two closures, $f = u \circ v$. We have $x \leq u(x) \leq u(v(x))$ and $x \leq v(x) \leq u(v(x))$, so that $\mathrm{I}(f) = \mathrm{I}(u) \cap \mathrm{I}(v)$. This implies that $g = \sup(u, v)$.

Assume E inductive. For any x, there is a maximal element y such that $x \leq y$. This element satisfies $f(y) = y$. Thus $\mathrm{I}_x(f)$ is non-empty. Assume moreover E well-ordered. Then $\mathrm{I}_x(f)$ has a least element, and any pair of closures has a supremum.

```
Definition Ixf x f :=  Zo (substrate r) (fun z => gle r x z & W z f = z).
Definition Jf f := (forall x, inc x (substrate r) -> exists y,
        least_element (induced_order r (Ixf x f)) y).

Lemma Exercise2_7g u v: (* 82 *)
  inc u (set_of_closures) -> inc v (set_of_closures) ->
   Jf (compose u v) ->
   has_supremum (closure_ordering) (doubleton u v).
Lemma Exercise2_7h u v: (* 15 *)
  inductive_set r -> worder r ->
  inc u (set_of_closures) -> inc v (set_of_closures) ->
   has_supremum (closure_ordering) (doubleton u v).

End Exercise27.
```

We shall give below an example where $\mathrm{I}(u) \cap \mathrm{I}(v)$ is empty. In this case, the pair has no upper bound (if w is an upper bound, then any non-empty set $\mathrm{I}(w)$ is a subset of the intersection). We shall also give an example of an inductive set, where there are upper bounds but no least upper bound.

Consider the ordinal sum $\mathrm{E} = \mathrm{N}_1 + \mathrm{N}_2$, where N_1 is the set of natural integers, and N_2 the set of natural integers with the reverse ordering. This is is an inductive set, as N_2 has a greatest element.

```
Definition NNstar :=
  order_sum2 Bnat_order (opposite_order Bnat_order).

Lemma Exercice2_7A1 r:  (* 1 *)
  (exists u, greatest_element r u) -> inductive_set r.
Lemma Exercice2_7A2 r r': (* 8 *)
```

```
  order r  -> order r' ->
  (exists u, greatest_element r' u)
  -> inductive_set  (order_sum2 r r').
Lemma Exercice2_7A3:  (* 11 *)
  order NNstar & substrate NNstar =  canonical_du2 Bnat Bnat
  & (forall x x', gle NNstar x x' <->
    (inc x (canonical_du2 Bnat Bnat) &
    inc x' (canonical_du2 Bnat Bnat) &
    ((Q x = TPa & Q x' = TPa & (P x) <=c (P x'))
      \/ (Q x <> TPa & Q x' <> TPa & (P x') <=c (P x))
      \/ (Q x = TPa & Q x' <> TPa)))).
Lemma Exercice2_7A4: inductive_set NNstar. (* 7 *)
```

If f is a function $\mathrm{N}_1 \mapsto \mathrm{N}_1$ it can be extended to E by putting $f(x) = x$ on N_2. If we have a closure, then the extension is a closure.

```
Definition extension_to_NNstar f :=
  BL (fun z=> Yo (Q z = TPa) (J (W (P z) f) TPa) z)
  (substrate NNstar) (substrate NNstar).
Lemma Exercice2_7A5 f: (* 14 *)
  function_prop f Bnat Bnat ->
  let g:= (extension_to_NNstar f) in let E:= (substrate NNstar) in
    ((forall x, inc x Bnat -> W (J x TPa) g = (J (W x f) TPa)) &
      (forall x, inc x E -> Q x = TPa ->
        (P (W x g)  = W (P x) f & Q (W x g) = TPa)) &
    (forall x, inc x E -> Q x <> TPa ->  W x g = x) &
    function_prop g E E).
Lemma Exercice2_7A6 f: is_closure f Bnat_order -> (* 49 *)
  is_closure  (extension_to_NNstar f) NNstar.
```

We define here some properties of even and odd numbers. We say that a number is even if the remainder in the division by two is zero. We shall use the relation $2n+1 \neq 2m$ to discriminate even and odd numbers. The successor of an even (resp. odd) number is odd (resp. even).

```
Definition even_int n := inc n Bnat & card_rem n \2c = \0c.
Definition odd_int n := inc n Bnat & ~ (even_int n).

Lemma even_odd_aux n m: (* 18 *)
  inc n Bnat -> inc m Bnat ->
  succ (n *c \2c) <> (m *c \2c).
Lemma even_odd_aux1 n m: (* 1 *)
  inc n Bnat -> inc m Bnat ->
  succ (\2c *c n) <> (\2c *c m).
Lemma even_odd_succ n: (* 28 *)
  (even_int n -> odd_int (succ n)) & (odd_int n -> even_int (succ n)).
```

Let $v(x)$ be the function defined on **N** by: if x is even, then $v(x) = x$, otherwise $v(x) = x+1$, and u the same function with "even" replaced by "odd". These are closures, and there is no upper bound. In fact, if w is an upper bound, for any x, we have $u(x) \leq w(x)$ and $v(x) \leq w(x)$. We show that this implies $x \neq w(x)$ (which is false for $x = w(0)$).

```
Lemma Exercice2_7A7: (* 1 *)
  (bl_axioms (fun z => Yo (even_int z) z (succ z)) Bnat Bnat).
Lemma Exercice2_7A8:  (* 1 *)
```

```
  (bl_axioms (fun z => Yo (even_int z) (succ z) z) Bnat Bnat).
Lemma Exercice2_7A9:  (* 15 *)
  is_closure (BL (fun z => Yo (even_int z) z (succ z)) Bnat Bnat) Bnat_order.
Lemma Exercice2_7A10:  (* 14 *)
  is_closure (BL (fun z => Yo (even_int z) (succ z) z) Bnat Bnat) Bnat_order.

Lemma Exercice2_7A11 x w: (* 5 *)
  let u :=BL (fun z => Yo (even_int z) (succ z) z) Bnat Bnat in
    let v :=BL (fun z => Yo (even_int z) z (succ z)) Bnat Bnat in
      inc x Bnat ->
      (W x u) <=c  w ->
      (W x v) <=c w ->
      x <> w.
```

The two closures u and v can be extended as u' and v' to the ordinal sum $E = N_1 + N_2$. Every upper bound w satisfies: if $w(y) = y$, then $y \in N_2$, because of Exercice2_7A11. It is easy to construct such functions: consider for instance the function f_y that maps x to x for $x \geq y$. This is a closure, and is an upper bound if $y \in N_2$.

Consider now any upper bound w. Let $k = w(0)$, where $0 \in N_1$ is the least element of E. We have $k \in N_2$, and, for $x \leq k$ we have $w(x) = k$. Let $k' = k+1$ (in N_2). We have $k' < k$. Define $f(x)$ to be k' if $x \leq k'$ and $w(x)$ otherwise. This is an upper bound and shows that w is not the least upper bound.

```
Lemma Exercice2_7A12: exists r, exists u, exists v, (* 156 *)
  order r & inductive_set r &
  inc u (set_of_closures r) & inc v (set_of_closures r)
  & ~  has_supremum (closure_ordering r) (doubleton u v).
```

¶ 8. An ordered set E is said to be *ramified* (on the right) if, for each pair of elements x, y of E such that $x < y$, there exists $z > x$ such that y and z are not comparable. E is said to be *completely ramified* (on the right) if it is ramified and has no maximal elements. Every antidirected set (§ 1, Exercise 22) is ramified.

(a) Let E be an ordered set and let a be an element of E. Let $\mathfrak{R}_a$ denote the set of ramified subsets of E which have a as least element. Show that $\mathfrak{R}_a$, ordered by inclusion, has a maximal element.

(b) If E is branched (§ 1, Exercise 24), show that every maximal element of $\mathfrak{R}_a$ is completely ramified.

(c) Given an example of a branched set which is not ramified. The branched set defined in § 1, Exercise 24 (c) is completely ramified.

(d) Let E be a set in which each interval $]\leftarrow, x]$ is totally ordered. Show that E has an antidirected cofinal subset (§ 1, Exercise 22) (use (b)).

The first point is trivial. Assume $x < y$ in an antidirected set. There exists z such that $x < z$ and the intervals $[y, \rightarrow [$ and $[z, \rightarrow [$ do not intersect. This implies that y and z are non-comparable and the set is ramified.

```
Definition is_ramified r :=
```

```
  forall x y, glt r x y -> exists z, glt r x z & ~ gle r y z&  ~ gle  r z y.

Definition is_ramified_c r :=
  is_ramified r & not (exists x, maximal_element r x).

Lemma Exercise2_8a r:  (* 5 *)
  order r -> anti_directed r -> is_ramified r.

Lemma and_distrib: forall (A B C: Prop), (* 7 *)
  ((A & B) <-> (A & C)) <-> (A -> (B <-> C)).
```

Let $\mathfrak{R}_a$ be the set of all subsets Z of E such that the induced ordering is ramified and has a as least element. We first rewrite this condition in terms of the ordering on E. This set has a maximal element, thanks to Zorn's Lemma: consider a totally ordered family A_i of elements of $\mathfrak{R}_a$. If the family is empty, it has $\{a\}$ as upper bound, otherwise it has $\bigcup A_i$ as upper bound.

```
Definition  Exercise2_8a_R r a :=
  Zo (powerset (substrate r))
  (fun z => is_ramified (induced_order r z) &
    least_element  (induced_order r z) a).

Lemma Exercise2_8b r a F: order r ->  (* 14 *)
  (inc F (Exercise2_8a_R r a) <->
  (sub F (substrate r)
    &  (forall x y, glt r x y -> inc x F -> inc y F ->
      exists z, glt r x z & ~ gle r y z&  ~ gle  r z y & inc z F)
    & inc a F
    & (forall z, inc z F -> gle r a z))).
Lemma Exercise2_8c r a: (* 31 *)
  order r -> inc a (substrate r) ->
  exists A, maximal_element (inclusion_suborder  (Exercise2_8a_R r a)) A.
```

TBC.

---

9. *An ordinal sum* $\sum_{\iota\in I} E_\iota$ *(§ 1, Exercise 3) is well-ordered if and only if* I *and each* E_ι *is well-ordered.*

One implication is orsum_wor. The other one is easy.

```
Lemma orsum_wo_pr r g:  (* 32 *)
  orsum_ax r g ->   orsum_ax2 g ->
  (forall i, inc i (domain g) -> nonempty (substrate (V i g))) ->
  (worder (order_sum r g) <->
    (worder r &  (forall i, inc i (domain g) -> worder (V i g)))).
```

---

10. *Let* I *be an ordered set and let* $(E_\iota)_{\iota\in I}$ *be a family of ordered sets, all equal to the same ordered set* E. *Show that the ordinal sum* $\sum_{\iota\in I} E_\iota$ *(§ 1, Exercise 3) is isomorphic to the lexicographic product of the sequence* $(F_\lambda)_{\lambda\in\{\alpha,\beta\}}$, *where the set* $\{\alpha,\beta\}$ *of two distinct elements is*

well-ordered by the relation whose graph is $\{(\alpha,\alpha),(\alpha,\beta),(\beta,\beta)\}$*, and where* $F_\alpha = I$ *and* $F_\beta = E$*. This product is called the* lexicographic product *of* E *by* I *and is written* E.I.

This is lemma order_prod_pr of Chapter 8.

¶ 11. * *Let* I *be a well-ordered set and let* $(E_\iota)_{\iota\in I}$ *be a family of ordered sets, each of which contains at least two distinct comparable elements. Then the lexicographic product of the* E_ι *is well-ordered if and only if each of the* E_ι *is well-ordered and* I *is* finite *(if* I *is infinite, construct a strictly decreasing infinite sequence in the lexicographic product of the* E_ι*).* *

Assume I well-ordered, and each E_i ordered. In the main text we have shown that if each E_i is totally ordered, so is the product. If each E_i is well-ordered, so is the product, provided that I is finite, proof by induction on the cardinal of I, the case I empty being trivial.

Let i be the least element of I, X be a non-empty subset of the product, and X_i the i-th projection. This is a non-empty subset of E_i and has a least element a. Let $\bar{X}$ the set of elements of X for which the i-th component is equal to a. If $x \in \bar{X}$ and $y \in X - \bar{X}$ then $x < y$. Denote by f' the restriction of f to $I - \{i\}$. We denote by $\bar{X}'$ the set of restrictions. By induction, the restriction product is well-ordered and this set has a least element x', that is the restriction of some element $x \in \bar{X}$. If $y \in \bar{X}$ then $x \leq y$ if and only if $x' \leq y'$ in the restriction product. This x is the least element of $\bar{X}$, hence of X.

We show here the converse. If one E_i is empty, so is the product, and nothing can be said of the other factors. Otherwise, E_i is isomorphic to a subset of the product, so, if the product is totally ordered, or well-ordered, so are the factors. Let J be the set of indices such that E_i is not small (has at least two elements). Assume the product well-ordered, hence totally ordered. Then E_i not small is equivalent to the condition "E_i has at least two distinct comparable elements". Consider two sequences x_i and y_i such that $x_i < y_i$ for $i \in J$ (and $x_i = y_i$ otherwise). Define $z(k)$ by $z(k)_i = x_i$, if $i < k$ and $z(k)_i = y_i$ otherwise. This is a decreasing function of k, and is strictly decreasing on J. Thus J is finite. We simplify the proof by assuming I = J, so that $z(k)$ is a strictly decreasing function.

We start with an auxiliary result: if $f : I \to J$ is strictly decreasing, both sets are well-ordered, then I is finite. Since I is totally ordered, every non-empty subset K of I has a greatest element (inverse image of the least element of $f(K)$). Since I is well-ordered, there is a morphism $\mathbf{N} \to s(I)$, or a morphism $I \to s(\mathbf{N})$, where $s(K)$ is some segment of K. Let g be the morphism. In the first case, the image has a greatest element $g(a)$. We have a contradiction since $g(a+1) > g(a)$. In the second case, the morphism may be surjective; same contradiction as above. Otherwise its range is finite and I is finite.

In the main text we have shown: if I is finite, and if each E_i is well-ordered, then the product is well-ordered. Conversely, if the product is well-ordered, either there exists i such that E_i is empty, or the set J of all indices is well-ordered. Condition orsum_ax2 says that t E_i is non-empty.

```
Lemma worder_decreasing_finite r r' f: (* 51 *)
  worder r -> worder r' ->
  (forall i, inc i (substrate r) -> inc (f i) (substrate r')) ->
  (forall i j, glt r i j -> glt r' (f j) (f i)) ->
  finite_set (substrate r).
```

```
Section Exercise2_11.
Variables (r g: Set).
Hypothesis oa:  orprod_ax r g.

Lemma orprod_total2: orsum_ax2 g ->(* 35 *)
  ((total_order (order_prod r g) ->
     (forall i, inc i (domain g) -> total_order (V i g)))
  &
  (worder (order_prod r g) -> (forall i, inc i (domain g) -> worder (V i g)))).

Lemma orprod_total3:  orsum_ax2 g  -> (* 2 *)
  ((forall i, inc i (domain g) -> total_order (V i g)) <->
   total_order (order_prod r g)).

Lemma orprod_total4:
  total_order (order_prod r g) ->
  (forall i, inc i (domain g) -> ~ (small_set (substrate (V i g)))) ->
  exists f1, exists f2,
   inc f1 (substrate (order_prod r g)) &
   inc f2 (substrate (order_prod r g)) &
   (forall i, inc i (domain g) -> glt (V i g) (V i f1) (V i f2)).

Lemma orprod_worder_bis:  (* 33 *)
  (forall i, inc i (domain g) -> ~ (small_set (substrate (V i g)))) ->
  ( ((forall i, inc i (domain g) -> worder (V i g)) &
       finite_set (substrate r))
    <-> worder (order_prod r g)).
```

¶ 12. *Let* I *be a totally ordered set and let* $(E_\iota)_{\iota\in I}$ *be a family of ordered sets indexed by* I. *Let* $R\{\!\{x,y\}\!\}$ *denote the following relation on* $E = \prod_{\iota\in I} E_\iota$*: "the set of indices* $\iota \in I$ *such that* $\mathrm{pr}_\iota x \neq \mathrm{pr}_\iota y$ *is well-ordered, and if* κ *is the least element of this subset of* I*, we have* $\mathrm{pr}_\kappa x < \mathrm{pr}_\kappa y$*". Show that* $R\{\!\{x,y\}\!\}$ *is an order relation between* x *and* y *on* E*. If the* E_ι *are totally ordered, show that the connected components of* E *with respect to the relation "*x *and* y *are comparable" (Chapter II, § 6, Exercise 10) are totally ordered sets. Suppose that each* E_ι *has at least two elements. Then* E *is totally ordered if and only if* I *is well-ordered and each* E_ι *is totally ordered (use Exercise 3); and* E *is then the lexicographic product of the* E_ι*.*

Assume that E is is a totally ordered set, A and B are two well-ordered sets of E. Let C be a subset of $A \cup B$; it is well-ordered. Consider a non-empty subset X. If X does not intersection both A and B it is contained in one of them, so X has a least element. Otherwise, the intersections have a least element a and b. Then $\inf(a,b)$ is the least element of X.

```
Definition olex_nsv r x y:=
  Zo (substrate r) (fun i =>  (V i x <> V i y)).
Definition olex_io r x y:=   (induced_order r (olex_nsv r x y)).

Definition olex_comp1_r r g x y :=
  worder (olex_io r x y) &
  let i := the_least_element (olex_io r x y) in glt (V i g) (V i x) (V i y).

Definition olex_comp2_r r g x y :=
```

```
  (inc x (prod_of_substrates g))
  & (inc y (prod_of_substrates g))
  & (x = y \/ olex_comp1_r r g x y).

Definition olex r g := graph_on  (olex_comp2_r r g) (prod_of_substrates g).
Definition olex_ax r g:= total_order r & substrate r = domain g & order_fam g.

Lemma union2_wor r A B C:  (* 38 *)
  total_order r -> sub A (substrate r) ->  sub B (substrate r) ->
  sub C (union2 A B) ->
  worder (induced_order r A) ->worder (induced_order r B) ->
  worder (induced_order r C).
Lemma olex_nsvS r x y: olex_nsv r x y = olex_nsv r y x.  (* 1 *)
Lemma olex_ioS r x y: olex_io r x y = olex_io r y x.  (* 1 *)
```

Let T_{xy} be the set of indices i such that $x_i \neq y_i$. If T_{xy} and T_{yz} are well-ordered so is T_{yz}. Let a, b and c be the least element of these sets. We have $c = \inf(a,b)$, and this shows that $R\{x,y\}$ is transitive, thus is an order relation.

```
Section Olex_basic.
Variables (r g: Set).
Hypothesis ax: olex_ax r g.

Lemma olex_R x:  (* 1 *)
  inc x (prod_of_substrates g) -> olex_comp2_r r g x x.
Lemma olex_sr: (* 1 *)
  substrate (olex r g) = (prod_of_substrates g).
Lemma olex_gle x y:  (* 1 *)
  gle (olex r g) x y <-> olex_comp2_r r g x y.
Lemma olex_nsve x y:  (* 9 *)
   inc x (prod_of_substrates g) -> inc y (prod_of_substrates g) ->
   ((x = y) <-> (olex_nsv r x y = emptyset)).
Lemma olex_nsve1 x y: (* 8 *)
  inc x (prod_of_substrates g) -> inc y (prod_of_substrates g) ->
  let r' := (olex_io r x y) in let i := the_least_element r' in
    x <> y -> worder r' -> least_element r' i.

Lemma olex_glt_aux x y: glt (olex r g) x y  ->  (* 13 *)
  let r' := (olex_io r x y) in
      let i := the_least_element r' in
        (inc i (substrate r) & forall j, glt r j i -> V j x = V j y).
Lemma olex_or: order  (olex r g).  (* 78 *)
```

Consider now the second point. The connected component of x is the class of x for the equivalence relation $x \equiv y$: there is a chain for $a \sim b$ (short for $a \leq b$ or $b \leq a$). In order to show that the connected component is totally ordered, it suffices to show that $x \equiv y$ implies that x and y are comparable. We proceed by induction on the length of the chain. If the chain contains only x and y, then $x \sim y$, and the result is obvious. Assume x is chained to z and $z \sim y$. By induction $x \sim y$. Note that $x \sim y$ is the same as: the set of indices such that $x_i \neq y_i$ is well-ordered; and, if this set is non-empty and has a least element j, then x_j and y_j are comparable. If all E_ι are totally ordered, this second condition becomes trivial, and $x \sim y$ is thus transitive.

Assume I well-ordered. The condition that T_{xy} is well-ordered becomes trivial. In that case, the ordering on E is the lexicographic ordering, so that E is totally ordered provided

that each E_i is totally ordered. Conversely, assume E totally ordered and each E_i has at least two elements. Consider two elements x and y in E such that $x_i \neq y_i$ whatever i. If $x = y$, then I is empty. Otherwise, to say that x and y are comparable implies I well-ordered. Then the ordering on E is the lexicographic ordering and each E_i is totally ordered.

```
Definition all_total g :=
 (forall i, inc i (domain g) -> total_order (V i g)).

Lemma olex_cc_comparable1  (r':= olex r g):  (* 39 *)
  all_total g ->
  forall x y,
      ocomparable r' x y <-> (inc x (substrate r') & inc y (substrate r') &
          worder (olex_io r x y)).
Lemma olex_cc_comparable2  (r':= olex r g): (* 11 *)
  all_total g -> transitive_r (ocomparable r').

Lemma olex_cc_tor (r':= olex r g):  (* 29 *)
  all_total g ->
  forall x, inc x (substrate r') ->
   total_order (induced_order r'
        (Exercice1.connected_comp  (ocomparable r') (substrate r') x)).


Lemma olex_lex: worder r -> olex r g = order_prod r g. (* 22 *)
Lemma olex_total1: worder r (* 2 *)
  -> all_total g -> total_order (olex r g).
Lemma olex_total2:  (* 49 *)
  (total_order (olex r g)
  -> (forall i, inc i (domain g) -> ~ (small_set (substrate (V i g))))
  -> (worder r & all_total g)).
End Olex_basic.
```

---

13. *(a) Let* $Is(\Gamma,\Gamma')$ *be the relation "*Γ *is an ordering (on* E*) and* Γ' *is an ordering (on* E'*), and there exists an isomorphism of* E*, ordered by* Γ*, onto* E'*, ordered by* Γ'*". Show that* $Is(\Gamma,\Gamma')$ *is an equivalence relation on every set whose elements are orderings. The term* $\tau_\Delta(Is(\Gamma,\Delta))$ *is an ordering called the* order-type *of* Γ *and denoted by* $\mathrm{Ord}(\Gamma)$*, or* $\mathrm{Ord}(E)$ *by abuse of notations. Two ordered sets are isomorphic if and only if their order-types are equal.*

(b) Let $R\{\lambda,\mu\}$ *be the relation: "*λ *is an order-type, and* μ *is an order-type and there exists an isomorphism of the set ordered by* λ *onto a subset of the set ordered by* μ*". Show that* $R\{\lambda,\mu\}$ *is a preorder relation between* λ *and* μ*. It will be denoted by* $\lambda \prec \mu$.

(c) Let I *be an ordered set and let* $(\lambda_\iota)_{\iota\in I}$ *be a family of order-types indexed by* I*. The order-type of the ordinal sum (§ 1, Exercise 3) of the family of sets ordered by the* λ_ι $(\iota \in I)$ *is called the* ordinal sum *of the order-types* λ_ι $(\iota \in I)$ *and is denoted by* $\sum_{\iota\in I}\lambda_\iota$*. If* $(E_\iota)_{\iota\in I}$ *is a family of ordered sets, the order type of* $\sum_{\iota\in I}E_\iota$ *is* $\sum_{\iota\in I}\mathrm{Ord}(E_\iota)$*. If* I *is the ordinal sum of a family* $(J_\kappa)_{\kappa\in K}$*, show that*

$$\sum_{\kappa\in K}\Big(\sum_{\iota\in J_\kappa}\lambda_\iota\Big) = \sum_{\iota\in I}\lambda_\iota.$$

(d) Let I *be a well-ordered set and* $(\lambda_\iota)_{\iota\in I}$ *be a family of order-types indexed by* I. *The order-type of the lexicographic product of the family of sets indexed by by the* λ_ι $(\iota \in I)$ *is called the* ordinal product *of the order-types* λ_ι $(\iota \in I)$ *and is denoted by* $\prod_{\iota\in I}\lambda_\iota$. *If* $(E_\iota)_{\iota\in I}$ *is a family of ordered sets, the order type of the lexicographic product of the family* $(E_\iota)_{\iota\in I}$ *is* $\prod_{\iota\in I}\mathrm{Ord}(E_\iota)$. *If* I *is the ordinal sum of a family of well-ordered sets* $(J_\kappa)_{\kappa\in K}$ *indexed by a well-ordered set* K, *show that*

$$\prod_{\kappa\in K}\Big(\prod_{\iota\in J_\kappa}\lambda_\iota\Big)=\prod_{\iota\in I}\lambda_\iota.$$

(e) We denote by $\lambda+\mu$ *(resp.* $\mu\lambda$*) the ordinal sum (resp. ordinal product) of the family* $(\xi_\iota)_{\iota\in J}$ *where* $J=\{\alpha,\beta\}$ *is a set with two distinct elements, ordered by the relation whose graph is* $\{(\alpha,\alpha),(\alpha,\beta),(\beta,\beta)\}$, *and where* $\xi_\alpha=\lambda$ *and* $\xi_\beta=\mu$. *Show that if* I *is a well-ordered set of order-type* λ *and if* $(\mu_\iota)_{\iota\in I}$ *is a family of order-types such that* $\mu_\iota=\mu$ *for each* $\iota\in I$ *then* $\sum_{\iota\in I}\mu_\iota=\mu\lambda$. *We have* $(\lambda+\mu)+\nu=\lambda+(\mu+\nu)$, $(\lambda\mu)\nu=\lambda(\mu\nu)$, *and* $\lambda(\mu+\nu)=\lambda\mu+\lambda\nu$ *(but in general* $\lambda+\mu\neq\mu+\lambda$, $\lambda\mu\neq\mu\lambda$ *and* $(\lambda+\mu)\nu\neq\lambda\nu+\mu\nu$*).*

(f) Let $(\lambda_\iota)_{\iota\in I}$ *and* $(\mu_\iota)_{\iota\in I}$ *be two families of order-types indexed by the same ordered set* I. *Show that if* $\lambda_\iota\prec\mu_\iota$ *for each* $\iota\in I$, *then* $\sum_{\iota\in I}\lambda_\iota\prec\sum_{\iota\in I}\mu_\iota$ *and (if* I *is well-ordered)* $\prod_{\iota\in I}\lambda_\iota\prec\prod_{\iota\in I}\mu_\iota$. *If* J *is a subset of* I, *show that* $\sum_{\iota\in J}\lambda_\iota\prec\sum_{\iota\in I}\lambda_\iota$ *and (if* I *is well-ordered and the* λ_ι *are non-empty)* $\prod_{\iota\in J}\lambda_\iota\prec\prod_{\iota\in I}\lambda_\iota$.

(g) Let λ^* *denote the order-type of the set ordered by the opposite of the ordering* λ. *Then we have*

$$(\lambda^*)^*=\lambda\qquad\text{and}\qquad\Big(\sum_{\iota\in I}\lambda_\iota\Big)^*=\sum_{\iota\in I^*}\lambda_\iota^*$$

where I^* *denotes the set* I *endowed with the opposite of the ordering given in* I.

Point (a). Lemmas orderIR, orderIS and orderIT show that the relation Is is an equivalence relation. The order-type is defined via an axiom in section 8.22.

Point (b). Lemmas OT_order_le_reflexive and OT_order_le_transitive show that R is a preorder.

Points (c) and (d). The first claim corresponds to lemma OT_sum_invariant3, and the second to OT_sum_assoc1. There are similar results in the case of a product. These lemmas will be specialized to ordinal numbers as orsum_invariant3 and osum_assoc1.

Point (e). The first result is OT_prod_pr1. It will be specialized to ordinals as oprod_pr1. The other properties are OT_sum_assoc3, OT_prod_assoc3, OT_sum_distributive3. There are similar results in the case of ordinals.

Point (f). The first result is OT_sum_increasing2; in the case of ordinals, the result becomes osum_increasing2. There are three other results.

Point (g) is implemented as OT_double_opposite and OT_opposite_sum.

¶ 14. *An* ordinal *is the order-type of a well-ordered set (Exercise 13).*

(a) Show that, if $(\lambda_\iota)_{\iota\in I}$ *is a family of ordinals indexed by a well-ordered set* I, *then the ordinal sum* $\sum_{\iota\in I}\lambda_\iota$ *is an ordinal;* **and that, if moreover,* I *is finite, then the ordinal product* $\prod_{\iota\in I}\lambda_\iota$ *is an ordinal (Exercise 11).*$_*$ *The order-type of the empty set is denoted by 0, and that of*

a set with one element by 1 (by abuse of language, cf § 3). Show that

$$\alpha + 0 = 0 + \alpha = \alpha \qquad \textit{and} \qquad \alpha.1 = 1.\alpha = \alpha$$

for every ordinal α.

(b) Show that the relation "λ *is an ordinal and* μ *is an ordinal and* $\lambda \prec \mu$*" is a* well-ordering *relation, denoted by* $\lambda \leq \mu$ *(Note that, if* λ *are* μ *are ordinals, the relation* $\lambda \prec \mu$ *is equivalent to "*λ *is equal to the order-type of a segment of* μ*" (no. 5, Theorem 3, Corollary 3): given a family* $(\lambda_\iota)_{\iota \in I}$ *of ordinals, consider a well-ordering in* I *and take the ordinal sum of the family of sets ordered by the* λ_ι*; finally use Proposition 2 of no 1.).*

(c) Let α *be an ordinal. Show that the relation "*ξ *is an ordinal and* $\xi \leq \alpha$*" is collectivizing in* ξ*, and that the set* O_α *of ordinals* $< \alpha$ *is a well-ordered set such that Ord*$(O_\alpha) = \alpha$*. We shall often identify* O_α *with* α*.*

(d) Show that for every family of ordinals $(\xi_\iota)_{\iota \in I}$ *there exists a unique ordinal* α *such that the relation "*λ *is an ordinal and* $\xi_\iota \leq \lambda$ *for all* $\iota \in I$*" is equivalent to* $\alpha \leq \lambda$*. By abuse of language,* α *is called the* least upper bound *of the family of ordinals* $(\xi_\iota)_{\iota \in I}$*, and we write* $\alpha = \sup_{\iota \in I} \xi_\iota$ *(it is the greatest element of the union of* $\{\alpha\}$ *and the set of the* ξ_ι*). The least upper bound of the set of ordinals* $\xi < \alpha$ *is either* α *or an ordinal* β *such that* $\alpha = \beta + 1$*. In the latter case* β *is said to be the* predecessor *of* α*.*

Point (a). The first two properties are OS_sum2 and OS_prod2. We have also osum0r, and variants.

Point (b). Lemma OT_ordinal_compat says that if x is a well-ordering, then $o(\mathrm{ord}(x))$ and x have the same order-type. We may identity $\mathrm{Ord}(x)$ with $o(\mathrm{ord}(x))$ when x is well-ordered. This means that $\leq_{\mathrm{Ord}}$ and $\leq_{\mathrm{ord}}$ share the same properties, where $\leq_{\mathrm{Ord}}$ is the relation introduced by Bourbaki, and $x \leq_{\mathrm{ord}} y$ the relation: x and y are two von Neumann ordinals such that $x \subset y$. Point (b) becomes the lemma wordering_ordinal_le.

Point (c). The lemmas set_ord_le_rw and set_ord_lt_rw say that for any ordinal α, the relations $x \in \alpha^+$ and $x \in \alpha$ are equivalent to $x \leq \alpha$ and $x < \alpha$. Lemma set_ord_lt_prop3 says that the ordinal of the set of $x < \alpha$, ordered by $\leq$ is α. This $O_\alpha = \alpha$ and $O'_\alpha = \alpha^+$.

Point (d). Consider a family $(\xi_i)_{i \in I}$ of ordinals and the set E of all ξ_i. The union of E, sometimes denoted sup(E), is the unique λ such that "λ is an ordinal and $x \leq \lambda$ for all $x \in E$". We have obviously $\sup(E) = \sup(\xi_i)$. We show here that it is the greatest element of the union of E and $\{\sup(E)\}$.

```
Lemma ord_sup_pr6 E: ordinal_set E -> (* 9 *)
  greatest_element (graph_on ordinal_le (tack_on E (\osup E)))
    (\osup E).
```

---

15. *(a) Let* α *and* β *be two ordinals. Show that the inequality* $\alpha < \beta$ *is equivalent to* $\alpha + 1 \leq \beta$*, and that it implies the inequalities* $\xi + \alpha < \xi + \beta$*,* $\alpha + \xi \leq \beta + \xi$*,* $\alpha\xi \leq \beta\xi$ *for all ordinals* ξ *and* $\xi\alpha < \xi\beta$ *if* $\xi > 0$*.*

(b) Deduce from (a) that there exists no set to which every ordinal belongs (use Exercise 14 (d)).

(c) Let α, β, μ *be three ordinals. Show that each of the relations* $\mu + \alpha < \mu + \beta$*,* $\alpha + \mu < \beta + \mu$ *implies* $\alpha < \beta$*; and that each of the relations* $\mu\alpha < \mu\beta$*,* $\alpha\mu < \beta\mu$ *implies* $\alpha < \beta$ *provided that* $\mu > 0$*.*

(d) Show that the relation $\mu+\alpha=\mu+\beta$, *implies* $\alpha=\beta$, *and that* $\mu\alpha=\mu\beta$, *implies* $\alpha=\beta$ *provided that* $\mu>0$.

(e) Two ordinals α *and* β *are such that* $\alpha\leq\beta$ *if and only if there exists an ordinal* ξ *such that* $\beta=\alpha+\xi$. *This ordinal is then unique and is such that* $\xi\leq\beta$; *it is written* $(-\alpha)+\beta$.

(f) Let α, β, ζ *be three ordinals such that* $\zeta<\alpha\beta$. *Show that there exists two ordinals* ξ, η *such that* $\zeta=\alpha\eta+\xi$ *and* $\xi<\alpha$, $\eta<\beta$ *(cf. No. 5, Theorem 3, Corollary 3). Moreover,* ξ *and* η *are uniquely determined by these conditions.*

Point (a). The first claim is ord_succ_lt2. See relations (8c) and (8d) in Chapter 8.

Point (b) is ordinal_not_collectivizing.

Point (c) is implemented in (8e); The "provided $\mu>0$" is obviously superfluous.

Point (d) is osum2_simpl and oprod2_simpl.

Point (e) is odiff_pr.

Point (f) is odivision_exists and ord_division_exists.

¶ 16. *An ordinal* $\rho>0$ *is said to be* indecomposable *if there exists no pair of ordinals* ξ, η *such that* $\xi<\rho$, $\eta<\rho$ *and* $\xi+\eta=\rho$.

(a) An ordinal ρ *is indecomposable if and only if* $\xi+\rho=\rho$ *for every ordinal* ξ *such that* $\xi<\rho$.

(b) If $\rho>1$ *is an indecomposable ordinal and if* α *is any ordinal* >0, *then* $\alpha\rho$ *is indecomposable, and conversely (use Exercise 15(f)).*

(c) If ρ *is indecomposable and if* $0<\alpha<\rho$, *then* $\rho=\alpha\xi$, *where* ξ *is an indecomposable ordinal (use Exercise 15(f)).*

(d) Let α *be an ordinal* >0. *Show that there exists a greatest indecomposable ordinal among the indecomposable ordinals* $\leq\alpha$ *(consider the decomposition* $\alpha=\rho+\xi$, *where* ρ *is indecomposable).*

(e) If E *is a set of indecomposable ordinals, deduce from (d) that the least upper bound of* E *(Exercise 14(d)) is an indecomposable ordinal.*

Point (a) (b) and (c) are lemmas indecomposable_prop, indecomposable_prod, and indecomposable_division.

Point (d). The hint is strange. One might consider the least ξ such that there is an indecomposable ρ such that $\alpha=\rho+\xi$. However, this does not give the greatest ρ, since we cannot simplify on the right. The proof is the following; consider the Cantor Normal Form of α, and its degree n. Then ω^n is the greatest power of ω that is $\leq\alpha$. The conclusion follows as indecomposable ordinals are powers of ω (see Exercise 12 of no 6 below).

¶ 17. *Given an ordinal* α_0, *a term* $f(\xi)$ *is said to be an* ordinal functional symbol (with respect to ξ) defined for $\xi\geq\alpha_0$ *if the relation "*ξ *is an ordinal and* $\xi\geq\alpha_0$*" implies the relation "*$f(\xi)$ *is an ordinal"; $f(\xi)$ is said to be* normal *if the relation* $\alpha_0\leq\xi<\eta$ *implies* $f(\xi)<f(\eta)$ *and if for each family* $(\xi_\iota)_{\iota\in I}$ *of ordinals* $\geq\alpha_0$ *we have* $\sup_{\iota\in I} f(\xi_\iota)=f(\sup_{\iota\in I}\xi_\iota)$ *(cf. Exercise 14(d)).*

(a) Show that for each ordinal $\alpha > 0$, $\alpha + \xi$ *and* $\alpha\xi$ *are ordinal functional symbols defined for* $\xi \geq 0$ *(use Exercise 15(f)).*

(b) Let $w(\xi)$ *be an ordinal functional symbol defined for* $\xi \geq \alpha_0$ *such that* $w(\xi) \geq \xi$ *and such that* $\alpha_0 \leq \xi < \eta$ *implies* $w(\xi) < w(\eta)$. *Also let* $g(\xi,\eta)$ *be a term such that the relation "*ξ *and* η *are ordinals* $\geq \alpha_0$*" implies the relation* $g(\xi,\eta)$ *is an ordinal such that* $g(\xi,\eta) > \xi$*". Define a term* $f(\xi,\eta)$ *with the following properties: (1) for each ordinal* $\xi \geq \alpha_0$, $f(\xi,1) = w(\xi)$*; for each ordinal* $\xi \geq \alpha_0$ *and each ordinal* $\eta > 1$, $f(\xi,\eta) = \sup_{0<\zeta<\eta} g(f(\xi,\zeta),\xi)$ *(use Criterion C60 of no. 2). Show that if* $f_1(\xi,\eta)$ *is another term with these properties then* $f(\xi,\eta) = f_1(\xi,\eta)$ *for all* $\xi \geq \alpha_0$ *and all* $\eta \geq 1$. *Prove that, for each ordinal* $\xi \geq \alpha_0$, $f(\xi,\eta)$ *is a normal functional symbol with respect to* η *(defined for* $\eta \geq 1$*). Show that* $f(\xi,\eta) \geq \xi$ *for all* $\eta \geq 1$ *and* $\xi \geq \alpha_0$ *and that* $f(\xi,\eta) \geq \eta$ *for all* $\xi \geq \sup(\alpha_0, 1)$ *and* $\eta \geq 1$. *Furthermore, for each pair* (α,β) *of ordinals such that* $\alpha > 0$, $\alpha \geq \alpha_0$ *and* $\beta \geq w(\alpha)$ *there exists a unique ordinal* ξ *such that*

$$f(\alpha,\xi) \leq \beta < f(\alpha,\xi+1),$$

and we have $\xi \leq \beta$.

(c) If we take $\alpha_0 = 0$, $w(\xi) = \xi + 1$, $g(\xi,\eta) = \xi + 1$ *then* $f(\xi,\eta) = \xi + \eta$. *If we take* $\alpha_0 = 1$, $w(\xi) = \xi$, $g(\xi,\eta) = \xi + \eta$ *then* $f(\xi,\eta) = \xi\eta$.

(d) Show that if the relations $\alpha_0 \leq \xi \leq \xi'$, $\alpha_0 \leq \eta \leq \eta'$ *imply* $g(\xi,\eta) \leq g(\xi',\eta')$*, then the relations* $\alpha_0 \leq \xi \leq \xi'$, $1 \leq \eta \leq \eta'$ *imply* $f(\xi,\eta) \leq f(\xi',\eta')$. *If the relations* $\alpha_0 \leq \xi \leq \xi'$, $\alpha_0 \leq \eta < \eta'$ *imply* $g(\xi,\eta) < g(\xi,\eta')$ *and* $g(\xi,\eta) \leq g(\xi',\eta)$*, then the relations* $\alpha_0 \leq \xi < \xi'$ *and* $\eta \geq 0$ *imply* $f(\xi,\eta+1) < f(\xi',\eta+1)$.

(e) Suppose that $w(\xi) = \xi$ *and that the relations* $\alpha_0 \leq \xi \leq \xi'$, $\alpha_0 \leq \eta < \eta'$ *imply* $g(\xi,\eta) < g(\xi,\eta')$ *and* $g(\xi,\eta) \leq g(\xi',\eta)$. *Suppose moreover, that for each* $\xi \geq \alpha_0$, $g(\xi,\eta)$ *is a normal functional symbol with respect to* η *(defined for* $\eta \geq \alpha_0$*), and that, whenever* $\xi \geq \alpha_0$, $\eta \geq \alpha_0$, *and* $\zeta \geq \alpha_0$, *we have the associativity relation*

$$g(g(\xi,\eta),\zeta) = g(\xi,g(\eta,\zeta)).$$

Show that, if $\xi \geq \alpha_0$, $\eta \geq 1$, *and* $\zeta \geq 1$, *we have then*

$$g(f(\xi,\eta),f(\xi,\zeta)) = f(\xi,\eta+\zeta)$$

("distributivity" of g *with respect to* f*) and*

$$f(f(\xi,\eta),\zeta) = f(\xi,\eta\zeta)$$

("associativity" of f*).*

Point (a) is osum_normal and oprod_normal.

Point (b): existence of f is given by ord_induction_exists and uniqueness ord_induction_unique. It is a normal function by ord_induction_p10. Existence and uniqueness of β is given by ord_induction_p11 and ord_induction_p12.

Point (c) is ord_induction_p13 and ord_induction_p14.

Point (d) is ord_induction_p15 and ord_induction_p17.

Point (e) is ord_induction_p18 and ord_induction_p19.

¶ 18. *In the definition procedure defined in Exercise 17 (b), take* $\alpha_0 = 1+1$ *(denoted by 2 by abuse of language),*

$$w(\xi) = \xi, \qquad g(\xi,\eta) = \xi\eta.$$

Denote $f(\xi,\eta)$ *by* ξ^η *and define* α^0 *to be 1 for all ordinals* α*. Also define* 0^β *to be 0 and* 1^β *to be 1 for all ordinals* $\beta \geq 1$*.*

(a) Show that if $\alpha > 1$ *and* $\beta < \beta'$*, we have* $\alpha^\beta < \alpha^{\beta'}$*, and that, for each ordinal* $\alpha > 1$*,* α^ξ *is a normal functional symbol with respect to* ξ*. Moreover, if* $0 < \alpha \leq \alpha'$*, we have* $\alpha^\beta \leq \alpha'^\beta$*.*

(b) Show that $\alpha^\xi.\alpha^\eta = \alpha^{\xi+\eta}$ *and* $(\alpha^\xi)^\eta = \alpha^{\xi\eta}$*.*

(c) Show that if $\alpha \geq 2$ *and* $\beta \geq 1$*,* $\alpha^\beta \geq \alpha\beta$*.*

(d) For each pair of ordinals $\beta \geq 1$ *and* $\alpha \geq 2$*, there exists three ordinals* ξ*,* γ*,* δ *such that* $\beta = \alpha^\xi\gamma + \delta$ *where* $0 < \gamma < \alpha$ *and* $\delta < \alpha^\xi$*, and these ordinals are uniquely by these conditions.*

Point (a) is opow_increasing2, opow_normal, opow_increasing4.

Point (b) is opow_sum and opow_prod.

Point (c) is opow_increasing5.

Point (d) is ord_ext_div_unique and ord_ext_div_exists.

19. * *Let* α *and* β *be two ordinals and let* E *and* F *be two well-ordered sets such that Ord*(E) = α *and Ord*(F) = β*. In the set* E^F *of mappings of* F *into* E *consider the subset* G *of mappings g such that* $g(y)$ *is equal to the least element of* E *for all but a* finite *number of elements* $y \in F$*. If* F^* *is the ordered set obtained by endowing* F *with the opposite order, show that* G *is a connected component with respect to the relation "x and y are comparable" (Chapter II, § 6, Exercise 10) in the product* E^{F^*} *endowed with the ordering defined in Exercise 12, and show that* G *is well-ordered. Furthermore, prove that Ord*(G) = α^β *(use the uniqueness property of Exercise17 (b)).* $_*$

We consider two orderings r and r' corresponding to E and F respectively. Let $\bar{r}'$ be the opposite ordering of r'. Consider the constant function defined on F that maps any element to r. If we apply the construction of Exercise 12, we get an ordering on E^F, the set of graphs of functions from F to E, that satisfies some properties. Let m be the least element of E, and $\bar{m}$ the constant function $F \to E$ with value m, and I_x the set of indices i such that x_i is not m. Let G be the set of functions $x : F \to E$ for which I_x is finite. We consider the ordering induced on G by that of E^F.

```
Section OlexPowBasic.
Variables (r r': Set).
Hypotheses (wor: worder r) (wor':worder r').
Definition olexp_g := cst_graph (substrate r') r.
Definition olexp_lE := the_least_element r.
Definition olexp' := olex (opposite_order r') olexp_g.
Definition olexp_I x := Zo (substrate r') (fun i => (V i x <> olexp_lE)).
Definition olexp_G:= Zo (set_of_gfunctions (substrate r') (substrate r))
  (fun x => finite_set (olexp_I x)).
Definition olexp := induced_order olexp'  olexp_G.

Lemma olexp_ax: olex_ax (opposite_order r') olexp_g.  (* 6 *)
Lemma olexp'_sr:  (* 3 *)
  substrate olexp' = set_of_gfunctions (substrate r') (substrate r).
Lemma olexp'_order: order olexp'. (* 1 *)
Lemma olexp_gleh x y:  (* 3 *)
```

```
 (induced_order (opposite_order r') (olex_nsv r' x y)) =
  olex_io (opposite_order r')  x y.

Lemma olexp'_gle x y: (* 20 *)
  gle olexp' x y <->
  (inc x (set_of_gfunctions (substrate r') (substrate r))
   & inc y (set_of_gfunctions (substrate r') (substrate r))
   & x = y \/
     let T := olex_nsv r' x y in
     let r'' := induced_order (opposite_order r') T in
       worder r'' &
       let i := the_least_element r'' in glt r (V i x) (V i y)).
```

We show here some trivial properties. If x and y are in G then T_{xy} is finite, hence well-ordered. We can then simplify the order relation on G. In fact $x < y$ means that there is j such that $x_j < y_j$, and $x_i = y_i$ whenever $j < i$. The set G is ordered by olexp. This is a total order (see olex_cc_comparable).

Assume F empty. Then G has a single element, the empty graph. Assume F non-empty; if E is empty, then G is empty. Assume E non-empty, so that E has a least element m, and $\bar{m}$ is the least element of G. Let's show that G is the connected component of $\bar{m}$. Note that $g \in G$ is comparable with m since $m \leq g$. On the other hand, consider an element in the connected component, and a chain x_i. We prove, by induction on the length of the chain, that each x_i is in G. All we need to show is that, if $x \in G$, x and y are comparable, then $y \in G$. This holds because T_{xy} is well-ordered for the restriction of the opposite order of a well-order, thus is finite.

```
Lemma olexp_Gs: sub olexp_G (substrate olexp'). (* 1 *)
Lemma olexp_lEp: nonempty (substrate r) ->  (* 3 *)
  (inc olexp_lE (substrate r)
  & (forall x, inc x (substrate r) -> gle r olexp_lE x)).

Lemma olexp_Gxy x y: inc x olexp_G -> inc y olexp_G -> (* 8 *)
  finite_set (olex_nsv (opposite_order r') x y).
Lemma olexp_Gxy1 x y: inc x olexp_G -> inc y olexp_G -> (* 8 *)
  worder (olex_io (opposite_order r') x y).
Lemma olexp_gle1 x y: inc x olexp_G -> inc y olexp_G -> (* 43 *)
  (gle olexp' x y <->
   ( x = y \/
     (exists j,
        inc j (substrate r') &
        glt r (V j x) (V j y)
      & forall i, glt r' j i -> V i x = V i y))).

Lemma olexp_sr: substrate olexp = olexp_G.  (* 1 *)
Lemma olexp_order: order olexp.  (* 1 *)
Lemma olexp_gle x y: (* 6 *)
 gle olexp x y <-> (inc x olexp_G & inc y olexp_G &    ( x = y \/
     (exists j,
        inc j (substrate r') &
        glt r (V j x) (V j y)
      & forall i, glt r' j i -> V i x = V i y))).
Lemma olexp_total: total_order olexp. (* 16 *)

Lemma olex_Fe: (substrate r' = emptyset) ->  is_singleton olexp_G. (* 1 *)
```

```
Lemma olex_nFe_Ee: (* 4 *)
  (substrate r' <> emptyset) -> (substrate r = emptyset) ->
   olexp_G = emptyset.
Lemma olexp_G_least  (m:= cst_graph (substrate r') olexp_lE):  (* 18 *)
  nonempty (substrate r) -> least_element olexp m.
Lemma olex_G_cc (* 69 *)
  (m:= cst_graph (substrate r') olexp_lE)
  (comp:= ocomparable olexp')
  (G := (Exercice1.connected_comp comp (substrate olexp') m)) :
  nonempty (substrate r) ->
  olexp_G = G.
```

Let's show that G is well-ordered. Consider a non-empty subset of X of G. We proceed by contradiction, assuming X has no least element. Let Y be any subset of X such that (a) $Y \subset X$, (b) Y is non-empty and (c) if $x \in Y$ and $y \in X - Y$ then $x < y$. In particular, Y has no least element. Consider a subset A of F such that (d) whenever x and y are in Y and $i \in A$, then $x_i = y_i$ and (e) that if $i \in I_x$ then either $i \in A$ or $i < j$ for all $j \in A$. We define J_x to be the set $I_x - A$. This set cannot be empty, for otherwise x would be the least element of Y. Since J_x is finite, it has a greatest element M_x. Let α be the least element of the form M_x for $x \in Y$. Let B be the set of all $x \in Y$ for which M_x is α, and C the set of all x_α for $x \in B$. This is a non-empty set, thus has a least element, β. Let Y' be the set of all elements $x \in B$ such that $x_\alpha = \beta$, and $A' = A \cup \{\alpha\}$. If $x \in Y'$ and $y \in Y - Y'$ then $x < y$. It follows that Y' and A' satisfy conditions (a) to (e). We construct by induction a sequence (Y_k, A_k) starting with $Y_0 = X$ and $A_0 = \emptyset$. This gives us a sequence α_k. We have $\alpha_k \in A_{k+1}$, and the relation "$\alpha < i$ for all $i \in A$" implies $\alpha_k < \alpha_n$ for $n < k$. Thus, the sequence α_k is strictly decreasing, in a well-ordered set, absurd.

```
Lemma olexp_worder: worder olexp. ( * 254 *)
End OlexPowBasic.
```

By abuse of notations, this ordering will be denoted E^F. if we replace E and F by isomorphic set, we get isomorphic powers. If a and b are two ordinals, $E = o(a)$ and $F = o(b)$, then we may consider the ordinal ord(E^F). We shall denote it a^b by abuse of notations.

```
Definition ord_powa x y := ordinal (olexp (ordinal_o x) (ordinal_o y)).


Lemma image_of_inf r r' f: worder r -> worder r' ->  (* 13 *)
  order_isomorphism f r r' -> nonempty (substrate r) ->
  W (the_least_element r) f = the_least_element r'.
Lemma fct_co_simpl_right f1 f2 g: (* 7 *)
  f1 \coP g -> f2 \coP g -> bijection g -> f1 \co g = f2 \co g -> f1 = f2.
Lemma fct_co_simpl_left f1 f2 g: (* 7 *)
  g \coP f1 -> g \coP f2 -> bijection g ->g \co f1 = g \co f2 -> f1 = f2.


Lemma OS_ord_powa a b: is_ordinal a -> is_ordinal b -> (* 1 *)
  is_ordinal (ord_powa a b).
Lemma opoxwa_invariant r1 r2 r3 r4:  (* 195 *)
   worder r1 -> worder r2 -> worder r3 -> worder r4 ->
   r1 \Is r2 -> r3 \Is r4 ->
   (olexp r1 r3) \Is (olexp r2 r4).
```

TODO. Sow that a^b coincides with the ordinal power if $a = 0$, $a = 1$, or $b = 0$. Show that a^b is increasing in b. Show that $a^b a = a^{b+1}$. Show that $a^b = \sup_{0<c<b}(a^c.a)$. We can then

use uniqueness. Note if b is a successor, say $c+1$, the sup is obtained at c, and the result is trivial. If b is a limit ordinal, the sup is, after simplification, $\sup_{0<c<b}(a^c)$. It is $\leq a^b$. Thus, the non-trivial point is to show that $a^b \leq \sup a^c$.

¶ 20. *A set* X *is said to be* transitive *if the relation* $x \in X$ *implies* $x \subset X$.

(a) If Y *is a transitive set, then so is* $Y \cup \{Y\}$. *If* $(Y_\iota)_{\iota\in I}$ *is a family of transitive sets, then* $\bigcup_{\iota\in I} Y_\iota$ *and* $\bigcap_{\iota\in I} Y_\iota$ *are transitive.*

(b) A set X *is a* pseudo-ordinal *if every transitive set* Y *such that* $Y \subset X$ *and* $Y \neq X$ *is an element of* X. *A set* S *is said to be* decent *if the relation* $x \in S$ *implies* $x \notin x$. *Show that every pseudo-ordinal is transitive and decent (consider the union of decent transitive subsets of* X *and use (a)). If* X *is a pseudo-ordinal, so is* $X \cup \{X\}$.

(c) Let X *be a transitive set and suppose that each* $x \in X$ *is a pseudo-ordinal. Then* X *is a pseudo-ordinal (note that, for each* $x \in X$, $x \cup \{x\}$ *is a pseudo-ordinal contained in* X*).*

(d) Show that $\emptyset$ *is a pseudo-ordinal and that every element of a pseudo-ordinal* X *is a pseudo-ordinal (Consider the union of the transitive subsets of* X *whose elements are pseudo-ordinals).*

(e) If $(X_\iota)_{\iota\in I}$ *is a family of pseudo-ordinals then* $\bigcap_{\iota\in I} X_\iota$ *is the least element of this family (with respect to the relation of inclusion). (Use (b).) Deduce that, if* E *is a pseudo-ordinal, the relation* $x \subset y$ *between elements* x, y *of* E *is a well-ordering relation.*

(f) Show that for each ordinal α *there exists a unique pseudo-ordinal* E_α *such that Ord*$(E_\alpha) = \alpha$ *(use (e) and Criterion C60). In particular the pseudo-ordinals whose order-type are* 0, 1, $2 = 1+1$, *and* $3 = 2+1$ *are respectively*

$$\emptyset, \quad \{\emptyset\}, \quad \{\emptyset, \{\emptyset\}\}, \quad \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$$

Note. The French version of the exercise has one more item. It says: if X and Y are two pseudo-ordinals, then either $X \in Y$ or $Y \in X$ or $X = Y$. The hint for item (c) is: "note that if $Y \neq X$ is transitive, then $Y \subset x$".

This exercise is implemented in section 4.1.2.

10.3 Section 3

¶ 1. *Let* E *and* F *be two sets, let* f *be an injection of* E *into* F *and let* g *be a mapping of* F *into* E. *Show that there exist two subsets* A, B *of* E *such that* $B = E - A$ *and two subsets* A', B' *of* F *such that* $B' = F' - A'$ *for which* $A' = f(A)$ *and* $B = g(B')$. *(Let* $R = E - g(F)$*) and put* $h = g \circ f$*; take* A *to be the intersection of the subsets* M *of* E *such that* $M \supset R \cup h(M)$*.)*

Note: The 1956 Edition of Bourbaki has: f and g injective. In this case, f induces a bijection $A \to A'$ and g induces a bijection $B' \to B$, thus E and F are equipotent. This is the Cantor-Bernstein theorem. The result is true if one of f or g is injective (by symmetry, one case implies the other). The hint works only if g is injective.

```
Lemma Exercise3_1 E F f g: (* 59 *)
  function_prop f E F -> function_prop g F E -> injection f ->
```

```
  exists A, exists A',
    sub A E & sub A' F & image_by_fun f A = A' &
    image_by_fun g (complement F A') = complement E A.

Lemma Exercise3_1b E F f g: (* 4 *)
  function_prop f E F -> function_prop g F E -> injection g ->
  exists A, exists A',
    sub A E & sub A' F & image_by_fun f A = A' &
    image_by_fun g (complement F A') = complement E A.
```

---

2. *If* E *and* F *are two distinct sets, show that* $E^F \neq F^E$. *Deduce that if* E *and* F *are the cardinals 2 and 4 (= 2 + 2), then at least one of the sets* E^F, F^E *is not a cardinal.*

Remember that $C = F^E$ (respectively $D = E^F$) is the set of functional graphs $E \to F$ (respectively $F \to E$). If F is non-empty, the set C is non-empty, and its elements are graphs with domain E. Thus, if both sets E and F are non-empty, C = D implies E = F. The same is true if exactly one of E, F is empty (since exactly one of C, D is empty), and if $E = F = \emptyset$.

The lemma power_2_4 says that, if E = 2 and F = 4, the two sets $\mathscr{F}(E;F)$ and $\mathscr{F}(F;E)$ are equipotent. In particular, C and D are equipotent, and have the same cardinal: card(C) = card(D). This implies that one of "C = card(C)" or "D = card(D)" must be false.

In our implementation a cardinal is a von Neumann ordinal. Assume that F^E is an ordinal. If this ordinal is non-zero, it contains the empty set as least element. Since the domain of the empty set is empty, we get $E = \emptyset$, case where $F^E = \{\emptyset\}$ is an ordinal. Otherwise F is non-empty. Thus, in our implementation E^F is an ordinal if and only if it is zero or one.

```
Lemma Exercise3_2a: forall E F, (* 13 *)
 set_of_gfunctions E F = set_of_gfunctions F E -> E = F.
Lemma Exercise3_2b:
  let E := \2c in let F := card_four in
  let s1 := set_of_gfunctions E F in let s2 :=set_of_gfunctions F E in
  ~ (is_cardinal s1) \/ ~ (is_cardinal s2).
Lemma Exercise3_2c: forall E F, (* 11 *)
  is_ordinal (set_of_gfunctions E F) -> E= emptyset \/ F = emptyset.
```

¶ 3. *Let* $(\mathfrak{a}_\iota)_{\iota\in I}$ *and* $(\mathfrak{b}_\iota)_{\iota\in I}$ *be two families of cardinals such that* $\mathfrak{b}_\iota \geq 2$ *for each* $\iota \in I$.

(a) Show that, if $\mathfrak{a}_\iota \leq \mathfrak{b}_\iota$ *for each* $\iota \in I$, *then*

$$\sum_{\iota\in I} \mathfrak{a}_\iota \leq \prod_{\iota\in I} \mathfrak{b}_\iota.$$

(b) Show that, if $\mathfrak{a}_\iota < \mathfrak{b}_\iota$ *for each* $\iota \in I$, *then*

$$\sum_{\iota\in I} \mathfrak{a}_\iota < \prod_{\iota\in I} \mathfrak{b}_\iota.$$

(Note that a product $\prod_{\iota\in I} E_\iota$ *cannot be the union of a family* $(A_\iota)_{\iota\in I}$ *such that* $\mathrm{Card}(A_\iota) < \mathrm{Card}(E_\iota)$ *for all* $\iota \in I$, *by observing that* $\mathrm{Card}(\mathrm{pr}_\iota(A_\iota)) < \mathrm{Card}(E_\iota)$.*).*

Assume that E_i is a family of cardinals, with $E_i \geq 2$. This means that E_i has at least two distinct elements, say a_i and b_i. We pretend that $\sum E_i \leq \prod E_i$ (this shows (a)). This is obvious if I is empty or reduced to a singleton. Assume that I has two elements. We consider the following function that maps the disjoint union of E_1 and E_2 to the product: elements a_1, a_2, b_1, b_2 are mapped to (a_1, a_2), (a_1, b_2), (b_1, a_2), and (b_1, b_2). Let's maps remaining elements x of E_1 to (x, a_2) and remaining elements y of E_2 to (a_1, y). This function is an injection.

Assume now that I has at least three elements. To $x \in E_j$ we associate the sequence $(x_i)_i$ as follows. We have $x_j = x$, and if $i \neq j$ we take $x_i = b_i$ (if $x = a_j$) and $x_i = b_i$ (otherwise). This function is injective: assume $(x_i)_i = (y_i)_i$. If $y \in E_j$, we have $x = y$ from $x_j = y_j$. Assume $y \in E_k$ with $k \neq j$. We have either $x = a_j$ and $y = b_k$ or $x \neq a_j$ and $y \neq b_k$ (consider otherwise an index distinct from i and j), and these two cases are impossible.

Assume that a_i is a family of cardinals, $a_i < b_i$. Let J be the set of indices such that $b_i \geq 2$. Note that $j \notin J$ implies $b_i = 1$ and $a_i = 0$. Thus $\sum_I a_i = \sum_J b_j$ and $\prod_I a_i = \prod_J b_j$. Thus we may assume J = I, and apply (a). We pretend $\sum_J a_j \neq \prod_J b_j$, proof by contradiction. For otherwise, there would exist a family of sets A_i such that $\mathrm{Card}(A_i) < \mathrm{Card}(E_i)$, and a bijection that maps the disjoint union of A_i onto the product. Let B_i be the image of A_i, so that $\mathrm{Card}(A_i) < \mathrm{Card}(E_i)$, and $\bigcup B_i = \prod E_i$. Let $C_i = \mathrm{pr}_i\langle B_i\rangle$ (the set of i-th components of elements of B_i, this is a subset of E_i). We have $\mathrm{Card}(C_i) \leq \mathrm{Card}(A_i)$. This implies $C_i \neq E_i$. In particular, there is $x_i \in E_i$, not in C_i. The family $(x_i)_i$ is in the product, thus in some B_i; its i-th component is in C_i, absurd.

This is implemented as compare_sum_prod2 and compare_sum_prod3 in section 8.17.

---

4. Let E be a set and let f be a mapping of $\mathfrak{P}(E) - \emptyset$ into E such that for each non-empty subset X of E we have $f(X) \in X$ ("Choice function").

(a) Let $\mathfrak{b}$ be a cardinal and let A be the set of all $x \in E$ such that $\mathrm{Card}(\overset{-1}{f}(x)) \leq \mathfrak{b}$. Show that if $\mathfrak{a} = \mathrm{Card}(A)$, then $2^{\mathfrak{a}} \leq 1 + \mathfrak{a}\mathfrak{b}$ (note that if $Y \subset A$ and $Y \neq \emptyset$ then $f(Y) \in A$).

(b) Let B be the set of all $x \in E$ such that, for each non-empty subset X of $\overset{-1}{f}(x)$, $\mathrm{Card}(X) \leq \mathfrak{b}$. Show that $\mathrm{Card}(B) \leq \mathfrak{b}$.

Discussion.

Let's write $g(x)$ instead of $\overset{-1}{f}(x)$. Statement (b) should be corrected as: if B is the set of all $x \in E$ such that $X \in g(b)$ implies $\mathrm{Card}(X) \leq b$, then $\mathrm{card}(B) \leq b$.

The Zermelo theorem asserts that there is a unique ordering on E such that $f(X)$ is the least element of X. Then $g(x)$ is the set of all subsets of E that have x as least element. Assume that $E = \mathbf{N}$. Then $g(x)$ is the translation of $g(0)$ and has cardinal $b_0 = 2^{\aleph_0}$. In case $b < b_0$ we have $A = \emptyset$, $a = 0$, and otherwise $A = \mathbf{N}$, $a = \alpha_0$. In both cases, the conclusion (a) is obvious. Conclusion (b) follows since, if b is finite, then B is empty, and otherwise $b = E$.

Assume E finite, and its elements are $x_0 > x_1 > x_2 > x_3$. Note that $g(x_i)$ is the set of all $\{x_i\} \cup Y$, where Y is a subset of $\{x_0, x_1, \ldots, x_{i-1}\}$ thus has 2^i elements. Note that $[x_i, x_0] \in g(x_i)$, and this interval has $i+1$ elements, so that B contains the elements $> x_b$ and there are b such elements (unless $b \geq \mathrm{card}(E)$, case where $B = E$). In particular result (b) is obvious. Consider (a): If $2^i \leq b < 2^j$, then A contains only x_j for $j \leq i$, thus has $j+1$ elements. The conclusion follows easily.

Consider point (a). Obviously, if Y is a non-empty subset of A, then $f(Y) \in Y$, thus $f(Y) \in A$.

In particular, if A is non-empty, we have $f(A) \in A$. Since f is a choice function, $f(\{x\}) = x$, so that f is surjective, and $f^{-1}(x)$ is never empty. Consider $b = 0$. We have to show $2^a \leq 1$, which has $a = 0$ as only solution. In fact, if $b = 0$, A is empty since no $f^{-1}(x)$ has cardinal zero. Consider $b = 1$. We have to show $2^a \leq 1 + a$, and this equation has only two solutions: $a = 0$ and $a = 1$. In fact, assume A non-empty so that $f(A) \in A$. Set $Y = f^{-1}(f(A))$. The two elements A and $\{f(A)\}$ are in Y. But Y has at most one element, so that these two elements are the same and A is a singleton.

Consider $b = 2$. We must show that $2^a \leq 1 + a$. This equation has only 0, 1, and 2 as solutions. We have to show that A has at most two elements. Assume that there are elements x and y such that $f^{-1}(x)$ and $f^{-1}(x)$ have two elements, say X, Y, and the singletons. Note that $f(\{x, y\})$ is one of x, y, let's say its x. Assume there is another element, say z in A. Then $f(\{x, z\}) = z$. It follows that $f(\{y, z\}) = y$, so that $Y = \{x, z\}$. This leaves no choice for $f(\{x, y, z\}$. Thus $A_2 - A_1$ has at most one element; and since A_1 has at most one element, the result is true.

---

5. *Let* $(\lambda_\iota)_{\iota\in I}$ *be a family of order-types (§2, Exercise 13), indexed by an ordered set* I. *Show that*

$$\mathrm{Card}\big(\sum_{\iota\in I}\lambda_i\big) = \sum_{\iota\in I}\mathrm{Card}(\lambda_i)$$

and that, if I *is well-ordered,*

$$\mathrm{Card}\big(\prod_{\iota\in I}\lambda_i\big) = \prod_{\iota\in I}\mathrm{Card}(\lambda_i).$$

According to Exercise 2.13, an order type λ_i is some particular ordered set $(E_i, \leq_i)$. The quantity $\mathrm{Card}(\lambda_i)$ is the cardinal of E_i. The quantity $\sum\lambda_i$ is some ordering on a set E order-isomorphic to the ordinal sum $S = \sum E_i$. We must show that Card(E) is the sum $s = \sum \mathrm{Card}(E_i)$. We show here that $\mathrm{Card}(\sum E_i) = t$. Since E is order-isomorphic to S it has the same cardinal. We then show that, if I is well-ordered and if each λ_i is an ordinal, so that $\sum\lambda_i$ is an ordinal, then $\mathrm{Card}(\sum\lambda_i) = s$.

```
Definition fam_card_sub f :=
  (L (domain f) (fun z => cardinal (substrate (V z f)))).

Lemma Exercise3_5a r f: orsum_ax r f -> (* 5 *)
  cardinal (substrate (order_sum r f)) = card_sum (fam_card_sub f).
Lemma Exercise3_5b r f: orprod_ax r f -> (* 5 *)
  cardinal (substrate (order_prod r f)) = card_prod (fam_card_sub f).
Lemma Exercise3_5c r f h: orsum_ax r f -> (* 2*)
  h \Is (order_sum r f) ->
  cardinal (substrate h) =  card_sum (fam_card_sub f).
Lemma Exercise3_5d: r f h: ordprod_ax  r f -> (* 2 *)
  h \Is (order_prod r f) ->
  cardinal (substrate h) =  card_prod (fam_card_sub f).

Lemma Exercise3_5e r g:  (* 15 *)
  worder r -> substrate r = domain g -> ordinal_fam g ->
  cardinal (ord_sum r g) =
    card_sum (L (domain g) (fun z => cardinal (V z g))).
Lemma Exercise3_5f r g:  (* 15 *)
  worder r -> substrate r = domain g -> ordinal_fam g ->
```

```
  finite_set (substrate r) ->
  cardinal (ord_prod r g) =
    card_prod (L (domain g) (fun z => cardinal (V z g))).
```

---

6. *Show that for every set* E *there exists* $X \subset E$ *such that* $X \notin E$ *(use Theorem 2 of no. 6).*

If this property were false, we would have: for all X, $X \subset E \implies x \in E$, this is $\mathfrak{P}(E) \subset E$, and implies $\mathrm{Card}(\mathfrak{P}(E)) \leq \mathrm{Card}(E)$. This contradicts Cantor. Note that we can choose for X the set of all $t \in E$ such that $t \notin t$.

```
Lemma Exercice3_6 E: exists X, sub X E & ~ (inc X E). (* 7 *)
Lemma Exercice3_6b E:  (* 4 *)
  let X:=  Zo E (fun t => ~(inc t t)) in sub X E & ~ (inc X E).
```

10.4 Section 4

1. *(a) Let* E *be a set and let* $\mathfrak{F}(E)$ *be the set of finite subsets of* E*. Show that* $\mathfrak{F}(E)$ *is the smallest subset* $\mathfrak{G}$ *of* $\mathfrak{P}(E)$ *satisfying the following conditions: (i)* $\emptyset \in \mathfrak{G}$*; (ii) the relation* $X \in \mathfrak{G}$ *and* $x \in E$ *imply* $X \cup \{x\} \in G$*.*

(b) Deduce from (a) that the union of two finite subsets A *and* B *is finite (consider the set of subsets* X *of* E *such that* $X \cup A$ *is finite; cf § 5; no 1 Proposition 1, Corollary 1).*

(c) Deduce from (a) and (b) that for every finite set E *the set* $\mathfrak{P}(E)$ *is finite (consider the set of subsets* X *of* E *such that* $\mathfrak{P}(X)$ *is finite; cf § 5; no 1 Proposition 1, Corollary 4).*

If we denote by $\mathfrak{F}(E)$ the set of finite subsets of E, and by $P(E,\mathfrak{G})$ the property: (i) $\emptyset \in \mathfrak{G}$; (ii) the relation $X \in \mathfrak{G}$ and $x \in E$ imply $X \cup \{x\} \in \mathfrak{G}$, then $P(E,\mathfrak{F})$ is true; by induction, $P(E,\mathfrak{G})$ implies $\mathfrak{F} \subset \mathfrak{G}$. As a consequence, the union of two finite sets is finite. The powerset of a finite set is finite (proof by induction: $\mathfrak{P}(X \cup \{x\})$ is the union of $\mathfrak{P}(X)$ and the image of $\mathfrak{P}(X)$ by the mapping $Y \mapsto (Y \cup \{x\})$; if $x \notin X$, the union is disjoint, and the cardinal is twice the cardinal of $\mathfrak{P}(X)$).

```
Definition set_of_finite_subsets E := Zo(powerset E)(fun X => finite_set X).

Definition set_of_finite_subsets_prop E F:=
  inc emptyset  F & forall x X, inc x E -> inc X F -> inc (tack_on X x) F.

Lemma set_of_finite_subsets_pr E: (* 12 *)
  set_of_finite_subsets_prop E (set_of_finite_subsets E) &
  (forall F, set_of_finite_subsets_prop E F -> sub(set_of_finite_subsets E) F).
Lemma finite_union2 x y: (* 18 *)
  finite_set x -> finite_set y -> finite_set (union2 x y).
Lemma finite_powerset x:  (* 50 *)
  finite_set x -> finite_set (powerset x).
```

---

2. *Show that a set* E *is finite if and only if every non-empty subset of* $\mathfrak{P}(E)$ *has a maximal element (with respect to inclusion). (To show that the condition is sufficient, apply it to the set* $\mathfrak{F}(E)$ *of finite subsets of* E*).*

Notice that if F is a finite subset of an infinite set E, there exists another finite subset F' of E, such that F is a strict subset of F'. Conversely, assume E finite, and proceed by induction. Say $E = F \cup \{a\}$, and let Y be a subset. If no element of Y contains a, the result is obvious by induction. Otherwise, let Z be the set of elements of Y that do contain a. By induction, Z has a maximal element.

```
Lemma finite_is_maximal_inclusion x: (* 46 *)
  finite_set x <->
  (forall y, sub y (powerset x) -> nonempty y -> exists z,
     inc z y & forall t, inc t y -> sub z t -> z = t).
```

---

3. *Show that if a well-ordered set* E *is such that the ordered set obtained by endowing* E *with the opposite ordering is also well-ordered, then* E *is finite (consider the greatest element* x *of* E *such that the segment* S_x *is finite.*

This is well_ordered_opposite.

---

4. *Let* E *be a finite set with* $n \geq 2$ *elements, and let* C *be a subset of* $E \times E$ *such that, for each pair* x, y *of distinct elements of* E*, exactly one of the two elements* (x, y), (y, x) *of* $E \times E$ *belongs to* C*. Show that there is a mapping* f *of the interval* $[1, n]$ *onto* E *such that* $(f(i), f(i+1)) \in C$ *for* $1 \leq i \leq n-1$ *(use induction on* n*).*

We consider the assumption H(C, E) that says that for each pair x, y of distinct elements of E, exactly one of the two pairs (x, y), (y, x) belongs to C. The Exercise assumes further that $C \subset E \times E$, but this is of no help. We consider the conclusion: there is a bijection $f : [1, \mathrm{Card}(E)] \to E$ such that $(f(i), f(i+1)) \in E$ (the Exercise requires f surjective, but $[1, \mathrm{Card}(E)]$ and E are two finite sets with the same cardinal, so bijectivity is equivalent to surjectivity). We show that, if the assumption implies the conclusion whenever E has cardinal $< n$, then the result holds for cardinal n as well, and conclude via cardinal_c_induction1.

The result is obvious if E is empty. Otherwise, fix $a \in E$, let E_1 be the set of all x such that $(x, a) \in C$, and E_2 the set of all x such that $(a, x) \in C$. If these sets have cardinal n_1 and n_2, we have $n_1 + n_2 + 1 = n$, and by induction we get two functions f_1 and f_2. Define f by $f_1(i)$ if $i \leq n_1$, $f_2(i - n_1 - 1)$ if $i > n_1 + 1$, and $f(n_1 + 1) = a$. This is the desired function.

```
Lemma Exercise4_4 n E C:  (* 182 *)
  inc n Bnat -> cardinal E = n -> sub C (coarse E) ->
  (forall x y, inc x E -> inc y E -> x <> y ->
    ((inc (J x y) C \/ inc (J y x) C) &
    (inc (J x y) C -> inc (J y x) C -> False))) ->
    exists f,  function_prop f (interval_Bnat \1c n) E &
      bijection f &
      (forall i, \1c <=c i -> i <c n ->
        inc (J (W i f) (W (succ i) f)) C).
```

¶ 5. *Let* E *be an ordered set for which there exists an integer* k *such that* k *is the greatest number of elements in a free subset* X *of* E *(§ 1, Exercise 5). Show that* E *can be partitioned into* k *totally ordered subsets (with respect to the induced ordering)*[2]*. The proof is in two steps:*

(a) If E *is finite and has* n *elements, use induction on* n*; let* a *be a minimal element of* E *and let* $\mathrm{E}' = \mathrm{E} - \{a\}$*. If there exists a partition of* E' *into* k *totally ordered sets* C_i $(1 \leq i \leq k)$*, let* U_i *be the set of all* $x \in \mathrm{C}_i$ *which are* $\geq a$*. Show that there is at least one index* i *for which a free subset* $\mathrm{E}' - \mathrm{U}_i$ *has at most* $k-1$ *elements. The proof of this is by* reduction ad absurdum*. For each* i*, let* S_i *be a free subset of* $\mathrm{E}' - \mathrm{U}_i$ *which has* k *elements, let* S *be the union of the sets* S_i*, and let* s_j *be the least element of* $\mathrm{S} \cap \mathrm{C}_j$ *for each index* $j \leq k$*; show that the* $k+1$ *elements* $a, s_1, \ldots, s_k$ *form a free subset of* E*.*

(b) If E *is arbitrary, the proof is by induction on* k*, as follows. A subset* C *of* E *is said to be* strongly related *in* E *if for each finite subset* F *of* E *there exists a partition of* F *into at most* k *totally ordered sets such that* $\mathrm{C} \cap \mathrm{F}$ *is contained in one of them. Show that there exists a maximal strongly related subset* C_0*, and that every free subset of* $\mathrm{E} - \mathrm{C}_0$ *has at most* $k-1$ *elements (Argue by contradiction, and suppose that there is a free subset* $\{a_1, \ldots, a_k\}$ *of* k *elements in* $\mathrm{E} - \mathrm{C}_0$*; Consider each set* $\mathrm{C}_0 \cup \{a_i\}$ $(1 \leq i \leq k)$*, and express the fact that it is not strongly related, thus introducing a finite subset* F_i *of* E *for each index* i*. Then consider the union* F *of the sets* F_i *and use the fact that* C_0 *is strongly related to obtain a contradiction).*

We start with a lemma. Assume that Y and T are two sets with k elements. Assume $\mathrm{T} \subset \bigcup \mathrm{Y}$, the elements of Y mutually disjoint. Assume moreover that for all $\mathrm{Z} \in \mathrm{Y}$ the set $\mathrm{Z} \cap \mathrm{T}$ has at most one element. Then it has exactly one element (consider the set of those that are non-empty; assume that there are m such sets. Each set has cardinal 1, so that the union, which is disjoint, has cardinal m. Thus $m = k$).

```
Lemma finite_set_minimal r: (* 7 *)
  order r ->finite_set (substrate r) -> nonempty (substrate r) ->
  exists x, minimal_element r x.
Lemma Exercise4_5a Y T k:   (* 47 *)
  cardinal Y = k -> cardinal T = k -> inc k Bnat ->
  sub T (union Y) ->
  (forall a b : Set, inc a Y -> inc b Y -> a = b \/ disjoint a b) ->
  (forall Z, inc Z Y -> small_set (intersection2 T Z)) ->
  (forall Z, inc Z Y -> is_singleton (intersection2 T Z)).
```

Denote by $p'_k(\mathrm{E})$ the property that each free subset of E has at most k element; and by $p_k(\mathrm{E})$ the property that moreover, there is one free subset of E with k elements. Denote by $q'_k(\mathrm{E},\mathrm{X})$ the property that X contains at most k elements which are mutually disjoint, totally ordered, whose union is E. Denote by $q_k(\mathrm{E},\mathrm{X})$ the property that $q'_k(\mathrm{E},\mathrm{X})$ holds, and moreover that X has exactly k elements, none of which is empty (so that X is a partition of E). Denote by $q'_k(\mathrm{E})$ and $q_k(\mathrm{E})$ the property that there exists X such that $q'_k(\mathrm{E},\mathrm{X})$ (respectively $q_k(\mathrm{X},\mathrm{E})$) holds. We must show: $p_k(\mathrm{E}) \implies q_k(\mathrm{E})$. In a first step we prove this in the case when E is finite. We then deduce that (for E finite), $p'_k(\mathrm{E}) \implies q'_k(\mathrm{E})$. We the prove the result by induction on k.

[2] This is called Dilworth's theorem in the French edition

```
Definition Exercise4_5_hyp r k :=
  (exists x, inc x (set_of_free_subsets r) & cardinal x = k) &
  (forall x, inc x (set_of_free_subsets r) -> (cardinal x) <=c k).
Definition Exercise4_5_conc r k :=
  exists X, partition X (substrate r) & cardinal X = k &
    (forall x, inc x X ->  total_order (induced_order r x)).
```

Part (a): case E finite by induction on the cardinal of E.

Assume the result true whatever k, for all ordered sets with less than n elements. Let E be an ordered set with n elements. If $n = 0$, the set E is empty and we have $k = 0$, so that this case is trivial. Otherwise, E is non-empty, thus has a minimal element, say a. Set $E' = E - \{a\}$, ordered by the ordering of E. If $\mathfrak{I}(E)$ is the set of free subsets of E, we have $\mathfrak{I}(E') \subset \mathfrak{I}(E)$, so that $k' \leq k$ (where k' is the least number of elements in an element of $\mathfrak{I}(E')$). If X is free in E with k elements then $X - \{a\} \in \mathfrak{I}(E')$ has at least $k-1$ elements. Thus, there are two cases to consider $k' = k-1$ or $k' = k$. In the first case, we get a partition of E' with $k-1$ elements. We add $\{a\}$ as a new set to this partition. This gives a partition of E with k sets. (This requires 100 lines of proof).

In the second case, we get a partition with k elements. For each C_i in the partition we consider U_i, the set of element of C_i that are $\geq a$, and $\bar{U}_i = U_i \cup \{a\}$. This is a totally ordered set. Let $V_i = E - \bar{U}_i$. Assume that for some i, all free subsets of V_i have less than k elements. Let T be a free subset of E with k elements. It has at most one element in U'_i, thus has $k-1$ elements in V_i. Thus, there is a partition of V_i into $k-1$ sets, and it suffices to add $\bar{U}_i$ to get a partition of E into k parts. (This requires 100 lines of proof).

We assume now that for each i there is a free subset S_i with k elements such that $S_i \subset V_i$. Write $S_i \cap C_j = \{s_{ij}\}$. Note that s_{ij} is never a so that $s_{ij} \leq a$ is false, since a is minimal. On the other hand $a \leq s_{ii}$ is equally false, since s_{ii} is in C_i and V_i. Let W_j be the set of all s_{ij}. This set is non-empty, totally ordered and finite, thus has a least element, say s_j. We have $s_j \leq s_{ij}$. Assume $s_i \leq s_j$. Then s_i is some s_{li} and $s_{li} \leq s_{lj}$. Since S_l is free, we get $s_{li} = s_{lj}$. We deduce $i = j$, for otherwise C_i and C_j are disjoint. This implies in particular that $l \to s_l$ is injective. Let A be the set of all s_l. It has k elements. Let $B = A \cup \{a\}$. It has more than k elements, yet is free, absurd. (This requires 150 lines of proof).

```
Lemma Exercise4_5b r k: finite_set (substrate r) -> (* 354 *)
  order r -> inc k Bnat ->  Exercise4_5_hyp r k -> Exercise4_5_conc r k.
```

Part (b): general case by induction on k.

Since singletons are free, $k = 0$ says that E is empty, and the conclusion is trivial. Assume that all free subsets are of cardinal $\leq k$, and there is a free subset X_0 of cardinal k. Assume (H): there is a totally ordered set C, such that $C \subset E$ and each free subset of $E - C$ has at most $k-1$ elements. Note that $X_0 \cap (E - C)$ has $k-1$ elements, since X_0 cannot have two elements in the totally ordered set C. Note also that C is non-empty. We can apply the induction assumption to $X - C$, partition it into $k-1$ parts and add C as a k-th set of the partition (This requires 100 lines of proof).

We say that C is strongly related if $C \subset E$, and for every finite subset F of E, there is a set X such that $q'_k(F,X)$ holds and for some $x \in X$ we have $C \cap F \subset x$. We pretend that there is a maximal strongly related set (for the $\subset$ ordering), apply application of the Zorn lemma. All we need to do is prove that if $\mathfrak{C}$ is a totally ordered set of strongly related sets, then $C = \bigcup \mathfrak{C}$ is strongly related. Consider a finite set F. For each $x \in C \cap F$ we select an element C_x of $\mathfrak{C}$ such that $x \in C_x$. The set of all $G = \{C_x, x \in C \cap F\}$ is finite and totally ordered, thus has a greatest

element C_0 (for all x we have $C_x \subset C_0$). We have $C \cap F = C_0 \cap F$, and the conclusion holds since C_0 is strongly related. The previous argument fails when G is empty, i.e., when $C \cap F$ is empty. But, in this case, all we need to do is to find a non-empty set X such that $q'_k(F,X)$ holds. Since F is finite, part (a) says that there is X such that $q'_k(F,X)$. If this gives $X = \emptyset$, then $F = \emptyset$, and we can use $\{\emptyset\}$ instead of X, since k is non-zero [recall that X is not required to be a partition]. (This requires 80 lines of proof).

We assume now that C_0 is a maximal strongly related set. Let a and b be two elements of C_0, and $F = \{a, b\}$. Since C_0 is strongly related, there exists a totally ordered set X such that $F \subset X$. Thus F is totally ordered, and C_0 is totally ordered. Assume that every free subset of X_0 has less than k elements. Then C_0 satisfies assumption (H) and the theorem is proved. On the contrary, assume that there is a free set with k elements $a_1, \ldots, a_k$ not in C_0. Let $C_i = C_0 \cup \{a_i\}$. By maximality of C_0, this is not a strongly related set. Thus, there exists a finite set F_i, such that, whenever $q'_k(F,X)$ holds, $C_i \cap F_i$ is not a subset of any element of X. Consider the union G of these F_i and the set of all a_i. This is a finite set. Since C_0 is a strongly related set, there exists a set X, such that $q'_k(G,X)$ holds, and $C_0 \cap G$ is a subset of some element Y_0 of X. Fix i. Consider T_i, the set of all $F_i \cap Y$ for $Y \in X$. These sets are totally ordered, and the union is F_i. There are at most k distinct elements in T; so that $q'_k(F_i, T_i)$ holds. Thus $C_i \cap F_i$ is not a subset of $F_i \cap Y$, whatever Y. Take $Y = Y_0$. We know that $C_0 \cap F_i$ is a subset of $F_i \cap Y$. This implies $a_i \notin Y$. Finally, each a_i (there are k such elements) belongs to at most one element X_j of X (there are $\leq k$ such elements), since the set of a_i is free, the sets X_j are totally ordered. This implies that each X_j (included Y) contains exactly one a_i. Contradiction. (This requires 110 lines of proof).

```
Lemma Exercise4_5c r k:  (* 27 *)
  finite_set (substrate r) -> order r -> inc k Bnat ->
  (forall x, inc x (set_of_free_subsets r) -> (cardinal x) <=c k) ->
  exists X, (cardinal X) <=c k &
    union X = (substrate r) &
    (forall a b, inc a X -> inc b X -> a = b \/ disjoint a b) &
    (forall x, inc x X ->  total_order (induced_order r x)).
Lemma Exercise4_5d r k: (* 294 *)
  order r -> inc k Bnat -> Exercise4_5_hyp r k -> Exercise4_5_conc r k.
```

¶ 6. *(a) Let* A *be a set and let* $(X_i)_{1 \leq i \leq m}$, $(Y_j)_{m+1 \leq j \leq m+n}$ *be two finite families of subsets of* A. *Let* h *be the least integer such that, for each integer* $r \leq m - h$ *and each subset* $\{i_1, \ldots, i_{r+h}\}$ *of* $r+h$ *elements of* $[1, m]$, *there exists a subset* $\{j_1, \ldots, j_r\}$ *of* r *elements of* $[m+1, m+n]$ *for which the union of the sets* X_{i_α} $(1 \leq \alpha \leq r + h)$ *meets each of the sets* Y_{j_β} $(1 \leq \beta \leq r)$ *(which implies that* $m \leq n + h$*). Show that there exists a finite subset* B *of* A *with at most* $n + h$ *elements such that every* X_i $(1 \leq i \leq m)$ *and every* Y_j $(m + 1 \leq j \leq m + n)$ *meets* B. *(Consider the order relation on the interval* $[1, m+n]$ *whose graph is the union of the diagonal and the set of pairs* (i, j) *such that* $1 \leq i \leq m$ *and* $m + 1 \leq j \leq m + n$ *and* $X_i \cap Y_j \neq \emptyset$*, and apply Exercise 5 to this ordered set.)*

Note: The sets X_i and Y_j must be non-empty, since otherwise no set can meet them all.

We first show that a non-empty set T such that $x \in T$ implies $x \leq m$ has a greatest element (when $m \in \mathbf{N}$). Thus, if all free subsets of E have cardinal $\leq n$, there exists an integer k such that p_k holds (where p_k is defined in the previous exercise). If $m = n + h$, and if there is a free subset with n elements, then $k = n + h$ for some h.

```
Lemma finite_bounded_greatest_B n T:  (* 14 *)
   inc n Bnat -> (forall m, inc m T -> m <=c n) ->
   nonempty T ->
  exists m, inc m T & forall k, inc k T -> k <=c m.
Lemma Exercise4_5A1 r n:
  inc n Bnat ->
  (forall x, inc x  (set_of_free_subsets r) ->
     (cardinal x) <=c n) ->
  exists k, inc k Bnat &  Exercise4_5_hyp r k.

Lemma Exercise4_5A2 r n h: (* 7 *)
  inc n Bnat -> inc h Bnat ->
  (forall x, inc x  (set_of_free_subsets r) ->
    (cardinal x) <=c (n +c h)) ->
  (exists x, inc x (set_of_free_subsets r) & (cardinal x = n)) ->
  exists k, inc k Bnat & k <=c h & Exercise4_5_hyp r (n +c k).
```

Let I be the interval $[1, m]$ and J the interval $[m+1, m+n]$. All we need to know is that I and J have m and n elements, and are disjoint.

```
Definition meet A B := nonempty (intersection2 A B).

Section Exercise46.
Variables A X Y m n :Set.
Hypothesis (nB: inc n Bnat) (mB: inc m Bnat).
Hypothesis Xpr:
  fgraph X & cardinal (domain X) = m & sub (range X) (powerset A) &
   forall i, inc i (domain X) -> nonempty (V i X).
Hypothesis Ypr:
  fgraph Y & cardinal (domain Y) = n & sub (range Y) (powerset A) &
   forall i, inc i (domain Y) -> nonempty (V i Y).
Hypothesis disdom: disjoint (domain X) (domain Y).

Definition E46_hprop h := forall r  Z, r <=c (card_sub m h) ->
   sub Z (domain X) -> cardinal Z = r +c h  ->
   exists T, sub T (domain Y) &  cardinal T = r &
   forall j, inc j T -> meet (V j Y) (unionb (restr X Z)).

Definition E46_hp h :=  inc h Bnat & h <=c m & E46_hprop h
  & forall l, inc l Bnat ->  l <=c m -> E46_hprop l -> h <=c l.

Definition E46_conc h := exists B, (cardinal B) <=c (n +c h)
  & finite_set B &  sub B A& (forall i, inc i (domain X) -> meet (V i X) B)
  &  (forall j, inc j (domain Y) -> meet (V j Y) B).
```

Note that p_m holds, so that there is a least h satisfying p_h. Note also that if $h \leq m$, taking $r = m - h$ and $Z = I$ yields a subset with r elements of J. Thus $m \leq n + h$.

Write $i < j$ if X_i meets Y_j. We can convert this into an order relation on $E = I \cup J$ by adding reflexivity. The condition X_{i_α} $(1 \leq \alpha \leq r + h)$ meets Y_{J_β} can be restated as: there is at least one α such that $i_\alpha \leq j_\beta$. The assumptions of (a) can be restated as h is the least integer satisfying property p_h: for any $r \leq m - h$ and any $Z \subset I$ of cardinal $r + h$, there is a subset T of J with r elements such that, if $j \in T$, there is $i \in Z$ such that $i \leq j$.

```
Definition E46_u := union2 (domain X) (domain Y).
```

```
Definition E46_order_rel x y :=
  x = y  \/ (inc x (domain X) & inc y (domain Y) & meet (V x X) (V y Y)).
Definition E46_order_r :=  graph_on E46_order_rel  E46_u.

Lemma Exercise4_6a:  (* 12 *)
  order  E46_order_r & (substrate E46_order_r = E46_u) &
  (forall x y, gle E46_order_r x y <->
    (inc x E46_u & inc y E46_u & E46_order_rel x y)).
Lemma Exercise4_6b h: h <=c m ->  (* 10 *)
  E46_hprop h -> m <=c (n +c h).
Lemma Exercise4_6c: exists h,  E46_hp h. (* 14 *)
Lemma Exercise4_6d h: E46_hprop h <->  (* 13 *)
  forall r Z, r <=c (card_sub m h) ->
   sub Z (domain X) -> cardinal Z = r +c h  ->
   exists T, sub T (domain Y) &  cardinal T = r &
   forall j, inc j T -> exists i, inc i Z & gle E46_order_r i j.
```

Assume p_h true. Consider a free subset K of E and write $Z = K \cap I$. Then K is the disjoint union of K and a subset L of J. If Z has at most h elements, then K has at most $h+n$ elements since L has at most n elements. But if Z has $r+h$ elements, there is a set T with r elements such that $L \cap T = \emptyset$, so that L has at most $n-h$ elements, and we get the same conclusion. Note that J is a free subset with n elements, so that the maximal number of elements in a free subset is $n+k$ for some k, with $k \leq h$. The same is obviously true if h is the least integer that satisfies p_h; and we can apply the previous exercise.

```
Lemma Exercise4_6e h K:
  inc h Bnat -> E46_hprop h -> inc K (set_of_free_subsets E46_order_r) ->
  (cardinal K) <=c (n +c h).

Lemma Exercise4_6f h: inc h Bnat -> E46_hprop h ->
  exists k, inc k Bnat & k <=c h &
    Exercise4_5_hyp E46_order_r  (n +c k).
Lemma Exercise4_6g h: E46_hp h -> (* 4 *)
  exists k, inc k Bnat & k <=c h &
    Exercise4_5_conc E46_order_r  (n +c k).
```

Assume that U is a non-empty totally ordered subset of E. If a and b are in U we have $a = b$, $a < b$ or $b < a$. In the case $a < b$ we have $a \in I$ and $b \in J$ so that U cannot have more than two elements. Consider the following quantity x_U. If U has two elements, say a and b with $a < b$, then x_U is an element of the intersection $X_a \cap Y_b$ (which is nonempty by definition of $<$). Otherwise U is a singleton say $U = \{i\}$ or $U = \{j\}$ with $i \in I$ and $j \in J$. We choose some element of X_i or Y_j (remember that we assume these sets to be nonempty). In any case, we have $x_U \in A$.

Assume that E is the union of $n+k$ totally ordered non-empty subsets. Let B be the set of all x_U for U in the union. This is a finite subset of A with at most $n+k$ elements. We have shown that there exists an index k with $k \leq h$, thus B has at most $n+h$ elements. By construction B meets any element of X_i and any Y_j.

```
Lemma Exercise4_6h h: E46_hp h ->  E46_conc h. (* 104 *)
End Exercise46.
```

(b) Let E *and* F *be two finite sets and let* $x \to A(x)$ *be a mapping of* E *into* $\mathfrak{P}(F)$. *Then there exists an injection* f *of* E *into* F *such that* $f(x) \in A(x)$ *for each* $x \in E$ *if and only if for*

each subset H *of* E, *we have* $\mathrm{Card}\big(\bigcup_{x\in H} A(x)\big) \geq \mathrm{Card}(H)$ *(the method of proof is analogous to that of (a), with* $h = 0$*).*

Write the statement as (P) $\iff$ (Q). The implication (P) $\implies$ (Q) is trivial. The converse is a particular case of (c) below.

```
Definition E46b_hyp E F A :=
  exists f, injection f & source f = E & target f = F &
  (forall x, inc x E -> inc (W x f) (A x)).
Definition E46b_conc E A :=
  forall H, sub H E ->
  (cardinal H) <=c (cardinal (union (fun_image H A))).

Lemma Exercise4_6i E F A:  E46b_hyp E F A -> E46b_conc E A. (* 10 *)
```

(c) With the hypotheses of (b), let G *be subset of* F. *Then there exists an injection* f *of* E *into* F *such that* $f(x) \in A(x)$ *for each* $x \in E$ *and such that* $f(E) \supset G$ *if and only if* f *satisfies the condition of (b) and for each subset* L *of* G *the cardinal of the set of all* $x \in E$ *such that* $A(x) \cap L \neq \emptyset$ *is* $\geq \mathrm{Card}(L)$. *(Let* $(a_i)_{1\leq i\leq p}$ *be the sequence of distinct elements of* G, *arranged in some order; let* $(b_j)_{p+1\leq j\leq p+m}$ *be the sequence of distinct elements of* F, *arranged in some order; and let* $(c_k)_{p+m+1\leq k\leq p+m+n}$ *be the sequence of distinct elements of* E, *arranged in some order. Consider the order relation on the set* $[1, p+m+n]$ *whose graph is the union of the diagonal and the set of pairs* (i, j) *such that either*

$$1 \leq i \leq p \text{ and } p+1 \leq j \leq p+m \text{ and } a_i = b_j,$$

$$\text{or } 1 \leq i \leq p \text{ and } p+m+1 \leq j \leq p+m+n \text{ and } a_i \in A(c_j),$$

$$\text{or } p+1 \leq i \leq p+m \text{ and } p+m+1 \leq j \leq p+m+n \text{ and } b_i \in A(c_j);$$

then apply Exercise 5.)

Lemma 4.6k below uses the assumptions of (b) and (c) and proof the conclusion of (b), thus is not optimal. Lemma 4.6l shows (b) by taking $G = \emptyset$, see discussion below.

```
Definition E46c_hyp E F A G :=
  exists f, injection f & source f = E & target f = F &
  sub G (image_of_fun f) &  (forall x, inc x E -> inc (W x f) (A x)).

Definition E46c_conc E A G :=
  (cardinal L) <=c (cardinal (Zo E (fun z => meet (A z) L)))

Lemma Exercise4_6j E F A G: E46c_hyp E F A G -> (* 18 *)
  (E46b_conc E A &  E46c_conc E A G).

Lemma Exercise4_6k E F A G: (* 223 *)
  finite_set F -> sub G F ->
  (forall x, inc x E -> sub (A x) F) ->
   E46b_conc E A ->  E46c_conc E A G -> E46b_hyp E F A.

Lemma Exercise4_6l E F A: (* 8 *)
  finite_set F -> (forall x, inc x E -> sub (A x) F) ->
   E46b_conc E A ->  E46b_hyp E F A.
```

The Lemma. Assume f injective, $L \subset G \subset f\langle E\rangle$. Then $L = f\langle K\rangle$ where K has same cardinal as L. If $x \in K$, then $f(x) \in A(x) \cap L$.

Conversely, assume F finite, G ⊂ F. Assume that (Q) is true; taking H = E shows that E is finite, so that there is no need to add "E finite" to the list of assumptions.

Consider the disjoint union R of E, F and G. This is the set of all x_e, y_f or z_g for $x \in$ E, $y \in$ F or $z \in$ G. The indices e, f and g are some arbitrary distinct constants. Since G ⊂ F, if $y \in$ G then y_f and y_g are two distinct elements of R. Assume that E, F and G have n, m and p elements. Then R has $n+m+p$ elements. We consider the following relation on R: for all x, $x \le x$, for all $x \in$ G, $x_g \le x_f$, for all $x \in$ E and $y \in$ F, such that $y \in$ A(x), we have $y_f \le x_e$; if moreover $y \in$ G we have also $y_g \le x_e$. This is easily seen to be an order relation.

The key relation is the following: if $x < y$ then the indices are (g,f) or (f,e) or (g,e). In particular F, considered as a subset of R is free.

Consider any free subset I. Let J be the set obtained by replacing all x_g by x_f. This is a free subset with the same number of elements. Let J_e and J_f be the intersections of J with E and F (considered as a subset of the disjoint union), and H and K the same sets (but in E and F). Since J is the disjoint union of J_e and J_f the cardinal of I will be Card(H) + Card(K). Apply the assumption (b). It says card H ≤ card H′ where H′ is some subset of F. Obviously H′ and K are disjoint; This yields Card(I] ≤ Card(F).

Exercise 5 says: there is a partition of K into some set $(U_i)_{i\in I}$. If $x \in$ F, (respectively $x \in$ E) there is a unique $i \in$ I such that $x_f \in U_i$ (resp $x_e \in$ I); This gives two injective functions f_E and f_F. Since Card(F) = Card(I), the function f_F is bijective. Then $f = f_F^{-1} \circ f_E$ is an injection. Assume $x \in$ E, let $y = f(x)$. Then $f_E(x)$ and $f_F(y)$belong to a same U_i, thus are comparable. This says $y \in$ A(x) and proves point (b).

Example. Assume A(x) = $\{a,d\}$, A(y) = $\{b\}$ and A(z) = $\{c\}$. Let E be the set $\{x,y,z\}$ and F the set $\{a,b,c,d\}$. There are are two functions f_a and f_d that are such that $f(t) \in$ A(t) for all t, with $f_a(x) = a$ and $f_d(x) = d$. These functions are injective. Take G = $\{c,d\}$. We have G ⊂ f_d(E); but this is false for f_a. Our function A satisfies the assumptions (b) and (c). Write $\bar{c}$ and $\bar{d}$ instead of a_1 and a_2 for the elements of G, a, b, c, d instead of b_3, b_4, b_5 and b_6, and x, y, z instead of c_7, c_8 and c_9. Instead of the interval [1,9] we consider the set of these nine elements, ordered by $\bar{c} < c < z$, $\bar{d} < d < x$, $a < x$ and $b < y$ (plus the relations that follow by reflexivity and transitivity). The four sets $E_1 = \{a,x\}$, $E_2 = \{\bar{d},d,x\}$, $E_3 = \{b,y\}$ and $E_4 = \{\bar{c},c,z\}$ are totally ordered. They do not form a partition, since x is duplicated. Let's remove x from one of these sets. For each $t \in$ E there is now a unique i such that $x \in E_i$, and this E_i contains a unique element of F, call it $f(t)$. This function is one of f_a or f_d, and is the solution to (b). But only in one case is it a solution to (c). Let's try to analyse condition (c). It says that $c \in$ A(t) for some t, and $d \in$ A(t) for some t (take for L a singleton). These two conditions are equivalent to G ⊂ $\bigcup$A(t), it is necessary since we want G ⊂ F(E) and F(E) ⊂ $\bigcup$A(t). It is not sufficient (move d from A(x) to A(z)). Take for L the whole set G. The condition becomes: there are two distinct elements t_1 and t_2 such that A(t_1) and A(t_2) meet G (in our case, x and z). This second condition is not sufficient (consider what happens when d is replaced by c in A(x)). In this example the conjunction of both cases is enough. Note that the ordering on the disjoint union of E, F and G is isomorphic to the ordering on the disjoint union of E and F (just forget elements that have a bar).

What we should do is the following: given sets X_i as a above, if X_1 contains $a < x$ and X_2 contains $\bar{d} < d < x$, we should keep x in X_2 and remove it from X_1.

7. *An element a of a lattice* E *is said to* irreducible *if the relation* sup(x,y) = a *implies either* $x = a$ *or* $y = a$.

(a) Show that in a finite lattice E *every element a can be written as* sup($e_1,\dots,e_n$) *where*

the e_i ($1 \le i \le n$) are irreducible.

(b) Let E *be a finite lattice and let* J *be the set of its irreducible elements. For each $x \in$ E let* $S(x)$ *be the set of all $y \in$ J which are $\le x$. Show that the mapping $x \to S(x)$ is an isomorphism of* E *onto a subset of* $\mathfrak{P}(J)$*, ordered by inclusion, and that* $S(\inf(x,y)) = S(x) \cap S(y)$.

We start with a lemma that will be useful for the next exercise. Let A and B be two subsets of E, ξ_A and ξ_B the characteristic functions. We have $A \subset B$ if and only if, for each $x \in E$ we have $\xi_A(x) \le \xi_B(x)$. In a lattice, $\sup(A \cup \{b\}) = \sup(\sup A, b)$ provided that $\sup A$ exists.

We introduce some definitions, and show that if x is not irreducible, it can be written as $\sup(a,b)$ where $a < x$ and $b < x$. We also show Exercise 8(a): if x is irreducible and $x \le \sup(a,b)$, then $x \le a$ or $x \le b$, provided that the lattice is distributive (we consider $\inf(x, \sup(a,b))$.

```
Lemma char_fun_sub A A' B:  sub A B -> sub A' B ->
  ((sub A A') <-> (forall x, inc x B ->
     (W x (char_fun A B)) <=c (W x (char_fun A' B)))).
Lemma supremum_singleton r x:  (* 2 *)
   order r -> inc x (substrate r) -> supremum r (singleton x) = x.

Definition sup_irred r x:=
  forall a b, inc a (substrate r) -> inc b (substrate r) ->
   x = sup r a b -> (x = a \/ x = b).

Definition set_of_irreds r := Zo (substrate r)(sup_irred r).

Definition E47S r x := Zo (substrate r)
   (fun z => (sup_irred r z) & (gle r z x)).

Section Irred_latice.

Variable r:Set.
Hypothesis lr:lattice r.

Lemma Exercise4_7a x: inc x (substrate r) -> (* 6 *)
 sup_irred r x \/ (exists a, exists b, glt r a x& glt r b x & x = sup r a b).
Lemma Exercise3_8a a: distributive_lattice3 r -> (* 10 *)
  sup_irred r a ->
  forall x y, inc x (substrate r) -> inc y (substrate r) ->
   gle r a (sup r x y) -> (gle r a x \/ gle r a y).
Lemma supremum_tack_on a b: (* 2 *)
  sub a (substrate r) -> has_supremum r a -> inc b (substrate r) ->
  supremum r (tack_on a b) =  sup r (supremum r a) b.
```

Consider now what happens when E is empty. Point (a) is trivial. In (b) we have to show there is some subset F of $\mathfrak{P}(J)$ isomorphic to E, and we can choose the empty set. In Exercise 8(c), we have to show that F is isomorphic to some set A (see discussion below); this set is empty if E is empty. Exercise 8(c) is trivial. In Exercise 8(d), the case $k = 0$ is trivial, and $k > 0$ implies that E has at least two elements. Therefore, we shall from now on (until the end of the next exercise) assume E non-empty.

We assume from now on that E is finite. Then every non-empty subset of E has a maximal and minimal element. In particular, E has a least element a, which is in J, and in every $S(x)$. Every subset of E has a supremum (which is a if the set is empty).

It follows that $S(x)$ is non-empty, and thus has a least upper bound. Obviously $\sup S(x) \le x$. We pretend that $x = \sup S(x)$, proof by contradiction. Let V be the set of elements x such

that $\sup \mathrm{S}(x) \neq x$. Assume that this set is non-empty, and consider a minimal element. It cannot be an irreducible element. Thus, assume $x = \sup(a, b)$, with $a < x$ and $b < x$. We cannot have $a \in \mathrm{V}$ so that $a \leq \sup(\mathrm{S}(a))$. Since $\mathrm{S}(a) \leq \mathrm{S}(b)$ it follows $a \leq \sup \mathrm{S}(t))$. We deduce $t \leq \sup(\mathrm{S}(t))$.

```
Hypothesis fs: finite_set (substrate r).
Hypothesis nes: nonempty (substrate r).

Definition E48P := complement (set_of_irreds r)
        (singleton (the_least_element r)).

Lemma Exercise4_7b: order r. (* 1 *)
Lemma finite_set_maximal1 U:  (* 8 *)
  sub U (substrate r) -> nonempty U ->
  exists x, inc x U & (forall y, inc y U -> gle r x y -> x = y).
Lemma finite_set_minimal1 U: (* 9 *)
  sub U (substrate r) -> nonempty U ->
  exists x, inc x U & (forall y, inc y U -> gle r y x -> y = x).

Lemma Exercise4_7c: exists a, least_element r a.  (* 6 *)
Lemma Exercise4_7d:  (* 6 *)
  inc (the_least_element r) (set_of_irreds r).
Lemma Exercise4_7e: sub E48P (substrate r).  (* 1 *)
Lemma Exercise4_7f: (* 1 *)
  set_of_irreds r = tack_on E48P (the_least_element r).
Lemma Exercise4_7g a:  inc a (substrate r) -> (* 4 *)
  inc (the_least_element r) (E47S r a).
Lemma Exercise4_7h a: (* 1 *)
   inc a (substrate r) -> nonempty  (E47S r a).
Lemma Exercise4_7i x: sub x (substrate r) -> (* 5 *)
  has_supremum r x.
Lemma Exercise4_7j a: inc a (substrate r) -> (* 4 *)
  gle r (supremum r (E47S r a)) a.
Lemma Exercise4_7k a b:  (* 2 *)
  gle r a b -> sub (E47S r a)  (E47S r b).
Lemma Exercise4_7l a: inc a (substrate r) -> (* 31 *)
  (supremum r (E47S r a)) = a.
Lemma Exercise4_7m a b:  (* 7 *)
  inc a (substrate r) -> inc b (substrate r) ->
  sub (E47S r a)  (E47S r b) -> gle r a b.
```

Statement (a) is equivalent to $x = \sup \mathrm{S}(x)$, together with the assertions that $\mathrm{S}(x)$ is finite and non-empty. Statement (b) is trivial.

```
Lemma Exercise4_7n a: inc a (substrate r) -> (* 6 *)
  exists S, finite_set S & nonempty S & sub S (substrate r) &
  (forall x, inc x S -> sup_irred r x) &
  supremum r S = a.
Lemma Exercise4_7o a b: (* 6 *)
  inc a (substrate r) -> inc b (substrate r) ->
  (E47S r (inf r a b)) = intersection2 (E47S r a) (E47S r b).
Lemma Exercise4_7p: (* 12 *)
  let tg := (set_of_irreds r) in
   order_morphism (BL (E47S r) (substrate r) (powerset tg))
   r (inclusion_order tg).
```

```
Lemma Exercise4_7q a: inc a (substrate r)  (* 1 *)
 -> sub (E47S r a) (set_of_irreds r).
```

¶ 8. *(a) Let* E *be a distributive lattice (§ 1, Exercise 16). If a is irreducible in* E *(Exercise 7), show that the relation* $a \leq \sup(x, y)$ *implies* $a \leq x$ *or* $a \leq y$.

We continue to assume that r is the ordering of a non-empty finite set, and is a lattice. We add the assumption that E. We have seen that this implies point (a). By induction, if X is a finite set and a irreducible, then $a \leq \sup X$ implies $a \leq x$ for some $x \in X$.

Let a be the least element of E,denote by $\bar{X}$ the set $X - \{a\}$. Let $P = \bar{J}$. Let $p(X)$ be the property: X is a non-empty subset of J, and $x \in X$, $y \in J$ and $y \leq x$ imply $y \in X$. Let $q(X)$ be the property: X is a subset of P, and $x \in X$, $y \in P$ and $y \leq x$ imply $y \in X$. If $p(X)$ is true, then $X = S(\sup X)$, for if $y \in S(\sup X)$, there is $x \in X$ such that $y \leq x$, hence $y \in X$. One deduces that $p(X)$ holds if and only if X has the form $S(x)$, and that $q(X)$ holds if and only if X has the form $\bar{S}(x)$.

```
Hypothesis dl3: distributive_lattice3 r.

Lemma Exercise4_8b a b: (* 8 *)
  inc a (substrate r) -> inc b (substrate r) ->
  (E47S r (sup r a b)) = union2 (E47S r a) (E47S r b).
Lemma Exercise4_8c1 t U: inc t (substrate r) -> (* 23 *)
  sup_irred r t ->  sub U (set_of_irreds r) ->  gle r t (supremum r U) ->
  nonempty U ->  exists x, inc x U & gle r t x.
Lemma Exercise4_8c U: sub U (set_of_irreds r) -> (* 12 *)
  (forall x y, inc y U -> sup_irred r x -> gle r x y -> inc x U) ->
   nonempty U ->
  U = (E47S r (supremum r U)).
Lemma Exercise4_8d:  (* 8 *)
  let p:= fun U => (sub U (set_of_irreds r) & nonempty U &
    (forall x y, inc y U -> sup_irred r x -> gle r x y -> inc x U)) in
  (forall x, inc x (substrate r) -> p (E47S r x)) &
  (forall U, p U -> exists x, (inc x (substrate r)) & U = (E47S r x)).
Lemma Exercise4_8e: (* 26 *)
  let comp:= fun X => complement X (singleton (the_least_element r)) in
  let p:= fun U => (sub U E48P &
    (forall x y, inc y U -> inc x E48P -> gle r x y -> inc x U)) in
  (forall x, inc x (substrate r) -> p (comp (E47S r x))) &
  (forall U, p U -> exists x, (inc x (substrate r)) & U = comp (E47S r x)).
```

(b) Let E *be a finite distributive lattice and let* J *be the set of its irreducible elements, ordered by the induced ordering. Show that the isomorphism* $x \to S(x)$ *of* E *onto a subset of* $\mathfrak{P}(J)$ *defined in Exercise 7 (b) is such that* $S(\sup(x, y)) = S(x) \cup S(y)$. *Deduce that if* J^* *is the ordered set obtained by endowing* J *with the opposite ordering, then* E *is isomorphic to the set* $\mathscr{A}(J^*, I)$ *of increasing mappings of* J^* *into* $I = \{0, 1\}$ *(§ 1, Exercise 6).*

The first point is obvious, the second one is wrong. In fact, both functions S and $\bar{S}$ are isomorphisms of E into a subset of $\mathfrak{P}(J)$ or $\mathfrak{P}(P)$. Define $A_1 = \mathscr{A}(J^*, I)$ and $A_2 = \mathscr{A}(P^*, I)$. If a set satisfies property p or q, its characteristic function is in A_1 or A_2. By composition, we get a morphism $E \to A_1$ and $E \to A_2$ (its is injective and order-preserving, not necessarily

surjective). If $f \in A_1$, then either f is the zero function, or $f(a) = 1$. Let A_3 be the subset of A_1 formed of all functions that are not constantly zero. The mapping that associates to each function f its restriction to P is an order isomorphism of A_3 to A_2. We show that E is isomorphic to A_2 and to A_3.

Note the special case $E = \emptyset$. Here A_1 has a single element, and $A_1 - \{0\}$ is empty, so that there is nothing to prove. If E is non-empty, then it has a least element, P and A_2 are well-defined.

We start with a bunch of definitions: the ordered sets J^* and P^*, the sets A_1 and A_2, the zero function and the set A_3. Note that I, considered as an ordered set, is the interval $[0,1]$ of **N**.

```
Definition E48I := doubleton \0c \1c.
Definition E48z := BL (fun z => \0c) (set_of_irreds r) E48I.
Definition E48Ps := opposite_order (induced_order r E48P).
Definition E48Js := opposite_order (induced_order r (set_of_irreds r)).
Definition E48Io :=  interval_Bnato \0c \1c.
Definition E48AJIo := increasing_mappings_order E48Js E48Io.
Definition E48APIo := increasing_mappings_order E48Ps E48Io.
Definition E48AJI := set_of_increasing_mappings E48Js E48Io.
Definition E48API := set_of_increasing_mappings E48Ps E48Io.
Definition E48AJImo:=
  induced_order E48AJIo (complement (substrate E48AJIo) (singleton E48z)).
```

The following lemmas are trivial (except for the characterization of $\mathscr{A}(J^*, I)$) and that of $\mathscr{A}(J^*, I) - \{0\}$

```
Lemma Exercise4_8f K: sub K (substrate r) ->  (* 3 *)
 let o :=  opposite_order (induced_order r K) in
   order o & (substrate o = K).
Lemma Exercise4_8g:   (* 2 *)
  order E48Js & substrate E48Js =  (set_of_irreds r)
  &  order E48Ps  & substrate E48Ps = E48P.
Lemma Exercise4_8h: order E48Io &  substrate  E48Io = E48I. (* 8 *)
Lemma Exercise4_8i K: sub K (substrate r) ->  (* 24 *)
  let o :=  opposite_order (induced_order r K) in
  let A:= set_of_increasing_mappings o E48Io in
 forall f, inc f A <->
 (is_function f & source f = K & target f =  E48I &
  (forall i j, inc i K -> inc j K -> gle r i j ->
     W j f = \1c -> W i f = \1c)).
Lemma Exercise4_8j K: sub K (substrate r) ->
  let o :=  opposite_order (induced_order r K) in
  let A:= set_of_increasing_mappings o E48Io in
  let no := increasing_mappings_order o E48Io in
  order no & substrate no = A.
Lemma Exercise4_8k: (* 2 *)
  order E48AJIo & substrate E48AJIo = E48AJI &
  order E48APIo & substrate E48APIo = E48API.
Lemma Exercise4_8l: inc E48z (substrate E48AJIo). (* 7 *)
Lemma Exercise4_8m:  (* 23 *)
 order E48AJImo & substrate E48AJImo = (complement E48AJI (singleton E48z))&
 forall f, inc f (substrate E48AJImo) <->
    (inc f E48AJI & W (the_least_element r) f = \1c).
```

We now show that the characteristic functions of $S(x)$ and $\bar{S}(x)$ are in A_3 and A_2.

```
Lemma  Exercise4_8n x: inc x (substrate r) ->  (* 13 *)
  inc (char_fun (E47S r x) (set_of_irreds r)) (substrate E48AJImo).
Lemma Exercise4_8o x: inc x (substrate r) -> (* 11 *)
  let comp:= fun X => complement X (singleton (the_least_element r)) in
  inc (char_fun (comp (E47S r x)) E48P) E48API.
Lemma Exercise4_8p f: (* 8 *)
   is_function f -> target f = E48I ->
   char_fun (inv_image_by_fun f (singleton \1c)) (source f) = f.
Lemma Exercise4_8q A A' B:  sub A B -> sub A' B -> (* 16 *)
  ((sub A A') <-> (forall x, inc x B ->
    gle E48Io (W x (char_fun A B)) (W x (char_fun A' B)))).
Lemma Exercise4_8r: order_isomorphic r E48APIo.  (* 53 *)
Lemma Exercise4_8s: order_isomorphic r E48AJImo. (* 60 *)
```

(c) With the hypothesis of (b), let P *be the set of elements of* J *other than the least element of* E. *For each* $x \in E$, *let* $y_1, \ldots, y_k$ *be the distinct minimal elements of the interval* $]x, \rightarrow[$ *in* E; *for each index* i, *let* q_i *be an element of* P *such that* $q_i \notin S(x)$ *and* $q \in S(y_i)$. *Show that no two elements* $q_1, \ldots, q_k$ *are comparable.*

(d) Conversely, let $q_1, \ldots, q_k$ *be* k *elements of* P, *no two of which are comparable. Let* $u = \sup(q_1, \ldots, q_k)$ *and let*

$$v_i = \sup_{1 \le j \le k, j \ne i} (q_j) \qquad (1 \le i \le k).$$

Show that $v_i < u$ *for* $1 \le i \le k$. *Let* $x = \inf(v_1, \ldots, v_k)$ *and let*

$$y_i = \inf_{1 \le j \le k, j \ne i} (v_j)$$

Show that $x < y_i$ *for each index* i, *and deduce that the interval* $]x, \rightarrow[$ *has at least* k *distinct minimal elements.*

For each x we consider the set $A(x)$ of all $a > x$ such that, whenever b satisfies $x < b \le a$ we have $b = a$. We consider the property $p(K)$ that says that K is a subset of P and the conditions "$a \in K$, $b \in K$, $a \le b$" imply $a = b$. Condition (c) can be restated as: for each x, there is a set $K(x)$ satisfying p, and a bijection $f : A(x) \to K(x)$. We choose $f(y)$ to be some z such that $z \in S(y) - S(x)$. It exists since S is strictly increasing. Note that z cannot be the least element of E, thus is in P. We have $\sup(x, f(y)) = y$ since y is minimal. Thus f is the desired condition.

Point (d) shows that, for any K that satisfies p, there is an x, such that $A(x)$ has at least as many elements as K. Consider $f : K \to E$ that maps x to $\sup(K - \{x\})$. Let $u = \sup K$. We have $\sup(x, f(x)) = u$. This implies $f(x) \le u$. We have $f(x) < u$, for otherwise we would have $x \le f(x)$. If K is not $\{x\}$ there an element y of $K - \{x\}$ such that $x \le y$, contradicting $p(K)$. Otherwise, we get $u = x$. If $x \ne y$ we have $x \le f(y)$; we deduce $\sup(f(x), f(y)) = u$. This implies injectivity of f. Let L be the image of f. This set has the same number of elements as K.

We define now $g(x) = \inf(L - \{x\})$, $v = \inf(L)$. For any $x \in L$, we have $v = \inf(x, g(x))$. From this we deduce $v < g(x)$, for otherwise we would have $g(x) \le x$. This says $\sup(g(x), x) = x$. Using distributivity, we get $\inf\{k(y)\}_y = x$, where $k(y) = \sup(y, x)$. Since x and y are distinct in L we have $k(y) = v$, $\{k(y)\}_y = \{v\}$, and $v = x$, absurd. Since the set of all z such that $v < z \le g(x)$ is non-empty, it has a minimal element, call it $h(x)$. We have $h(x) \in A(v)$. Note that h is injective; if $h(x) = h(y)$ we have $h(x) \le g(x)$ and $h(x) \le g(y)$, thus $h(x) \le \inf(g(x), g(y))$. But $x \ne y$ implies $\inf(g(x), g(y)) = \inf(L) = v$. Thus, $A(v)$ has at least as many elements as L.

```
Definition all_uncomp_inP K :=
  sub K E48P & forall x y, inc x K -> inc y K -> gle r x y -> x = y.

Definition minimal_in_int x a :=
   glt r x a &  (forall b, glt r x b -> gle r b a -> b = a).

Definition set_of_minimal x :=  Zo (substrate r) (minimal_in_int x).

Lemma Exercise4_7c1:  exists a, greatest_element r a. (* 6 *)
Lemma Exercise4_7i1 x: sub x (substrate r) -> (* 5 *)
  has_infimum r x.
Lemma Exercise4_8u: infimum r emptyset = the_greatest_element r.
Lemma infimum_tack_on a b: (* 2 *)
  sub a (substrate r) -> has_infimum r a -> inc b (substrate r) ->
  infimum r (tack_on a b) =  inf r (infimum r a) b.
Lemma distributive_rec x T:
  inc x (substrate r) -> sub T (substrate r) ->
  sup r x (infimum r T) = infimum r (fun_image T (fun z => sup r z x)).

Lemma Exercise4_8t x: inc x (substrate r) -> (* 41 *)
  exists K, all_uncomp_inP K & equipotent K (set_of_minimal x).
Lemma Exercise4_8v K: all_uncomp_inP K -> exists x, (* 151 *)
  inc x (substrate r) &
   (cardinal K) <=c (cardinal (set_of_minimal x)).

End Irred_lattice.
```

¶ 9. A subset A of a lattice E is said to be a *sublattice* if for each pair (x, y) of elements of A, $\sup_E(x, y)$ and $\inf_E(x, y)$ belong to A.

(a) Let $(C_i)_{1\le i\le n}$ be a finite family of totally ordered sets and let $E = \prod_{i=1}^{n} C_i$ be their product. Let A be a sublattice of E. Show that A cannot have more than n irreducible elements (Exercise 7) no two of which are comparable (The proof is by *reductio ad absurdum*. Suppose that there exist $r > n$ irreducible elements $a_1, \dots, a_n$ in A, no two of which are comparable. Consider the elements $u = \sup(a_1, \dots, a_n)$ and

$$v_j = \inf_{1\le j\le r, j\ne i}(a_j)$$

of A. By projecting onto the factors, show that $u = v_i$ for some index i, and hence that two of the a_i are comparable).

(b) Conversely, let F be a finite distributive lattice, let P be the set of irreducible elements of F other than the least element of F, and suppose that n is the greatest number of elements in a free subset of P (§ 1, Exercise 5). Show that F is isomorphic to a sublattice of a product of n totally ordered sets (Apply Exercise 5, which shows that P is the union of n totally ordered sets P_i with no elements in common. Let C_i be the totally ordered set obtained by adjoining a least element to P_i $(1 \le i \le n)$. With each $x \in F$ associate the family $(x_i)_{1\le i\le n}$ where x_i is the least upper bound in C_i of the sets of elements of P_i which are $\le x$.)

¶ 10. (a) An ordered set E is isomorphic to a subset of a product of n totally ordered sets if and only if the graph of the ordering on E is the intersection of the graphs on n total orderings

on E. (To show that the condition is necessary, show that if $F = \prod_{i=1}^{n} F_i$ is a product of n totally ordered sets, the the graph of the product ordering on F is the intersection of n graphs of lexicographic orderings on F.)

(b) An ordered set E is isomorphic to a subset of the product of two totally ordered sets if and only if the ordering Γ on E is such that there exists another ordering Γ' on E with the property that any two distinct elements of E are comparable with respect to exactly one of the orderings Γ and Γ'.

(c) Let A be a finite set of n elements. Let E be the subset of $\mathfrak{P}(A)$ consisting of all subsets $\{x\}$ and $A-\{x\}$ as x runs through A. Show that n is the smallest integer m such that E, ordered by inclusion, is isomorphic to a subset of a product of m totally ordered sets (use (a)).

¶ 11. Let A be a set and let $\mathfrak{R}$ be a subset of the set $\mathfrak{F}(A)$ of finite subsets of A. $\mathfrak{R}$ is set to be *mobile* if it satisfies the following condition:
(MO) If X, Y are two distinct elements of $\mathfrak{R}$ and if $z \in X \cap Y$, then there exists $Z \subset X \cap Y$ belonging to $\mathfrak{R}$ such that $z \notin Z$.
A subset P of A is then said to be *pure* if it contains no set belonging to $\mathfrak{R}$.

(a) Show that every pure subset of A is contained in a maximal pure subset of A.

(b) Let M be a maximal pure subset of A. Show that for each $x \in \complement M$ there exists a unique finite subset $E_M(x)$ of M such that $E_M(x) \cup \{x\} \in \mathfrak{R}$. Moreover, if $y \in E_M(x)$, the set $(M \cup \{x\}) - \{y\}$ is a maximal pure subset of A.

(c) Let M, N be two maximal pure subsets of A, such that $N \cap \complement M$ is finite. Show that Card(M) = Card(N). (Proof by induction on the cardinal of $N \cap \complement M$, using (b).)

(d) Let M, N be two maximal pure subsets of A, and put $N' = N \cap \complement M$, $M' = M \cap \complement N$. Show that $M' \subset \bigcap_{x \in N'} E_M(x)$. * Deduce that Card(M) = Card(N) (by virtue of (c), we are reduced to the case where N' and M' are infinite; show then that $\mathrm{Card}(M') \leq \mathrm{Card}(N')$). *

10.5 Section 5

1. Prove the formula

$$\sum_{k=q+1}^{n-p+q+1} \binom{n-k}{p-q-1}\binom{k-1}{q} = \binom{n}{p},$$

where $p \leq n$ and $q < p$ (generalize the argument of no. 8, Corollary to Proposition 14).

2. If $n \geq 1$, prove the relation

$$\binom{n}{0} - \binom{n}{1} + \binom{n}{2} - \ldots + (-1)^n \binom{n}{n} = 0$$

(Define a one-to-one correspondence between the set of subsets of $[1, n]$ which have an even number of elements, and the set of subsets of $[1, n]$ which have an odd number of elements. Distinguish between the cases n even and n odd.)

3. Prove the relations

$$\binom{n}{0}\binom{n}{p} + \binom{n}{1}\binom{n-1}{p-1} + \binom{n}{2}\binom{n-2}{p-2} + \cdots + \binom{n}{p}\binom{n-p}{0} = 2^p \binom{n}{p},$$

$$\binom{n}{0}\binom{n}{p}-\binom{n}{1}\binom{n-1}{p-1}+\binom{n}{2}\binom{n-2}{p-2}-\cdots+(-1)^p\binom{n}{p}\binom{n-p}{0}=0.$$

(Consider the subsets of p elements of $[1,n]$ which contain a given subset of k elements ($0\leq k\leq p$), and use Exercise 2 for the second formula.)

4. *Prove Proposition 15 of no. 8 by defining a bijection of the set of mappings u of $[1,h]$ into $[0,n]$ such that*

$$\sum_{i=1}^{h}u(x)\leq n$$

onto the set of strictly increasing mappings of $[1,h]$ into $[1,n+h]$.

5. * (a) Let E be a distributive lattice and let f be a mapping of E into a commutative semigroup M (written additively) such that

$$f(x)+f(y)=f(\sup(x,y))+f(\inf(x,y))$$

for all x, y in E. Show that for each finite subset I of E, we have

$$f(\sup(\mathrm{I}))+\sum_{2n\leq\mathrm{Card(I)}}\Big(\sum_{\mathrm{H}\subset\mathrm{I},\mathrm{Card(H)}=2n}f(\inf(\mathrm{H}))\Big)=\sum_{2n+1\leq\mathrm{Card(I)}}\Big(\sum_{\mathrm{H}\subset\mathrm{I},\mathrm{Card(H)}=2n+1}f(\inf(\mathrm{H}))\Big)$$

(By induction on Card(I)).) *

(b) In particular let A be a set, let $(\mathrm{B}_i)_{i\in\mathrm{I}}$ be a finite family of finite subsets of A and let B be the union of the B_i. For each subset H of I, put $\mathrm{B_H}=\bigcap_{i\in\mathrm{H}}\mathrm{B}_i$. Show that

$$\mathrm{Card(B)}+\sum_{2n\leq\mathrm{Card(I)}}\Big(\sum_{\mathrm{Card(H)}=2n}\mathrm{Card(B_H)}\Big)=\sum_{2n+1\leq\mathrm{Card(I)}}\Big(\sum_{\mathrm{Card(H)}=2n+1}\mathrm{Card(B_H)}\Big).$$

6. Prove the formula

$$\binom{n+h}{h}=1+\binom{h}{1}\binom{n+h-1}{h}-\binom{h}{2}\binom{n+h-2}{h}+\ldots+(-1)^h\binom{h}{h}\binom{n}{h}.$$

(If F denotes the set of mappings u of $[1,h]$ into $[0,n]$ such that $\sum_{x=1}^{h}u(x)\leq n$, consider for each subset H of $[1,h]$ the set of all $u\in\mathrm{F}$ such that $u(x)\geq 1$ for each $x\in\mathrm{H}$, and use Exercise 5.)

7. (a) Let $\mathrm{S}_{n,p}$ denote the number of mappings of $[1,n]$ *onto* $[1,p]$. Prove that

$$\mathrm{S}_{n,p}=p^n-\binom{p}{1}(p-1)^n+\binom{p}{2}(p-2)^n-+\ldots+(-1)^{p-1}\binom{p}{p-1}.$$

(Note that $p^n=\mathrm{S}_{n,p}=\binom{p}{1}\mathrm{S}_{n,p-1}+\binom{p}{2}\mathrm{S}_{n,p-2}+\ldots+\binom{p}{p-1}$ and use Exercise 3.)

(b) Prove that $\mathrm{S}_{n,p}=p(\mathrm{S}_{n-1,p}+\mathrm{S}_{n-1,p-1})$ (method of no. 8, Proposition 13).

(c) Prove that

$$\mathrm{S}_{n+1,n}=\frac{n}{2}(n+1)!\qquad\text{and}\qquad\mathrm{S}_{n+2,n}=\frac{n(3n+1)}{24}(n+2)!$$

(consider the elements r of $[1,n]$ whose inverse image consists of more than one element).

(d) If $\mathrm{P}_{n,p}$ is the number of partitions into p parts of a set of n elements, show that $\mathrm{S}_{n,p}=p!\mathrm{P}_{n,p}$.

8. Let p_n be the number of permutations of a set E with n elements such that $u(x) \neq x$ for all $x \in E$. Show that

$$p_n = n! - \binom{n}{1}(n-1)! + \binom{n}{2}(n-2)! - \cdots + (-1)^n$$

* and hence that $p_n \sim n!/e$ as $n \to \infty$ * (same method as in Exercise 7 (a)).

9. (a) Let E be a set with qn elements. Show that the number of partitions of E into n subsets each of q elements is equal to

$$(qn)!/(n!(q!)^n).$$

(b) Suppose that $E = [1, qn]$. Show that the number of partitions of E into n subsets each of q elements, no one of which is an interval is equal to

$$\frac{(qn)!}{n!(q!)^n} - \frac{(qn-q+1)!}{1!(n-1)!(q!)^{n-1}} + \frac{(qn-2q+2)!}{2!(n-2)!(q!)^{n-2}} - \cdots + (-1)^n$$

(same method as in Exercises 7 and 8).

10. Let $q_{n,k}$ be the number of strictly increasing mappings u of $[1,k]$ into $[1,n]$ such that for each even (resp. odd) x, $u(x)$ is even (resp. odd). Show that $q_{n,k} = q_{n-1,k-1} + q_{n-2,k}$ and deduce that

$$q_{n,k} = \binom{[\frac{n+k}{2}]}{k}.$$

¶ 11. Let E be a set with n elements and let S be a set of signs such that S is the disjoint union of E and a set consisting of a single element f. Suppose that f has weight 2 and that each element of E has weight 0 (Chapter I, Appendix Exercise 3).

(a) Let M be the set of significant words in $L_0(S)$ which contain each element of E exactly once. Show that if u_n is the number of elements in M, then $u_{n+1} = (4n-2)u_n$, and deduce that

$$u_n = 2.6\ldots(4n-6) \qquad (n \geq 2)$$

(This is the number of products of n different terms with respect to a non-associative law of composition).

(b) let x_i be the ith of the elements of E which appear in a word of M. Show that the number v_n of words of M, for which the sequence x_i is given, is equal to $\binom{2n-2}{n-1}/n$ and satisfies the relation

$$v_{n+1} = v_1 v_n + v_2 v_{n-1} + \cdots + v_{n-1} v_2 + v_n v_1.$$

¶ 12. (a) let p and q be two integers ≥ 1, let $n = 2p+q$, let E be a set with n elements and let $N = \binom{n}{p} = \binom{n}{p+q}$. Let $(X_i)_{1 \leq i \leq N}$ (resp $(Y_i)_{1 \leq i \leq N}$) be the sequence of all subsets of E which have p (resp $p+q$) elements arranged in a certain order. Show that there exists a bijection ϕ of $[1,n]$ onto itself such that $X_{\phi(i)} \subset Y_i$ for all i. (The method is analogous to that of Exercise 6 of § 4: observe that for each $r \leq N$ the number of sets Y_j which contain at least one of X_1, …, X_r is $\geq r$).

(b) Let h, k be two integers ≥ 1, let n be an integer such that $2h+k < n$, let E be a set with n elements and let $(X_i)_{1 \leq i \leq r}$ be a sequence of distinct subsets of E, each having h elements. Show that there exists a sequence $(Y_j)_{1 \leq j \leq r+1}$ of distinct subsets of E, each having $h+k$ elements, such that each Y_j contains at least one X_i and each X_i is contained in at least one Y_j (by induction on n, using (a)).

¶ 13. Let E be set with $2m$ elements, let q be an integer $< m$, and let $\mathscr{F}$ be the set of all subsets $\mathfrak{S}$ of $\mathfrak{P}(E)$ with the following property: if X and Y are two distinct elements of $\mathfrak{S}$ such that $X \subset Y$, then $Y - X$ has at most $2q$ elements.

(a) Let $\mathfrak{M} = (A_i)_{1\leq i\leq p}$ be an element of $\mathscr{F}$ such that $p = \mathrm{Card}(\mathfrak{M})$ is as large as possible. Show that $m - q \leq \mathrm{Card}(A_i) \leq m + q$ for $1 \leq i \leq p$ (Argue by contradiction. Suppose, for example, that there exists indices i such that $\mathrm{Card}(A_i) < m - q$ and consider those of the A_i for which $\mathrm{Card}(A_i)$ has the least possible value $m-q-s$ (where $s \geq 1$). Let $A_1, \ldots, A_r$, say, these sets. Let $\mathfrak{G}$ be the set of subsets of E each of which is the union of some A_i $(1 \leq i \leq r)$ and a subset of $2q+1$ elements contained in $E - A_i$. Show that $\mathfrak{G}$ contains at least $r+1$ elements (cf. Exercise 12), and that if $B_1, \ldots, B_{r+1}$ are $r+1$ distinct elements of $\mathfrak{G}$, the set whose elements are B_j ($1 \leq j \leq r+1$ and A_i ($r+1 \leq i \leq p$) belongs to $\mathscr{F}$, contrary to the hypothesis.)

(b) Deduce from (a) that the number of elements p of each $\mathfrak{S} \in \mathscr{F}$ satisfies the inequality

$$p \leq \sum_{k=0}^{2q} \binom{2m}{m-q+k}.$$

(c) Establish results analogous to those of (a) and (b) when $2m$ or $2q$ is replaced by an uneven number.

¶ 14. Let E be a finite set with n elements, let $(a_j)_{1\leq j\leq n}$ be the sequence of elements of E arranged in some order, and let $(A_i)_{1\leq i\leq m}$ be a sequence of subsets of E.

(a) For each index j, let k_j be the number of indices i such that $a_j \in A_i$, and let $S_i = \mathrm{Card}(A_i)$. Show that

$$\sum_{j=1}^{n} k_j = \sum_{i=1}^{m} s_i.$$

(b) Suppose that for each subset $\{x, y\}$ of two elements of E, there exists exactly one index i such that x and y are contained in A_i. Show that, if $a_j \notin A_i$, then $S_i \leq k_j$.

(c) With the hypotheses of (b), show that $m \geq n$ (Let k_n be the least of the numbers k_j. Show that we may suppose that, whenever $i \leq k_n$, $j \leq k_n$, and $i \neq j$, we have $a_j \notin A_i$ and $a_n \notin A_j$ for all $j \geq k_n$.)

(d) With the hypotheses of (b), show that $m = n$ if and only if one of the following two alternatives is true: (i) $A_1 = \{a_1, a_2, \ldots, a_{n-1}\}$, $A_i = \{a_{i-1}, a_n\}$ for $i = 2, \ldots, n$; (ii) $n = k(k-1)+1$; each A_i has k elements, and each element of E belongs to exactly k set A_i.

¶ 15. Let E be a finite set, let $\mathfrak{L}$ and $\mathfrak{C}$ be two disjoint non-empty subsets of $\mathfrak{P}(E)$, and let λ, h, k, l be four integers ≥ 1 with the following properties: (i) for each $A \in \mathfrak{L}$ and each $B \in \mathfrak{C}$, $\mathrm{Card}(A \cap B) \geq \lambda$; (ii) for each $A \in \mathfrak{L}$, $\mathrm{card}(A) \geq h$; (iii) for each $B \in \mathfrak{C}$, $\mathrm{card}(B) \leq k$; (iv) for each $x \in E$ the number of elements of $\mathfrak{L} \cup \mathfrak{C}$ which contain x is exactly l. Show that $\mathrm{Card}(E) \leq hk/\lambda$. (Let $(a_i)_{1\leq i\leq n}$ be the sequence of distinct elements of E arranged in some order, and for each i let r_i be the number of elements of $\mathfrak{L}$ to which a_i belongs. Show that, if $\mathrm{Card}(\mathfrak{L}) = s$ and $\mathrm{Card}(\mathfrak{C}) = t$, then we have

$$\sum_{i=1}^{n} r_i \leq sh, \qquad \sum_{i=1}^{n} (l - r_i) \leq tk, \qquad \sum_{i=1}^{n} r_i(l - r_i) \geq \lambda st.)$$

For $\mathrm{Card}(E)$ to be equal to hk/λ it is necessary and sufficient that for each $A \in \mathfrak{L}$ and each $B \in \mathfrak{C}$ we have $\mathrm{card}(A) = h$, $\mathrm{card}(B) = k$, $\mathrm{Card}(A \cap B) = \lambda$ and that there exists an $r \leq l$ such that for each $x \in E$ the number of elements of $\mathfrak{L}$ to which x belongs is equal to r.

16. Let E be a finite set with n elements, let $\mathfrak{D}$ be a non-empty subset of $\mathfrak{P}(E)$, and let λ, k, l be three integers ≥ 1 with the following properties: (i) if A and B are distinct elements of $\mathfrak{D}$, then $\mathrm{Card}(A \cap B) = \lambda$; (ii) for each $A \in \mathfrak{D}$, $\mathrm{card}(A) \leq k$; (iii) for each $x \in E$ the number of elements of $\mathfrak{D}$ to which x belongs is equal to l. Show that

$$n(\lambda - 1) \leq k(k-1)$$

and that if $n(\lambda-1) = k(k-1)$ then $\lambda = k$ and $\mathrm{Card}(\mathfrak{D}) = n$. (Given $a \in E$, let $\mathfrak{L}$ be the set of all $A - \{a\}$ where $A \in \mathfrak{D}$ and $a \in A$, and let $\mathfrak{C}$ be the set of all $A \in \mathfrak{D}$ such that $a \notin A$. Apply the results of Exercise 15 to $\mathfrak{L}$ and $\mathfrak{C}$.

¶ 17. Let i, h, k be three integers such that $i \geq 1$, $h \geq i$, $k \geq i$. Show that there exists an integer $m_i(h,k)$ with the following properties: for each finite set E with at least $m_i(h,k)$ elements, and each partition $(\mathfrak{X}, \mathfrak{Y})$ of the set $\mathfrak{F}_i(E)$ of subsets of i elements of E, it is impossible that every subset of h elements of E contains a subset $X \in \mathfrak{X}$ and that every subset of k elements of E contains a subset $Y \in \mathfrak{Y}$; in other words, if every subset of h elements of E contains some $X \in \mathfrak{X}$ there exists a subset A of k elements of E such that every subset of i elements of A belongs to $\mathfrak{X}$ (Proof by induction. Show that we may take $m_1(h,k) = h+k-1$, $m_i(i,k) = k$ and $m_i(h,i) = h$ and finally $m_i(h,k) = m_{i-1}(m_i(h-1,k), m_i(h,k-1)) + 1$. If E is a set with $m_i(h,k)$ elements if $a \in E$ and $E' = E - \{a\}$, show that if the proposition were false, then every subset of $m_i(h-1,k)$ elements of E' would contain a subset X' of $i-1$ elements such that $X' \cup \{a\} \in \mathfrak{X}$, and that every subset of $m_i(h,k-1)$ elements of E' would contain a subset Y' of $i-1$ elements such that $Y' \cup \{a\} \in \mathfrak{Y}$.

18. (a) Let E be a finite ordered set with p elements. If m, n are two integers such that $mn < p$, show that E has either a totally ordered subset of m elements or else a free subset (§ 1, Exercise 5) of n elements (use § 4, Exercise 5).

(b) Let h, k be two integers ≥ 1 and let $r(h,k) = (h-1)(k-1)+1$. Let I be a totally ordered set with at least $r(h,k)$ elements. Show that, for each finite sequence $(x_i)_{i\in I}$ of elements of a totally ordered set E, there exists either a subset H of h elements of I such that the sequence $(x_i)_{i\in H}$ is increasing, or else a subset K of k elements of I such that the sequence $(x_i)_{i\in H}$ is decreasing. (Use (a) applied to $I \times E$.)

10.6 Section 6.

1. *A set* E *is infinite if and only if for each mapping* f *of* E *into* E *there exists a non-empty set* S *of* E *such that* $S \neq E$ *and* $f(S) \subset S$.

Assume E non-empty, and let $f : E \to E$. Fix $x \in E$, and define by induction $g(n+1) = f(g(n))$, with $g(0) = x$. Let G be the range (or target) of g and $S = f(G)$. Then G and S are non-empty and stable by f. If $x \notin S$, then $S \neq E$. Otherwise, $x = f(g(m)) = g(m+1)$. Let $n = m+1$. Then $x = g(n)$ and by induction on i, $g(i) = g(i+n)$. By induction on k, we have $g(i) = g(i+kn)$. By Euclidean division, every element of S has the form $g(i)$ for $i < n$. This shows that S is finite. If E is infinite, we deduce $S \neq E$.

Conversely, assume that for every function $f : E \to E$ there is a non-trivial subset of E invariant by f. This implies E non-empty. Assume E finite. There is a bijection $T : [0, n[\to E$, where $n \neq 0$. Let f be the function $i \mapsto i+1$ (modulo n). This induces a function g on E, so that there is a set S such that $g(S) \subset S$. Since this set is not empty, it contains an element $T(i)$. By induction it contains all $T(i+j \pmod n)$, thus all elements of E.

```
Lemma Exercise_6_1 E: infinite_set E <-> (* 130 *)
  (forall f, is_function f -> source f = E -> target f = E ->
    exists F, sub F E & nonempty F & F <> E & sub (image_by_fun f F) F).
```

---

2. *Show that, if* $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$ *and* $\mathfrak{d}$ *are four cardinals such that* $\mathfrak{a} < \mathfrak{c}$ *and* $\mathfrak{b} < \mathfrak{d}$ *then* $\mathfrak{a}+\mathfrak{b} < \mathfrak{c}+\mathfrak{d}$ *and* $\mathfrak{a}\mathfrak{b} < \mathfrak{c}\mathfrak{d}$. *(cf. Exercise 21 (c)).*

Exercise 21.c gives an example where $\mathfrak{a}^{\mathfrak{b}} < \mathfrak{c}^{\mathfrak{d}}$ is false. In this case, we use commutativity, and may assume $\mathfrak{c} \leq \mathfrak{d}$. We may also assume $\mathfrak{d}$ infinite, for otherwise all four cardinals are finite. We use the property that $A+B = AB = B$ if B is an infinite cardinal and $0 < A \leq B$. We use it first with $B = \mathfrak{d}$, in order to simplify the goal to $\mathfrak{a}+\mathfrak{b} < \mathfrak{d}$ and $\mathfrak{a}\mathfrak{b} < \mathfrak{d}$. This is obvious if none of $\mathfrak{a}$, $\mathfrak{b}$ is infinite. Otherwise, we apply the previous rule (using commutativity if needed).

```
Lemma exercice6_2 a b c d:
  a <c c -> b <c d -> ((a +c b) <c (c +c d) & (a *c b) <c (c *c d)).
```

---

3. *If* E *is an infinite set, the subsets of* E *which are equipotent to* E *is equipotent to* $\mathfrak{P}(E)$ *(use Proposition 3 of no. 4).*

If n is the cardinal of E, we have $n = n+n$ so that there is a bijection $f : E_1 \cup E_2 \to E$, where $E_1 = E \times \{\alpha\}$ and $E_2 = E \times \{\beta\}$. For each subset X of E, let $\tilde{X} = X \times \{\alpha\}$), and let $g(X) = f\langle \tilde{X} \cup E_2 \rangle$. This a subset of E, and its cardinal is at least the cardinal of $f(E_2)$, which is the cardinal of E, so that $g(X)$ is equipotent to E, thus is in the set Q of subsets of E equipotent to E. The conclusion follows from the injectivity of g and the relation $Q \subset \mathfrak{P}(E)$.

```
Lemma Exercise6_3 E: infinite_set E -> (* 52 *)
  (powerset E) \Eq (Zo (powerset E) (fun z => z \Eq E)).
```

---

4. *If* E *is an infinite set, the set of all partitions of* E *is equipotent to* $\mathfrak{P}(E)$ *(associate a subset of* $E \times E$ *with each partition of* E*).*

Let ϖ be a partition, and $\tilde{\varpi}$ be the union of all $A \times A$ with $A \in \varpi$. We know that $\tilde{\varpi} \supset \tilde{\varpi}'$ is an order (see chapter one), so that the function $\varpi \mapsto \tilde{\varpi}$ is injective. Let Q be the set of partitions; this shows $\mathrm{Card}(Q) \leq \mathrm{Card}(\mathfrak{P}(E \times E))$, hence $\mathrm{Card}(Q) \leq \mathrm{Card}(\mathfrak{P}(E)$.

Conversely, consider the mapping $f : I \mapsto \{I, E-I\}$. Note that $f(I) = f(E-I)$ so that f is obviously not injective. Since E is infinite there exists $y \in E$. Let $F = E - \{y\}$. Then f is injective on $\mathfrak{P}(F)$. Assume moreover I non-empty. Then $f(I)$ is a partition of E. All that remains to do is to show that $\mathrm{Card}(\mathfrak{P}(E-\{y\}) - \{\emptyset\}) = \mathrm{Card}(\mathfrak{P}(E))$. This shows that the set of all mappings of E onto F and the set of all mappings of E into F are equipotent.

```
Lemma infinite_powerset E: (* 7 *)
  infinite_set E -> infinite_set (powerset E).
Lemma Exercise6_4 E: infinite_set E -> (* 58 *)
  (set_of_partition_set E) \Eq (powerset E).
```

---

5. *If* E *is an infinite set, the set of all permutations of* E *is equipotent to* $\mathfrak{P}$(E). *(Use Proposition 3 of No. 4 to show that, for each subset* A *of* E *whose complementary does not consist of a single element, there exists a permutation* f *of* E *such that* A *is the set of elements of* E *which are invariant under* f*.)*

Let G be the set of functions defined on a subset X of E. Then card(F^E) ≤ Card(G) ≤ Card($\mathfrak{P}$(E × F)). If E is non-empty, F infinite, Card(E) ≤ Card(F) then F × E is equipotent to F. Thus Card(G) ≤ Card($\mathfrak{P}$(F)). In particular, if E is an infinite set, and P is the set of permutations of E, we have Card(P) ≤ Card($\mathfrak{P}$(E)).

```
Lemma infinite_doubleton F: (* 3 *)
  infinite_set F -> exists a, exists b, (inc a F & inc b F & a <> b).
Lemma product2_infinite3 E F: nonempty E -> (* 7 *)
   (cardinal E) <=c (cardinal F) -> infinite_set F ->
   (product F E) \Eq F.

Lemma Exercise6_5a E F: (* 2 *)
   (cardinal (set_of_functions E F))
   <=c  (cardinal (set_of_sub_functions E F)).
Lemma Exercise6_5b E F:  (* 9 *)
  (cardinal (set_of_sub_functions E F))
   <=c (cardinal (powerset (product E F))).
Lemma Exercise6_5d E: infinite_set E -> (* 20 *)
   (cardinal (Zo (set_of_functions E E) bijection))
   <=c (cardinal (powerset E)).
```

Let's consider the property H(E): There is a permutation of E with no fixed point. By composition, if E is equipotent to F, then H(F) holds. The property H(**N**) holds: define $f(x)$ by $x-1$ if x is odd and $x+1$ otherwise. Since 0 is even, we apply $x-1$ only to non-zero integers, so that f has no fix-point. Note that $f(f(x)) = x$ so that f is a bijection.

Assume that A is a finite set, not reduced to a singleton. Then H(A) holds. We may assume that A is the interval $[0, n[$, with $n \in \mathbf{N}$. The case $n = 0$ is trivial, the case $n = 1$ is excluded. Let $f(x) = 0$ if $x = n-1$ and $f(x) = x+1$ otherwise. Then f is surjective, and bijective as A is finite. It has no fix-point since $n \neq 1$. Note that the set of non-fix-points of a bijection cannot be a singleton: if z is not invariant, then $f(z)$ is not invariant either, and these elements are distinct.

Any non-singleton has a permutation without fix-point: If it is finite, we apply the previous result; otherwise the set is the disjoint union of copies of **N**, on each copy we consider a permutation, and glue them together.

```
Definition no_fix_perm E f :=
  inc f (set_of_functions E E) & bijection f & set_of_invariants f = emptyset.

Lemma Exercice6_5e E F: (* 20 *)
  equipotent E F -> (exists f, no_fix_perm F f)  ->
  exists g,  no_fix_perm E g.
Lemma Exercice6_5f: exists f, no_fix_perm Bnat f. (* 31 *)
Lemma Exercice6_5g E: finite_set E ->  (* 41 *)
  is_singleton E \/ exists f, no_fix_perm E f.
```

```
Lemma Exercice6_5h E: (* 39 *)
   is_singleton E \/ exists f, no_fix_perm E f.
```

We want to show that Card(P) ≥ Card($\mathfrak{P}$(E)). Let R be the set of subsets of E that are not singletons. Note that $\mathfrak{P}$(E) is the disjoint union of R and a set equipotent to E. Since $\mathfrak{P}$(E) is infinite and its cardinal is > Card(E) it follows that Card(R) = Card($\mathfrak{P}$(E)). If X ∈ R there is a permutation of X without fix-points. Extent this function to E by defining $f(x) = x$ for $x \in$ E − X. This induced a surjection P → R.

```
Lemma Exercice6_5i E: infinite_set E -> (* 70 *)
  equipotent (Zo (set_of_functions E E) bijection) (powerset E).
```

---

6. *Let* E, F *be two infinite sets such that* Card(E) ≤ Card(F)*. Show that (i) the set of all mappings of* E *onto* F*, (ii) the set of all mappings of* E *into* F*, and (iii) the set of all mappings of subsets of* E *into* F *are all equipotent to* $\mathfrak{P}$(F)*.*

Note. If Card(E) < Card(F) there is no surjective function E → F. For this reason, we assume Card(F) ≤ Card(E). Then E is the disjoint union of two sets that are respectively equipotent to E and F. In F is non-empty, E is equipotent to E × F.

If F is empty, there is no function E → F, and the result is false. If F has a single element, the set of functions E → F is equipotent to E and the result is false. The result is true if F has at least two elements.

Assume f_1 : G → E and f_2 : E − G → F be two surjective functions, where G is some subset of E. If f is any function, we define a function g as follows. If $x \in$ G, we define $g(x) = f(f_1(x))$, otherwise $g(x) = f_2(x)$. Note that g is surjective since f_2 is surjective. The mapping $f \to g$ is injective (since f_1 is surjective). This shows that the sets in (i) and (ii) are equipotent.

Let A be the set of functions E → F, and B the set of functions X → F, where X is a subset of E, and let C be the powerset of E. We have A ⊂ B so that Card(A) ≤ card(B). Let a and b be two distinct elements of F; if X ⊂ E we consider the function that maps an element of X to a, other elements of E to b. This yields an injection C → A and shows Card(C) ≤ card(A). To each element of B we associate its graph, a subset of E × F. This shows Card(B) ≤ Card($\mathfrak{P}$(E × F)) = Card(C). Thus, A, B and C are equipotent.

```
Section Exercise6_6.
Variables E F: Set.
Hypothesis Einf: infinite_set E.
Hypothesis leFE:  (cardinal F) <=c (cardinal E).
Hypothesis Finf:  exists a, exists b, (inc a F & inc b F & a <> b).

Lemma Exercise6_6a: (* 24 *)
  exists G, sub G E & G \Eq E &  (complement E G) \Eq F.
Lemma Exercise6_6b: (* 33 *)
  let A := set_of_functions E F in let B := Zo A surjection in
       cardinal A = cardinal B.
Lemma Exercise6_6c: (* 27 *)
  let s1 := set_of_functions E F in
   let s2 := set_of_sub_functions E F in
     let s3 := powerset E in
```

```
       (s1 \Eq s3 & s2 \Eq s3).
End Exercise6_6.
```

Here is now the code of Exercise 5.

```
Lemma Exercice_6_5 E: infinite_set E -> (* 28 *)
  (Zo (set_of_functions E E) bijective) \Eq (powerset E).
```

---

7. *Let* E, F *be two infinite sets such that* Card(E) < Card(F). *Show that the set of all subsets of* F *which are equipotent to* E *and the set of all injections of* E *into* F *are both equipotent to the set* F^E *of all mappings of* E *into* F *(for each mapping* f *of* E *into* F*, consider the injection* $x \mapsto (x, f(x))$ *of* E *into* E × F*).*

Note The result is true if Card(E) ≤ Card(F), F infinite. If E is empty, the three sets are singletons, and the result is true. Otherwise F is equipotent to E × F.

Let A be the set of functions E → F, B the set of injections E → F et C the set of subsets of F equipotent to E. To $f \in B$ we associate the range of its graph. This set is equipotent to E, this is in C, and all elements of C have this form. We deduce Card(C) ≤ Card(B) (this is true, whatever E and F).

Consider a bijection $g : F \times E \to F$. If $f : E \to F$ is a function, we consider $h : x \to g((f(x), x))$. This is an injection and $f \to h$ is injective. This shows that A and B have the same cardinal.

We consider now a bijection $g : E \times F \to F$. If f is any function E → F, G is graph, then $g\langle G\rangle$ is a subset of F equipotent to E. The mapping $f \to G$ is injective and so is the mapping $f \to g\langle G\rangle$. This shows Card(A) ≤ Card(C).

```
Lemma equipotent_source_graph f: is_function f ->
  (graph f) \Eq  (source f).
Lemma image_by_fun_injective f u v: (* 8 *)
  injective f -> sub u (source f) -> sub v (source f) ->
  image_by_fun f u =  image_by_fun f v -> u = v.
Lemma Exercise6_7a  E F: (* 25 *)
  let A := set_of_functions E F in let B := Zo A injective in
    let C := Zo (powerset F)(fun x =>  x \EQ E) in
    (cardinal C) <=c (cardinal B).
Lemma Exercise6_7b E F: infinite_set F -> (* 68 *)
  (cardinal E) <=c (cardinal F) ->
  let A := set_of_functions E F in let B := Zo A injection in
    let C := Zo (powerset F)(fun x => equipotent x E) in
      (cardinal A = cardinal B &  cardinal A = cardinal C).
```

---

8. *Show that the set of well-orderings on an infinite set* E *(and* a fortiori *the set of orderings on* E*) is equipotent to* $\mathfrak{P}(E)$ *(Use Exercise 5).*

Consider the following sets. A is the set of orderings, B the set of well-orderings, C the set of permutations and D the powerset of E and D_2 the powerset of E × E. Let a, b, etc

their cardinals. We have $c \leq b$, see below; we have obviously $b \leq a \leq d_2$. If E is infinite we have $d = d_2 = c$, which proves the theorem. The non-trivial point is $c \leq$: we have to find an injection function $f \to \Gamma$ that maps a permutation f of E onto a well-ordering. By Zermelo, there is a well-ordering $\leq$. Consider $f(x) \leq f(y)$. This is a well-ordering Γ, and f is an order isomorphism $\Gamma \to \leq$. By uniqueness of isomorphisms of well-orderings this mapping is injective.

```
Lemma Exercice6_8a r r' f: (* 2 *)
  order_isomorphism f r r' -> order_morphism f r r'.
Lemma Exercice6_8b E:  (* 59 *)
  (cardinal (Zo (set_of_functions E E) bijection))  <=c
  (cardinal (Zo (powerset (coarse E)) (fun r => worder r & substrate r = E))).
Lemma Exercice6_8c E: infinite_set E -> (* 17 *)
  let s1 := Zo (powerset (coarse E)) (fun r => order r & substrate r = E) in
    let s2 := Zo (powerset (coarse E)) (fun r = worder r & substrate r = E) in
  (s1 \Eq s2 & s2 \Eq  (powerset E)).
```

---

9. *Let* E *be a non-empty well-ordered set in which every element x other then the least element of* E *has a predecessor (the greatest element of* $]\leftarrow, x[$*). Show that* E *is isomorphic to either* **N** *or an interval* $[0, n[$ *of* **N** *(remark that every segment* $\neq$ E *is finite by using Proposition 6 of No. 5; then use Theorem 3 of § 2, no. 5).*

We first restate Theorem 3 as: if E and E′ are well-ordered sets, either they are isomorphic, or E is isomorphic to a segment S_x of E′ or E′ is isomorphic to a segment S_x of E. We then state a lemma that says that a segment S_x of **N** (with the induced order) is nothing else than the open interval $[0, x[$ (with its usual ordering). From this, it follows that, any well-ordered set is isomorphic to **N** or to an interval of **N**, or else there is a segment S_x of E which is isomorphic to **N**.

Let's prove the exercise. The assumption $E \neq \emptyset$ is not needed (it suffices to take $n = 0$). The previous remark shows that either the conclusion is true, or there exists some x and a isomorphism $f : S_x \to \mathbf{N}$. Note that x cannot be the least element of E, since this would imply $S_x = \emptyset$ hence $\mathbf{N} = \emptyset$. On the other hand, if $y \in S_x$ then then $y < f^{-1}(1 + f(y))$, so that S_x cannot have a greatest element. There is no need to use Proposition 6 of No. 5.

```
Lemma Exercise6_9a n: inc n Bnat -> (* 14 *)
   (interval_Bnatco n = induced_order Bnat_order (segment Bnat_order n)).
Lemma Exercise6_9  r: worder r -> (* 26 *)
  (forall x, inc x (substrate r) ->
    (least_element r x \/
    (exists y, greatest_element (induced_order r (segment r x)) y))) ->
  r \Is Bnat_order
  \/ (exists n, inc n Bnat & r \Is (interval_Bnatco n)).
```

¶ 10. *(c) Show that for each infinite cardinal* $\mathfrak{a}$*, the least upper bound* λ *of the set of ordinals* $W(\mathfrak{a})$ *is an initial ordinal* ω_α*, and that* $\mathfrak{a} = \aleph_\alpha$ *(consider the least ordinal* μ *such that* $\omega_\mu \geq$

λ*); in other words* ω_α *is the least ordinal* ξ *such that* $\mathrm{Card}(\xi) = \aleph_\alpha$. *For each ordinal* α *the mapping* $\xi \mapsto \aleph_\xi$, *defined on* $\mathrm{O}'(\alpha)$, *is an isomorphism of the well-ordered set* $\mathrm{O}'(\alpha)$ *onto the well-ordered set of cardinals* $\leq \aleph_\alpha$; *in particular* $\aleph_{\alpha+1}$ *is the least cardinal* $> \aleph_\alpha$. *Show that if* α *has no predecessor, then for every strictly increasing mapping* $\xi \mapsto \sigma_\xi$ *of an ordinal* β *into* α *such that* $\alpha = \sup_{\xi<\beta} \sigma_\xi$, *we have*

$$\sum_{\xi<\beta} \aleph_{\sigma_\xi} = \aleph_\alpha.$$

(d) Deduce from (c) that ω_ξ *is a normal ordinal functional symbol (§ 2, Exercise 17).*

Point (a) is ord_bound_coll.

Point (b): the quantity ω_x is denoted by omega_fct. The two main properties are aleph_pr1 and aleph_pr4. Lemma aleph_pr6 says that this is a strict increasing function.

Point (c). The first claim is aleph_pr7. We deduce that ω_x is a cardinal, and $\aleph_x = \omega_x$. The isomorphism is aleph_pr9. Note that it maps $\mathrm{O}'(\alpha)$ onto the set of inifinite cardinals $\leq \alpha_\alpha$, the word "infinite" is missing in Bourbaki. Lemma aleph_pr10 asserts that $\aleph_{\alpha+1}$ is the cardinal successor of $\aleph_\alpha$. Point (d) is aleph_pr11.

The last claim in (c) is a bit strange. We must assume α non-zero, for otherwise $\beta = 0$, and the sum is zero. Thus α is a limit ordinal, and σ a cofinal function. Since the cardinal sum is commutative, we can restate it as: $\sum_{i\in \mathrm{E}} \aleph_i = \aleph_\alpha$ whenever E is a subset of E such that $\alpha = \sup \mathrm{E}$.

The exercise considers an ordinal α such that "α has no predecessor". We have to add the additional condition that α is not zero, so that α is a limit ordinal. The claim considers a "strictly increasing mapping $\xi \mapsto \sigma_\xi$ of an ordinal β into α". Notice first that Bourbaki makes an abuse of language: he considers a strictly increasing mapping such that $\xi < \beta$ implies $\sigma_\xi < \alpha$. If E is the range of σ we have $\mathrm{E} \subset \alpha$. On the other hand, any subset E of α is a well-ordered set (ordered by $\leq_{\mathrm{Ord}}$) and if β is its ordinal, there exists a unique order isomorphism σ such that E is the range of σ. Thus σ is uniquely defined by E.

The claim is "if $\alpha = \sup_{\xi<\beta} \sigma_\xi$ then $\sum_{\xi<\beta} \aleph_{\sigma_\xi} = \aleph_\alpha$." Note that sup and $\sum$ depend only on the family σ_ξ or $\aleph_{\sigma_\xi}$, not on the ordering. Thus the statement is: for any set of ordinals E such that we have $\sum_{i\in \mathrm{E}} \aleph_i = \aleph_\alpha$. Denote the sum by S_E. The claim $\mathrm{S}_\mathrm{E} \leq \aleph_\alpha$ follows from the fact that each term of the sum satisfies this relation, and the number of terms in the sum is Card(E), which is at most Card(α) which is less than $\aleph_\alpha$. Thus $\mathrm{S}_\mathrm{E} \leq \aleph_\alpha^2 = \aleph_\alpha$. On the other hand $\mathrm{S}_\mathrm{E} \geq \aleph_j$ for any j such that $j < \alpha$ since there is some i such that $i \in \mathrm{E}$ and $j \leq i$. Now $\sup_{j<\alpha} \aleph_j = \aleph_\alpha$ just says that ω is a normal ordinal function.

We show here that $x \to \aleph_x$ is an order isomorphism $\mathrm{O}'(y) \to \mathrm{T}(y)$, where $\mathrm{T}(y)$ is the set of infinite cardinals $\leq \aleph_y$, for any ordinal y. Since ω_x is a cardinal, we have $\aleph_x = \omega_x$. Since $\leq_{\mathrm{Card}}$ and $\leq_{\mathrm{Ord}}$ are the same, and ω_x is strictly increasing, all we need to show is that the function is well-defined and surjective.

```
Lemma aleph_pr9 x: is_ordinal x -> (* 35 *)
  let y:= (omega_fct x) in
  let src := (succ_o x) in
   let trg := Zo (set_of_cardinals_le y) infinite_c  in
  order_isomorphism
    (BL (fun z => (omega_fct z)) src trg)
    (graph_on ordinal_le src)(graph_on cardinal_le trg).
```

¶ 11. *(a) Show that the ordinal* ω *is the least ordinal* >0 *which has no predecessor, that* ω *is indecomposable (§ 2, Exercise 16), and that for each ordinal* $\alpha>0$, $\alpha\omega$ *is the least indecomposable ordinal which is* $>\alpha$ *(note that* $n\omega=\omega$ *for each integer* n*). Deduce that*

$$(\alpha+1)\omega=\alpha\omega \text{ for each } \alpha>0.$$

(b) Deduce from (a) that an ordinal is indecomposable if and only if it is of the form ω^β *(use Exercise 18 (d) of § 2).*

Point (a). Lemma omega0_limit1 says that ω is limit ordinal and omega0_limit2 says that it is the least limit ordinal. Lemma indecomp_omega says that ω is indecomposable. Lemma indecomposable_prod2 says that $\alpha\omega$ is the least indecomposable ordinal $>\alpha$. The last claim is indecomposable_prod3.

Point (b). This relies on the existence of the Cantor normal form (see next exercise). Let α be a non-zero ordinal, write it as $\alpha=\omega^\beta+r$, where r is a sum of powers of ω. By uniqueness of the representation, $r<\alpha$. Thus, if α is indecomposable we must have $\alpha=\omega^\beta$. Conversely, ω^β is indecomposable, since $\omega^a+\omega^b=\omega^b$ for $a<b$.

Note that ω^β is the greatest indecomposable ordinal $\leq\alpha$. This shows Exercise 16 (d) and (e) of no. 2.

.

¶ 12. (a) Show that for each ordinal α and each ordinal $\gamma>1$, there exists two finite sequences of ordinals (λ_i) and (μ_i) $(1\leq i\leq k)$ such that

$$\alpha=\gamma^{\lambda_1}\mu_1+\gamma^{\lambda_2}\mu_2+\ldots+\gamma^{\lambda_k}\mu_k,$$

where $0<\mu_i<\gamma$ for each i, and $\lambda_i>\lambda_{i+1}$ for $1\leq i\leq k-1$ (use Exercise 18 (d) of § 2 and Exercise 3 of § 4). Moreover the sequences (λ_i), (μ_i) are uniquely determined by these conditions. In particular there exists a unique finite decreasing sequence $(\beta_j)_{1\leq j\leq m}$ such that

$$\alpha=\omega^{\beta_1}+\omega^{\beta_2}+\cdots+\omega^{\beta_m}.$$

Let $\phi(\alpha)$ denote the greatest ordinal ω^{β_1} in this sequence.

(b) For each integer n let $f(n)\leq n!$ be the greatest number of elements in the set of ordinals of the form $\alpha_{\sigma(1)}+\alpha_{\sigma(2)}+\cdots+\alpha_{\sigma(n)}$, where $(\alpha_i)_{1\leq i\leq n}$ is an arbitrary sequence of n ordinals and σ runs through the set of permutations of the interval $[1,n]$. Show that

$$f(n)=\sup_{1\leq k\leq n-1}(k.2^{k-1}+1)f(n-k). \tag{1}$$

(consider first the case where all the $\phi(\alpha_i)$ are equal and show that the largest possible number of distinct ordinals of the desired form is equal to n, by using Exercise 16 (a) of § 2. Then use induction on the number of ordinals α_i for which $\phi(\alpha_i)$ takes the least possible value among the set of ordinals $\phi(\alpha_j)$ $(1\leq j\leq n)$.) Deduce from (1) that for $n\geq 20$ we have $f(n)=81f(n-5)$.

(c) Show that the $n!$ ordinals $(\omega+\sigma(1))(\omega+\sigma(2))\ldots(\omega+\sigma(n))$ where σ runs through the set of permutations of the interval $[1,n]$, are all distinct.

Point (a) is the subject of the section 8.12 (existence and uniqueness of the Cantor Normal Form). Consider the product form (18d) where all exponents are one, except ν_k, and take

$\mu_1 = 1$. If n is an integer, we have $n\cdot(\omega+1) = \omega+n$, so that we get $\alpha = (\omega+\mu_k)\cdots(\omega+\mu_3)(\omega+\mu_2)$. Such a form is unique (when it exists) and it implies trivially (c).

¶ 13. *(a) Let $w(\xi)$ be an ordinal functional symbol (§2, Exercise 17), defined for $\xi \geq \alpha_0$ and such that the relation $\alpha_0 \leq \xi < \xi'$ implies $w(\xi) < w(\xi')$. Show that, if $\xi \geq \alpha_0$, then $w(\xi+\eta) \geq w(\xi)+\eta$ for every ordinal η (argue by contradiction). Deduce that there exists α such that $w(\xi) \geq \xi$ for all $\xi \geq \alpha$ (take α to be the least indecomposable ordinal $\geq \alpha_0$; cf Exercise 11 (a)).*

(b) Let $f(\xi,\eta)$ be the ordinal functional symbol defined in § 2, Exercise 17(b). Suppose that the relations $\alpha_0 \leq \xi \leq \xi'$ and $\alpha_0 \leq \eta \leq \eta'$ imply $g(\xi,\eta) \leq g(\xi',\eta')$ so that the relations $\alpha_0 \leq \xi \leq \xi'$ and $1 \leq \eta \leq \eta'$ imply $f(\xi,\eta) \leq f(\xi',\eta')$ (§ 2, Exercise 17(d)). Show that for each ordinal β there exist at most a finite number of ordinals η for which the equation $f(\xi,\eta) = \beta$ has at least one solution (Note that if ξ_1 is the least solution of $f(\xi,\eta_1) = \beta$ and if ξ_2 is the least solution of $f(\xi,\eta_2) = \beta$ then the relation $\eta_1 < \eta_2$ implies $\xi_1 > \xi_2$.)

(c) A critical ordinal *with respect to f is any infinite ordinal $\gamma > \alpha_0$ such that $f(\xi,\gamma) = \gamma$ for all ξ such that $\alpha_0 \leq \xi < \gamma$. Show that a critical ordinal (with respect to f) has no predecessor. If there exists a set A of ordinals such that $f(\xi,\gamma) = \gamma$ for all $\xi \in A$, and if γ is the least upper bound of A, show that γ is a critical ordinal.*

(d) Let $h(\xi) = f(\xi,\xi)$ (defined for $\xi \geq \alpha_0$); define inductively $\alpha_1 = \alpha_0 + 2$, $\alpha_{n+1} = h(\alpha_n)$ for $n \geq 1$. Show that that the least upper bound of the sequence (α_n) is a critical ordinal with respect to f.

(e) Show that the least upper bound of every set of critical ordinals with respect to f is again a critical ordinal, and that every critical ordinal is indecomposable (note that $f(\xi,\eta+1) \geq w(\xi)+\eta \geq \xi+\eta$ for all $\xi \geq \alpha_0$).

(a) is ord_sum_increasing5.
(b) is ord_inductionp_p20.
(c) is critical_limit, sup_critical.
(d) is sup_critical2.
(e) is sup_critical3.

¶ 14. *(a) Show that if $\alpha \geq 2$ and if β has no predecessor, then α^β is an indecomposable ordinal (cf § 2, Exercise16 (a)); if α is finite and if $\beta = \omega\gamma$, then $\alpha^\beta = \omega^\gamma$; if α is infinite and if π is the greatest indecomposable ordinal $\leq \alpha$, then $\alpha^\beta = \pi^\beta$ (use Exercise 11).*

(b) An ordinal δ is critical with respect to the functional symbol $f(\xi,\eta) = \xi\eta$ if and only if, for each α such that $1 < \alpha \leq \delta$, the equation $\delta = \alpha^\xi$ has a solution; the unique solution ξ of this equation is then indecomposable (Use Exercise 13 (e), together with Exercise 18 (d) of § 2). Conversely, for each $\alpha > 1$ and each indecomposable ordinal π, α^π is a critical ordinal with respect to $\xi\eta$ (use Exercise 13 (c)). Deduce that δ is a critical ordinal with respect to $\xi\eta$ if and only if δ is of the form ω^{ω^μ} (cf. Exercise 11 (b)).

(c) For an ordinal ϵ to be critical with respect to the functional symbol $f(\xi,\eta) = \xi^\eta$, i.e. such that $\gamma^\epsilon = \epsilon$ for each γ satisfying $2 \leq \gamma \leq \epsilon$, it is sufficient that $2^\epsilon = \epsilon$. Show that the least critical ordinal ϵ_0 with respect to ξ^η is countable (cf. Exercise 13 (d)).

(a). Let α and β be two ordinals. If $\beta = 0$ then $\alpha^\beta = 1$ is indecomposable. Assume that β has no predecessor, $\beta = \omega\beta'$ for some β'. We compute α^β via formula (17e) of Section 8.12).

It says $\alpha^\beta = \omega^\gamma$ for some γ, thus is indecomposable. If α is finite, then $\gamma = \beta'$ (second part of (17e)). Assume α infinite of degree n, then $\gamma = n\beta$ (first part of (17e)). Let $\pi = \omega^n$ so that $\alpha^\beta = \pi^\beta$. Since $\omega^n \leq \alpha < \omega^{n+1}$, it follows that π is the greatest power of ω (i.e., the greatest indecomposable ordinal) which is $\leq \alpha$.

(b). Let's show that if α^π has the form ω^μ, for some indecomposable μ, whenever α and π are > 1, and π is indecomposable. Note that we have to exclude the case $\pi = 1$. Now $\pi = \omega^n$ for some non-zero n, and we can apply formula (17e). Assume α is infinite, of degree β, then $\alpha^\pi = \omega^{\beta\pi}$. We know that $\beta\pi$ is indecomposable. If α is finite, then $\alpha^\pi = \omega^{\pi'}$ where $\pi = \omega\pi'$, and π' is again indecomposable.

Consider the following properties of y: (P1) says that y is critical for the product, (P2) says that $y = x^z$ has a solution in z whenever $1 < x \leq z$, (P3) says that $y = x^z$ has a solution in z which is indecomposable whenever $1 < x \leq z$, (P4) says that $y = \omega^{\omega^\mu}$. In the main text we have shown that (P1), (P3), and (P4) are equivalent. Obviously (P3) implies (P2).

Let's show that (P2) implies the other relations. Consider a such that $1 \leq a < y$, and n such that $y = a^n$. Assume first n infinite. Then $1 + n = n$ (proof: consider the CNF of n, and the sum of two CNFs). In this case $a \cdot y = a^{1+n} = a^n = y$. Assume n finite. Assume first $n \geq 3$. Let $m = n - 1$, so that $a^m \leq y$ and there is p with $y = (a^m)^p$. The cases $p = 0$ and $p = 1$ are excluded, so that $p \geq 2$ and $mp \geq m + 2$. Thus $y \geq ay$, this concludes the proof. Cases $n < 2$ are excluded so that $n = 2$ and $y = a^2$. If $b = a + a$, we have $b \leq y$, so that $y = b^m$ for some m. Simple considerations show $m = 2$, so that $y = a^2 = (a + a)^2$.

This is impossible: assume first a not a successor, of degree n. The same holds for $b = a + a = a \cdot 2$. Relation (17b) says $a^2 = \omega^n \cdot a$, and $b^2 = \omega^n \cdot b$. From $a^2 = b^2$, after simplifying by ω^n, we get $a = b$ absurd. If a is a successor, then so is b. We compute squares via (16c). Comparing coefficients shows that this is impossible too.

(c). There is a misprint in the English Edition. One should read: ϵ is critical when $\gamma^\epsilon = \epsilon$ for each γ satisfying $2 \leq \gamma < \epsilon$. Lemma ord_epsilon_p10 says that if $2^\epsilon = \epsilon$, then ϵ is critical. The least critical ordinal is ω, and if ϵ is not ω, it is critical if and only if it is an ϵ-ordinal. There are many countable critical ordinals, since ϵ_α is countable whenever α is countable.

```
Lemma rev_succ_pr x: is_ordinal x -> (* 32 *)
  x <o \omega \/ x = \1o +o x.
Lemma ord_square_inj a: is_ordinal a ->  (* 73 *)
   a ^o \2o =  (a *o \2o) ^o \2o -> a = \0o.
Lemma critical_product_pr2: (* 102 *)
  let CP := critical_ordinal \1o ord_prod2 in
  let p1 := fun y => infinite_o y & is_ordinal y &
       (forall z, \1o <o z -> z <=o y ->
          exists t, is_ordinal t &  y = z ^o t) in
  forall y, CP y <-> p1 y.

Lemma critical_product_pr3 a b: (* 65 *)
 \1o <o a -> \1o <o b ->
 ord_indecomposable b ->
 critical_ordinal \1o ord_prod2 (a ^o b).
```

¶ 15. *Let γ be an ordinal > 1, and for each ordinal α let $L(\alpha)$ denote the set of exponents λ_i in the expression for α given in Exercise 12 (a).*

(a) Show that $\lambda_i \leq \alpha$ for each $\lambda_i \in L(\alpha)$ and that $\lambda_i = \alpha$ for one of these ordinal only if $\alpha = 0$ or if α is a critical ordinal with respect to ξ^η (Exercise 14 (c)).

(b) Define $L_n(\alpha)$ *by induction on n as follows:* $L_1(\alpha) = L(\alpha)$ *and* $L_n(\alpha)$ *is the union of the sets* $L(\beta)$ *as* β *runs through* $L_{n-1}(\alpha)$*. Show that there exists an integer* n_0 *such that* $L_{n+1}(\alpha) = L_n(\alpha)$ *whenever* $n \geq n_0$*, and that the elements of* $L_n(\alpha)$ *are then either 0 or critical ordinals with respect to* ξ^η *(Argue by contradiction: for each n, consider the set* $M_n(\alpha)$ *of elements* $\beta \in L_n(\alpha)$ *such that* $\beta \notin L(\beta)$*, and assume that* $L_n(\alpha)$ *is not empty for any n; use (a) to obtain a contradiction.)*

Consider an ordinal $b \geq 2$. Consider the expansion of x in base b. This is a function X; defined on an interval $[0, n[$, such that X_i is a pair (e_i, c_i), where e_i is a increasing decreasing sequence of ordinals. With the notations of Exercice 12, $\lambda_1 = e_{n-1}$, $\lambda_2 = e_{n-2}$, etc, up to $\lambda_k = e_0$. Let $L(x)$ be the set of all e_i. In the case $x = 0$, we have $n = 0$ and $L = \emptyset$. Assume $x > 0$, and $m = n - 1$. We have $e_i < e_m$ if $i < m$. Ler $e = e_m$ so that $x = b^e c + d$. We deduce $x \geq b^e \geq e$, and equality holds only if $c = 1$, $d = 0$, and $b^e = e$. We have already noticed that this relation is equivalent to e being critical for exponentiation; this means that either e is a an ϵ-ordinal, or $e = \omega$ and b is finite. Let's call this critical.

We show here the following three properties: if x is critical, then $L(x) = \{x\}$, otherwise, $y \in L(x)$ implies $y < x$; and $L(x)$ is a finite set of ordinals.

```
Section Exercise6_15.
Variable (b: Set).
Hypothesis bg2: \2c <=o b.

Definition the_cnf_b x :=
  choose (fun z => is_pair z & inc (Q z) Bnat
    & (CNF_ax b (P z) (Q z)) & x = (CNF_b b (P z) (Q z))).
Definition the_cnf_expos x :=
  let X := P (the_cnf_b x) in
  fun_image (domain X) (fun z => P (V z X)).
Definition b_critical x := b ^o x = x.

Lemma the_cnf_b_p x: is_ordinal x ->  (* 2 *)
  let z := (the_cnf_b x) in (is_pair z  & inc (Q z) Bnat
    & (CNF_ax b (P z) (Q z)) & x = (CNF_b b (P z) (Q z))).
Lemma the_cnf_e_p1 x n X: (* 6 *)
  inc n Bnat -> CNF_ax b X n -> x = CNF_b b X n ->
   the_cnf_expos x =  fun_image (domain X) (fun z => P (V z X)).
Lemma the_cnf_expos_zero: the_cnf_expos \0o = emptyset. (* 21 *)
Lemma greatest_expo_first n X k: (* 15 *)
  inc n Bnat -> CNF_ax b X (succ n) -> k <c n ->
  P (V k X) <o P (V n X).
Lemma the_cnf_e_p2 n X: (* 12 *)
  inc n Bnat -> CNF_ax b X (succ n) ->
  P (V n X) <=o  (CNF_b b X (succ n)).
Lemma the_cnf_e_p3 n X (x:= CNF_b b X (succ n)): (* 32 *)
  inc n Bnat -> CNF_ax b X (succ n) ->
  ((P (V n X) = x ->  b_critical x)
   & (b_critical x -> (n = \0c & (P (V n X) = x)))).
Lemma the_cnf_e_p4 x (y:=the_cnf_expos x): is_ordinal x -> (* 31 *)
  (finite_set y &
   ordinal_set y &
   (b_critical x  -> y = singleton x) &
   (~ (b_critical x) -> forall a, inc a y -> a <o x)).
```

Let's now define by induction $L_n(x)$ to be $L(x)$ if $n = 0$, and the union of the $L_n(y)$ for $y \in L_{n-1}(x)$ otherwise. This is a finite set of ordinals. Let M_n be the set of all non-critical

elements of $L_n(x)$. If M_n is empty then $L_k = L_n$ whenever $k \geq n$ and this set contains only critical elements. We pretend that at least one M_n is empty, for otherwise it would contain a greatest element y_n, and the sequence y_n is strictly decreasing.

```
Definition the_cnf_expos_rec x:=
 induction_defined (fun z => union (fun_image z the_cnf_expos))
    (the_cnf_expos x).
Definition the_cnf_expos_rec_nc x n :=
  Zo (W n (the_cnf_expos_rec x)) (fun z => ~ (b_critical z)).

Lemma the_cnf_e_p5 x n (y := W n (the_cnf_expos_rec x)): (* 17 *)
   is_ordinal x -> inc n Bnat -> (finite_set y & ordinal_set y).
Lemma the_cnf_e_p6 x n (f :=  (the_cnf_expos_rec x)): (* 29 *)
  is_ordinal x -> inc n Bnat ->
  the_cnf_expos_rec_nc x n = emptyset ->
  ( (forall a, inc a (W n f) -> b_critical a)
  & (forall k, inc k Bnat -> n <=c k -> W k f = W n f)).
Lemma the_cnf_e_p6 x n (f :=  (the_cnf_expos_rec x)): (* 93 *)
  is_ordinal x -> inc n Bnat ->
  the_cnf_expos_rec_nc x n = emptyset ->
  ( (forall a, inc a (W n f) -> b_critical a)
  & (forall k, inc k Bnat -> n <=c k -> W k f = W n f)).
End Exercise6_15.
```

---

16. Every totally ordered set has a well-ordered cofinal subset (§ 2, Exercise 2). The least of the ordinals Ord(M) of the well-ordered cofinal subsets M of E is said the *final character* of E.

(a) An ordinal ξ is said to be *regular* if it is equal to its final character, and *singular* otherwise. Show that every infinite regular ordinal is an initial ordinal ω_α (Exercise 10). Conversely, every initial ordinal ω_α, whose index α is either 0 or has a predecessor, is a regular ordinal. An initial ordinal ω_α whose index α has no predecessor is singular if $0 < \alpha < \omega_\alpha$; in particular, ω_ω is the least infinite singular initial ordinal.

(b) An initial ordinal ω_α is said to be *inaccessible* if it is regular and its index α has no predecessor. Show that if $\alpha = 0$, then $\omega_\alpha = \alpha$; in other words, α is a critical ordinal with respect to the normal functional symbol ω_η (Exercise 10 (d) and 13 (c)).[3] Let κ be the least critical ordinal with respect to this functional symbol. Show that ω_κ is singular, with final character ω (cf Exercise 13(d)). In other words, there exists no inaccessible ordinal ω_α such that $0 < \alpha \leq \kappa$ (At present, it is not known whether or not there exist inaccessible ordinals other than ω).

(c) Show that there exists only one regular ordinal which is cofinal in a given totally ordered set E; this ordinal is equal to the final character of E, and if E is not empty and has no greatest element, it is an initial ordinal. If $\omega_{\bar\alpha}$ is the final character of ω_α, then $\bar\alpha \leq \alpha$; and ω_α is regular if and only if $\alpha = \bar\alpha$.

(d) Let ω_α be a regular ordinal and let I be a well-ordered set such that $\mathrm{Ord}(I) < \omega_\alpha$. Show that, for each family $(\xi_\iota)_{\iota\in I}$ of ordinals such that $\xi_\iota < \omega_\alpha$ for all $\iota \in I$, we have $\sum_{\iota\in I} \xi_\iota < \omega_\alpha$.

By exercise2_2b for every ordered set $(E, \leq)$ there exists a well-ordered subset X that has no upper bound; if E is totally ordered, it follows that X is cofinal. Consider now all ord(X)

[3] The condition is obviously $\alpha > 0$; moreover critical has to be understrood as $f(x) = x$.

where X is cofinal in E and well-ordered; this is a set of ordinals, thus has a least element, called the *final character* of E.

Assume now that x is an ordinal and let $(E, \leq)$ be its ordering. This is a total ordering. We define $c(x)$, the final character of x, as the final character of E. Note that any subset of X is well-ordered by $\leq$, so that the final character of x is the least ord(X), where X is cofinal in x.

```
Lemma Exercise6_16a r: total_order r -> exists X, (* 6 *)
  cofinal_set r X & (worder (induced_order r X)).
Lemma cofinality'_pr1 r: total_order r -> (* 7 *)
  (nonempty (cofinality' r) & ordinal_set (cofinality' r)).
Lemma intersection_sub1 A B C:
   A = union2 B C -> (forall x, inc x C -> exists y, inc y B & sub y x)
   -> intersection A = intersection B.
Lemma cofinal_trans r x y: (* 4 *)
   order r -> cofinal_set r x -> cofinal_set (induced_order r x) y ->
   cofinal_set r y.

Lemma cofinal_image r r' f x:  (* 10 *)
    order_isomorphism f r r' -> cofinal_set r x ->
    cofinal_set r' (image_by_fun f x).
Lemma worder_image r r' f A: (* 51 *)
  order_isomorphism f r r' -> sub A (substrate r) ->
  let oa := (induced_order r A) in
  let ob := (induced_order r' (image_by_fun f A)) in
  worder oa  -> (worder ob & ordinal oa = ordinal ob).
```

We show here that the cofinality of $a + b$ is that of b (whenever b is non-zero); and the cofinality of $a \cdot b$ is that of b (whenever a is non-zero, and b is a limit ordinal). Let x be a non-zero ordinal and $\omega^a \cdot b$ the last term in the Cantor Normal Form of x. If $a = 0$, then $\mathrm{cf}(x) = 1$. Otherwise $\mathrm{cf}(x) = \mathrm{cf}(\omega^a)$. This is ω in the case where a is successor. Otherwise, consider the CNF of a. Write $a = c + \omega^m$. Let $d = \omega^m$. We have $\mathrm{cf}(x) = \mathrm{cf}(\omega^d)$. There is no easy way to evaluate this. We prove here the following: a regular ordinal is zero, one, ω, or ω^n, where n is a limit ordinal.

```
Lemma cofinality_sum a b: (* 24 *)
   is_ordinal a -> is_ordinal b -> b <> \0o ->
   cofinality (a +o b) = cofinality b.
Lemma regular_indecomposable x: (* 8 *)
   regular_ordinal x -> (x = \0o \/ ord_indecomposable x).
Lemma cofinality_prod a b: (* 20 *)
   is_ordinal a -> is_ordinal b -> a <> \0o ->
   limit_ordinal b ->
   cofinality (a *o b) = cofinality b.
Lemma cofinality_prod_omega a: is_ordinal a -> a <> \0o -> (* 3 *)
   cofinality (a *o \omega) = \omega.
Lemma regular_indecomposable1 x: (* 11 *)
   regular_ordinal x -> (x = \0o \/ x= \1o \/ x =\omega \/
   exists y, limit_ordinal y & x = \omega ^o y).
```

17. *A cardinal $\aleph_\alpha$ is said to be* regular *(resp.* singular*) if the initial ordinal ω_α is regular (resp. singular). For $\aleph_\alpha$ to be regular if is necessary and sufficient that for every family $(\mathfrak{a}_\iota)_{\iota\in I}$ of cardinals such that* $\mathrm{Card}(I) < \aleph_\alpha$ *and $\mathfrak{a}_\iota < \aleph_\alpha$ for all $\iota \in I$, we have*

$$\sum_{\iota\in I} \mathfrak{a}_\iota < \aleph_\alpha.$$

$\aleph_\omega$ *is the least singular cardinal.*

This is infinite_regular_pr.

¶ 18. *(a) For each ordinal* α *and each cardinal* $\mathfrak{m} \neq 0$ *we have* $\aleph_{\alpha+1}^{\mathfrak{m}} = \aleph_\alpha^{\mathfrak{m}} \cdot \aleph_{\alpha+1}$ *(reduce to the case where* $\mathfrak{m} < \aleph_{\alpha+1}$ *and consider the mappings of the cardinal* $\mathfrak{m}$ *into the ordinal* $\omega_{\alpha+1}$*).*

(b) Deduce from (a) that, for each ordinal γ *such that* $\mathrm{Card}(\gamma) \leq \mathfrak{m}$ *we have* $\aleph_{\alpha+\gamma}^{\mathfrak{m}} = \aleph_\alpha^{\mathfrak{m}} \cdot \aleph_{\alpha+\gamma}^{\mathrm{Card}(\gamma)}$ *(by transfinite induction on* γ*).*

(c) Deduce from (b) that, for each ordinal α *such that* $\mathrm{Card}(\alpha) \leq \mathfrak{m}$*, we have* $\aleph_\alpha^{\mathfrak{m}} = 2^{\mathfrak{m}} \cdot \aleph_\alpha^{\mathrm{Card}(\alpha)}$.

This exercice corresponds to lemmas infinite_power2, infinite_power5 and infinite_power6, and formulas (21a), (21c) and (21d). If $\alpha = 0$, formula (c) reduces to $\aleph_\alpha^{\mathfrak{m}} = 2^{\mathfrak{m}}$, so that we have to add the condition: "either α non-zero or $\mathfrak{m}$ infinite".

¶ 19. (a) Let α and β be two ordinals such that α has no predecessor, and let $\xi \to \sigma_\xi$ be a strictly increasing mapping of the ordinal ω_β into the ordinal α such that $\sup_{\xi<\omega_\beta} \sigma_\xi = \alpha$. Show that

$$\aleph_\alpha^{\aleph_\beta} = \prod_{\xi<\omega_\beta} \aleph_{\sigma_\xi}.$$

(With each mapping f of the ordinal ω_β into the ordinal ω_α associate an injective mapping $\bar{f}$ of ω_β into the set of all ω_{σ_ξ} $(\xi < \omega_\beta)$ such that $f(\zeta) \leq \bar{f}(\zeta)$ for all $\zeta < \omega_\beta$. Calculate the cardinal of the set of mappings f associated with the same $\bar{f}$ and observe that

$$\mathfrak{m} = \prod_{\xi<\omega_\beta} \aleph_{\sigma_\xi} \geq 2^{\mathrm{Card}(\omega_\beta)}$$

and $\mathfrak{m} \geq \aleph_\alpha$ (cf. § 3, Exercise 3).)

(b) Let $\bar{\alpha}$ be the ordinal such that $\omega_{\bar{\alpha}}$ is the final character of ω_α. Show that $\aleph_\alpha^{\aleph_{\bar{\alpha}}} > \aleph_\alpha$ and that if there exists $\mathfrak{n}$ such that $\aleph_\alpha = \mathfrak{n}^{\aleph_\gamma}$ then $\gamma < \bar{\alpha}$ (use (a) and Exercise 3 of § 3).

(c) Show that if $\lambda < \bar{\alpha}$ then

$$\aleph_\alpha^{\aleph_\lambda} = \sum_{\xi<\alpha} \aleph_\alpha^{\aleph_\lambda}$$

(argue as in Exercise 18 (a)).

(a) This is lemma infinite_increasing_power1. Note that it suffices for σ to be simply increasing. Bourbaki proposes here another proof.

(b) The first point can be restated as $\kappa^{\mathrm{cf}(\kappa)} > \kappa$. The second point follows from $\mathrm{cf}(n^x) > x$; note that $n \geq 2$.

```
Lemma exercise_6_19b a     (* 20 *)
  (ba := ord_index (cofinality (omega_fct a)))
  (x := omega_fct a) (y := omega_fct ba):
  is_ordinal a ->
  (x <c x ^c y &
   (forall n c, is_cardinal n -> is_ordinal c -> x = n ^c (omega_fct c) ->
       c <o ba)).
```

(c). Let $y = \aleph_\lambda$. Condition $\lambda < \bar{\alpha}$ is equivalent to $y < \aleph_{\bar{\alpha}} = \mathrm{cf}(\omega_\alpha)$. Since α is a limit ordinal, the condition becomes $y < \mathrm{cf}(\alpha)$ and the result is (20g), infinite_power7e.

¶ 20. (a) For a cardinal $\mathfrak{a}$ to be regular (Exercise 17) it is necessary that for every cardinal $\mathfrak{b} \neq 0$ we should have

$$\mathfrak{a}^{\mathfrak{b}} = \mathfrak{a}. \sum_{\mathfrak{m}<\mathfrak{a}} \mathfrak{m}^{\mathfrak{b}}.$$

(Use Exercise 19 and consider separately the cases (i) $\mathfrak{b}$ is finite, (ii) $\aleph_0 \leq \mathfrak{b} < \mathfrak{a}$, (iii) $\mathfrak{b} \geq \mathfrak{a}$; also use Exercise 3 of § 3.) The generalized continuum hypothesis implies that the above condition is also sufficient.

(b) Show that if a cardinal $\mathfrak{a}$ is such that $\mathfrak{a}^{\mathfrak{m}} = \mathfrak{a}$ for every cardinal $\mathfrak{m}$ such that $0 < \mathfrak{m} < \mathfrak{a}$, then $\mathfrak{a}$ is regular (use Exercise 3 of § 3).

(c) Show that the proposition "for every regular cardinal $\mathfrak{a}$ and every cardinal $\mathfrak{m}$ such that $0 < \mathfrak{m} < \mathfrak{a}$, we have $\mathfrak{a} = \mathfrak{a}^{\mathfrak{m}}$ is equivalent to the generalized continuum hypothesis (use (a)).

(a) is infinite_power7h.

¶ 21. *An infinite cardinal $\mathfrak{a}$ is said to be* dominant *if for each pair of cardinals $\mathfrak{m} < \mathfrak{a}$, $\mathfrak{n} < \mathfrak{a}$ we have $\mathfrak{m}^{\mathfrak{n}} < \mathfrak{a}$.*

(a) For $\mathfrak{a}$ to be dominant it is sufficient that $2^{\mathfrak{m}} < \mathfrak{a}$ for every cardinal $\mathfrak{m} < \mathfrak{a}$.

(b) Define inductively a sequence $(\mathfrak{a}_n)$ of cardinals as follows: $\mathfrak{a}_0 = \aleph_0$, $\mathfrak{a}_{n+1} = 2^{\mathfrak{a}_n}$. Show that the sum $\mathfrak{b}$ of the sequence $\mathfrak{a}_n$ is a dominant cardinal. $\aleph_0$ and $\mathfrak{b}$ are the two smallest dominant cardinals.

(c) Show that $\mathfrak{b}^{\aleph_0} = \aleph_0^{\mathfrak{b}} = 2^{\mathfrak{b}}$ (Note that $2^{\mathfrak{b}} \leq \mathfrak{b}^{\aleph_0}$). Deduce that $\mathfrak{b}^{\aleph_0} = (2^{\mathfrak{b}})^{\mathfrak{b}}$, although $\mathfrak{b} < 2^{\mathfrak{b}}$ and $\aleph_0 < \mathfrak{b}$.

a) is card_dominant_pr1.

b) Lemma next_dominant_pr says that the supremum of the sequence is dominant, whatever the value of $\mathfrak{a}_0$, and is the least dominant greater than $\mathfrak{a}_0$. Lemma card_dominant_pr2 says that $\aleph_0$ is dominant so that it is the least dominant. It follows that $\mathfrak{b}$ is the second least dominant. Note that $\sum x_i = \sup x_i$ if the supremum is infinite and the index set not too big;

c) is card_dominant_pr4.

¶ 22. A cardinal $\aleph_\alpha$ is said to be *inaccessible* if the ordinal ω_α is inaccessible (Exercise 16 (b)). We have then $\omega_\alpha = \alpha$ if $\omega_\alpha \neq \omega_0$. A cardinal $\mathfrak{a}$ is said to be *strongly inaccessible* if it is inaccessible and dominant.

(a) The generalized continuum hypothesis implies that every inaccessible cardinal is strongly inaccessible.

(b) For a cardinal $\mathfrak{a} \geq 3$ to be strongly inaccessible it is necessary and sufficient that, for each family $(\mathfrak{a}_\iota)_{\iota\in I}$ of cardinals such that

$$\mathrm{Card}(I) < \mathfrak{a} \text{ and } \mathfrak{a}_\iota < \mathfrak{a}$$

for all $\iota \in I$, we should have $\prod_{\iota\in I} \mathfrak{a}_\iota < \mathfrak{a}$.

(c) For an infinite cardinal $\mathfrak{a}$ to be strongly inaccessible it is necessary and sufficient that it should be dominant (Exercise 21) and that it should satisfy one of the following two conditions: (i) $\mathfrak{a}^{\mathfrak{b}} = \mathfrak{a}$ for every cardinal $\mathfrak{b}$ such that $0 < \mathfrak{b} < \mathfrak{a}$; (ii) $\mathfrak{a}^{\mathfrak{b}} = \mathfrak{a}.2^{\mathfrak{b}}$ for every cardinal $\mathfrak{b} > 0$ (Use Exercises 20 and 21).

Let's write WI and SI wherever Bourbaki uses "inaccessible" and "strongly inaccessible". A cardinal $x = \aleph_\alpha$ is WI if α is regular and not a successor. We do not consider $\aleph_0$ as inaccessible. Thus, x is WI if either $x = \aleph_0$ or is weakly inaccessible. In the second case we know that $\omega_\alpha = \alpha$. Similarly x is SI if either $x = \aleph_0$ or is inaccessible.

(a) is inaccessible_weak_strong (this lemma excludes the case of $\aleph_0$ but $\aleph_0$ is WI and SI).

¶ 23. Let α be an ordinal > 0. A mapping f of the ordinal α into itself is said to be *divergent* if for each ordinal $\lambda_0 < \alpha$ there exists an ordinal $\mu_0 < \alpha$ such that the relation $\mu_0 \leq \xi < \alpha$ implies $\lambda_0 \leq f(\xi) < \alpha$. (this condition may be written as $\lim_{\xi \to \alpha, \xi < \alpha} f(\xi) = \alpha$)

(a) Let ϕ be a strictly increasing mapping of an ordinal β into α such that

$$\phi(\sup_{\zeta<\gamma} \zeta) = \sup_{\zeta<\gamma} \phi(\zeta) \text{ for all } \gamma < \beta,$$

and such that

$$\sup_{\zeta<\beta} \phi(\zeta) = \alpha.$$

(if we extend ϕ be defining $\phi(\beta) = \alpha$, the conditions above signify that ϕ is continuous) Then there exists a divergent mapping f of α into itself, such that $f(\xi) < \xi$ for all ξ satisfying $0 < \xi < \alpha$, if and only if there exists a divergent mapping of β into itself of the same type.

(b) Deduce from (a) that there exists a divergent mapping of ω_α into itself, such that $f(\xi) < \xi$ for all ξ satisfying $0 < \xi < \alpha$ if and only if the final character of ω_α is ω_0. (If ω_α is a regular ordinal $> \omega_0$, defined inductively a strictly increasing sequence (η_n) as follows: $\eta_1 = 1$, and η_{n+1} is the least ordinal ζ such that $f(\xi) > \eta_n$ for all $\xi \geq \zeta$.)

(c) Let $\omega_{\bar{\alpha}}$ be the final character of ω_α (Exercise 16). Show that, if $\bar{\alpha} > 0$ and if f is a mapping of ω_α into itself such that $f(\xi) < \zeta$ for all ξ such that $0 < \xi < \omega_\alpha$, then there exists an ordinal λ_0 such that the set of solutions to the equation $f(\xi) = \lambda_0$ has a cardinal $\geq \aleph_{\bar{\alpha}}$.

¶ 24. Let $\mathfrak{F}$ be a set of subsets of a set E such that for every $A \in \mathfrak{F}$ we have $\mathrm{Card}(A) = \mathrm{Card}(\mathfrak{F}) = \mathfrak{a} \geq \aleph_0$. Show that E has a subset P such that $\mathrm{Card}(P) = \mathfrak{a}$ and such that no set of $\mathfrak{F}$ is contained in P (If $\mathfrak{a} = \aleph_\alpha$, define by transfinite induction two injective mappings $\xi \to f(\xi)$, $\xi \to g(\xi)$ of ω_α into E such that the sets $P = f(\omega_\alpha)$ and $Q = g(\omega_\alpha)$ do not intersect and such that each of them meets every subset $A \in \mathfrak{F}$.)

(b) Suppose, moreover, that for each subset $\mathfrak{G}$ of $\mathfrak{F}$ such that $\mathrm{Card}(\mathfrak{G}) < \alpha$, the complement in E of the union of the sets $A \in \mathfrak{G}$ has cardinal $\geq \mathfrak{a}$. Show that E then has a subset P such that $\mathrm{Card}(P) = \mathfrak{a}$ and such that, for each $A \in \mathfrak{G}$, $\mathrm{card}(P \cap A) < \mathfrak{a}$ (similar method).

¶ 25. (a) Let $\mathfrak{F}$ be a covering of an infinite set E. The *degree of disjointness* of $\mathfrak{F}$ is the least cardinal $\mathfrak{c}$ such that $\mathfrak{c}$ is *strictly greater* than the cardinals $\mathrm{Card}(X \cap Y)$ for each pair of distinct sets $X, Y \in \mathfrak{F}$. If $\mathrm{Card}(E) = \mathfrak{a}$ and $\mathrm{Card}(\mathfrak{F}) = \mathfrak{b}$, show that $\mathfrak{b} \leq \mathfrak{a}^{\mathfrak{c}}$ (note that a subset of E of cardinal $\mathfrak{c}$ is contained in at most one set of $\mathfrak{F}$).

(b) Let ω_α be an initial ordinal and let F be set such that $2 \leq \mathfrak{p} = \mathrm{Card}(F) < \aleph_\alpha$. Let E be the set of mappings of segments of ω_α, other then ω_α itself, into F. The we have $\mathrm{Card}(E) \leq \mathfrak{p}^{\aleph_\alpha}$.

For each mapping f of ω_α into F, let K_f be the subset of E consisting of the restrictions of f to the segments of ω_α (other than ω_α itself). Show that the set $\mathfrak{F}$ of subsets of K_f is a covering of E such that $\mathrm{Card}(\mathfrak{F}) = \mathfrak{p}^{\aleph_\alpha}$ and that its degree of disjointness is equal to ω_α.

(c) Let E be an infinite set of cardinal $\mathfrak{a}$ and let $\mathfrak{c}$, $\mathfrak{p}$ be two cardinal > 1 such that $\mathfrak{p} < \mathfrak{c}$, $\mathfrak{p}^{\mathfrak{m}} < \mathfrak{a}$ for all $\mathfrak{m} < \mathfrak{c}$ and $\mathfrak{a} = \sum_{\mathfrak{m}<\mathfrak{c}} \mathfrak{p}^{\mathfrak{m}}$. Deduce from (b) that there exists a covering $\mathfrak{F}$ of E consisting of sets of cardinal $\mathfrak{c}$, with degree of disjunction equal to $\mathfrak{c}$ and such that $\mathrm{Card}(\mathfrak{F}) = \mathfrak{p}^{\mathfrak{c}}$. In particular, if E is countably infinite, there exists a covering $\mathfrak{F}$ of E by infinite sets such that $\mathrm{card}(\mathfrak{F}) = 2^{\aleph_0}$ and such that the intersection of any two sets of $\mathfrak{F}$ is *finite*.

¶ 26. Let E be an infinite set and let $\mathfrak{F}$ be a set of subsets of E such that for $A \in \mathfrak{F}$ we have

$$\mathrm{card}(A) = \mathrm{card}(\mathfrak{F}) = \mathrm{card}(E) = \mathfrak{a} \geq \aleph_0.$$

Show that there exists a partition $(B_\iota)_{\iota\in I}$ of E such that

$$\mathrm{card}(I) = \mathrm{card}(B_\iota) = \mathfrak{a}$$

for all $\iota \in I$ and such that $A \cap B_\iota \neq \emptyset$ for all $A \in \mathfrak{F}$ and all $\iota \in I$. (With the notation of Exercise 24 (a), consider first a surjective mapping f of ω_α into $\mathfrak{F}$ such that for each $A \in \mathfrak{F}$ the set of all $\xi \in \omega_\alpha$ such that $f(\xi) = A$ has cardinal equal to $\mathfrak{a}$. Then, by transfinite induction, define a bijection g of ω_α onto E such that $g(\xi) \in f(\xi)$ for every $\xi \in \omega_\alpha$.)

¶ 27. Let L be an infinite set and let $(E_\lambda)_{\lambda\in L}$ be a family of sets indexed by L. Suppose that for each integer $n > 0$ the set of $\lambda \in L$ such that $\mathrm{card}(E_\lambda) > n$ is equipotent to L. Show that there exists a subset F of the product $E = \prod_{\lambda\in L} E_\lambda$, such that $\mathrm{Card}(F) = 2^{\mathrm{Card}(L)}$, and such that F has the following property: for each finite sequence $(f_k)_{1\leq k\leq n}$ of distinct elements of F there exists $\lambda \in L$ such that the elements $f_k(\lambda) \in E_\lambda$ $(1 \leq k \leq n)$ are all distinct. (Show first that there exists a partition $(L_j)_{j\in\mathbf{N}}$ of L such that $\mathrm{Card}(L_j) = \mathrm{Card}(L)$ for all j, and such that $\mathrm{Card}(E_\lambda) \geq 2^j$ for each $\lambda \in L_j$. Hence reduce to the case where L is the sum of the countable family of sets X^j $(j \geq 1)$, where X is an infinite set, and $E_\lambda = 2^j$ for each $\lambda \in X^j$. With each mapping $g \in 2^X$ of X into 2, associate the element $f \in E$ such that $f(\lambda) = (g(x_1), \ldots, g(x_j))$ whenever $\lambda = (x_k)_{1\leq k\leq j} \in X^j$; show that the set F of elements $f \in E$ so defined has the required property.)

¶ 28. Let E be an infinite set and let $(\mathfrak{X}_i)_{1\leq i\leq m}$ be a finite partition of the set $\mathfrak{F}_n(E)$ of subsets of E having n elements. Show that there exists an index i and an infinite subset F of E such that every subset of F with n elements belongs to $\mathfrak{X}_i$. (Proof by induction on n. For each $a \in E$ show that there exists an index $j(a)$ and an infinite subset $M(a)$ of $E - \{a\}$ such that, for every subset A of $M(a)$ with $n-1$ elements, $\{a\} \cup A$ belongs to $\mathfrak{X}_{j(a)}$. Then define a sequence (a_i) of elements of E as follows: a_1 is an arbitrary element of E, a_2 is an arbitrary element of $M(a_1)$, a_3 is defined in terms of $M(a_1)$ and a_2 in the same way as a_2 was defined in terms of E and a_1, and so on. Show that the set F of elements of a suitable subsequence of the sequence (a_i) satisfies the required conditions).

29. In an ordered set E, every finite union of Noetherian subsets (with respect to the induced ordering) is Noetherian.

(b) An ordered set E is Noetherian if and only if for each $a \in E$, the interval $]a, \rightarrow [$ is Noetherian.

(c) Let E be an ordered set such that the ordered set obtained by endowing E with the opposite ordering is Noetherian. Let u be a letter and let $\mathrm{T}\{u\}$ be a term. Show that there exists a set U and a mapping f of E onto U such that for each $x \in \mathrm{E}$ we have $f(x) = \mathrm{T}\{f^{(x)}\}$, where $f^{(x)}$ denotes the mapping of $]\leftarrow, x[$ onto $]\leftarrow, f(x)[$ which coincides with f on this interval. Furthermore U and f are determined uniquely by this condition.

(d) Let E be a Noetherian ordered set such that every finite subset of E has a least upper bound in E. Show that, if E has a least element, then E is a complete lattice (§ 1, Exercise 11); and that if E has no least element, the set E' obtained by adjoining a least element to E (§ 1, no. 7, Proposition 3) is a complete lattice.

30. Let E be a lattice such that the set obtained by endowing E with the opposite ordering is Noetherian. Show that every element $a \in \mathrm{E}$ can be written as $\sup(e_1, e_2, \dots, e_n)$ where $e_1, \dots, e_n$ are irreducible (§ 4, Exercise 7; show first that there exists an irreducible element e such that $a = \sup(e, b)$ if a is not irreducible). Generalize Exercise 7 (b) of § 4 to E; also generalize Exercises 8(b) and 9(b) of § 4.

¶ 31. Let A be an infinite set and let E be the set <of all infinite subsets of A, ordered by inclusion. Show that E is completely ramified (§ 2, Exercise 8) but not antidirected (§ 1, Exercise 23) and that E has an antidirected cofinal subset F. (Consider first the set $\mathfrak{D}(\mathrm{A})$ of countable infinite subsets of A (which is cofinal in E) and let $\mathrm{Z} = \mathrm{R}_0(\mathfrak{D}(\mathrm{A}))$ (§ 1, Exercise 23). Write Z in the form $(z_\lambda)_{\lambda \in \mathrm{L}}$, where L is a well-ordered set, and take F to be a set of countable subsets X_n^λ, where λ runs through a suitable subset of L, $n \in \mathbf{N}$, $\mathrm{X}_m^\lambda \supset \mathrm{X}_n^\lambda$ whenever $m \leq n$, $\mathrm{X}_n^\lambda - \mathrm{X}_{n+1}^\lambda$ is infinite for all $n \geq 0$ and $\cup_{n \in \mathbf{N}} \mathrm{X}_n^\lambda = \emptyset$; the X_n^λ, are to be defined by transfinite induction in such a way that the images of the sets X_n^λ, under the canonical mapping $r : \mathfrak{D}(\mathrm{A}) \to \mathrm{Z}$ (§ 1, Exercise 23) are mutually disjoint and form a cofinal subset of Z.)

¶ 32. * Let (M_n), (P_n) be two sequences of mutually disjoint finite sets (not all empty), indexed by the set $\mathbf{Z}$ of rational integers. Let $\alpha_n = \mathrm{Card}(\mathrm{M}_n)$, $\beta_n = \mathrm{card}(\mathrm{P}_n)$. Suppose that there exists an integer $k > 0$ such that for each $n \in \mathbf{Z}$ and each integer $l \geq 1$ we have

$$\alpha_n + \alpha_{n+1} \cdots + \alpha_{n+l} \leq \beta_{n-k} + \beta_{n-k+1} + \cdots + \beta_{n+l+k},$$

$$\beta_n + \beta_{n+1} \cdots + \beta_{n+l} \leq \alpha_{n-k} + \alpha_{n-k+1} + \cdots + \alpha_{n+l+k}.$$

Let M be the union of the family (M_n) and let P be the union of the family (P_n). Show that there exists a bijection ϕ of M onto P such that

$$\phi(\mathrm{M}_n) \subset \bigcup_{i=n-k-1}^{n+k+1} \mathrm{P}_i \text{ and } \overset{-1}{\phi}(\mathrm{P}_n) \subset \bigcup_{i=n-k-1}^{n+k+1} \mathrm{M}_i$$

for each $n \in \mathbf{Z}$ (consider a total ordering on each M_n (resp. P_n) and take M (resp. P) to be the ordinal sum (§ 1, Exercise 3) of the family $(\mathrm{M}_n)_{n \in \mathbf{Z}}$ (resp. $(\mathrm{P}_n)_{n \in \mathbf{Z}}$). If n_0 is an index such that $\mathrm{M}_{n_0} \neq \emptyset$ consider the isomorphisms of M onto P which transform the least element of M_{n_0} into one of the elements of $\cup_{j=n_0-k}^{n_0+k} \mathrm{P}_j$ and show that one of these isomorphisms satisfies the required conditions. Let δ be the least of the numbers

$$\beta_{n-k} + \beta_{n-k+1} + \cdots + \beta_{n+l+k} - (\alpha_n + \alpha_{n+1} \cdots + \alpha_{n+l}),$$

$$\alpha_{n-k} + \alpha_{n-k+1} + \cdots + \alpha_{n+l+k} - (\beta_n + \beta_{n+1} \cdots + \beta_{n+l})$$

for all $n \in \mathbf{Z}$ and all $l \geq 1$. If $n \in \mathbf{Z}$ and $l \geq 1$ are such that, for example $\beta_{n-k} + \beta_{n-k+1} + \cdots + \beta_{n+l+k} = \delta + \alpha_n + \alpha_{n+1} \cdots + \alpha_{n+l}$ we may take ϕ to be such that the least element of P_{n-k} is the image under ϕ of the least element of M_n. *

¶ 33. Soient $\mathfrak{a}$, $\mathfrak{b}$ deux cardinaux tels que $\mathfrak{a} \geq 2$, $\mathfrak{b} \geq 1$, l'un au moins des deux étant infini. Soient E un ensemble, $\mathfrak{F}$ une partie de $\mathfrak{P}(E)$, telle que $\text{card}(\mathfrak{F}) > \mathfrak{a}^{\mathfrak{b}}$ et $\text{card}(X) \leq \mathfrak{b}$ pour tout $X \in \mathfrak{F}$. On se propose de montrer qu'il existe une partie $\mathfrak{G} \subset \mathfrak{F}$ telle que $\text{card}(\mathfrak{G}) > \mathfrak{a}^{\mathfrak{b}}$ et que deux quelconques des ensembles appartenant à $\mathfrak{G}$ aient *la même intersection.* On pourra procéder de la façon suivante.

a) soit $\mathfrak{c}$ le plus petit des cardinaux $> \mathfrak{a}^{\mathfrak{b}}$ et soit Ω le plus petit ordinal de cardinal $\mathfrak{c}$. On considère une application injective $\nu \mapsto X(\nu)$ de Ω dans $\mathfrak{F}$ et on pose $M = \cup_{\nu\in\Omega} X(\nu)$; on peut supposer que $\text{card}(M) = \mathfrak{c}$, et il y a donc une bijection $\nu \mapsto x_\nu$ de Ω sur M, ordonnant M.

b) Pour tout $\nu \in \Omega$ soit ρ_ν l'ordinal type d'ordre du sous-ensemble $X(\nu)$ de M (on a $\text{Card}(\rho_\nu) \geq \mathfrak{b}$) et soit $\mu \mapsto y_\mu^{(\nu)}$ l'unique application bijective croissante de ρ_ν sur $X(\nu)$. On note M_μ l'ensemble des $y_\mu^{(\nu)}$ lorsque ν parcourt Ω. Montrer qu'il existe au moins un ordinal μ tel que $\text{Card}(M_\mu) = \mathfrak{c}$. On désigne par α le *plus petit* de ces ordinaux; la réunion de M_γ pour $\gamma < \alpha$ a un cardinal $\leq \mathfrak{a}^{\mathfrak{b}} < \mathfrak{c}$.

c) Montrer qu'il existe une partie $N_0 \subset \Omega$ telle que $\text{Card}(N_0) = c$ et que l'application $\nu \mapsto y_\alpha^{(\nu)}$ de N_0 dans M soit injective. Montrer, par récurrence ur β, qu'il existe une partie $N_\beta \subset N_0$ de cardinal $\mathfrak{c}$ telle que l'élément $y_\lambda^{(\nu)} = z_\lambda$ soit indépendant de ν pour $\nu \in N_\beta$ et pour tout $\lambda \leq \beta$. Montrer que l'intersection N des N_β pout $\beta < \alpha$ a pour cardinal $\mathfrak{c}$ (considérer son complémentaire). Soit Q l'ensemble des z_λ pour $\lambda < \alpha$.

d) Pour tout $\nu \in N$, on définit par récurrence un ordinal λ_ν par la condition suivante : c'est le plus petit ordinal dans N tel que $y_\alpha^{(\lambda_\nu)}$ soit un majorant strict, dans M, de la réunion des $X(\lambda_\mu)$ pour $\mu < \nu$, $\mu \in N$. Montrer que pour $\mu < \nu$ dans N on a $(\lambda_\mu) \cap (\lambda_\nu) = Q$.

Chapter 11

Compatibility

This chapter explains the differences between the current version and the previous one.

11.1 Changes to Chapter 5

11.2 Pseudo Ordinals

This section explains the previous implementation of pseudo-ordinals, it has been replaced by section 4.1.2.

Consider the following properties:

(1) $\emptyset \in \mathrm{E}$.

(2) E is transitive.

(3) The relation "$x \in \mathrm{E}$ and $y \in \mathrm{E}$ and ($x = y$ or $x \in y$)" is a well-ordering of E.

(4) The relation "$x \in \mathrm{E}$ and $y \in \mathrm{E}$ and $x \subset y$" is a well-ordering of E.

(5) The relation "$x \in \mathrm{E}$ and $y \in \mathrm{E}$ and $x \in y$" is a strict well-ordering of E.

(6) E is asymmetric for $\in$.

(7) $\forall x \in \mathrm{E}, x =]\leftarrow, x[$.

(8) Every transitive subset of E is E or an element of E.

(9) $\forall x \in \mathrm{E}, \forall y \in \mathrm{E} \implies$ exactly one of $x \in y, y \in x, x = y$.

(10) $\forall x \in \mathrm{E}, f(x) = f\langle]\leftarrow, x[\rangle$.

Different variants of ordinals can be found in the litterature. The 1956 Edition of Bourbaki considered (1), (2) and (3), the current edition uses (8); the most common definition is (2) and (5) (for instance [9, 1]. The von Neumann definition (see [12]) uses (4) and (7).

Let's start with a few comments. Condition (6) say that E is decent, condition (5) is equivalent to (3)(6), it implies (9). We shall see that the orderings defined by (3), (4), (5) are the same, they define the natural ordering $o(\mathrm{E})$ of relation (OP).

According to von Neumann, an ordinal is a well-ordered set satisfying (7), where $]\leftarrow, x[$ is the set of elements $y \in \mathrm{E}$ satisfying $y < x$. The relation $x =]\leftarrow, x[$ means: for any element x of E, the relation $y \in x$ is equivalent to $y \in \mathrm{E}$ and $y < x$. From $y \in \mathrm{E}$ we deduce that E is transitive. On the other hand, if x and y belong to E, $y \in x$ is equivalent to $y < x$, so that $a \leq b$

is equivalent to $a = b$ or $a \in b$. If $x \leq y$ we have $]\leftarrow, x[\subset]\leftarrow, y[$, thus $x \subset y$. Conversely, if $x \subset y$, since E is totally ordered, we have $x \leq y$ or $y \leq x$. In the second case we have $y \subset x$ thus $x = y$ by extensionality. Thus we have shown: $a \leq b$ if and only if $a \subset b$. As a consequence, the ordering of E satisfies (3) and (4). It satisfies (5) since $x \notin x$ is a consequence of $x \notin]\leftarrow, x[$. A *numeration* of an ordered set E is function f satisfying (10). This means that $f(x)$ is the set of all $f(y)$ for $y < x$. Relation (7) says that the identity function is a numeration. By transfinite induction, every well-ordered set has a numeration; relation (OP) follows from the fact that the image of a numeration is an ordinal.

In the 1956 version, Bourbaki defined an ordinal via (1), (2), and (3). Consider a set W such that $W = \{\emptyset, W\}$. According to the axiom of foundation, no such set exists; according to AFA (anti-foundation axiom) there is a unique set satisfying this conditition. These two axioms being independent of the Bourbaki theory, whether or not W exists is undecidable. This set is not decent, but it satisfies (1), (2) and (3), thus was an ordinal according to the 1956 Edition of Bourbaki. It is isomorphic to the set $\{\emptyset, \{\emptyset\}\}$, contradicting uniqueness of (OP). According to (1), the empty set is not an ordinal, contradicting existence. Note that the least element of a non-empty ordinal cannot have elements (by transitivity), thus must be empty.

We start with the definition of [9]. There are four conditions K_1, K_2, K_3 and K_4. Conditions K_1 and K_4 say that E is decent and transitive. Condition K_2 says that $x \in y$ is a transitive relation on E. We show here that K_1 and K_2 say that the relation "$x \in y$ or $x = y$" is an ordering on E. Condition K_3 will imply that $\in$ is a well-ordering.

```
Definition Kordinal a :=
  ( (forall  x y, inc x a -> inc y a -> inc x y -> inc y x -> False)
   & (forall x y z, inc x a -> inc y a -> inc z a ->
       inc x y -> inc y z -> inc x z)
   & (forall z, sub z a -> nonempty z ->
       exists x, (inc x z & forall y, inc y z -> inc x y \/ x = y))
   & (forall x y, inc x a -> inc y x -> inc y a)).
```

Condition K_3 implies that $x \in y$ is a strict total ordering (at least one of $x \in y$, $y \in x$ or $x = y$ is true). It implies that $x \in y$ is equivalent to $x \subsetneq y$, and that $x \subset y$ is equivalent to "$x \in y$ or $x = y$". Thus $\subset$ is a well-ordering.

```
Lemma Kordinal_asymmetric: forall E, Kordinal E -> asymmetric_set E.
Lemma Kordinal_decent: forall E, Kordinal E -> decent_set E.
Lemma Kordinal_irreflexive: forall E , Kordinal E -> inc E E -> False.
Lemma Kordinal_transitive: forall E, Kordinal E -> transitive_set E.
Lemma Kordinal_trichotomy: forall E x y, Kordinal E ->
  inc x E -> inc y E -> (inc x y \/ inc y x \/ x = y).
Lemma Kordinal_inclusion: forall E, Kordinal E ->
  forall x y, inc x E -> inc y E -> (inc x y = strict_sub x y).
Lemma Kordinal_inclusion1: forall E, Kordinal E ->
  forall x y, inc x E -> inc y E -> (sub x y = (inc x y \/ x = y)).
Lemma Kordinal_inclusion2:  forall E,
  Kordinal E -> worder (inclusion_suborder E).
```

We now show that K_3 implies that $\in$ is a well-ordering, so that the definition of [9] is equivalent to (2), (3) and (6), in other terms: E is transitive, asymmetric and is a well-ordering.

```
Lemma trans_sym_order: forall x,
  (forall y,inc y x ->  transitive_set y) -> asymmetric_set x ->
```

```
  (order (ordinal_oa x) & substrate (ordinal_oa x) = x). (* 18 *)
Lemma Kordinal_elt1:  forall E,
  Kordinal E -> worder (ordinal_oa E).
Lemma Kordinal_pr:  forall E,
  Kordinal E =
  (transitive_set E & worder (ordinal_oa E) & asymmetric_set E). (* 24 *)
```

If the set E is ordered by $\in$, the segment $]\leftarrow, x[$ is the set of all elements y in E such that $y \in x$. This is the intersection of x and E. If E is transitive, this is x. In particular, if E is a K-ordinal, we have $x =]\leftarrow, x[$. The same is true if we consider the ordering $\subset$.

We deduce that K-ordinals satisfy the von Neumann condition (7). The converse is true by the argument explained above (relation (7) says that E is transitive and that $x \in y$ is equivalent to $x \subsetneq y$. This implies that E is asymmetric and that the ordering induced by $\subset$ is the same as that of $\in$).

```
Lemma Kordinal_segment1: forall E x, decent_set E ->
  inc x E -> segment (ordinal_oa E) x = intersection2 x E.
Lemma Kordinal_segment2: forall E x, decent_set E -> transitive_set E->
  inc x E -> segment (ordinal_oa E) x = x.
Lemma Kordinal_segment3: forall E x, Kordinal E ->
  inc x E -> segment (ordinal_oa E) x = x.
Lemma Kordinal_segment4: forall E x, Kordinal E ->
  inc x E -> segment (inclusion_suborder E) x = x.

Lemma Kordinal_pr2:  forall E,
  Kordinal E =
    (worder (inclusion_suborder E) &
    (forall x, inc x E -> segment (inclusion_suborder E) x = x)).
```

Every element of a K-ordinal is transitive since we can replace $\in$ by $\subsetneq$ which is transitive. Every transitive subset of E is a K-ordinal (this comes directly from the definition; it is also a consequence of the fact that a subset of a well-ordered set is well-ordered). Thus, every element of a K-ordinal is a K-ordinal. We now state: (9) is true whenever x and y are K-ordinals (there is no need to add the restrictions $x \in$ E and $y \in$ E). In fact, the intersection of two ordinals is a segment of each one. Since the sets are well-ordered, a segment is the whose set or of the form $]\leftarrow, x[$, and we know that this is x. Hence we get $x \cap y = x$ or $x \cap y \in x$, and similarly, $x \cap y = y$ or $x \cap y \in y$; the conclusion follows since $x \cap y \notin x \cap y$. It follows that, if E is a K-ordinal, X a transitive strict subset of E, then X is a K-ordinal, and X $\in$ E, so that E is an ordinal in the Bourbaki sense.

```
Lemma Kordinal_sub_trans: forall E x, Kordinal E -> inc x E ->
  transitive_set x.
Lemma Kordinal_sub_ordinal: forall E x, Kordinal E -> sub x E ->
  transitive_set x -> Kordinal x.
Lemma Kordinal_inc_ordinal: forall E x, Kordinal E -> inc x E ->
  Kordinal x.
Lemma Kordinal_trichotomy1 : forall x y,
  Kordinal x -> Kordinal y ->
  (inc x y \/ inc y x \/ x = y). (* 29 *)
```

Bourbaki defines an ordinal as a set E that satisfies (8). The following result, can be restated (thanks to Kordinal_pr) as: a Bourbaki ordinal is transiticve, asymmetric, and well-ordered by $\in$. This property has been prove in the main text, section 4.1.2.

```
Lemma Kordinal_pr3:  forall E,
  Kordinal E = Bordinal E.
```

11.2.1 Cardinals

This is how Bourbaki defines a cardinal, and the cardinals zero, one and two.

```
(*
  Definition cardinal x := choose (fun z => equipotent x z).
  Definition card_zero := cardinal emptyset.
  Definition card_one := cardinal (singleton emptyset).
  Definition card_two := cardinal (two_points).
*)
```

Using von Neumann ordinals for cardinals makes some theorems easier.

Theorem one [3, p. 159] says that the ordering between cardinals is a well-ordering. The idea is the following. Let E be a set of cardinals, and A its union. Consider a well-ordering on A. Let $\phi(x)$ be the smallest segment of A equipotent to x (if $x \in$ E, x is a subset of A hence isomorphic, hence equipotent, to a segment of A; hence ϕ is well-defined). The relation $a \leq_{\mathrm{Card}} b$ on E is equivalent to $\phi(a) \subset \phi(b)$ (if $a \leq_{\mathrm{Card}} b$ then a is isomorphic to a subset of $\phi(b)$, hence a is isomorphic to a segment u of $\phi(b)$; by definition $\phi(a) \subset u$, hence $\phi(a) \subset \phi(b)$; converse is easy). From this, one deduces that the relation $a \leq_{\mathrm{Card}} b$ is an order on E. This is a well-ordering, since the set of segments is well-ordered. Assume $a \leq_{\mathrm{Card}} b$ and $b \leq_{\mathrm{Card}} a$. If we consider the doubleton $\{a, b\}$, we have $a \leq b$ and $b \leq a$ for the induced order, hence $a = b$.

If a is equipotent to a subset of b, and conversely then a is equipotent to b. This is equivalent to the Cantor-Bernstein theorem. Since $a \leq_{\mathrm{Card}} b$ is a well-ordering it is a total ordering.

```
Lemma cardinal_le5: forall E, (forall x, inc x E -> is_cardinal x) ->
  substrate (graph_on cardinal_le E) = E.
Theorem wordering_cardinal_le: worder_r  cardinal_le.  (* 137 *)
Lemma cardinal_le7: forall a b c,
  equipotent b c -> equipotent_to_subset a b ->
  equipotent_to_subset a c.
Lemma cardinal_antisymmetry2: forall a b,
  equipotent_to_subset a b -> equipotent_to_subset b a ->
  equipotent a b.
Lemma cardinal_le_total_order: forall a b,
  equipotent_to_subset a b \/ equipotent_to_subset b a.
```

For every cardinal $\mathfrak{a}$, the set of objects of the form Card(b) for $b \in \mathfrak{P}(\mathfrak{a})$ is the set of cardinals $\leq \mathfrak{a}$.

Since $\leq_{\mathrm{Card}}$ is a well-ordering, thus total, a finite cardinal is always smaller than an infinite cardinal. Fix some infinite cardinal $\mathfrak{b}$ (for instance the cardinal of $\mathbb{N}$), and consider the subset of $\{\mathfrak{a}, \mathfrak{a}$ is cardinal and $\mathfrak{a} \leq_{\mathrm{Card}} \mathfrak{b}\}$ formed of finite cardinals. It contains all finite cardinals. It is independent of $\mathfrak{b}$. It will be called the set of natural integers and denoted by Bnat or **N**.

```
Definition set_of_cardinals_le a:=
  fun_image(powerset a)(fun x => cardinal x).
Definition Bnat := Zo(set_of_cardinals_le (cardinal nat))
  (fun z => finite_c z).
```

This is the old definiton of the predecessor function.

```
Definition predc n := choose (fun m => is_cardinal m & n = succ m).
Lemma predc_pr0: forall n, is_cardinal n -> n <> card_zero ->
  (is_cardinal (predc n) & n = succ (predc n)).
Theorem exists_predc: forall n, inc n Bnat -> n <> card_zero ->
  exists_unique (fun m => inc m Bnat & n = succ m).
```

11.2.2 The von Neumann Proof

We have shown that for any well-ordered set E there exists an isomorphism f, defined by transfinite induction, whose target is some ordinal E′. Von Neumann calls a function satisfying (10) a *numeration* (we shall use here a functional graph, instead of a function). He shows that it is unique, that it exists, and defines an ordinal as the image of the graph.

```
Definition numeration r f:=
  fgraph f &  domain f = substrate r &
  forall x, inc x (domain f) ->  V x f = image_by_graph f (segment r x).
Definition  numerable r :=
   worder r & exists f, numeration r f.
Definition the_numeration r := transfinite_defined r target.
```

The proofs are rather straightforward (compare with the proof of existence of a function by transfinite induction). The key relation is that, if n(E) is the numeration of E, and F a segment of E then n(F) is the restriction of n(E) to F. If all segments of E can be numerated, we can extent these numerations to a numeration of E; otherwise, there is a least non-numerable segment; all its segments can be numerated, so that the segment can be numerated; absurd.

```
Lemma worder_numerable: forall r,
  worder r -> numeration r (graph (the_numeration r)).
Lemma numeration_unique: forall r f f',
  worder r -> numeration r f -> numeration r f' -> f = f'.
Lemma sub_numeration: forall  r f x,
  let r' := induced_order r (segment r x) in
  worder r ->  numeration r f -> inc x (substrate r) ->
  numeration r' (restr f (segment r x)).
Lemma segments_numerables: forall r,
  let r' := fun x => induced_order r (segment r x) in
  worder r -> (forall x, inc x (substrate r)-> numerable (r' x)) ->
  numerable r. (* 64 *)
Lemma worder_numerable_bis: forall r,
  worder r ->  numerable r. (* 22 *)
```

11.2.3 Pseudo-ordinals and the type nat

Our implementation of Bourbaki in Coq relies on the fact that a set is a type, and if a is a set, $a \in$ B means $a = \mathscr{R}b$ for some b of type B (where $\mathscr{R}$ denotes Ro). This is an abstract construction. If we define a type A with two constructors B and C, then $\mathscr{R}$B ∈ A and $\mathscr{R}$C ∈ A. We assume $\mathscr{R}$ injective; since B ≠ C by construction, the set A has two distinct elements $\mathscr{R}$B and $\mathscr{R}$C. The only property of $\mathscr{R}$B is that it is one of the two elements of A. A property of the form $\mathscr{R}$B = ∅ is undecidable.

Let's now define a more complicated type, nat, denoted ℕ; it has a constant constructor O, and another constructor S that is a function on ℕ. This means that, whenever x is of type

$\mathbb{N}$, then Sx is also of type $\mathbb{N}$. This set satisfies the principle of induction (that says under which condition a property is true for the elements of this set), and we can define a function by induction. Later on, Bourbaki introduces **N**, the set of finite cardinals. It satisfies the principle of induction (see section 5.4), and functions can be defined by induction (Chapter six, section 7.2), as a variant of definition by transfinite induction of the well-ordered set **N**. In the next section, we shall show that $\mathbb{N}$ and **N** are isomorphic; in this section we compare $\mathbb{N}$ and the collection of finite pseudo-ordinals (this is in fact a set, but it will not be used).

Since Sn is of type $\mathbb{N}$ whenever n is of type $\mathbb{N}$, we can define a function s such that $s(a) \in \mathbf{N}$ whenever $a \in \mathbb{N}$: if $a \in \mathbb{N}$ and $a = \mathscr{R}(b)$ then $s(a) = \mathscr{R}(\mathrm{S}(b))$. As noted above, a property of the form $\mathscr{R}\mathrm{O} = \emptyset$ is undecidable. Although the exact value of $s(\mathscr{R}\mathrm{O})$ is unknown, we can show some properties of s: for instance, s is injective and not surjective (there is a unique way to construct an object of type $\mathbb{N}$; so that Sx is never O and Sx = Sy implies $x = y$). The Coq parser and pretty printer identify O and 0, SO and 1, SSO and 2. In order to avoid confusion, we shall write **0**, **1** and **2** for the cardinals (note that $\mathbf{0} = \emptyset$).

Let's define by induction a function f by $f(\mathbf{0}) = \mathscr{R}\mathrm{O}$ and $f(n+\mathbf{1}) = \mathscr{R}(\mathrm{S}(\mathscr{B}(f(n))))$ where $a = \mathscr{B}(f(n))$ is defined by $\mathscr{R}a = f(n)$. For instance $f(\mathbf{1}) = \mathscr{R}(1)$. The property "$f$ is the identity function" (more precisely $f = \mathscr{R}$) is undecidable (it cannot be proved; we hope that adding it as an axiom does not make the theory contradictory).

The type $\mathbb{N}$ is called nat in Coq; it has an order relation, noted $\leq$, and two operations + and $*$, that correspond, via the bijection f, to comparison, sum and product of finite cardinals. We shall import all theorems about natural integers from the Coq library by identification of **N** and $\mathbb{N}$. Given that $\emptyset = f(\mathbf{0}) = \mathscr{R}0 = \mathscr{R}\mathrm{O}$, we may assume $\mathscr{R}\mathrm{O} = \emptyset$. The relation $f(\mathbf{1}) = \mathscr{R}(1)$ suggest that $\mathscr{R}(1)$ should be **1**, but this is a set defined via the axiom of choice, as a set with one element; it could be $\{\emptyset\}$, it could also be any other set. We will add the relation $\mathscr{R}(\mathrm{SO}) = \{\emptyset\}$ as axiom. As a consequence $\mathscr{R}(\mathrm{SO})$ is unlikely to be a cardinal, but it will allow us to construct a function card, such that $\mathrm{card}(x) = 1$ whenever x is a singleton, i.e., whenever $\mathrm{Card}(x) = \mathbf{1}$. The two axioms relating $\mathscr{R}$, S and O have been introduced by Carlos Simpson in the following way:

```
(*
   Axiom nat_realization_0 : forall x : Set, ~ inc x (Ro 0).
   Axiom nat_realization_S :
          forall (n : nat) (x : Set),
          inc x (Ro (S n)) = (inc x (Ro n) \/ x = Ro n).
   Lemma nat_zero_emptyset : Ro 0 = emptyset.
*)
```

These axioms are useless, hence have been withdrawn. On the other hand, we can define a function that shares exactly the same properties. The first axioms defines a set $\mathscr{R}0$ that contains no element, hence is the emptyset. The second axioms defines $\mathscr{R}(\mathrm{S}n)$, that is equal (by extensionality) to $\mathrm{T}(\mathscr{R}n)$. Thus, we define natR denoted by $\mathscr{R}_{\mathbb{N}}$, via $\mathscr{R}_{\mathbb{N}}0 = \emptyset$ and $\mathscr{R}_{\mathbb{N}}(\mathrm{S}n) = \mathrm{T}(\mathscr{R}_{\mathbb{N}}n)$.

```
Fixpoint natR (n:nat) :=
  match n with 0 =>  emptyset
             | S p => tack_on (natR p) (natR p)
  end.
```

The conclusion of Exercise 20 is: *In particular the pseudo-ordinals whose order-type are* $0, 1, 2 = 1+1,$ *and* $3 = 2+1$ *are respectively*

$$\emptyset, \quad \{\emptyset\}, \quad \{\emptyset,\{\emptyset\}\}, \quad \{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}.$$

```
Lemma value_R_0: natR 0 = emptyset.
Lemma value_R_1: natR 1 = singleton emptyset.
Lemma value_R_2: natR 2 = doubleton (singleton emptyset) emptyset.
Lemma value_R_3: let tripleton a b c := tack_on (doubleton a b) c in
  natR 3 = tripleton (doubleton (singleton emptyset) emptyset)
  (singleton emptyset) emptyset.
```

If a is a pseudo-ordinal, then so is $\mathrm{T}(a)$. By induction, we deduce that if n is of type nat, $\mathscr{R}_{\mathbb{N}}n$ is a finite pseudo-ordinal.

```
Lemma pseudo_ordinal_Rnat: forall i,  pseudo_ordinal (natR i).
Lemma finite_Rnat: forall i, is_finite_set (natR i).
```

Define a relation $\leq$ on nat by the properties: $\forall x, x \leq x$ and $\forall xy,\ x \leq y \implies x \leq \mathrm{S}(y)$. This is reflexive and transitive. Showing that it is an order is not completely trivial (see the Coq library). One can show that the relation $x < y$, defined by $\mathrm{S}x \leq y$, is equivalent to $x \leq y$ and $x \neq y$.

It is clear by induction that if $x \leq y$ then $\mathscr{R}_{\mathbb{N}}x \subset \mathscr{R}_{\mathbb{N}}y$. If $x < y$ then $\mathscr{R}_{\mathbb{N}}(\mathrm{S}x) \subset \mathscr{R}_{\mathbb{N}}y$ hence $\mathscr{R}_{\mathbb{N}}x \in \mathscr{R}_{\mathbb{N}}y$. If $\mathscr{R}_{\mathbb{N}}x \subset \mathscr{R}_{\mathbb{N}}y$, then $x \leq y$, for otherwise we would have $y < x$, hence $\mathscr{R}_{\mathbb{N}}y \in \mathscr{R}_{\mathbb{N}}x$, hence $\mathscr{R}_{\mathbb{N}}x \in \mathscr{R}_{\mathbb{N}}x$, absurd. In the same fashion, $x < y$ is equivalent to $\mathscr{R}_{\mathbb{N}}(\mathrm{S}x) \subset \mathscr{R}_{\mathbb{N}}y$. Note that, if $\mathscr{R}_{\mathbb{N}}i = \mathscr{R}_{\mathbb{N}}j$, then $\mathscr{R}_{\mathbb{N}}i \subset \mathscr{R}_{\mathbb{N}}j$ and $\mathscr{R}_{\mathbb{N}}j \subset \mathscr{R}i$, this $i \leq j$ and $j \leq i$; hence $i = j$. This shows injectivity of $\mathscr{R}_{\mathbb{N}}$.

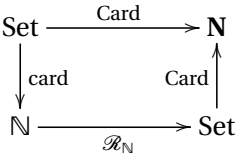
```
Lemma Rnat_le_implies_sub : forall i j,  i <= j -> sub (natR i) (natR j).
Lemma Rnat_lt_implies_inc : forall i j,  i < j -> inc (natR i) (natR j).
Lemma Rnat_lt_implies_strict_sub : forall i j,
  i < j -> strict_sub (natR i) (natR j).
Lemma Rnat_sub_le : forall i j,  sub (natR i) (natR j) = (i <= j).
Lemma Rnot_inc_itself: forall i, ~ (inc (natR i)(natR i)).
Lemma Rnat_inc_lt : forall i j,  inc (natR i) (natR j) = (i < j).
```

As a consequence, if $i < j$ then $\mathrm{Card}(\mathscr{R}_{\mathbb{N}}i) <_{\mathrm{Card}} \mathrm{Card}(\mathscr{R}_{\mathbb{N}}j)$, and if $\mathrm{Card}(\mathscr{R}_{\mathbb{N}}i) = \mathrm{Card}(\mathscr{R}_{\mathbb{N}}j)$, then $i = j$. From this, we deduce that each finite cardinal is of the form $\mathrm{Card}(\mathscr{R}_{\mathbb{N}}(i))$ for a unique i.

```
Lemma cardinal_Rnat_lt: forall i j,
  i< j -> cardinal_lt (cardinal (natR i)) (cardinal (natR j)).
Lemma  cardinal_Rnat_inj: forall i j,
  cardinal (natR i) = cardinal (natR j) -> i = j.
Lemma exists_nat_cardinal: forall a, is_finite_c a ->
  exists_unique(fun i:nat => cardinal (natR i)=a).
```

We can now introduce a function $\mathrm{card}(n)$ making the following diagram commutative

(Finite cardinals)



In this diagram, $\mathbb{N}$, Set and **N** are types; $\mathbb{N}$ is the set of natural numbers (as a Coq type), and **N** is the collection of cardinals (the objects x such that x is a cardinal do not form a set,

neither a type, we call it a *collection*). We use the notation **N** to emphasize that if x is a finite set, then its cardinal is a member of the set of finite cardinals. If x is not finite, we define card(x) to be 0, in this case the diagram does not commute. If however n is finite we have Card($\mathscr{R}_{\mathbb{N}}$(card(n))) = Card(n). By uniqueness, we have card($\mathscr{R}_{\mathbb{N}}(i)$) = i for every nat i. If A is a finite set, then Card A = Card B implies card A = card B. The converse is true if both sets are finite.

```
Definition cardinal_nat x := choosenat(fun i => cardinal (natR i) = x).
Lemma cardinal_nat_cardinal: forall x,
  cardinal_nat (cardinal x) = cardinal_nat x.
Lemma cardinal_nat_pr: forall x, is_finite_set x ->
  cardinal (natR (cardinal_nat x)) = cardinal x.
Lemma cardinal_nat_pr1: forall i, cardinal_nat(natR i) = i.
Lemma cardinal_nat_finite_eq: forall a b, is_finite_set a ->
  cardinal a = cardinal b -> cardinal_nat a = cardinal_nat b.
Lemma cardinal_nat_finite_eq1: forall a b,
  is_finite_set a -> is_finite_set b ->
  cardinal_nat a = cardinal_nat b -> cardinal a = cardinal b.
```

We have card **0** = 0, card **1** = 1 and card **2** = 2. Note that, if s is the successor function, then **3** = s(**2**) and 3 = S2, this implies card **3** = 3.

```
Lemma cardinal_nat_emptyset: cardinal_nat emptyset = 0.
Lemma cardinal_nat_singleton: forall x, cardinal_nat (singleton x) = 1.
Lemma cardinal_nat_doubleton: forall x y,
  x <> y -> cardinal_nat (doubleton x y) = 2.
Lemma cardinal_nat_zero: cardinal_nat card_zero = 0.
Lemma cardinal_nat_one: cardinal_nat card_one = 1.
Lemma cardinal_nat_two: cardinal_nat card_two = 2.
```

11.2.4 Bijection between nat and the integers

Denote by $s(n)$ the successor of n. We have $s(0) = 1$, and $a + s(n) = s(a + n)$ (associativity of the sum). We have $a.s(b) = ab + a$ and $a.0 = 0$. By induction we deduced that the sum and product of two integers are integers.

```
Lemma plus_via_succ: forall a n,
  card_plus a (succ n) = succ (card_plus a n).
Lemma Bnat_stable_plus: forall a b, inc a Bnat -> inc b Bnat ->
  inc (card_plus a b) Bnat.
Lemma mult_via_plus: forall a b, is_cardinal a ->
  card_mult a (succ b) = card_plus (card_mult a b) a.
Lemma Bnat_stable_mult: forall a b, inc a Bnat -> inc b Bnat ->
  inc (card_mult a b) Bnat.
```

We define now by induction a function $\mathscr{N}$ that associates a cardinal to each nat; by construction $\mathscr{N}(0) = \mathbf{0}$ and $\mathscr{N}(\mathrm{S}n) = s(\mathscr{N}(n))$. This function converts addition and multiplication on nat to cardinal sum and cardinal product (induction on $\mathbb{N}$). This function is injective (because succ is injective). By induction on **N**, this function is surjective. More precisely, every finite cardinal has the form $\mathscr{N}(n)$. Assume $a \leq b$; then $\mathscr{N}(a) \leq_{\mathrm{Card}} \mathscr{N}(b)$; conversely, if this relation holds then $\mathscr{N}(a) + x = \mathscr{N}(b)$ for some x. By surjectivity $x = \mathscr{N}(c)$, by injectivity $a + c = b$ which implies $a \leq b$. By injectivity of $\mathscr{N}$ we deduce that $a < b$ and $\mathscr{N}(a) <_{\mathrm{Card}} \mathscr{N}(b)$ are equivalent.

Let g be the inverse of $\mathcal{N}$; such a function exists using the axiom of choice. By uniqueness this function is card, hence $\mathrm{card}(\mathcal{N}(x)) = x$ and $\mathcal{N}(\mathrm{card}(x)) = x$.

```
Fixpoint nat_to_B (n:nat) :=
  match n with 0 => card_zero | S m => succ (nat_to_B m) end.

Lemma nat_B_0: nat_to_B 0 = card_zero.
Lemma nat_B_1: nat_to_B 1 = card_one.
Lemma nat_B_2:  nat_to_B 2 = card_two.
Lemma nat_B_S: forall n, nat_to_B (S n) = succ (nat_to_B n).
Lemma inc_nat_to_B: forall n, inc (nat_to_B n) Bnat.
Lemma nat_B_plus: forall a b,
  nat_to_B (a+b) = card_plus (nat_to_B a) (nat_to_B b).
Lemma nat_B_mult: forall a b,
  nat_to_B (a*b) = card_mult (nat_to_B a) (nat_to_B b).

Lemma nat_B_inj: forall a b,
  nat_to_B a = nat_to_B b -> a = b.
Lemma nat_to_B_surjective: forall n, inc n Bnat -> exists m,
  nat_to_B m = n.
Lemma nat_B_le: forall a b,
  (a<= b) =  cardinal_le (nat_to_B a) (nat_to_B b).
Lemma nat_B_lt: forall a b,
  (a< b) =  cardinal_lt (nat_to_B a) (nat_to_B b).
Lemma nat_B_lt: forall a b,
  (a< b) =  cardinal_lt (nat_to_B a) (nat_to_B b).
Lemma nat_B_lt0: forall b,
  (0<b) = cardinal_lt card_zero (nat_to_B b).
Lemma nat_to_B_pr: forall n, inc n Bnat ->
  nat_to_B (cardinal_nat n) = n.
Lemma nat_to_B_pr1: forall n,
  cardinal_nat(nat_to_B n) = n.
```

We finish with the definition of the power function on $\mathbb{N}$.

```
Fixpoint pow (n m:nat) {struct m} : nat :=
  match m with
  | O => 1
  | S p => (pow n p) * n
  end
where "n ^ m" := (pow n m) : nat_scope.
```

11.2.5 Ordinals

```
Lemma transfinite_aux2: forall r p s, worder r -> (* 70 *)
  (forall z, inc z s -> is_segment r z) ->
  (forall z, inc z s -> (exists f : correspondenceC,
                transfinite_def (induced_order r z) p f)) ->
  exists f : correspondenceC, transfinite_def (induced_order r (union s)) p f.

Lemma order_morphism_pr1: forall f r r',
  order r -> order r' -> is_function f -> substrate r = source f ->
  substrate r' = target f ->x
  (forall x y, inc x (source f) -> inc y (source f) ->
    gle r x y = gle r' (W x f) (W y f))
  -> order_morphism f r r'.
```

etc

```
Definition ord_Bnat n := order_type(interval_Bnat_co n)
Lemma Bordinal_ord_Bnat: forall n, inc n Bnat -> Bordinal (ord_Bnat n).
Lemma cardinal_ord_Bnat: forall n, inc n Bnat ->
  cardinal (substrate (ord_Bnat n)) = n.
Lemma ord_Bnat_injective: forall n m, inc n Bnat -> inc m Bnat ->
  (ord_Bnat n = ord_Bnat m) -> n = m.
Lemma finite_ordinal: forall x, Bordinal x -> is_finite_set (substrate x)
  ->  x = ord_Bnat (cardinal (substrate x)).
Lemma ord_zero_card: ord_Bnat card_zero = ord_zero.
Lemma ord_one_card: ord_Bnat card_one = ord_one.
Lemma ord_two_card_pr1: forall x,
  Bordinal x -> card_two = cardinal (substrate x) ->
  x = ord_Bnat card_two.
Lemma ord_two_card: ord_Bnat card_two = ord_two.
```

The order-type of the order-sum of a family of order-types will be called the *ordinal sum* and denoted by $\sum_{i\in I}\lambda_i$ (abuse of notations here). The order-type of the order-product of a family will be called the *ordinal product.* If arguments are order-types so is the result of the operation.

```
Definition ord_sum r g := ordinal (order_sum r g).
Definition ord_prod r g := ordinal (order_product r g).
Definition ord_sum2 a b := ordinal (order_sum2 a b).
Definition ord_prod2 a b := ordinal (order_prod2 a b).

Lemma ord_sum_type: forall r g,
  order r -> substrate r = domain g -> fgraph g ->
  (forall i, inc i (domain g) -> order (V i g))
  -> is_order_type (ord_sum r g).
Lemma ord_prod_type: forall r g,
  worder r -> substrate r = domain g -> fgraph g ->
  (forall i, inc i (domain g) -> order (V i g))
  -> is_order_type (ord_prod r g).
```

In Exercice 14(c) Bourbaki say that if α is an ordinal, then the relation "ξ is an ordinal and $\xi \leq_{\mathrm{Ord}} \alpha$" is collectivizing in ξ. This means that there is a set O'_α whose elements are all sets ξ such that $\xi \leq_{\mathrm{Ord}} \alpha$ (note that this implies that ξ is an ordinal). There is also a set O_α whose elements are all sets ξ such that $\xi <_{\mathrm{Ord}} \alpha$. This set is well-ordered by $\leq_{\mathrm{Ord}}$ and $\mathrm{Ord}(O_\alpha) = \alpha$. Notice that the first set is the set of all order-types of segments S of α, and the second set is obtained by considering the strict segments, which are of the form $]\leftarrow, x[$. The mapping $x \mapsto \mathrm{Ord}(]\leftarrow, x[)$ is the desired isomorphism. The important property here is that two distinct segments of a well-ordered sets are never isomorphic.

```
Definition set_of_ordinal_le a:=
   fun_image (set_of_segments a)  (fun z => order_type (induced_order a z)).
Definition set_of_ordinal_lt a:=
  fun_image (substrate a)(fun z => order_type (induced_order a (segment a z))).

Lemma segments_iso2: forall a A B, worder a ->
   inc A  (set_of_segments a) -> inc B  (set_of_segments a)  ->
   order_isomorphic (induced_order a A)(induced_order a B) -> A = B.
```

```
Lemma set_ord_le_prop: forall a, is_ordinal a ->
  (forall x, inc x (set_of_ordinal_le a) = ordinal_le x a).
Lemma set_ord_lt_prop: forall a, is_ordinal a ->
  (forall x, inc x (set_of_ordinal_lt a) = ordinal_lt x a). (* 26 *)
Lemma set_ord_lt_prop2: forall a, is_ordinal a ->
  order_isomorphism (BL (fun z => order_type (induced_order a (segment a z)))
       (substrate a) (set_of_ordinal_lt a))
    a (graph_on ordinal_le (set_of_ordinal_lt a)).  (* 62 *)
Lemma set_ord_lt_prop3: forall a, is_ordinal a ->
  order_type (graph_on ordinal_le (set_of_ordinal_lt a)) = a.
```

For every family of ordinals $(\xi_i)_{i\in I}$, there exists a unique ordinal α such that "λ is an ordinal and $\xi_i \leq \lambda$ for all $i \in I$" is equivalent to $\alpha \leq \lambda$. It is called the least upper bound or supremum (by abuse of language, since $\leq$ is an ordering without graph). Let $Q(\lambda)$ be the property that $\xi_i \leq \lambda$ for all $i \in I$, and let $P(\lambda)$ be the property "λ is an ordinal and $Q(\lambda)$". Whatever Q, there is at most one ordinal α such that $\alpha \leq \lambda$ is equivalent to $P(\lambda)$ (by antisymmetry of $\leq$). Assume $P(\lambda)$ true for some λ. Then the supremum is the least such λ. In what follows, we consider a set of ordinals rather that a family; then supremum of the family is the supremum of the range. It exists, since the ordinal sum of the family is an upper bound.

```
Definition ord_sup_pr E x:=
   is_ordinal x &
   forall y, ordinal_le x y =
    (is_ordinal y & forall i, inc i E -> ordinal_le i y).
Definition ord_sup E := choose (fun x => ord_sup_pr E x).
Definition ord_supf f := ord_sup (range f).
Lemma ord_sup_pr1: forall E y, Bordinal y ->
  (forall i, inc i E -> ordinal_le i y) ->
  exists x, (ord_sup_pr E x & ordinal_le x y).
Lemma ord_sup_pr2: forall E, (forall i, inc i E -> Bordinal i) ->
  ord_sup_pr E (ord_sup E).
```

Consider a family $(\lambda_i)_{i\in I}$ of ordinals. We can well-order the index set and obtain the ordinal sum λ. The next lemma shows that each i, $\lambda_i \leq \lambda$. Thus, there exists a strictly increasing mapping $\lambda_i \to x_i$ such that λ_i is isomorphic to a segment $]\leftarrow, x_i[$ of λ. There is a least x_i, hence a least λ_i. Thus $\leq_{\mathrm{Ord}}$ is a well-ordering.

Consider a family $(\mathfrak{a}_i)_{i\in I}$ of cardinals. We can well-order the union E. Each $\mathfrak{a}_i$ is equipotent to at least one segment of E. Let x_i be the least endpoint of such a segment. Thus, there exists a strictly increasing mapping $\mathfrak{a}_i \to x_i$ such that $\mathfrak{a}_i$ is equipotent to a segment $]\leftarrow, x_i[$. There is a least x_i, hence a least $\mathfrak{a}_i$. Thus $\leq_{\mathrm{Card}}$ is a well-ordering.

In the current version, we simplified these two theorems as follows. First, we note that, if $\mathfrak{a}_i$ is any cardinal, and λ_i is the ordinal of $]\leftarrow, x_i[$, then λ_i depends only on $\mathfrak{a}_i$, but not on the other elements of the family or the ordering of E. Thus we can assume $\mathfrak{a}_i = \lambda_i$. Then $\leq_{\mathrm{Card}}$ is a well-ordering as being the restriction of $\leq_{\mathrm{Ord}}$ to cardinals. We can simplify the proof further by assuming $\lambda_i =]\leftarrow, x_i[$. Then $\leq_{\mathrm{Ord}}$ is a well-ordering as being the same ordering as that of λ (when suitably restricted). Note that, in the case of von Neumann ordinals, an ordinal is a set with a natural ordering, while Bourbaki considers ordered sets).

```
Lemma ordinal_le_sum: forall r g j,
  worder r -> substrate r = domain g -> fgraph g ->
  (forall i, inc i (domain g) -> worder (V i g)) -> inc j (domain g)
  -> ordinal_le (ordinal (V j g)) (ordinal (order_sum r g)). (* 27 *)
```

11.3 Introduction to Chapter 6

Our current work is based on the ssreflect library, which redefined a certain number of functions on natural numbers. For instance the type of $a \leq b$ is now bool instead of Prop. Below is a theorem zerop that says that a number is zero or positive. This is restated in ssrat by the lemma posnP that says that the boolean value of $n ==$ or $0 < n$ are mutually exclusive (one is true, the other one is false).

We start this chapter with a list of some definitions and theorems, extracted from the Coq standard library implemening $\mathbb{N}$. Consider for instance zerop. We show its type, not its value which is irrelevant for the use we shall make of it; this value is a proof that for every n (of type nat), one of A or B is true. This is summarized by the notation {A} + {B} (certified disjoint union). The heavyside function is not part of the library, it is an example of how this construction can be used. The underscores in the definition represent the two proofs. The Coq parser and pretty-printer interpret this in the same fashion as if zerop n then 0 else 1. A property is decidable if it can be shown true of false. For us, all properties are decidable since we have an axiom that says so. It is however useful to know that equality and inequality are decidable. We state also some theorems such as if $n \leq m$ is false then $n > m$ is true. In this case, the result is a consequence of the fact that one of the properties is true.

```
(*
  Definition zerop n : {n = 0} + {0 < n}.
  Definition lt_eq_lt_dec n m : {n < m} + {n = m} + {m < n}.
  Definition gt_eq_gt_dec n m : {m > n} + {n = m} + {n > m}.
  Definition le_lt_dec n m : {n <= m} + {m < n}.
  Definition le_le_S_dec n m : {n <= m} + {S m <= n}.
  Definition le_ge_dec n m : {n <= m} + {n >= m}.
  Definition le_gt_dec n m : {n <= m} + {n > m}.
  Definition le_lt_eq_dec n m : n <= m -> {n < m} + {n = m}.

Definition heavyside n := match (zerop n) with left _ => 0 | right _ => 1 end.

  Theorem dec_le : forall n m, decidable (n <= m).
  Theorem dec_lt : forall n m, decidable (n < m).
  Theorem dec_gt : forall n m, decidable (n > m).
  Theorem dec_ge : forall n m, decidable (n >= m).
  Theorem not_eq : forall n m, n <> m -> n < m \/ m < n.
  Theorem not_le : forall n m, ~ n <= m -> n > m.
  Theorem not_gt : forall n m, ~ n > m -> n <= m.
  Theorem not_ge : forall n m, ~ n >= m -> n < m.
  Theorem not_lt : forall n m, ~ n < m -> n >= m.
*)
```

Here we show how and addition and multiplicaton behave with respect to ordering.

```
(*
  Lemma plus_reg_l : forall n m p, p + n = p + m -> n = m.
  Lemma plus_le_reg_l : forall n m p, p + n <= p + m -> n <= m.
  Lemma plus_lt_reg_l : forall n m p, p + n < p + m -> n < m.

  Lemma plus_le_compat_l : forall n m p, n <= m -> p + n <= p + m.
  Lemma plus_le_compat_r : forall n m p, n <= m -> n + p <= m + p.
  Lemma le_plus_l : forall n m, n <= n + m.
  Lemma le_plus_r : forall n m, m <= n + m.
```

```
  Theorem le_plus_trans : forall n m p, n <= m -> n <= m + p.
  Theorem lt_plus_trans : forall n m p, n < m -> n < m + p.
  Lemma plus_lt_compat_l : forall n m p, n < m -> p + n < p + m.
  Lemma plus_lt_compat_r : forall n m p, n < m -> n + p < m + p.
  Lemma plus_le_compat : forall n m p q, n <= m -> p <= q -> n + p <= m + q.
  Lemma plus_le_lt_compat : forall n m p q, n <= m -> p < q -> n + p < m + q.
  Lemma plus_lt_le_compat : forall n m p q, n < m -> p <= q -> n + p < m + q.
  Lemma plus_lt_compat : forall n m p q, n < m -> p < q -> n + p < m + q.

  Lemma mult_O_le : forall n m, m = 0 \/ n <= m * n.
  Lemma mult_le_compat_l : forall n m p, n <= m -> p * n <= p * m.
  Lemma mult_le_compat_r : forall n m p, n <= m -> n * p <= m * p.
  Lemma mult_le_compat :
    forall n m p (q:nat), n <= m -> p <= q -> n * p <= m * q.
  Lemma mult_S_lt_compat_l : forall n m p, m < p -> S n * m < S n * p.
  Lemma mult_lt_compat_r : forall n m p, n < m -> 0 < p -> n * p < m * p.
  Lemma mult_S_le_reg_l : forall n m p, S n * m <= S n * p -> m <= p.
  Lemma plus_le_reg_l : forall n m p, p + n <= p + m -> n <= m.
  Lemma plus_lt_reg_l : forall n m p, p + n < p + m -> n < m.
  Lemma mult_S_le_reg_l : forall n m p, S n * m <= S n * p -> m <= p.
*)
```

Here we give some properties of subtraction.

```
(*
   Lemma minus_n_n : forall n, 0 = n - n.
   Lemma minus_n_O : forall n, n = n - 0.
   Lemma le_plus_minus : forall n m, n <= m -> m = n + (m - n).
   Lemma plus_minus : forall n m p, n = m + p -> p = n - m.
   Lemma minus_plus : forall n m, n + m - n = m.
   Theorem le_minus : forall n m, n - m <= n.
   Lemma minus_plus_simpl_l_reverse : forall n m p, n - m = p + n - (p + m).
   Lemma minus_Sn_m : forall n m, m <= n -> S (n - m) = S n - m.
   Lemma le_plus_minus : forall n m, n <= m -> m = n + (m - n).
   Lemma le_plus_minus_r : forall n m, n <= m -> n + (m - n) = m.
   Lemma lt_minus : forall n m, m <= n -> 0 < m -> n - m < n.
   Lemma lt_O_minus_lt : forall n m, 0 < n - m -> m < n.
   Theorem not_le_minus_0 : forall n m, ~ m <= n -> n - m = 0.
*)
```

Additional theorems about integers.

```
Theorem lt_to_plus: forall a b:nat, a<b = exists c:nat, 0<c & c+a=b.
Lemma mult_lt_le_compat : forall n m p q,
  0<q -> n < m -> p <= q -> n * p < m * q.
Lemma mult_le_lt_compat : forall n m p q,
  0< m -> n <= m -> p < q -> n * p < m * q.
Lemma zero_lt_oneN: 0 < 1.
Lemma lt_n_succ_leN: forall a b,  a < b -> S a <= b.
Lemma power_x_0N: forall a, a ^ 0 =  1.
Lemma power_0_0N:  0 ^ 0 = 1.
Lemma power_x_1N: forall a,  a ^ 1 = a.

Lemma plus_simplifiable_leftN: forall a b b':nat,
  a + b = a + b' -> b = b'.
Lemma plus_simplifiable_rightN: forall a b b':nat,
```

```
  b + a = b' + a -> b = b'.
Lemma Sn_is_1plus: forall n, S n = 1 + n.
Lemma Sn_is_plus1: forall n, S n = n +1.
Lemma lt_i_n : forall i n, i< n  -> 1 <= n-i.
Lemma double_subN: forall n p, p <= n -> n - (n- p) = p.
Lemma nonzero_suc: forall n, 0<>n -> exists m, n = S m.
Lemma mult_S_lt_reg_l : forall n m p, S n * m < S n * p -> m < p.
Lemma mult_lt_reg_l : forall n m p, 0<>n -> n * m < n * p -> m < p.
Lemma mult_lt_reg_r : forall n m p, 0<>n -> m * n <  p * n -> m < p.
Lemma minus_wrong: forall n m, n<=m -> n-m = 0.
Lemma pred_minus: forall n m, m<n -> n-m =  S(n - S m).

Lemma plus_n_Sm_subSn: forall n m, n + S m - n - 1 = m.
Lemma plus_n_Sm_subSm: forall n m, n + S m - m - 1 = n.

Lemma minus_SnSi: forall i n, i < S n -> S n - i - 1 = n - i.
Lemma double_compl_nat:forall i n, i< n ->
  i = n - (n - i - 1) -1.
Lemma double_compl_ex:forall i n, i< n -> (n - i - 1) < n.
Lemma plus_reg_r : forall n m p, n + p = m + p -> n = m.
```

11.4 The axiom of choice

Let E be any set, $p(x)$ be a property, $F = \{x \in E, p(x)\}$ and $z = \bigcup F$. If there is a unique x in E that satisfies p, then $F = \{x\}$, and $p(z)$ holds. This quantity z is denote by select p E. Assume that p dependens on a parameter y and $p(x)$ implies $x \in f(y)$. Then select (p y) (f y) is the same as choose p y, and some calls to choose have been replaced by this trick.

Here are the old definitions. In the case of supremum_pr1 we need the assumption that r is an order.

```
Definition the_least_element r := choose (least_element r).
Definition the_greatest_element r := choose (greatest_element r).
Definition supremum r X  := choose (least_upper_bound r X).
Definition infimum r X  := choose (greatest_lower_bound r X).


Lemma supremum_pr1 X r:
  has_supremum r X ->
  least_upper_bound r X (supremum r X).
Lemma infimum_pr1  X r:
  has_infimum r X ->
  greatest_lower_bound r X (infimum r X).
```

We introduced gge and ggt corresponding to ≥ and >, then removed them. Some statements had to be changed, and some lemmas became uselesss.

```
Definition gge r x y := gle r y x.
Definition ggt r x y := gle r y x &  x <> y.
Lemma opposite_gge r x y:
  gge (opposite_order r) x y <-> gle r x y.
Lemma glt_inva r x y:
  ggt (opposite_order r) x y <-> glt r x y.
Lemma ggt_inva r x y:
```

```
  glt (opposite_order r) x y <-> ggt r x y.
Lemma ggt_invb r x y:  ggt r x y <-> glt r y x.
```

These definitions have been simplified

```
Definition is_sup_fun r f x := least_upper_bound r (image_of_fun f) x.
Definition is_inf_fun r f x := greatest_lower_bound r (image_of_fun f) x.
Definition is_sup_graph r f x := least_upper_bound r (range f) x.
Definition is_inf_graph r f x := greatest_lower_bound r (range f) x.

Lemma infinite_nonempty x: infinite_c x -> nonempty x.
```

11.5 Theorems removed from Chapter 6

We study here the function pow on the type nat.

```
Lemma power_of_sumN: forall a b c, a ^ (b+c) = (a ^ b) *(a ^ c).
Lemma nat_B_pow: forall n m,
  nat_to_B (n ^ m) = card_pow (nat_to_B n)(nat_to_B m).
Lemma power_of_prodN: forall a b c,
  (a * b)  ^ c =  (a ^ c) *  (b ^ c).
Lemma power_1_xN: forall a,  1 ^ a = 1.
Lemma nat_not_zero_pr: forall a,  a <> 0 -> nat_to_B a <> card_zero.
Lemma power_0_x: forall a, a <> 0 -> 0 ^ a = 0.
Lemma non_zero_apowbN: forall a b, 0 < a -> 0 < a ^ b.
Lemma finite_power_lt1N: forall a a' b, a < a' -> 0 <  b ->  a ^ b < a' ^ b.
Lemma finite_power_lt2N: forall a b b',
  b < b' -> 1 < a ->  a^ b < a ^ b'.
Lemma mult_simplifiable_leftN: forall a b b':nat,
  0 <>a -> a * b = a * b' -> b = b'.
Lemma mult_simplifiable_rightN: forall a b b',
  0 <>a -> b * a = b' * a -> b = b'.
```

If a and b are integers and $a \leq b$ there is a unique integer c such that $b = a + c$, it is called the *difference* and denoted by $b - a$. The operation is called *subtraction.* There is a Coq function, denoted by sub, defined for all integers, whose value is zero for $a > b$; we extend our function so that they share the same behavior.

```
Definition card_sub a b :=
   Yo (cardinal_le b a)
     (choose (fun c => inc c Bnat & card_plus b c = a)) card_zero.
Lemma card_sub_pr: forall a b, inc a Bnat-> inc b Bnat ->
  cardinal_le b a -> card_plus b (card_sub a b) = a.
Lemma card_sub_rpr: forall a b, inc a Bnat-> inc b Bnat ->
  cardinal_le b a -> card_plus (card_sub a b) b = a.

Lemma minus_n_nC: forall a, inc a Bnat -> card_sub a a = card_zero.
Lemma prec_pr1: forall a, inc a Bnat -> a <> card_zero
  -> predc a = card_sub a card_one.
Lemma sub_le_symmetry: forall  a b, inc a Bnat -> inc b Bnat ->
  cardinal_le b a -> cardinal_le (card_sub a b) a).
```

These two lemmas we initially use to show that every infinite set has a infinitae countable subset.

```
Lemma morphism_range: forall f a b,
  order_morphism f a b -> equipotent (substrate a) (range (graph f)).
Lemma morphism_range1: forall f a b,
  order_morphism f a b -> cardinal (substrate a) = cardinal (range (graph f)).
```

Other removed theorems from chapter 6

```
Lemma nat_B_sub: forall a b,
  nat_to_B (a-b) = card_sub (nat_to_B a) (nat_to_B b).
Lemma card_sub_pr4N: forall a b a' b',
  a<=b -> a' <= b' ->  (b-a) + (b'-a') = (b+b')- (a+a').
Lemma card_sub_associativeN: forall a b c,
  (b +c) <= a -> (a-b) -c = a - (b+c).
Lemma nat_B_pred: forall a, 0 <> a -> nat_to_B (pred a) = predc (nat_to_B a).
Lemma prec_is_cardinal_prec: forall a, inc a Bnat ->
  a <> card_zero -> cardinal_nat (prec a) = pred (cardinal_nat a).
Lemma card_sub_associative1N: forall a b,
  (S b) <= a -> pred (a - b) = a  - S b.
```

11.5.1 Division

Given two integers a and $b \neq 0$, there is a unique pair of integers (q, r) satisfying $a = bq + r$ and $r < q$. Thus q and r are defined via the choose function like this.

```
Definition card_rem0 a b:=
  choose (fun r => inc r Bnat  & exists q, inc q Bnat & division_prop a b q r).
Definition card_quo0 a b:=
  choose (fun q => inc q Bnat & exists r, inc r Bnat & division_prop a b q r).
```

In order to simplify proofs, we define the quotient and remainder in the case $b = a$ as $q = 0$ and $r = a$. This means that $a = bq + r$ holds in any case.

```
Definition card_rem a b := variant \0c a (card_rem0 a b) b.
Definition card_quo a b := variant \0c \0c (card_quo0 a b) b.
```

Euclidean division is curiously defined in Coq. The following two lemmas say that if $b > 0$, then for every a we have $\{x : \mathbb{N} | \exists y : \mathbb{N},\ \mathrm{Z}\}$ where Z is the division property $a = bq + r$ and $r < b$, and (x, y) is (q, r) or (r, q). This expression is a type; from it one can extract q and the associated property (namely that there is r such that Z).

```
(*
  Lemma quotient :
    forall n,
      n > 0 ->
      forall m:nat, {q : nat | exists r : nat, m = q * n + r /\ n > r}.
  Lemma modulo :
    forall n,
      n > 0 ->
      forall m:nat, {r : nat | exists q : nat, m = q * n + r /\ n > r}.
*)
```

The ssreflect library has a clever definition of quotient and remainder (the first definition defines quotient and remainder, the second defines only the quotient). The two functions divn and modn are deduced from these recursive function. We give one theorem that showq how these quantities can be used.

```
Definition edivn_rec d := fix loop (m q : nat) {struct m} :=
  if m - d is m'.+1 then loop m' q.+1 else (q, m).
Definition modn_rec d := fix loop (m : nat) :=
  if m - d is m'.+1 then loop m' else m.

Lemma divn_eq : forall m d, m = m %/ d * d + m %% d.
```

In the previous version of this document, we explained another implementation of these operations. It is not used anymore. These lemmas show existence and uniqueness of division.

```
Lemma least_int_prop0: forall p:nat->Prop,
  ~(p 0) -> (exists x, p x) -> (exists x, p (S x) & ~ p x).

Lemma division_prop_nat: forall a b q r, 0 <>b ->
  (a=b*q+r & r<b) = (b*q <= a & a < b* (S q) & r = a - (b*q)).
Lemma Ndivision_unique: forall a b q q' r r', 0 <> b ->
  a = b* q + r -> r < b -> a = b* q' + r' -> r'< b ->
  (q = q' & r = r').
Lemma Ndivision_existence: forall a b, 0 <> b ->
  exists q, exists r, (a = b* q + r & r < b).
Lemma division_result_integer: forall a b q r, inc a Bnat-> inc b Bnat ->
  b <> card_zero -> division_prop a b q r -> is_cardinal q ->
  (inc q Bnat & inc r Bnat).
```

In the definition that follows $b = 0$ is replaced by $b = 1$ so that the quotient and remainder is well-defined for all arguments. Later on, we modified the definition of quotient and remainder (see above). These two variants give different results. However, we say that b divides a only when b is non-zero.

```
Definition Nquo a b :=
  cardinal_nat (card_quo (Ro (nat_to_B a))
    (Yo (b = 0) card_one (Ro (nat_to_B b)))).
Definition Nrem a b :=
  cardinal_nat (card_rem (Ro (nat_to_B a))
    (Yo (b = 0) card_one (Ro (nat_to_B b)))).
Definition Ndivides b a:= 0 <> b & Nrem a b = 0.

Lemma Ndivision_exists: forall a b,  0 <> b ->
  (a = b* (Nquo a b) + (Nrem a b) & (Nrem a b < b)).
Lemma Ndivision_pr: forall a b q r, 0 <> b ->
  a = b* q + r -> r < b -> (q = Nquo a b & r = Nrem a b).

Lemma Ndivision_pr_q: forall a b q r, 0 <> b ->
  a = b* q + r -> r < b -> q = Nquo a b.
Lemma Ndivision_pr_r: forall a b q r, 0 <> b ->
  a = b* q + r -> r < b -> r = Nrem a b.
```

All properties true for Nquo and Nrem are true for card_quo and card_rem. For this reason, we shall only prove our theorems for the case of type nat.

```
Lemma nat_B_division: forall a b, 0 <> b ->
  (nat_to_B (Nquo a b) =  card_quo (nat_to_B a) (nat_to_B b) &
    nat_to_B (Nrem a b) =  card_rem (nat_to_B a) (nat_to_B b)).
Lemma nat_B_quo: forall a b, 0 <> b ->
```

```
  nat_to_B (Nquo a b) =  card_quo (nat_to_B a) (nat_to_B b).
Lemma nat_B_rem: forall a b, 0 <> b ->
  nat_to_B (Nrem a b) =  card_rem (nat_to_B a) (nat_to_B b).
```

Now some consequences when division is exact. Bourbaki says: every multiple a' of a multiple a of b is a multiple of b. One can restate this as: if b divides a, then b divides ac.

```
Lemma inc_quotient_bnat:forall a b, inc a Bnat-> inc b Bnat -> b <> card_zero ->
  inc (card_quo a b) Bnat.
Lemma inc_remainder_bnat:forall a b,
    inc a Bnat-> inc b Bnat -> b <> card_zero ->
    inc (card_rem a b) Bnat.

Lemma Ndivides_pr: forall a b,
  Ndivides b a -> a =  b * (Nquo a b).
Lemma Ndivides_pr1: forall a b,  0 <> b ->  Ndivides b (b *a).
Lemma Ndivides_pr2: forall a b q,  0 <> b ->
  a = b * q -> q = Nquo a b.
Lemma one_divides_all: forall a, Ndivides 1 a.
Lemma Ndivides_pr3: forall a b q,
  Ndivides b a -> q = Nquo a b -> a =  b * q.
Lemma Ndivides_pr4: forall b q, 0 <> b ->
  Nquo (b * q) b = q.
Lemma Ndivision_itself: forall a, 0 <> a ->
  (Ndivides a a & Nquo a a = 1).
Lemma Ndivides_itself: forall a, 0 <> a -> Ndivides a a.
Lemma Nquo: forall a, 0 <> a -> Nquo a a = 1.
Lemma Ndivision_of_zero: forall a, 0 <> a ->
  (Ndivides a 0 & Nquo 0 a = 0).
Lemma Ndivides_trans: forall a b a', Ndivides a a'-> Ndivides b  a
  -> Ndivides b a'.
Lemma Ndivides_trans1: forall a b a', Ndivides a a'-> Ndivides b a
  -> Nquo a' b =  (Nquo a' a) *(Nquo a b).
Lemma Ndivides_trans2: forall a b c,
  Ndivides b a-> Ndivides b (a *c).
Lemma non_zero_mult: forall a b, 0 <> a -> 0 <> b -> 0 <> (a*b).
Lemma Nquo_simplify:  forall a b c, 0 <> b -> 0 <> c ->
  Nquo (a * c) (b * c) = Nquo a b.
```

If b divides a and a', it divides the sum and the difference.

```
Lemma divides_and_sum:  forall a a' b,  Ndivides b a ->  Ndivides b a'
  -> (Ndivides b  (a + a') &
    Nquo  (a + a')  b =  (Nquo a b) + (Nquo a' b)).
Lemma distrib_prod2_subN: forall a b c, c<= b->
  a * (b-c) = (a*b) - (a*c).
Lemma divides_and_difference:  forall a a' b, a' <= a ->
  Ndivides b a ->  Ndivides b a'
  -> (Ndivides b  (a -a') &
    (Nquo a' b) <= (Nquo a b) &
    Nquo  (a - a')  b =  (Nquo a b) -  (Nquo a' b)).
```

The following lemma may have some intererest, but is currently unused. Assume $a = bq + r$ with $r < b$, where a and b are integers, b is non-zero. The last relation says that r is an integer. The quantity bq is also an integer, so that q is finite. If q is a cardinal, we deduce that q is an integer.

```
Lemma division_result_integer: forall a b q r,
  inc a Bnat-> inc b Bnat ->
  b <> card_zero -> division_prop a b q r -> is_cardinal q ->
  (inc q Bnat & inc r Bnat).

Lemma lt_a_power_b_aN: forall a b,  1< b -> a < pow b a.
```

11.6 Finite sequences and lists

If P⦃i⦄ is equivalent to $i \in \mathrm{I}$, where I is a finite set of integers, then $(x_i)_{i\in \mathrm{I}}$ may be written as $(x_i)_{\mathrm{P}\{i\}}$. In fact, such a notation can be used whatever I. As an example one can see $(t_i)_{a\le i\le b}$. The sum of such a family may be denoted by $\sum_{i=a}^{b} t_i$.

Lists are defined in Coq by

```
Inductive list (A : Type) : Type :=
    nil : list A
 | cons : A -> list A -> list A
```

A list can be either empty (this is nil), or of the form cons A a b where a is of type A and b is a list of type A. The parameter A is often implicit. The expression cons A a b is denoted by $a::b$. There are many functions in the standard library that deal with lists. For instance, seq can produce the list containing 1, 2, 3. Given a list containing x_1, x_2 and x_3 (of type A) it is possible to create the list containing $(1,x_1)$, $(2,x_2)$, and $(3,x_3)$ (of type $\mathbb{N}\times$A) then the set of all these values. This is a finite sequence (i.e., a functional graph, with domain $\{1,2,3\}$). In this section, we explain how to convert operations defined by Bourbaki for finite sequences (like sum and product) into operations on Coq lists.

11.6.1 Lists as functions

Given a function g, we define here the list L containing $g(0)$, $g(1)$, $g(2)$ up to $g(n-1)$. The list has length n; it is stored in natural order[1]: On the diagram below, the mapping $g \mapsto$ L is denoted by fl. Conversely given a L of length n, we define a function g that returns the k-th element of the list, and 0 if $k \ge n$. It will be denoted by lf on the diagram below.

We consider a variant of lf where L is a list of sets (the default value is then $\emptyset$) and, later on, a variant of fl, where g is a function in the Bourbaki sense

```
Fixpoint fct_to_list_rev (A:Type) (f: nat->A)(n:nat): list A :=
  match n with O => nil
    | S m => (f m) ::(fct_to_list_rev f m) end.

Definition fct_to_list A f n := rev (fct_to_list_rev (A:=A) f n).

Definition list_to_fct (a: list nat) :=
  fun n => nth n a 0.

Definition list_to_fctB (a: list Set) :=
  fun n => nth n a emptyset.
```

[1] In the previous version, we used the other order: $g(n-1)$ was the head of the list

```
Lemma card_interval_c0_pr: forall n,
  cardinal_nat (interval_co_0a (nat_to_B n)) = n.
Lemma list_extens: forall (A:Type) (l1 l2 : list A) (u:A),
   length l1 = length l2 ->
   (forall i, i< length l1 -> nth i l1 u = nth i l2 u) -> l1 = l2.

Lemma fct_to_list_length :   forall A (f:nat->A) n,
  length (fct_to_list f n) = n.


Lemma list_to_fct_pr0: forall a l1 l2,
  list_to_fct (l2 ++ a :: l1) (length l2) = a.
Lemma list_to_fct_pr0B: forall a l1 l2,
  list_to_fctB (l2 ++ a :: l1)  (length l2) = a.
Lemma list_to_fct_pr: forall (A:Type) (f:nat->A) (u:A) n i,
  i<n -> nth i (fct_to_list f n) u = f i.
Lemma list_to_fct_pr1: forall f n i,
  i<n -> list_to_fct (fct_to_list f n) i = f i.
Lemma list_to_fct_pr1B: forall f n i,
  i<n -> list_to_fctB (fct_to_list f n) i = f i.
Lemma list_to_fct_pr3: forall l2 l1,
  fct_to_list (list_to_fct (l2++l1)) (length l2) = l2.
Lemma list_to_fct_pr4: forall l,
  fct_to_list (list_to_fct l) (length l) = l.
Lemma list_to_fct_pr3B: forall l2 l1,
  fct_to_list (list_to_fctB (l2++l1)) (length l2) = l2.
Lemma list_to_fct_pr4B: forall l,
  fct_to_list (list_to_fctB l) (length l) = l.
```

Note that if g and g' agree on $[0, n-1]$ then $\mathsf{fl}(g) = \mathsf{fl}(g')$. On the other hand, if L is a list of size n and $\mathrm{L}' = a$::L, if the associated functions are g and g', then $g'(n) = a$, and g and g' agree on $[0, n-1]$.

```
Lemma fct_to_list_unique: forall (A:Type) (f g: nat-> A) n,
  (forall i, i< n -> f i = g i) -> fct_to_list f n = fct_to_list g n.

Lemma app_nth3 : forall A (a:A),
  forall l' d n, n >= 1 -> nth n (a::l') d = nth (n-1) l' d.
```

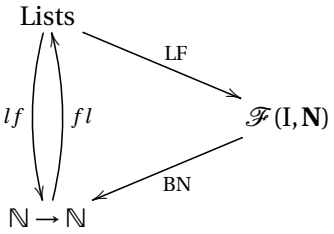
Given a list L of elements of $\mathbb{N}$, of length n, if $g = \mathsf{lf}(\mathrm{L})$, we construct a function $\mathrm{G} : [0, n[\to \mathbf{N}$ via $g(\mathrm{card}(i)) = \mathrm{card}(\mathrm{G}(i))$. The mapping $\mathrm{L} \mapsto \mathrm{G}$ will be denoted by LF on the diagram below. Similarly, given a list L of sets, we construct $\mathrm{G} : [0, n[\to \mathrm{E}$ via $g(\mathrm{card}(i)) = \mathrm{G}(i)$. This is well-defined if all elements of the list belong to the set E, see later.

```
Definition list_to_f (l: list nat):=
  BL (fun n => nat_to_B (list_to_fct l (cardinal_nat n)))
  (interval_co_0a (nat_to_B (length l))) Bnat.
Definition list_to_fB (l: list Set) E:=
  BL (fun n => list_to_fctB l (cardinal_nat n))
  (interval_co_0a  (nat_to_B (length l))) E.

Lemma list_to_f_axioms: forall (l: list nat),
  transf_axioms (fun n =>(Ro (nat_to_B (list_to_fct l (cardinal_nat n)))))
  (interval_co_0a (nat_to_B (length l))) Bnat.
Lemma list_to_f_function: forall (l: list nat),
```

```
  is_function (list_to_f l).
Lemma list_to_f_source: forall (l: list nat),
  source (list_to_f l) = (interval_co_0a (nat_to_B (length l))).
Lemma list_to_f_target: forall (l: list nat),
  target (list_to_f l) = Bnat.
Lemma list_to_f_W: forall (l: list nat) n,
  inc n (interval_co_0a (nat_to_B (length l))) ->
  W n (list_to_f l) = nat_to_B (list_to_fct l (cardinal_nat n)).
Lemma list_to_f_W1: forall (l: list nat) n,
  n < length l ->
  W (nat_to_B n) (list_to_f l) = nat_to_B (list_to_fct l n).
Lemma list_to_f_W2: forall (l: list nat) n,
  n < length l ->
  cardinal_nat(W (nat_to_B n) (list_to_f l)) = list_to_fct l n.
```

(Finite Lists)



Given a function $[0,n[\to \mathbf{N}$, we can construct a function $\mathbb{N}\to\mathbb{N}$, by extending the function with zero, and using the natural isomorphism between $\mathbb{N}$ and **N**. It will be denoted by BN on the diagram. The composition fl $\circ$ BN is the inverse of LF.

```
Definition back_to_nat f n:=
  cardinal_nat (Yo (inc (nat_to_B n) (source f))
    (W (nat_to_B n) f) card_zero).

Lemma back_to_nat_pr: forall f n, inc  (nat_to_B n) (source f) ->
  back_to_nat f n =  cardinal_nat (W  (nat_to_B n) f).
Lemma back_to_nat_pr1: forall f n k,
  source f = (interval_co_0a (nat_to_B k)) ->
  n < k -> back_to_nat f n =  cardinal_nat (W (nat_to_B n) f).
Lemma back_to_nat_pr2: forall (l: list nat) n,
  n < (length l) -> back_to_nat (list_to_f l) n = list_to_fct l n.

Lemma list_to_f_pr1: forall f n, is_function f -> target f = Bnat ->
  source f = (interval_co_0a (nat_to_B n)) ->
  f = list_to_f (fct_to_list (back_to_nat f) n).
Lemma list_to_f_pr2: forall l,
  fct_to_list (back_to_nat (list_to_f l)) (length l) = l.
```

Given a list L and a predicate P, we define P(L) to be true if every element of the list satisfies P. Given a predicate with two arguments, we say that the list satisfies the predicate whenever $P(a,b)$ is true if a comes before b. This means that, if f is the function associated to the list, then $i<j$ implies $P(f(i),f(j))$.

```
(*
  Fixpoint single_list_prop (A:Type) (L: list A) (q: A->Prop) :=
    match L with nil => True | a :: b => q a /\ single_list_prop b q end.
```

```
  Fixpoint double_list_prop (A:Type) (L: list A) (q: A->A->Prop) :=
    match L with nil => True
        | a :: b => single_list_prop b (q a) /\ double_list_prop b q
  end.
*)
```

These definitions changed in Version 2. The two-arguments case was unused, and removed; the single-argument case has been replaced by an inductive object.

```
Inductive list_prop (A:Type)  (q: A->Prop) : list A -> Prop :=
  | list_prop_nil: list_prop  q nil
  | list_prop_cons: forall (a:A)(l:list A),
    q a -> list_prop q l -> list_prop q (a::l).

Lemma list_prop1: forall A (q: A->Prop),  list_prop q nil.
Lemma list_prop2: forall A a b (q: A->Prop),
  q a -> (list_prop q b) = (list_prop q (a::b)).
Lemma list_prop3: forall A a b (q: A->Prop),
  ~ (q a) -> ~(list_prop q (a::b)).
Lemma list_prop_app: forall A a b (q: A->Prop),
  (list_prop q a) -> (list_prop q b)
  -> (list_prop q (a++b)).
Lemma list_prop_refine: forall A L (p q: A->Prop),
  (forall a, p a -> q a) ->  list_prop p L ->  list_prop q L.
Lemma list_prop_nth: forall  A (q: A->Prop) L u n,
  list_prop q L -> n < length L ->
  q (nth n L u).
```

The contraction $C_{fv}(L)$ of a list L is inductively defined by $C_{fv}(a::L) = f(a, C_{fv}(L))$, the value of the empty list being v. If $f(a,b) = b \cup \{a\}$, we call this the *range* of the list and denote it by $r(L)$. We write $L \subset E$ if $P_E(L)$ holds, where $P_E(x)$ is $x \in E$; this means that every element of the list belongs to E.

```
Fixpoint contraction (A B: Type) (L: list A) (f: A-> B->B) (v: B):B :=
  match L with | nil => v
    | a :: b => f a (contraction b f v) end.

Definition list_range l := contraction l (fun a b => tack_on b a) emptyset.
Definition list_subset L E := list_prop (fun x => inc x E) L.
```

The range of a list is the smallest set E such that $L \subset E$. We show $L \subset r(L)$ by induction. We have $r(L) \subset r(a::L)$. We then use the fact that if P implies Q, then P(L) implies Q(L), where P is $x \in r(L)$ and Q is $x \in r(a::L)$. We can now state: if $L \subset E$ is a list of length n, there is an associated function $[0,n[\rightarrow E$.

```
Lemma list_range_pr: forall L, list_subset L (list_range L).
Lemma list_range_pr1: forall L E, list_subset L E -> sub (list_range L) E.

Lemma list_to_fB_axioms: forall l E, list_subset l E ->
  transf_axioms (fun n => (list_to_fctB l (cardinal_nat n)))
  (interval_co_0a  (nat_to_B (length l))) E.
Lemma list_to_fB_function:  forall l E, list_subset l E ->
  is_function (list_to_fB l E).
Lemma list_to_fB_W: forall l E n, list_subset l E ->
```

```
  inc n (interval_co_0a (nat_to_B (length l))) ->
  W n (list_to_fB l E) = list_to_fctB l (cardinal_nat n).
Lemma list_to_fB_W1: forall  l E n, list_subset l E ->
  n < length l ->
  W (nat_to_B n) (list_to_fB l E) = list_to_fctB l n.
Lemma fct_to_rev: forall (A:Type) (f:nat->A) n,
  rev (fct_to_list f n) =  fct_to_list(fun i=> f(n-i-1)) n.
```

More properties of intervals.

```
Lemma partition_tack_on_intco: forall a, inc a Bnat ->
  partition_fam (variantLc (interval_co_0a a)
    (singleton a)) (interval_co_0a (succ a)).
Lemma interval_co_0a_restr: forall a f, inc a Bnat ->
  (L (interval_co Bnat_order card_zero a) f
    = (restr (L (interval_co Bnat_order card_zero (succ a)) f)
      (interval_co_0a a))).
```

Let L_1 and L_2 be two lists, $L = L_1 ++ a::L_2$, Let G_1 and G be the functions associated to L_1 and L. If L_1 is a list of size n, then $G(n) = a$, and G and G_1 agree on $[0, n-1]$. In fact, G_1 is the restriction of G to the interval $[0, n[$, and if L_2 is empty, then G is the function obtained from G_1 by adding the relation $G(n) = a$.

```
Lemma length_app1: forall (A:Type) (a:A) l l',
  length l < length (l ++ a :: l').
Lemma length_app2:  forall (A:Type) (a:A) l ,
  nat_to_B (length (l++a::nil)) = succ (nat_to_B (length l)).

Lemma list_to_f_cons0: forall a l l',
  W (nat_to_B  (length l)) (list_to_f (l++ a:: l')) = nat_to_B a.
Lemma list_to_f_cons1: forall a l l' n, n < length l ->
  W (nat_to_B n) (list_to_f (l ++ a :: l')) =  W (nat_to_B n) (list_to_f l).
Lemma list_to_f_cons2: forall a l l',
  list_to_f l = restriction (list_to_f (l++ a :: l'))
               (interval_co_0a (nat_to_B (length l))).
Lemma list_to_f_cons3: forall a l,
  list_to_f (l++a::nil) = tack_on_f (list_to_f l)
  (nat_to_B (length l))  (nat_to_B a).

Lemma list_subset_cons: forall a l l' E,
  list_subset l E -> inc a E -> list_subset l' E ->
  list_subset (l'++a::l) E.
Lemma list_to_f_consB0: forall a l l' E,
  list_subset l E -> inc a E -> list_subset l' E ->
  W (nat_to_B  (length l)) (list_to_fB (l++ a:: l') E) = a.
Lemma list_to_f_consB1: forall a l l' n E, n< length l ->
  list_subset l E -> inc a E ->  list_subset l' E ->
  W (nat_to_B n) (list_to_fB (l++ a :: l') E) = W (nat_to_B n) (list_to_fB l E).

Lemma list_to_f_consB0: forall a l E, list_subset l E -> inc a E ->
  W (nat_to_B  (length l)) (list_to_fB (a :: l) E) = a.
Lemma list_to_f_consB1: forall a l n E, n< length l ->
  list_subset l E -> inc a E ->
  W (nat_to_B n) (list_to_fB (a :: l) E) =  W (nat_to_B n) (list_to_fB l E).
Lemma list_to_f_consB2: forall a l l' E,
```

```
  list_subset l E -> inc a E ->  list_subset l' E ->
  list_to_fB l E = restriction (list_to_fB (l++ a :: l') E)
              (interval_co_0a (nat_to_B (length l))).
Lemma list_to_f_consB3: forall a l E,
  list_subset l E -> inc a E ->
  list_to_fB (l++a::nil) E
  = tack_on_f  (list_to_fB l E) (nat_to_B (length l))  a.
```

We denote by LFB the variant of lf that converts a list $L \subset E$ into a function $I \to E$. The source of this function is an interval $[0, n[$; we shalll call this an iid function. We denote by FLB the variant of fl which is the inverse of LFB, i.e. $\mathsf{LFB}_E(\mathsf{FLB}(f)) = f$ and $\mathsf{FLB}(\mathsf{LFB}_E(L)) = L$, whenever f is a function whose source is $[0, n[$ and is target is E, and whenever $L \subset E$.

```
Definition fct_to_listB1 f n:=
  fct_to_list (fun n => W (nat_to_B n) f) n.
Definition fct_to_listB f := fct_to_listB1 f (cardinal_nat (source f)).
Definition iid_function f :=
  is_function f & exists n, source f = interval_co_0a (nat_to_B n).

Lemma list_to_fB_pr:  forall l E, list_subset l E ->
  iid_function (list_to_fB l E).
Lemma fct_to_list_lengthB :  forall f,  iid_function f ->
  nat_to_B (length (fct_to_listB f)) = cardinal (source f).
Lemma fct_to_listB_pr0: forall f i,
  iid_function f -> i< cardinal_nat (source f) ->
  list_to_fctB (fct_to_listB f) i = W (nat_to_B i) f.

Lemma fct_to_listB_pr1: forall l E, list_subset l E ->
  fct_to_listB(list_to_fB l E) = l.
Lemma fct_to_listB_pr2: forall f, iid_function f ->
   list_subset (fct_to_listB f) (target f).
Lemma fct_to_listB_pr3: forall f, iid_function f ->
  list_to_fB  (fct_to_listB f) (target f) = f.
```

11.6.2 Contracting lists

We show here the following. Assume that $f(n)$ is a cardinal for all n. Let $F(n)$ be the cardinal sum of the family $i \mapsto f(i)$ on $[0, n-1]$. Then $F(n+1) = f(n) + F(n)$, and there is a similar relation for the product. The same formula holds if $F(n+1)$ is the cardinal sum of the graph of the function f and $F(n)$ is cardinal sum of the graph of the restriction of f to $[0, n-1]$. We apply this to the case where f is $\mathsf{LF}(\mathsf{L}{+}{+}\mathsf{a}{::}\mathsf{nil})$; its restriction is $\mathsf{LF}(\mathsf{L})$, and $f(n) = a$.

```
Lemma induction_on_sum: forall a f,  inc a Bnat ->
  (forall a, inc a Bnat -> is_cardinal (f a)) ->
  let iter := fun n=> cardinal_sum (L (interval_co_0a n)f)
    in card_plus (iter a) (f a) = (iter (succ a)).
Lemma induction_on_prod: forall a f,  inc a Bnat ->
  (forall a, inc a Bnat -> is_cardinal (f a)) ->
  let iter := fun n=> cardinal_prod (L (interval_co_0a n ) f)
    in card_mult (iter a) (f a) = (iter (succ a)).
Lemma induction_on_sum1: forall f n,
  is_function f -> source f = interval_co_0a (succ n) -> inc n Bnat ->
  (forall a, inc a (source f) -> is_cardinal (W a f)) ->
  card_plus (cardinal_sum (graph (restriction f (interval_co_0a n))))
```

```
  (W n f) = cardinal_sum (graph f).
Lemma induction_on_prod1: forall f n,
  is_function f -> source f = interval_co_0a (succ n) -> inc n Bnat ->
  (forall a, inc a (source f) -> is_cardinal (W a f)) ->
  card_mult (cardinal_prod (graph (restriction f (interval_co_0a n))))
  (W n f) = cardinal_prod (graph f).
```

Denote by S(L) the cardinal sum of the family $\mathsf{LF}(\mathrm{L})$. The induction principle says S(L++ *a*::nil) = S(L) + *a*. By associativity we get S(L′ + +L) = S(L′) + S(L). We have S(nil) = 0 and S(*a*::nil) = *a*. If we take L′ = *a*::nil, the associativity formula gives S(*a*::L) = *a* + S(L). **Note**: in version 2, we changed the ordering of the elements of the list. This changes the properties of S; we proved the previous formula by using commutativity (rather than associativity).

```
Lemma induction_on_sum2: forall a l,
  card_plus (cardinal_sum (graph (list_to_f l))) (nat_to_B a)
  = cardinal_sum (graph (list_to_f (l++a::nil))).
Lemma induction_on_prod2: forall a l,
  card_mult (cardinal_prod (graph (list_to_f l))) (nat_to_B a)
  = cardinal_prod (graph (list_to_f (l++a::nil))).

Lemma induction_on_sum0:
  cardinal_sum (graph (list_to_f nil)) = card_zero.
Lemma induction_on_sum5: forall a,
  cardinal_sum (graph (list_to_f (a::nil))) = nat_to_B a.
Lemma induction_on_prod0:
  cardinal_prod (graph (list_to_f nil)) = card_one.
Lemma induction_on_prod5: forall a,
  cardinal_prod (graph (list_to_f (a::nil))) = nat_to_B a.

Lemma induction_on_sum4: forall l l',
  card_plus (cardinal_sum (graph (list_to_f l)))
  (cardinal_sum (graph (list_to_f l')))
  = cardinal_sum (graph (list_to_f (l++l'))).
Lemma induction_on_prod4: forall l l',
  card_mult (cardinal_prod (graph (list_to_f l)))
  (cardinal_prod (graph (list_to_f l')))
  = cardinal_prod (graph (list_to_f (l++l'))).
```

We define here the sum and product of a list of integers as a contraction. We shall denote this by Σ(L) and Π(L). This operation is related to the previous one by $\mathscr{N}(\Sigma(\mathrm{L})) = \sum \mathsf{LF}(\mathrm{L})$ and $\mathscr{N}(\Pi(\mathrm{L})) = \prod \mathsf{LF}(\mathrm{L})$.

```
Definition list_sum l := contraction (rev l) plus 0.
Definition list_prod l := contraction (rev l) mult 1.

Lemma list_sum_pr: forall l,
  nat_to_B (list_sum l) = cardinal_sum (graph (list_to_f l)).
Lemma list_prod_pr: forall l,
  nat_to_B (list_prod l) = cardinal_prod (graph (list_to_f l)).
```

If we denote by $a{+}{+}b$ the concatenation of two lists, then $\mathrm{C}_{fv}(a{+}{+}b) = f(\mathrm{C}_{fv}(a), \mathrm{C}_{fv}(b))$ provided that the result is true for the empty list, i.e., $f(v,b) = b$ for all b, and if f is associative. As a consequence $\Sigma(a{+}{+}b) = \Sigma(a) + \Sigma(b)$ and $\Pi(a{+}{+}b) = \Pi(a) \cdot \Pi(b)$. This is a general property of contractions of an associative function f

Denote by $\Sigma'_n(f)$ the expression $\Sigma(\mathsf{fl}(f,n))$. This is the sum of the list of the values $f(i)$ for $i<n$. If we denote by f_1+f_2 the function $i\mapsto f_1(i)+f_2(i)$ then we have $\Sigma'_n f+\Sigma'_n g=\Sigma'_n(f+g)$. There are similar formulas for the product. We have two induction formulas; the trivial one is $\Sigma'_{n+1}(f)=f(n)+\Sigma'_n(f)$; the non-trivial one is $\Sigma'_{n+1}(f)=f(0)+\Sigma'_n(f\circ \mathrm{S})$, where $(f\circ \mathrm{S})(i)=f(i+1)$.

```
Lemma contraction_assoc: forall  (A :Type) (L1 L2: list A)
  (f: A-> A->A) (v: A),
  (forall a b c, f a (f b c) = f (f a b) c) ->
  (forall a, f v a = a) ->
  (contraction (L1++L2) f v) = f (contraction L1 f v)(contraction L2 f v).


Lemma list_sum_single: forall a, list_sum (a::nil) = a.
Lemma list_prod_single: forall a, list_prod (a::nil) = a.
Lemma list_sum_app: forall a b, list_sum (a++b) = (list_sum a)+ (list_sum b).
Lemma list_sum_cons: forall a b, list_sum (a::b) = a + (list_sum b).
Lemma list_sum_consr: forall a b, list_sum (a++(b::nil)) = (list_sum a)+ b.
Lemma list_prod_app: forall a b,
  list_prod (a++b) = (list_prod a)* (list_prod b) .
Lemma list_prod_cons: forall a b, list_prod (a::b) = a*(list_prod b).
Lemma list_prod_consr: forall a b, list_prod (a++(b::nil)) = (list_prod a)* b.


Definition fct_sum f n:= list_sum (fct_to_list f n).
Definition fct_prod f n:= list_prod(fct_to_list f n).

Lemma fct_sum0: forall f, fct_sum f 0 = 0.
Lemma fct_prod0: forall f, fct_prod f 0 = 1.
Lemma fct_sum_rec: forall f n, fct_sum f (S n) = (fct_sum f n) + (f n).
Lemma fct_prod_rec: forall f n, fct_prod f (S n) = (fct_prod f n) * (f n).
Lemma fct_sum_rec1: forall f n,
  fct_sum f (S n) = (f 0) + (fct_sum (fun i=> f (S i)) n).
Lemma fct_prod_rec1: forall f n,
  fct_prod f (S n) = (f 0) * (fct_prod (fun i=> f (S i)) n).
Lemma fct_sum_plus: forall f g n,
  (fct_sum f n) +  (fct_sum g n) = fct_sum (fun i=> (f i) + (g i)) n.
Lemma fct_prod_mult: forall f g n,
  (fct_prod f n) *  (fct_prod  g n) =fct_prod (fun i=> (f i) * (g i)) n.
```

We show here some trivial results. The sum of a constant function is the product, and the sum is unchanged if we replace the list by its reverse. A bit more complicated: the reverse of the list associated to a function f is the list associated to $i\mapsto f(n-i-1)$.

```
Lemma fct_sum_const: forall n m, fct_sum (fun _ => m) n = n *m.
Lemma fct_prod_const: forall n m, fct_prod (fun _ => m) n = pow m n.
Lemma list_sum_rev: forall l, list_sum l = list_sum (rev l).
Lemma list_prod_rev: forall l, list_prod l = list_prod (rev l).
Lemma fct_sum_rev: forall f n,
  fct_sum f n =  fct_sum (fun i=> f(n-i-1)) n.
Lemma fct_prod_rev: forall f n,
  fct_prod f n =  fct_prod (fun i=> f(n-i-1)) n.
Lemma fct_to_rev: forall (A:Type) (f:nat->A) n,
  rev (fct_to_list f n) =  fct_to_list(fun i=> f(n-i-1)) n.
```

We consider here the inverse of BN: if f is a function of type nat→nat, we construct a function $[0,n[\to \mathbf{N}$. It is the composition of fl and LF. We shall denote it by NB. The first theorem is a statement about NB, the two others are statements about the graph of NB. The third theorem is deduced from the second by applying $[0,n+1[= [0,n]$.

```
Lemma l_to_fct: forall f n,
  BL (fun p => nat_to_B(f (cardinal_nat p))) (interval_co_0a (nat_to_B n))
  Bnat = list_to_f (fct_to_list f n).
Lemma l_to_fct1: forall f n,
  L (interval_co_0a (nat_to_B n)) (fun p => nat_to_B(f (cardinal_nat p)))
  = graph (list_to_f (fct_to_list f n)).
Lemma l_to_fct2: forall f n,
  L (interval_Bnat card_zero (nat_to_B n))
  (fun p => nat_to_B(f (cardinal_nat p)))
  = graph (list_to_f (fct_to_list f (S n))).
```

11.6.3 Iterated functions

Note: all useful results of this section have been moved to section 11.6.2. The remaining trivial results are given without comment.

```
Definition function_on_nat f :=
  fun m =>nat_to_B (f (cardinal_nat m)).

Lemma inc_function_on_nat_Bnat : forall f n,
  inc (function_on_nat f n) Bnat.
Lemma function_on_nat_pr : forall f n,
  cardinal_nat(function_on_nat f n) = f (cardinal_nat n).
Lemma function_on_nat_pr1 : forall f n,
  function_on_nat f (nat_to_B n) = nat_to_B (f n).
```

11.6.4 Factorial

In a previous version, we defined the factorial function as shown below. This definition is equivalrnt to the one provided by ssrnat.

```
Fixpoint factorial (n:nat) : nat :=
  match n with
  | 0 => 1
  | S p => (factorial p) *  S p
  end.

Lemma factorial0: factorial 0 = 1.
Lemma factorial1: factorial 1 = 1.
Lemma factorial2: factorial 2 = 2.

Lemma factorial_succ: forall n,  factorial (S n) = (factorial n) * (S n).
Lemma factorial_nonzero: forall n, 0 <> factorial n.

Lemma factorial_prop: forall f, f 0 = 1 ->
  (forall n, f (S n) =  (f n) * (S n)) ->
   forall x, f x = factorial x.
Lemma factorial_prop1: forall n, factorial n = fct_prod S n.
```

We prove here: if J ⊂ I, the product $\prod f_i$ restricted to J divides the product $\prod f_i$ restricted to I. We used this result to show that $n!$ divides $m!$. This requires 44 lines of proof, but proving by induction on c that $b!$ divides $(b+c)!$ requires only 3 lines (for that vasr of nat, ten lines in the case of Bnat).

```
Lemma divides_restriction_product: forall f x, fgraph f ->
     (forall i, inc i (domain f) -> is_finite_c (V i f)) ->
     (forall i, inc i (domain f) -> (V i f) <> card_zero) ->
     is_finite_set (domain f) -> sub x (domain f) ->
     BNdivides (cardinal_prod (restr f x)) (cardinal_prod f).

Lemma quotient_of_factorials: forall a b, b <= a ->
  Ndivides (factorial b) (factorial a).
Lemma quotient_of_factorials1: forall a b, b <= a ->
  Ndivides (factorial (a - b)) (factorial a).
Lemma tack_on_nat: forall a b, is_finite_set (tack_on a b) ->
  ~ (inc b a) -> cardinal_nat (tack_on a b) = S (cardinal_nat a).
```

11.6.5 The binomial coefficient

In the previous version, the binomial function was defined by induced as follows.

```
Fixpoint binom (n p:nat) {struct n} : nat :=
  match n, p with
    | 0, 0 => 1
    | 0, S m => 0
    | S q, 0 => 1
    | S q, S m =>  (binom q (S m)) + (binom q m)
end.
```

11.7 Removed theorems

The lemmas and definition shown here existed in previous version, but have been withdrawn.

A correspondence Γ = (G, E, E), whose graph G is an order on E, is also called an order by Bourbaki (this definition is in fact never used).

```
Definition order_c r :=
  is_correspondence r & source r = target r & source r = substrate (graph r)
  & order (graph r).
Theorem order_cor_pr: forall f,
  is_correspondence f ->
  order_c f =
  (source f = target f & source f = (domain (graph f)) &
    compose_graph (graph f)(graph f) = graph f &
    intersection2 (graph f) (opposite_order (graph f))
    = diagonal (substrate (graph f))).
```

Here is is the original definition of a product order. One can notice that f is uniquely defined by g. This iargument has beeen removed in the new version.

```
Definition product_order_r (f g:Set): EEP :=
```

```
  fun x x' =>
   inc x (productb f) & inc x' (productb f) &
   forall i, inc i (domain f) -> gle (V i g) (V i x)(V i x').

Definition product_order f g:=
  graph_on (product_order_r f g)(productb f).


Definition axioms_product_order f g:=
  fgraph f & fgraph g & domain f = domain g &
  (forall i, inc i (domain f) -> order (V i g)) &
  (forall i, inc i (domain f) -> substrate (V i g) = V i f).


Lemma order_product_order: forall f g,
  axioms_order_product f g -> order (product_order f g).
Lemma related_product_order: forall f g x x',
  axioms_product_order f g ->
  related(product_order f g) x x' =
  (inc x (productb f) & inc x' (productb f) &
  forall i, inc i (domain f) -> related (V i g) (V i x)(V i x')).
Lemma substrate_product_order: forall f g,
  axioms_product_order f g -> substrate(product_order f g) = productb f.
Lemma product_order_def: forall f g,  axioms_product_order f g ->
  image_by_fun (prod_of_products_canon f f)(product_order f g)
   = (productb g).  (* 36 *)
```

These are the original definitions of the lexicograph orced

```
Definition lexicographic_order_r (r f g:Set): EEP :=
  fun x x' =>
    inc x (productb f) & inc x' (productb f) &
    forall j, least_element (induced_order r (Zo (domain f)
      (fun i =>  V i x <> V i x'))) j -> glt (V j g) (V j x)(V j x').

Definition lexicographic_order_axioms r f g:=
  worder r & substrate r = domain f &
  fgraph f & fgraph g & domain f = domain g &
  (forall i, inc i (domain f) -> order (V i g)) &
  (forall i, inc i (domain f) -> substrate (V i g) = V i f).


Definition graph_order_r(x y g:Set): EEP :=
  fun z z' =>
    inc z (set_of_gfunctions x y) & inc z' (set_of_gfunctions x y) &
    forall i, inc i x-> related g (V i z)(V i z').
Definition graph_order x y g :=
  graph_on(graph_order_r x y g) (set_of_gfunctions x y).
Definition function_order x y r :=
  graph_on(fun u v => function_order_r x y r (sof_value x y u)
        (sof_value x y v))
  (set_of_functions x y).
```

Given two sets A and B, two distinct elements α and β, if I is the set that contains α and β, there is a family $(X_\iota)_{\iota\in I}$ such that such that $A = X_\alpha$ and $B = X_\beta$. This family is Lvariant. We shall denote it by $X_{\alpha\beta}(A,B)$. We show here uniqueness of the family.

```
Lemma two_terms_bij1: forall a b x y f,
  y <> x -> fgraph f -> domain f = doubleton x y -> V x f = a -> V y f = b ->
  range f  =  doubleton a b -> f = Lvariant x y a b.
```

Here are some trivial lemmas.

```
Lemma source_pfs:forall y x,
  source (partition_fun_of_set y x) = y.
Lemma target_pfs: forall y x,
  target (partition_fun_of_set y x) = powerset x.
Lemma source_graph_of_function: forall x y,
  source (graph_of_function x y) = set_of_sub_functions x y.
Lemma target_graph_of_function: forall x y,
  target (graph_of_function x y) =   (set_of_graphs x y).
Lemma sup_interval_co_0a: forall n, inc n Bnat ->
  supremum Bnat_order (interval_co_0a (succ n)) = n.
```

11.7.1 Other lemmas

The first lemma here is obvious. The second is unused.

```
Lemma cardinal_two_is_doubleton: exists x, exists x',
  x <> x' &  \2c = doubleton  x x'.
Lemma cardinal_equipotent1 x y: is_cardinal x -> is_cardinal y ->
  x \Eq y -> x = y.
Lemma card_lt_succ_le1 a b: inc b Bnat ->
  a <=c (succ b) -> a <> (succ b) -> a <=c b.
```

11.7.2 Definition of a function by induction

We explain here the initial implementation of section 7.2, more precisely the case when a function f is defined by (IND0), i.e., $f(0) = a$ and $f(n+1) = h(n, f(n))$ for $n \in \mathbf{N}$, or variants of this formulation.

In Version 1 we had the following two definitions (compare with induction_defined0_set and induction_defined1_set). They are of the form choose IND0 and choose IND1'. We have two theorems saying that these objects satisfy (IND0) and (IND1') respectively, and two others stating existence and uniqueness of (IND0), and existence of (IND1'). Together with these four theorems, we show a variant of integer_induction_stable and the Bourbaki variant of (IND0).

```
Definition induction_defined1 E h a:= choosef(fun f=>
  is_function f & source f = Bnat & target f = E & W card_zero f = a &
  forall n,  inc n Bnat -> W (succ n) f = h n (W n f)).
Definition induction_defined2 E h a p:= choosef(fun f=>
  is_function f & source f = Bnat & target f = E & W card_zero f = a &
  forall n,  cardinal_lt n p -> W (succ n) f = h n (W n f)).

Lemma integer_induction_stable: forall E g  a,
  inc a E -> is_function g -> source g = E -> target g = E ->
  sub (target (induction_defined g a)) E.
Lemma induction_with_var: forall E h a,
  is_function h -> source h = product Bnat E -> target h = E -> inc a E ->
  exists_unique (fun f=>  is_function f & source f = Bnat & target f= E &
```

```
    W card_zero f = a
    &  forall n, inc n Bnat -> W (succ n) f = W (J n (W n f)) h).


Lemma induction_with_var1: forall E h a,
  (forall n x, inc n Bnat -> inc x E -> inc (h n x) E) -> inc a E ->
  exists_unique (fun f=>  is_function f & source f = Bnat & target f= E &
    W card_zero f = a
    &  forall n, inc n Bnat -> W (succ n) f = h n (W n f)).

Lemma induction_with_var2: forall E h a p,
  (forall n x, inc n Bnat -> inc x E -> cardinal_lt n p -> inc (h n x) E)
  -> inc a E -> inc p Bnat ->
  exists f,  is_function f & source f = Bnat & target f= E &
    W card_zero f = a
    &  forall n, cardinal_lt n p -> W (succ n) f = h n (W n f).
Lemma induction_defined_pr2: forall E h a p,
  (forall n x, inc n Bnat -> inc x E -> cardinal_lt n p -> inc (h n x) E)
  -> inc a E -> inc p Bnat ->
  let f :=  induction_defined2 E h a p in is_function f &
    source f = Bnat &  target f = E & W card_zero f = a &
    forall n,  cardinal_lt n p -> W (succ n) f = h n (W n f).

Lemma induction_defined_pr1: forall E h a,
  (forall n x, inc n Bnat -> inc x E -> inc (h n x) E)
  -> inc a E ->
  let f :=  induction_defined1 E h a in is_function f &
    source f = Bnat &  target f = E & W card_zero f = a &
    forall n,  inc n Bnat -> W (succ n) f = h n (W n f).
```

The current definition does not use the choose function anymore.

```
Definition induction_defined0 h a := choose(fun f=>
  source f = Bnat & surjection f & W card_zero f = a &
  forall n,  inc n Bnat -> W (succ n) f = h n (W n f)).
Definition induction_defined s a:= choose(fun f=>
  source f = Bnat &  surjection f & W \0c f = a &
  forall n, inc n Bnat -> W (succ n) f = s (W n f)).
Definition induction_defined1 h a p := choose(fun f=>
  source f = Bnat & surjection f & W \0c f = a &
  (forall n,  n <c p -> W (succ n) f = h n (W n f)) &
  (forall n, inc n Bnat -> ~ (n <=c p) -> W n f = a)).

Definition induction_defined_set s a E:= choose(fun f=>
  is_function f &source f = Bnat & target f = E & W \0c f = a &
  forall n, inc n Bnat -> W (succ n) f = s (W n f)).

Definition induction_defined0_set h a E:= choose(fun f=>
  is_function f & source f = Bnat & target f = E & W \0c f = a &
  forall n, inc n Bnat -> W (succ n) f = h n (W n f)).

Definition induction_defined1_set h a p E := choose(fun f=>
  is_function f &source f = Bnat & target f = E & W \0c f = a &
  (forall n, n <c p -> W (succ n) f = h n (W n f)) &
  (forall n, inc n Bnat -> ~ (n <=c p) -> W n f = a)).
```

11.7.3 Intervals

We give here the original proof that the intersection of two intervals is an interval.

Let's say that an interval is of type B if it is bounded, of type L' if it is left unbounded, of type R' if it is right unbounded, of type U if it is $]\leftarrow,\rightarrow[$. Let's say that an interval is of type L if it is of type L' or U, of type R if it is of type R' or U.

Let's write $L' \cap L' = L'$ as a short-hand for: the intersection of two intervals of type L' is an interval of type L', this is lemma intersection_i3 and will be explained later. If we consider the reverse ordering, an interval remains an interval, but the lemmas shown here are more precise (they say for instance that the opposite of L' is R').

```
Lemma opposite_interval_cc: forall r a b,
  order r -> interval_cc r a b = interval_cc (opposite_order r) b a.
Lemma opposite_interval_oo: forall r a b,
  order r -> interval_oo r a b = interval_oo (opposite_order r) b a.
Lemma opposite_interval_oc: forall r a b,
  order r -> interval_oc r a b = interval_co (opposite_order r) b a.
Lemma opposite_interval_co: forall r a b,
  order r -> interval_co r a b = interval_oc (opposite_order r) b a.
Lemma opposite_bounded_interval: forall r x, order r ->
  is_bounded_interval r x -> is_bounded_interval (opposite_order r) x.
Lemma opposite_interval_ou: forall r a,
  order r -> interval_ou r a = interval_uo (opposite_order r) a.
Lemma opposite_interval_cu: forall r a,
  order r -> interval_cu r a = interval_uc (opposite_order r) a.
Lemma opposite_interval_uu: forall r,
  order r -> interval_uu r = interval_uu (opposite_order r).
Lemma opposite_interval_uo: forall r a,
  order r -> interval_uo r a = interval_ou (opposite_order r) a.
Lemma opposite_interval_uc: forall r a,
  order r -> interval_uc r a = interval_cu (opposite_order r) a.
Lemma opposite_unbounded_interval: forall r x, order r ->
  is_unbounded_interval r x -> is_unbounded_interval (opposite_order r) x.
Lemma opposite_interval: forall r x, order r ->
  is_interval r x -> is_interval (opposite_order r) x.
```

There are 9 types of intervals, thus 81 cases to consider. The case of intervals of type U is trivial, so that the number of cases is really 64. The new proof replaces bounded intervals by unbounded intervals, so that there are only 16 cases to consider. Let's start with these ones.

Case $L' \cap R' = R' \cap L' = B$. Consider $X =]\leftarrow, x[\cap]y, \rightarrow[$. If the intersection is non-empty, there is a such that $y \leq a \leq x$, thus $y \leq x$, and $X =]y, x[$. Otherwise $X =]x, x[$. Similarly, if we consider intervals that contain the end-point x or y, the intersection is empty, or an interval that contains the end-point x or y.

Case $L' \cap L' = L'$. Consider $X(b) =]\leftarrow, b[$ and $Y(b) =]\leftarrow, b]$. Let $d = \inf(b, c)$. We have $X(d) \subset X(b) \cap X(c) \subset Y(d)$. If d is in the intersection, then the intersection is $Y(d)$, otherwise it is $X(d)$. Replacing one of $X(b)$ or $X(c)$ by $Y(b)$ or $Y(c)$ is similar. Note: the intersection of two closed intervals is empty or closed, and the intersection of two open intervals is open, only when the order is total.

Using the reverse order, it follows $R' \cap R' = R'$, and this covers all unbounded intervals. All remaining cases are similar. We must consider what happens on the left, and what happens on the right. The big part of the proof (300 lines) consists in showing that the intersection

of two bounded intervals is a bounded interval. This does not follow directly from our new theorem, but is easy (for instance, if X is a subinterval of $[a,b]$, of the form $]\leftarrow,x[$, it is $[a,x[$).

```
Lemma intersection_interval1: forall r x y,
  lattice r ->is_closed_interval r x -> is_closed_interval r y  ->
    is_bounded_interval r (intersection2 x y).
Lemma intersection_interval2: forall r x y,
  lattice r ->is_open_interval r x -> is_open_interval r y  ->
  is_bounded_interval r (intersection2 x y).     (* 39 *)
Lemma intersection_interval3: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r
      (intersection2(interval_co r a b)(interval_co r a' b')) (* 19 *)
Lemma intersection_interval4: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r (intersection2(interval_oc r a b)(interval_oc r a' b')).
Lemma intersection_interval5: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r
     (intersection2(interval_co r a b)(interval_oc r a' b')). (* 30 *)
Lemma intersection_interval6: forall r x y,
  lattice r ->is_semi_open_interval r x -> is_semi_open_interval r y  ->
    is_bounded_interval r (intersection2 x y).
Lemma intersection_interval7: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r
      (intersection2(interval_cc r a b)(interval_oo r a' b')). (* 30 *)
Lemma intersection_interval8: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r
      (intersection2(interval_cc r a b)(interval_oc r a' b')). (* 18 *)
Lemma intersection_interval9: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r
      (intersection2(interval_oo r a b)(interval_oc r a' b')). (* 34 *)
Lemma intersection_interval10: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r (intersection2(interval_oo r a b)(interval_co r a' b')).
Lemma intersection_interval11: forall r a b a' b',
  lattice r -> inc a (substrate r) -> inc a' (substrate r)  ->
  inc b (substrate r)  -> inc b' (substrate r) ->
  is_bounded_interval r (intersection2(interval_cc r a b)(interval_co r a' b')).
Lemma intersection_interval12: forall r x y, lattice r ->
  is_bounded_interval r x -> is_bounded_interval r y ->
  is_bounded_interval r (intersection2 x y).
```

We consider now the case of unbounded intervals.

```
Lemma intersection_interval13: forall r x,
```

```
  is_interval r x -> intersection2 x (interval_uu r) = x.
Lemma intersection_interval14: forall r x y, lattice r ->
  is_left_unbounded_interval r x -> is_left_unbounded_interval r y ->
  is_left_unbounded_interval r (intersection2 x y).  (* 18 *)
Lemma intersection_interval15: forall r x y, lattice r ->
  is_right_unbounded_interval r x -> is_right_unbounded_interval r y ->
  is_right_unbounded_interval r (intersection2 x y).
Lemma intersection_interval16: forall r x y, lattice r ->
  is_left_unbounded_interval r x -> is_right_unbounded_interval r y ->
  is_bounded_interval r (intersection2 x y). (* 19 *)
Lemma intersection_interval17: forall r x y, lattice r ->
  is_unbounded_interval r x -> is_unbounded_interval r y ->
  is_interval r (intersection2 x y).
Lemma intersection_interval18: forall r x y, lattice r ->
  is_left_unbounded_interval r x -> is_bounded_interval r y ->
  is_bounded_interval r (intersection2 x y). (* 97 *)
Lemma intersection_interval19: forall r x y, lattice r ->
  is_right_unbounded_interval r x -> is_bounded_interval r y ->
  is_bounded_interval r (intersection2 x y).
Lemma intersection_interval20: forall r x y, lattice r ->
  is_unbounded_interval r x -> is_bounded_interval r y ->
  is_bounded_interval r (intersection2 x y).
```

The result is now obvious.

```
Theorem intersection_interval: forall r x y,
  lattice r -> is_interval r x -> is_interval r y ->
  is_interval r (intersection2 x y).
```

These are the original tactics used in this section.

```
Ltac uf_interval :=
  uf interval_cc;  uf interval_oo;  uf interval_co; uf interval_oc;
  uf interval_uu;  uf interval_uo;  uf interval_ou;
  uf interval_uc;  uf interval_cu.
Ltac zztac:=
  set_extens; Ztac; ee; match goal with H: inc _ (Zo _ _) |- _ => clear H end.
Ltac zztac2:= uf_interval; set_extens ;
  [  match goal with H: inc _  (intersection2 _ _) |- _ =>
       nin (intersection2_both H)  end ;
     match goal with
       H1:(inc _ (Zo _ _)), H2 :(inc _ (Zo _ _)) |- _
      => nin (Z_all H1); nin (Z_all H2); clear H1; clear H2; ee end;Ztac
    |
      Ztac; match goal with H: inc _ (Zo _ _) |- _ => clear H end;
      app intersection2_inc; Ztac; uf glt;ee; try order_tac].
Ltac eq_aux:= match goal with
    H: is_cardinal ?a |- cardinal ?b = ?a => wr (cardinal_le4 H); aw
  | H: is_cardinal ?a |- ?a = cardinal ?b => wr (cardinal_le4 H); aw
  | H: is_cardinal ?a |- cardinal ?a = ?b => wr (cardinal_le4 H); aw
end.
```

Chapter 12

Theorems, Notations, Definitions

List of all theorems of Bourbaki, with their Coq equivalent.

Theorems of Chapter 1

Proposition 1 (order_cor_pr) « A correspondence Γ between E and E is an ordering on E if and only if ...» , [14].

Proposition 2 (decreasing_composition) says that $u(v(x)) \geq x$ and $v(u(x)) \geq x$ and decreasing imply $u \circ v \circ u = u$ and $v \circ u \circ v = v$, [24].

Proposition 3 (adjoin_greatest) says that we can add a greatest element to an ordered set, [27].

Proposition 4 (compare_inf_sup1 and compare_inf_sup2) characterizes the supremum and infimum of a subset, [33].

Proposition 5 (sup_increasing and inf_decreasing) says that $\sup_A$ and $\inf_A$ are increasing functions of the set A, [34].

Proposition 6 (sup_increasing2 and inf_decreasing2) says that $\sup f$ and $\inf f$ are increasing functions of the function f, [34].

Proposition 7 (sup_distributive1 and inf_distributive2) asserts associativity of sup, [35].

Proposition 8 (sup_in_product and inf_in_product) characterizes supremum and infimum in a product, [36].

Proposition 9 (sup_induced2 and 3 variants) characterizes supremum of a subset of a subset, [36].

Proposition 10 (right_directed_maximal and left_directed_minimal) says that «in a right directed ordered set E, a maximal element a is the greatest element of E», [38].

Proposition 11 (total_order_monotone_injective and total_order_increasing_morphism) characterizes increasing functions and morphism on totally ordered sets, [39].

Proposition 12 (sup_in_total_order and inf_in_total_order) characterizes supremum and infimum in a totally ordered set, [40].

Proposition 13 (intersection_interval) says that in a lattice, the intersection of two intervals is an interval, [41].

Theorems of Chapter 2

Proposition 1 (well_ordered_segment) says that «in a well-ordered set E, every segment of E other than E itself is an interval $]\leftarrow, a[$, where $a \in E$, [46].

Proposition 2 (set_of_segments_iso_is and set_of_segments_worder) studies $x \mapsto]\leftarrow, x[$ [47].

Proposition 3 (worder_merge) studies the supremum of compatible well-orderings, [48].

Lemma 1 (order_merge1 and order_merge2) is a helper for Proposition 3.

Lemma 2 (transfinite_principle1 and transfinite_principle2) is a helper for C59.

Criterion C59 (transfinite_principle) is the principle of transfinite induction, [49].

Criterion C60 (transfinite_definition and transfinite_definition_stable) (Definition of a mapping by transfinite induction), [50].

Lemma 3 (Zermelo_aux) is a helper for Theorem 1.

Theorem 1 (Zermelo) says that «every set E can be well-ordered», [53].

Proposition 4 (Zorn_aux) is a generalization of Zorn's lemma [54].

Theorem 2 (Zorn_lemma) says «every inductive ordered set has a maximal element», [54].

Corollary 1 (inductive_max_greater).

Corollary 2 (maximal_in_powerset and minimal_in_powerset).

Theorem 3 (isomorphism_worder) studies existence and uniqueness of an isomorphism between two well-ordered sets, [54].

Lemma 4 (increasing_function_segments), [54].

Corollary 1 (unique_isomorphism_onto_segment), [55].

Corollary 2 (bij_pair_isomorphism_onto_segment), [55].

Corollary 3 (isomorphic_subset_segment) [55].

Theorems of Chapter 3

Proposition 1 cardinal_equipotent «two sets X and Y are equipotent if and only if their cardinals are equal», [69].

Theorem 1 (cardinal_le_wor) says that the ordering between cardinals is a well-ordering, [384].

Corollary 1 (card_le_to_ell).

Corollary 2 (cardinal_leA).

Proposition 2 (cardinal_supremum) says that a family of cardinals has a supremum, [78].

Proposition 3 (surjective_cardinal_le) says «if there exists a surjection f of X onto Y, then Card(Y) ≤ Card(X), [78].

Proposition 4 (cprod_pr and csum_pr) says that the cardinal sum or cardinal product of the family $\mathrm{Card}(E_\iota)$ is the cardinal of the sum or the product of the sets E_ι; [79].

Corollary (csum_pr1).

Proposition 5 (csum_An, cprod_An, csum_Cn, cprod_Cn and cprod_sum_Dn) asserts commutativity, associativity and distributivity of sum and products, [79].

Corollary. Application to the case of 2 or 3 arguments.

Proposition 6 (csum_zero_unit and cprod_one_unit) says that one can remove 0 in a sum and 1 in a product, [83].

Corollary 1 (csum0r, csum0l, cprod1r, cprod1l).

Corollary 2 (sum_of_ones and sum_of_same).

Proposition 7 (cprodnz) says that a cardinal product is non-zero if and only if each factor is non-zero, [84].

Proposition 8 (succ_injective) asserts injectivity of the successor function, [84].

Proposition 9 (cpow_pr) says that a^b remains unchanged if letters are replaced by equipotent sets, [84].

Proposition 10 cpow_pr3 says that a^b is a product where all factors are the same [84].

Corollary 1 (cpow_sum).

Corollary 2 (cpow_prod).

Corollary 3 (cpow_prod2).

Proposition 11 (cpowx0 and variants), states $\mathfrak{a}^0 = 1$, $\mathfrak{a}^1 = \mathfrak{a}$, $1^{\mathfrak{a}} = 1$, and $0^{\mathfrak{a}} = 0$, [85].

Proposition 12 (card_powerset) says $\mathrm{Card}(\mathfrak{P}(X)) = 2^X$, [85].

Proposition 13 (cardinal_le_when_complement) states that «$\mathfrak{a} \geq \mathfrak{b}$ if and only if there exists a cardinal $\mathfrak{c}$ such that $\mathfrak{a} = \mathfrak{b} + \mathfrak{c}$», [85].

Proposition 14 (csum_increasing and cprod_increasing) says the sum and product are increasing functions, [86].

Corollary 1 (csum_Mlele, cprod_Mlele).

Corollary 2 (cpow_Mlele).

Theorem 2 (cantor) says $X < 2^X$, [87].

Corollary (cantor_bis).

Theorems of Chapter 4

Proposition 1 (is_finite_succ) says that «a cardinal a is finite if and only if $a+1$ is finite», [91].

Proposition 2 (le_finite_finite, cpre_pr) says that (if n is an integer), if $a \leq n$ then a is an integer, if $n > 0$ there is a unique m with $m+1 = n$ and $a < m+1$ is equivalent to $a < n$, [92].

Corollary 1 (sub_finite_set).

Corollary 2 (strict_sub_smaller).

Corollary 3 (finite_image).

Corollary 4 (bijective_if_same_finite_c_inj, bijective_if_same_finite_c_surj).

Criterion C61 (cardinal_c_induction and variants) (principle of induction), [95]

Proposition 3 (finite_subset_directed_bounded, finite_subset_lattice_inf, finite_subset_lattice_sup, finite_subset_torder_greatest, finite_subset_torder_least) gives some properties of a finite subset of an ordered set [97].

Corollary 1 (finite_set_torder_greatest, finite_set_torder_worder)

Corollary 2 (finite_set_maximal).

Theorem 1 (maximal_inclusion) says that every nonempty set which is of finite character has a maximal element, [113].

Theorems of Chapter 5

Proposition 1 (finite_sum_finite and finite_product_finite) says that a finite sum or product of integers is an integer, [104].

Corollary 1 (finite_union_finite).

Corollary 2 (finite_product_finite_set).

Corollary 3 (BS_pow).

Corollary 4 (finite_powerset).

Proposition 2 (card_lt_pr) says that $a < b$ if and only if there is c such that $0 < c$ and $b = c + a$; [104].

Proposition 3 (finite_sum_lt and finite_product_lt) says that $\sum a_i < \sum b_i$ and $\prod a_i < \prod b_i$ if $a_i \leq b_i$ for each i and $a_j < b_j$ for some j, [104].

Corollary 1 (cpow_Mltle).

Corollary 2 (cpow_Mlelt).

Corollary 3 (csum_simplifiable_left and variants).

Corollary 4 (cdiff_pr and others).

Proposition 4 (restr_plus_interval_isomorphism) says that $x \mapsto x + a$ is a bijection (order isomorphism) $[0, b] \to [a, a + b]$, [109].

Proposition 5 (cardinal_interval) gives the cardinal of $[a, b]$, [110].

Proposition 6 (finite_ordered_interval) asserts that every finite totally ordered set is isomorphic to a unique interval $[1, n]$, [110].

Proposition 7 (char_fun_union and others) states properties of the characteristic function of a set, [112].

Theorem 1 (cquorem_pr) asserts existence and uniqueness of Euclidean division, [113].

Proposition 8 is expansion to base b [115].

Proposition 9 (shepherd_principle) says that if f is a function from a set with cardinal $\mathfrak{a}$ onto a set with cardinal $\mathfrak{b}$, and if all set $f^{-1}\langle\{x\}\rangle$ have the same cardinal $\mathfrak{c}$, then $\mathfrak{a} = \mathfrak{b}\mathfrak{c}$, [120].

Proposition 10 (number_of_injections_pr) gives the number of injections from a finite set into another one, [121].

Corollary (number_of_permutations).

Proposition 11 (number_of_partitions) gives the number of partitions with p_i elements, [123].

Corollary 1 (binomial7).

Corollary 2 (cardinal_set_of_increasing_functions).

Proposition 12 (sum_of_binomial) $\sum_p \binom{n}{p} = 2^n$ [125].

Proposition 13 is the binomial formula (is a definition in Coq) [125].

Proposition 14 (cardinal_pairs_lt and cardinal_pairs_le) counts the number of pairs (i, j) such $1 \leq i \leq j \leq n$ or $1 \leq i < j \leq n$, [133].

Corollary (sum_of_i).

Proposition 15 counts the number of monomials, [134].

Theorems of Chapter 6

Theorem 1 «The relation 'x is an integer' is collectivizing » (see inc_Bnat), [71].

Criterion C62 (restatement on C61, not shown in Coq).

Criterion C63 (integer_induction), [141].

Lemma 1 (infinite_greater_countable) «Every infinite set, E contains a set equipotent to **N**».

Lemma 2 (equipotent_N2_N) «The set $\mathbf{N} \times \mathbf{N}$ is equipotent to **N**».

Theorem 2 (equipotent_inf2_inf) «for every infinite cardinal $\mathfrak{a}$, we have $\mathfrak{a} = \mathfrak{a}^2$» [144]

Corollary 1 (power_of_infinite).

Corollary 2 (finite_family_product).

Corollary 3 (notbig_family_sum1).

Corollary 4 (sum2_infinite, product2_infinite).

Proposition 1 (countable_subset, countable_product countable_union) states properties of countable sets [145].

Proposition 2 (countable_finite_or_N) «Every countable infinite set E is equipotent to **N**», [145].

Proposition 3 (infinite_partition) says that every infinite set E has a partition X_i where X_i is equipotent to E and the index set to **N**, [145].

Proposition 4 (countable_inv_image) says that if f is a function from E onto F, such that F is infinite and $f^{-1}\langle\{x\}\rangle$ is countable for any $x \in$ F, then F is equipotent to E, [145].

Proposition 5 (infinite_finite_subsets) says that the set of finite subsets of an infinite set E is equipotent to E, [146].

Proposition 6 (increasing_stationary) characterizes stationary sequences, [147].

Corollary 1 (decreasing_stationary).

Corollary 2 (finite_increasing_stationary).

Proposition 7 (noetherian_induction) (Principle of Noetherian induction), [148].

Symbols

$x \wedge y$ is often replaced by "and". The Coq equivalent is /\.

$x \vee y$ is often replaced by "or". The Coq equivalent is \/.

$\neg x$ is often replaced by "not". The Coq equivalent is ~.

$(a|b)c$ is a Bourbaki notation, meaning the relation obtained by replacing b by a in c.

R$\{x\}$ is a Bourbaki notation, meaning that R is a relation that may depend on x. If R is a relation that depends on y, it is also $(x|y)$R.

τ_x(R) is a Bourbaki notation, it is the generic element satisfying R$\{x\}$.

$x \implies y$ is represented in Coq by x -> y.

$x \to y$ is a Coq notation meaning the type of functions from type a to type b.

$x = y$ is equality. We use it as synonym to $\iff$.

x : y is a Coq notation meaning that x is of type y.

$f(x)$ is the value of the function f at point x, parentheses are sometimes omitted.

$f\langle x\rangle$ is the value of f on the set x, see fun_image, image_by_graph, image_by_fun.

$\overset{-1}{f}\langle x\rangle$, see inverse_image.

$(\forall x)$P and forall x, p are similar constructions.

$(\exists x)$P and exists x, p are similar constructions.

$(\exists! x)$P means sometimes exists_unique.

$x \in y$, $x \ni y$ (is element of): see inc and elt.

$x \subset y$ (is subset of): see sub.

$\emptyset$ (empty set): see emptyset.

$\{x,$R$\}$ (set of x such that R): see Zo.

$\{x\}$, $\{x, y\}$: see singleton or doubleton.

$a - b$, $a \setminus b$, $\complement a$: see complement.

(x, y) (ordered pair): see J.

$\bigcup X$, $\bigcup_{\iota\in I} X_\iota$, see union.

$a \cup b$, $a \cap b$, see union2, intersection2.

$A \times B$, $u \times v$, $R \times R'$, see product, ext_to_prod, prod_of_relation.

$f \circ g$, see fcompose, gcompose, compose_graph, compose, composeC.

Δ_A, see diagonal.

G^{-1} see inverse_graph , inverse_fun or inverseC.

$x \mapsto y$ or $x \to y$ is the function that maps x to y, for instance $x \mapsto \sin x$ (source and target are implicit).

$\boldsymbol{x \to T}$ $(\boldsymbol{x \in A}, \boldsymbol{T \in C})$, is the function with source A, target C that maps x to T.

$(f_x)_{x\in A}$ is a shorthand for $x \to f(x)$ $(x \in A)$; see above, the piece $\boldsymbol{T \in C}$ is implicit.

$\hat{f}$, see extension_to_parts.

F^E, set of graphs of functions from E to F, see set_of_gfunctions.

$\mathscr{F}(E;F)$, set of functions from E to F, see see set_of_functions.

$\Phi(E,F)$, set of functions from a subset of E to F, see set_of_sub_functions.

f_x, f_y sometimes denotes the mappings $y \mapsto f((x,y))$ or $x \mapsto f((x,y))$, implemented as first_partial_fun, second_partial_fun.

$\tilde{f}$, sometimes denotes the mappings $x \mapsto f_x$ or $y \mapsto f_y$. Implemented as first_partial_function, second_partial_function.

$f \mapsto \tilde{f}$, implemented as first_partial_map, second_partial_map, is a bijection from $\mathscr{F}(B \times C; A)$ into $\mathscr{F}(B; \mathscr{F}(C; A)$ or $\mathscr{F}(C; \mathscr{F}(B; A))$.

$\prod_{\iota\in I} X_\iota$, product of a family of sets, see producttt.

$(x_\iota)_{\iota\in I}$ denotes an element of a product indexed by I.

$x \overset{r}{\sim} y$ is sometimes used instead of $r(x,y)$, especially when r is the graph of an equivalence relation.

$g_E(\sim)$, the graph of $\sim$ on E, see graph_on.

$\sim_f$ may denote eq_rel_associated f.

$\bar{x}$, may denote the equivalence class of x, see class.

$\hat{x}$ may denote a representative of the equivalence class x.

$E/\sim$, E/R (quotient set of E) see quotient.

R/S (quotient of two equivalence relations) see quotient_of_relations.

X_f sometimes means $f^{-1}\langle f\langle X\rangle\rangle$, see inverse_direct_value.

R_A see induced_relation.

$x \succ_r y$, $y \prec_r x$, notations used when x and y are related by a preorder relation,[9].

$x \le y$, $y \ge x$, $x < y$, $x > y$: notations used when x and y are related by an order relation, see gle, gge, glt, ggt.

$x \le_r y$, $y \ge_r x$, notations used when x and y are related by an order relation.

$x \le_{\mathrm{ord}} y$ is the orderings of ordinals, see ordinal_le.

$x \le_{\mathrm{Card}} y$ is the ordering of cardinals, see cardinal_le.

$x \le_{\mathbf{N}} y$ is the ordering of integers, see Bnat_le.

$f \mapsto G_f$ see graph_of_function.

$\varpi \mapsto \tilde{\varpi}$ see graph_of_partition.

$\sup(x,y)$, $\sup_E X$, $\sup X$, $\sup_{x\in A} f(x)$ see supremum, sup, sup_graph.

$\inf(x,y)$, $\inf_{\mathrm{E}} \mathrm{X}$, infX, $\inf_{x\in\mathrm{A}} f(x)$ see infimum, inf, sup_graph.

$[a,b]$, $[a,b[$, $]a,b]$, $]a,b[$, $[a,\rightarrow[$, $]a,\rightarrow[$, $]\leftarrow,b]$, $]\leftarrow,b]$, $]\leftarrow,\rightarrow[$, see interval.

$\mathfrak{r} \leq_{\mathrm{Card}} \mathfrak{n}$, order on cardinals, see cardinal_le.

$g^{(x)}$ is the restriction of g to $]\leftarrow,x[$ see restriction_to_segment

$\mathrm{Card}(x)$, $\mathrm{card}(x)$, is the cardinal of x, see cardinal.

0, 1, 2, 3, 4: see card_zero or card_three, or ord_zero

$\sum_{\iota\in\mathrm{I}} \mathfrak{a}_\iota$, $\mathsf{P}_{\iota\in\mathrm{I}} \mathfrak{a}_\iota$: cardinal sum or cardinal product of a family of cardinals, see card_sum and card_prod.

$\mathfrak{a}+\mathfrak{b}$, $\mathfrak{a}.\mathfrak{b}$, $\mathfrak{a}\mathfrak{b}$, is the cardinal sum or cardinal product of two cardinals, see card_sum2 card_prod2

$\mathrm{E}_1 + \mathrm{E}_2$ denotes also the ordinal sum, see ordinal_sum.

$\mathrm{X}_{xy}(a,b)$ is the family $x \mapsto a$ and $y \mapsto b$, see page 74.

$\mathrm{X}(a,b)$ is $\mathrm{X}_{\alpha,\beta}(a,b)$, for some fixed α and β.

$a_x \cup b_y$ is the disjoint union of $\mathrm{X}_{xy}(a,b)$, i.e., $a\times\{x\}\cup b\times\{y\}$.

$\mathfrak{a}^{\mathfrak{b}}$ is the cardinal power of two cardinals, see card_pow.

a^b is the power of two integers, see pow.

$\mathbb{N}$, $\mathbf{N}$, set of integers, see Bnat.

$a-b$ is the difference in $\mathbf{N}$, see card_diff; can also means minus, the difference in $\mathbb{N}$; it may denote the prdinal difference, see ord_diff:

$[a,b]$ is an interval on $\mathbf{N}$, [107]

$\sum_{i=a}^{b} t_i$ is $\sum_{i\in[a,b]} t_i$.

a::b is cons a b, it is the list obtained by putting the element a in front of the list b.

ϕ_{A} is the characteristic function on E.

a/b is the quotient of a and b.

$n!$ is the factorial of n.

$\binom{n}{p}$, see binom.

x^+ see succ_o.

$\mathrm{ord}(x)$ see ordinal.

$r \leq_{\mathrm{ord}} r'$ see order_le

$x \leq_{\mathrm{ord}} y$ $x <_{\mathrm{ord}} y$ see ordinal_le.

$x \ll y$ see ord_negl

$x \sharp y$ see ord_natural_sum

Letters

$\mathscr{B}$ see Bo.

$\mathscr{C}_{\mathrm{C}}(a,b)$, $\mathscr{C}_{\mathrm{T}}(p,q)$, $\mathscr{C}(p)$: see by_cases a b, chooseT and choose.

$\mathrm{C}_{xy}a$ stands for constant_function x y a, it is the constant function from x to y with value a.

$\mathrm{C}_{\mathrm{R}}x$ may denote the equivalence class of x for R, see class.

$\mathrm{Coll}_{\boldsymbol{x}}\boldsymbol{R}$ says that R is collectivizing in x.

$\mathscr{E}$, see Set.

$\mathscr{E}_x(\mathrm{R})$ appears in the English version where $\{x,\mathrm{R}\}$ is used in the French version; see Zo.

I_A, see identity.

I_{xy} see inclusionC, canonical_injection.

$\mathscr{I}(E,T,f)$ says that f is defined by transfinite induction, see transfinite_def.

$\mathscr{L}_X f$, $\mathscr{L} f$, $\mathscr{L}_{A;B} f$ (creating functions): see L, acreate, BL.

$\mathscr{M} f$, $\mathscr{M}_{A;B} f$ (inverse of $\mathscr{L}$) see bcreate1 and bcreate.

$\mathbb{N}$, **N**, set of integers, see Bnat.

$\mathscr{N}$, is the bijection from $\mathbb{N}$ onto **N**, see nat_to_B.

$o(E)$, $o'(E)$, see ordinal_o.

$\mathfrak{P}(x)$, see powerset.

$\mathrm{pr}_1 z$, $\mathrm{pr}_2 z$, $\mathrm{pr}_\iota f$, $\mathrm{pr}_J f$ (projections), see P, Q, pr_i, pr_j.

$\mathscr{R} x$ see Ro.

$R_{ab} f$ (restriction) see restriction2.

$\mathscr{V}(x,f)$, $\mathscr{V}_f x$ (value of a function): see V.

$\mathscr{W}_f x$ (value of a function): see W.

$\mathscr{X}(f,y)$, see Xo.

$\mathscr{Y}(P,x,y)$ see Yo.

$\mathscr{Z}(x,P)$ see Zo.

¶ is not defined. We use it as a paragraph separator.

Words

acreate f, $\mathscr{L} f$, is the correspondence associated to the Coq function f.

agrees_on x f f', agreeC x f f' is the property that for all $a \in x$, $f(a)$ and $f'(a)$ are defined and equal.

antisymmetric_r r says that the relation r is antisymmetric, [9].

asymmetric_set E says that if $x \in E$ and $y \in E$, at least one of $x \in y$ and $y \in x$ is false, [60].

axioms_product_order f g is the condition under which product_order f g is an order, [20].

back_to_nat f n returns the value of f (defined on a subset of **N**) as is if f were defined on $\mathbb{N}$, [401].

bcreate f A B, $\mathscr{M}_{A;B} f$, is a kind of inverse of $\mathscr{L}$.

bcreate1 f, $\mathscr{M} f$, is a kind of inverse of $\mathscr{L}$.

bijective f, bijectiveC f, means that f is a bijection.

binom n p, $\binom{n}{p}$, is the binomial coefficient, [125].

BL f a b, $\mathscr{L}_{A;B} f$, fun_function f a b, is function from A to B whose graph is $\mathscr{L}_A f$.

Bnat or **N** is the set of all integers, [91].

Bnat_le x y, Bnat_lt x y are the relations $x \leq y$ or $x < y$ on **N**, [94].

Bnat_order, is the ordering on **N**, [94].

Set or $\mathscr{E}$ is the type of sets.[1]

Bo, $\mathscr{B}$, is an inverse of $\mathscr{R}$.

bounded_above r X, bounded_below r X, bounded_both r X, mean that X is bounded for r (from above, below or both), [29].

by_cases a b, $\mathscr{C}_C(a,b)$, defines an object by applying a if P is true, and b if P is false.

[1] Changed to Set in version 2

canonical_doubleton_order x y is the well-ordering on the canonical doubleton, [**??**].

canonical_du2 x y is the canonical disjoint union of two sets. [150].

canonical_injection x y, I_{xy}, is the inclusion map on $x \subset y$.

canon_proj r, is the mapping $x \mapsto \bar{x}$ from E onto E/R, where E/R is the quotient set of r.

cantor_mon b x i is the value of one monomial in a CNF.

cardinal x is some set equipotent to x, [69].

card_diff a b, $a - b$ is the difference of the two cardinals. [106].

card_divides b a says that $a = bq$ for some q, [113].

card_dominant c says that x is a domoiant cardinal, [229]

card_five, card_ten, or 5 and 10, are the cardinals $4+1$ and $5+5$.

card_quo a b and card_rem a b are the quotient and remainder in the division of a by b [113].

card_pow a b, $\mathfrak{a}^{\mathfrak{b}}$, is the cardinal is the set of functions from y into x. [84].

card_prod x, $\mathsf{P}_{\iota\in I}\,\mathfrak{a}_\iota$, is the cardinal of the product of the family of sets, [79].

card_prod2 a b, $\mathfrak{a}.\mathfrak{b}$ or $\mathfrak{a}\mathfrak{b}$, is the cardinal product of a family of two elements, [80].

card_sum x, $\sum_{\iota\in I}\mathfrak{a}_\iota$, is the cardinal of the disjoint of the family of sets, [79].

card_sum2 a b, $\mathfrak{a}+\mathfrak{b}$, is the cardinal sum of a family of two elements, [80].

card_three, card_four, or 3 and 4, are the cardinals $2+1$ and $3+1$, [92]

card_zero, card_one,card_two, or 0, 1, 2, are the cardinals of the empty set, a singleton, or a doubleton with two distinct elements, [70].

cardinal x, Card(x) is some set equipotent to x, [69].

cardinal_fam x, says that x is a family of cardinals.

cardinal_le x y, $x \leq_{\mathrm{Card}} y$, says that x and y are two cardinals such that x is equipotent to a subset of y, [75].

cardinal_lt x y is $x \leq_{\mathrm{Card}} y$ and $x \neq y$, [75].

cardinal_nat x maps every set equipotent to the n-th ordinal to the natural number n, [388].

cardinal_set x, says that x is a sets whose elements are cardinals. [78].

char_fun A B is the characteristic function of A, as a mapping from B into $\{0,1\}$, [112].

choose p, $\mathscr{C}(p)$, is some x such that $p(x)$ is true, the empty set if no x satisfies p.

choosef p is some function f such that $p(f)$ is true, the identity on the empty set if no f satisfies p.

choosenat p, $\mathscr{C}_{\mathbb{N}}(p)$, is some integer i such that $p(i)$ is true, zero if no i satisfies p, [387].

chooseT p q, $\mathscr{C}_{\mathrm{T}}(p,q)$, is our basic axiom of choice.

class r x is the class of x for the equivalence relation r.

CNF_graph f says that f is a functional graph, whose domain has the form $[0,n[$,[180]

CNF_r X n, CNF_s X m k, CNF_s0 X M, CNF_m X M, CNF_change_n f n x, CNF_se X e, CNF_mp X Y pY n m, CNFq_sum are operations on CNF. [180], [188], [189], [195];, [199]

CNFr_ax f, CNFq_ax b X, CNFb_ax b X, CNF_axn X n, CNFp_ax1, CNFp_ax, CNFp_ax4 X p n x, are constraints of a CNF; [180],[183], [187], [192], [195].

CNFrv f, CNFbv b X, CNFv X, CNFp_value1, CNFp_value2 CNFpv1x , CNFpv X p are the value of a CNF, [180],[183], [192]

CNF_exponents X, CNF_coefficients is the set fo exponnts of coefficients of a CNF, [197]

CNFq_extension b x e is the extension of a CNF, [198]

coarse x is $x \times x$.

coarser x is the order associated to coarser_c, [13].

coarser_c f g, coarser_covering I f J g, two definitions that say for all $j \in$ J there is $i \in$ I such that $g_j \subset f_i$ or for all $g_j \in g$ there is $f_i \in f$ such that $g_j \subset f_i$.

coarser_preorder is the order induced by $\subset$ on preorders, [20].

cofinal_function f x y , cofinal_function_si f x y; cofinal_function_si_normal f x y; says that f is a cofinal function $y \to x$, possibly stricly increasing, possibly normal, [210].

cofinal_set r A says that x in the substrate of r there is an $y \in$ A such that $x \leq_r y$, [28].

cofinal_ordinal x y says that every element of x is bounded above by an element of y and vice-versa for the ordinal ordering, [162].

cofinality_c x defines the cofinality of a cardinal, [207]

cofinality x defines the cofinality of an orrdinal, [**??**]

coinitial_set r A says that x in the substrate of r there is an $y \in$ A such that $x \geq_r y$, [28].

common_extension_order_axiom g are the conditions on g for which an order can be put on the union, [48].

common_extension_order g h says that h is an order that on the union of the substrate of the g_i that coincides with the restriction, [48].

common_worder_axiom g are the conditions on g for which an well-ordering can be put on the union of the family [48].

compatible_with_equiv_p p r means that $p(x)$ and $x \overset{r}{\sim} y$ implies $p(y)$.

compatible_with_equiv f r means that $x \overset{r}{\sim} y$ is equivalent to $f(x) = f(y)$.

compatible_with_equivs f r r' means that $x \overset{r}{\sim} y$ is equivalent to $f(x) \overset{r'}{\sim} f(y)$.

complement a b, $a - b$, $a \setminus b$, $\complement b$, is the set of element of a not in b.

composableC f g, composable f g is the condition on correspondences (resp. functions) f and g for $f \circ g$ to be a correspondence (resp. function).

compose_graph f g, $f \circ g$, composition of two graphs.

compose f g, composeC f g, $f \circ g$, is the composition of two functions.

constant_graph s x is the graph of the constant function with domain s and value x.

contraction A B L f v: assume $f : A \times B \to B$ is a function, v an object of type B. If L is a list of type A C(L) is defined by $C(a :: b) = f(a, C(b))$ and $C(\text{nil}) = v$,[402].

correspondenceC is a data type with three slots, source, target and graph.

corr_value f associates to a correspondence f its triple (G, A, B).

countable_set x says that x is equipotent to a subset of $\mathbf{N}$, [145].

countable_ordinal x says that x is coutable and an ordinal,[162].

covering f x, covering_f I f x, covering_s f x, three variants of a family of sets (defined by f and I) whose union contains x.

cpred x is cardinal predecessor of x; it is the cardinal y such that $y + 1 = x$, is x if x is infinite, [92].

critical_ordinal y, says that y is critical for a function defined by induction [176]

csum_of_small1 x f, csum_of_small2 x z f, csum_of_small x y z f, sas that f is a family of cardinas, that are $< x$, indexed by z, whose sum is y, [207]

cofinality_aux r, cofinality' r, cofinality_alt, are variantof cofinality, [210]

csum_to_increasing_fun y n p is the function $[0,p] \to [0,n]$ that maps i to $\sum_{j\leq i} y_j$, [136].

cut x p is the set of all x that satisfy p.

cut r x is $r\langle x\rangle$.

decent_set x says that no element y of x satisfies $y \in y$, [60].

decreasing_map f r r', decreasing_fun f r r' is a function such that $x \leq_r y$ implies $f(x) \geq_{r'} f(y)$, [22].

decreasing_sequence f r says that f is a decreasing function with source **N** and target r, [147].

diagonal A, Δ_A, is the set of all (x,x) such that $x \in A$.

diagonal_application A is the diagonal mapping $x \mapsto (x,x)$ of A into Δ_A.

diagonal_graphp I E is the set of graphs of constant functions from I to E.

disjoint x y means $x \cap y = \emptyset$.

disjoint_union f, disjoint_union_fam f are two variants of the disjoint union of the family of sets f.

domain f is the set of x for which there is an y with $(x,y) \in f$, it is $\mathrm{pr}_1\langle f\rangle$.

doubleton x y, $\{x,y\}$, is a set with elements x and y.

doubleton_fam f x y means that f is a functional graph defined on a doubleton whose range is $\{x,y\}$,[74].

double_list_prop A L Q says that all elements x and y of the list L of type A satisfy the predicate $Q(x,y)$, whenever x comes before y, [401].

EEE is a shorthand for the type Set → Set → Set.

EEP is a shorthand for the type Set → Set → Prop.

elt x y, $x \ni y$, is the same as $y \in x$.

empty_function, empty_functionC is the identity on $\emptyset$.

empty_function_tg F, is the funciton define on the empty set with target F, [27].

emptyset, $\emptyset$, is a set without elements.

eq_rel_associated f is the graph of the equivalence relation $f(x) = f(y)$.

eqmod x y B x and y have the same remainder in the division by B, [119].

equipotent x y means that there is a bijection from x into y.

equipotent_ex x y denoted a bijection from x into y (it exists if x and y are equipotent),[72].

equipotent_to_subset x y means that x is equipotent to a subset of x, [75].

equivalence_associated f is the equivalence relation $f(x) = f(y)$.

equivalence_associated_o r is the equivalence relation $x \prec_r y$ and $y \prec_r x$, [15]

equivalence_r r, equivalence_re r x, says that the relation r is an equivalence relation (in x).

equivalence_corr r says that the correspondence r is associated ton an equivalence.

exists_unique p,$(\exists!x)p$, (this notation is not in Bourbaki) means that there exists a unique x such that $p(x)$.

expansion_value f b assume that f is a functional graph, defined for $i < k$ and such that $f_i < b$; the value is $\sum f_i b^i$ [116].

extends g f, extendsC g f says $g(x) = f(x)$ whenever $f(x)$ is defined.

extends_in E F is the relation extends in $\Phi(E,F)$, [12].

extension_order E F is the order associated to extends_in E F, [12].

ext_map_prod I X Y g is the function $(x_\iota)_{\iota\in I} \mapsto (g_\iota(x_\iota))_{\iota\in I}$ from $\prod_I X_\iota$ into $\prod_I Y_\iota$.

ext_to_prod u v is the function $(x, y) \mapsto (u(x), v(y))$, sometimes denoted $u \times v$.

extension_to_parts f, denotes the function $x \mapsto f\langle x\rangle$, from $\mathfrak{P}(A)$ into $\mathfrak{P}(B)$.

factorial n, $n!$, is the factorial function, [121].

fct_to_list A f n is the list of type A containing $f(i)$ for $i < n$, [399]

fct_sum f n, fct_prod f n, is the sum or product of the values $f(k)$ for $k < n$, computed via fct_to_list, [405].

finer_equivalence s r, comparison of equivalences, $x \overset{s}{\sim} y$ implies $x \overset{r}{\sim} y$.

finite_c x means that x is a finite cardinal, [71].

finite_int_fam f says that f is a functional graph, with a finite domain and whose range is a subset of the set of integers, [103].

finite_o x means that x is a finite ordinal, [68].

finite_set x means that x is a finite set, [68].

first_proj g is the function $x \mapsto \mathrm{pr}_1 x\ (x \in g)$.

first_proj_equiv x y, first_proj_equivalence x y, is the equivalence associated to first_proj on the set $x \times y$.

fcompose f g, $f \circ g$, composition of two graphs, without assumption.

fcomposable f g says that graphs g and $f \circ g$ have the same domain.

fgraph f says that f is a functional graph.

functional_graph f says that f is a functional graph.

function_order E F G, function_order_r E F G, is the order defined on $\mathscr{F}(E;F)$ by $\forall x, f(x) \prec_G g(x)$, [21].

function_prop f s t, function_prop_sub f s t. This is the property that f is a function from s into t, or into a subset of t.

fun_image x f, $f\langle x\rangle$, is the value of f on the set x.

fun_on_quotient r f, function_on_quotient r f b, function_on_quotients, fun_on_quotients r r' f, the function obtained from f on passing to the quotient of r (or r and r').

fun_set_to_prod E X is the canonical bijection between $(\prod X_\iota)^E$ and $\prod X_\iota^E$.

gcompose f g, $f \circ g$, composition of two graphs, assumes that range g is a subset of domain f.

gge r x y, $x \geq y$, says that $y \leq x$, [11]

ggt r x y, $x > y$, says that $x \geq y$ and $x \neq y$, [11]

gle r x y, $x \leq y$, says that x and y are related by r, [11]

glt r x y, $x < y$, says $x \leq y$ and $x \neq y$, [11]

graph f is a part of a correspondence.

graph_of_function X Y is the function $f \mapsto G_f$ defined on $\Phi(E,F)$, where G_f is the graph of f, [19].

graph_of_partition, is the function $\varpi \mapsto \tilde{\varpi}$, see partition_relation_set, [19]

graph_on r X is the graph of the relation r restricted to X.

graph_order E F G, graph_order_r E F G is the order defined on F^{E} by $\forall x, f(x) \prec_{\mathrm{G}} g(x)$, [21].

greatest_element r a is the property that a is the greatest (unique maximal) element of the substrate of the order r, [25].

greatest_lower_bound r X a is the property that a is the greatest element of the set of lower bounds of X, [30].

has_infimum r X says that X has a infimum, [30].

has_inf_graph r f says that the image of the graph f has an infimum, [33].

has_supremum r X says that X has an supremum, [30].

has_sup_graph r f says that the image of the graph f has a supremum, [33].

identity A, I_{A}, is is the graph of the identity function on the set A.

identity_fun A, I_{A}, is the identity function on the set A.

IM stands for the image of a function. Its axioms implement the Scheme of Selection and Union.

image_by_fun f A, $f\langle \mathrm{A} \rangle$, is $\{t, \exists x \in \mathrm{A}, t = f(x)\}$.

image_by_graph f x, $f\langle \mathrm{A} \rangle$, is $\{t, \exists x \in \mathrm{A}, (x, t) \in f\}$.

image_of_fun f, is the image of f.

inaccessible_w x, inaccessible x, says that x is a (weakly) inaccessible cardinal, [214], [229]

inc x y or $x \in y$ means that x is an element of y.

inclusionC x y, I_{xy}, it is the inclusion map on $x \subset y$ as a Coq function.

inclusion_order A, is the order induced by $\subset$ on $\mathfrak{P}(\mathrm{A})$ [10].

inclusion_suborder A, is the order induced by $\subset$ on A [10].

increasing_map f r r', increasing_fun f r r' is a function such that $x \leq_r y$ implies $f(x) \leq_{r'} f(y)$, [22].

increasing_pre f r r' says that f is an increasing function for preorders r and r', [264].

increasing_sequence f r says that f is an increasing function with source **N** and target r, [147].

induced_relation R A, R_{A}, is the equivalence induced by R on A.

induced_order R A, R_{A}, is the order induced by R on A [10].

induction_defined s a is the function f defined on **N** by $f(0) = a$ and $f(n+1) = s(f(n))$, [142].

induction_defined s a is the function f defined by $f(0) = a$ and $f(n+1) = s(f(n))$, [142].

induction_defined1 E h a is the function f defined by $f(0) = a$ and $f(n+1) = h(n, f(n))$, [142].

induction_defined2 E h a p is the function f defined for $n < p$ by $f(0) = a$ and $f(n+1) = h(n, f(n))$, [142].

induction_term s a is the term f defined by $f(0) = a$ and $f(n+1) = s(f(n))$, [121].

inductive_set r means that r is an order whose substrate is inductive, [53].

inf r x y, $\inf(x, y)$, is the greatest lower bound of pair $\{x, y\}$ (if it exists), [30].

infimum r X, $\inf_{\mathrm{E}} \mathrm{X}$, is the greatest lower bound of X (if it exists), [30].

infinite_c x means that x is a not a finite set, [71].

infinite_o x means that x is equipotent to x^+, [68].

infinite_set x means that x is a not a finite set, [71].

inf_graph r f, $\inf_{x\in A} f(x)$, is the greatest lower bound of the image of the graph f (if it exists), [33].

injective f, injectiveC f, means that f is an injection.

in_same_coset f is the relation "there exists i such that $x \in f(i)$ and $y \in f(i)$" between x and y.

intersection X, $\bigcap X$, is the intersection of a set of sets.

intersectiont I f, intersectionf x f, intersectiont g, $\bigcap_{\iota\in I} X_\iota$ is the set of elements a such that for all $\iota \in I$ we have $a \in X_\iota$.

intersection2 X Y, $X \cap Y$, is the intersection of two sets.

intersection_covering, intersection of coverings, .

interval_oo r a b, interval_oc r a b, interval_ou r a, interval_co r a b, interval_cc r a b, interval_cu r a interval_uo r bn, interval_uc r b interval_uu r; Intervals, [40].

interval_Bnat a b, interval_co_0a c is the interval $[a,b]$ or $[0,c[$ as a subset of $\mathbf{N}$.[107]

interval_Bnato a b, interval_Bnatco a is the interval $[a,b]$ or $[0,c[$ as an ordered set.[109]

inverse_direct_value f X, X_f, is $f^{-1}\langle f\langle X\rangle\rangle$.

inverse_graph G, $\overset{-1}{G}$, inverse graph of the graph G.

inverse_fun f or inverseC a b f H, $\overset{-1}{f}$, inverse of the function f.

inverse_image x f, $\overset{-1}{f}\langle x\rangle$, is the inverse value of f on the set x.

inv_image_relation f r, is the inverse image of the relation r under the function f.

inv_image_by_graph f x, inv_image_by_fun r x, $\overset{-1}{f}\langle x\rangle$, direct image of a set by the inverse function

inv_corr_value t associates to a $t = (G,A,B)$ its correspondence f.

inv_graph_canon G is the bijection $(x,y) \mapsto (y,x)$ from G to G^{-1}.

is_a_successor x says that x is an ordinal successor, [67].

is_antisymmetric r says that the graph r is antisymmetric, [9].

is_base_ten_expansion f k says that f is a functional graph on the interval $[0,k[$, with values in the interval $[0,9]$, [120]

is_bounded_interval r x, see interval.

is_cardinal x says that x is of the form $\mathrm{Card}(x)$, [69].

is_class r x says that x is an equivalence class for r.

is_closed_interval r x, see interval.

is_correspondence f says that f is associated to a triple (G,A,B).

is_equivalence r says that the graph r is an equivalence.

is_expansion f b k say that f is a functional graph, defined for $i < k$ and such that $f_i < b$, [116].

is_finite_c x, is_finite_set y: a cardinal x is finite if $x \neq x+1$, a set is finite if its cardinal is finite, [92].

is_function f says that f is a function in the sense of Bourbaki.

is_graph f says that f is a set of pairs.

is_graph_of g r is true if g is the graph of the relation r.

is_infinite_c x means that x is a cardinal that is not finite, [71].

is_inf_fun r f x, is_inf_graph r f x, says that x is the supremum of the image of the function of graph f for the order r, [32].

is_interval r x, see interval.

is_left_inverse r f means that r is a retraction or left-inverse of f, and $r \circ f$ is the identity.

is_left_unbounded_interval r x, see interval.

is_open_interval r x, see interval.

is_ordinal x, says that x is an ordinal, [61].

is_reflexive r says that the graph r is reflexive.

is_restriction f g says that f is the restriction of g to some set.

is_right_inverse s f means that s is a section or right-inverse of f, and $f \circ s$ is the identity.

is_right_unbounded_interval r x, see interval.

is_segment r s, says that s is the interval $]\leftarrow, x[$ or the whole substrate of a well ordered relation r, [45].

is_semi_open_interval r x, see interval.

is_singleton x means that x is a singleton.

is_sup_fun r f x, is_sup_graph r f x, says that x is the supremum of the image of the function or graph f for the order r, [32].

is_symmetric r says that the graph r is symmetric.

is_unbounded_interval r x, see interval.

is_transitive r says that the graph r is transitive.

J x y, or (x, y), is an ordered pair, formed of two items x and y.

nil is the empty list.

L X f, fcreate X f, $\mathscr{L}_X f$ is the graph formed of all $(x, f(x))$ with $x \in X$.

largest_partition x is the set of all singletons of x.

lattice r, is a relation for which $\sup(x, y)$ and $\inf(x, y)$ exist, [38].

least_element r a is the property that a is the least (unique minimal) element of the substrate of the order r, [25].

least_fixedpoint_ge f x y says that y the least fixed-point of f that is $\geq x$, [212]

least_ordinal p x is the least ordinal that satisfies p, provided that $p(x)$ holds, [62].

least_upper_bound r X a is the property that a is the least element of the set of upper bounds of X, [30].

left_directed r means that each doubleton is bounded below, [37].

left_inverseC, left inverse of a Coq function.

lexicographic_order r f g, lexicographic_order_r r f g, lexicographic_order_axioms r f g: assume that f is a family of sets with index I, g is a family of orders with index I such that the substrate of g_ι is f_ι, and r is a well-ordering on I; these conditions are the axioms; they allow to define an ordering and an order relation on the product of the family f, [56].

LHS is the left hand side of an equality.

limit_ordinal x, is a non-zero ordinal that is not a successor, [67].

list_range L is the smallest set containing all elements of the list, [402]

list_subset L E says that all elements of the list L belong to the set E, [402]

list_sum L, list_prod L, is the sum or product of the element of the list L [405].

list_to_fct L, list_to_f L, list_to_fctB L, list_to_fB L E, converts the list L into a mapping $\mathbb{N} \to \mathbb{N}$, or a function with source $[0, n[$ and target **N** or E.[399], [400].

lower_bound r X x says that for all $y \in X$, we have $x \leq_r y$, [29].

Lvariant a b x y, variant a x y, Lvariantc x y, these are functions whose range is the doubleton $\{x, y\}$.

maximal_element r a says that $x \leq_r a$ implies $x = a$, [25].

merge_int n m is a bijection $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$, [143].

minimal_element r a says that $x \geq_r a$ implies $x = a$, [25].

monotone_fun f r r' is a, increasing or decreasing function, [22].

mutually_disjoint f says that for all distinct i and j, $f(i)$ and $f(j)$ are disjoint.

mutually_cofinal x y says x and y are set of ordinals such each element of one set is bounded by an element of the other set.

nat, $\mathbb{N}$ is the type of natural integers in Coq.

nat_to_B, $\mathcal{N}$ is the canonical bijection from nat to the set of finite cardinals, [388].

natR n is maps a natural number onto a pseudo-ordinal, [386]

neq x y, $x \neq y$, is inequality.

normal_ofs f u, normal_ofs1 f u, normal_ofs2 f u says that f is a norma functional symbol defined for $\geq u$, [163]

Nquo a b and Nrem a b are the quotient and remainder in the division of a by b, [397].

number_of_injections b a is $a!/(a-b)!$, [121].

of_finite_character x says that x is of finite character, [98].

omega_fct x is the initial ordinal of index x, [205]

one_point is the basic singleton.

opposite_relation r is the relation $r(y, x)$ between x and y, [9].

opposite_order r is inverse graph of r, [9].

oprod_comm, oprod2_comm_P4 , oprod2_comm_P1 are helper definition for the sudy of $x \cdot y = y \cdot x$.

ord_diff a b, is the difference between two ordinals, [167].

ord_div_pr a b c q r, says that $a = bq + r$, $q < r$ and $q < c$, [169]

ord_ext_div_pr a b x y z says that $b = a^x y + z$ with $z < a^x$, $0 < y < a$ [179]

ord_indecomposable z says that z is an indecomposable ordinal, [170].

ord_index x is y such that $x = \aleph_y$, [206]

ord_induction_defined, (and others of the form ord_induction_???), ordinal induction for normal ordinal functionals, [171]

ord_natural_sum x y, $x \sharp y$, is the natural sum of two ordinals [199]

ord_negl x y, $x \ll y$, says that $x + y = y$, [182]

ord_omega; ω is the least infinite ordinal, [72]

ord_pair_le a b is the canonical ordering of pairs of ordinals.

ord_pow a b is the ordinal power, [177].

ord_prod r g, ord_prod2 a b, is the product of a family g whose index set is ordered by r, or the ordinal product of two ordinals a and b, [155].

ord_prod_expansion defines an ordinal product indexed by an interval $[0, n[$, .

ord_sum r g, ord_sum2 a b, is the ordinal sum of a family g whose index set is ordered by r, or the ordinal sum of two ordinals a and b, [155].

ord_sum_expansion defines an ordinal sum indexed by an interval $[0, n[$, .

ord_sup_pr E x says that x is the least ordinal such that $y \leq x$, whenever $y \in$ E.[66]

ord_zero, ord_one, ord_two are 0, 1 and 2, considered as ordinal, [70], [66].

order r says that the graph r is an order, [10].

order_associated r is the order associated to a preorder by passing on the quotient of "$x \prec y$ and $y \prec x$", [15]

order_axioms r s is the condition on which r is an order on s [17].

order_c f says that the graph of the function r is an order, [10].

order_fam f says that f is a family of ordered sets, [48].

order_isomorphic r r' says that there is an order-isomorphism f from $r \to r'$, [58].

order_isomorphism f r r' says that f is a bijective increasing function from the support of the order r onto the support of the order r', [17].

order_le r r' says $r \leq r'$ when r and r' are orderings.

order_morphism f r r' says that f is an injective increasing function from the support of the order r into the support of the order r', [17].

order_prod r g, order_prod_r r g, is the lexicographic order of the family g whose domain is the well-ordered set r, [149].

orprod_ax r g are the conditions for the order product to exist, [149].

order_prod2 r r', $E_1 \cdot E_2$, is ordinal product of two sets. [151].

order_r r says that the relation r is an order, [9].

order_re r x says that the relation r is an order on x, [10].

order_sum r g, $\sum_{\iota \in I} X_\iota$, is the ordinal sum of the family of of orders $g(i)$, the index set being ordered by r, [150].

order_sum2 r r', $E_1 + E_2$, is ordinal sum of two sets. [151].

order_with_greatest r a is the order obtained from r by adjoining a greatest element a, [28]

ordinal x is some ordinal y such that $o(y)$ is order-isomorphic to x, [64].

ordinal_epsilon x, epsilon_fct a, epsilon_fam b says that x is an ϵ-ordinal, is the least ϵ-ordinal that is $\geq a$ or is the ϵ-ordinal of rank b, [201].

ordinal_fam f says that f is a family of ordinals, [155].

ordinal_interval a b, is the set of ordinals x such that $a \leq x < b$, [163].

ordinal_le r r', ordinal_lt r r', denoted $x \leq_{\mathrm{ord}} y$ and $x <_{\mathrm{ord}} y$, is the ordering on ordinals [64]

ordinal_o E, ordinal_oa E (denoted $o(E)$ and $o'(E)$) are the relations $x \subset y$ or "$x \in y$ or $x = y$ defined on E; they coincide if E is an ordinal, [60].

ordinal_set x, says that x is a sets whose elements are ordinals. [62].

ord_zero, ord_one, ord_two, ord_omega are 0, 1, 2 and ω considered as ordinal numbers. [70]

orsum_ax r g are the conditions for the order sum to exist, [150].

P z, $\mathrm{pr}_1 z$ denotes x if z is the pair (x, y).

partial_fun1 f y, partial_fun1 f x, partial functions.

partition y x, partition_s y x,partition_fam f x, thee variants that say that y or f is a partition of x.

partition_fun_of_set Y X is the canonical injection from Y int $\mathfrak{P}(X)$, (if Y is a partition of X then $Y \in \mathfrak{P}(\mathfrak{P}(X))$) [13].

partition_relation f x is the equivalence relation associated to the partition f of x.

partition_relation_set y x, $\tilde{\omega}$, is the graph of the equivalence associated to the partition $y = \omega$ of x, [14].

partition_with_complement X A, is the partition of X formed of A and its complementary set.

partition_with_pi_elements p E f says that the sets $f(i)$ are of cardinal p_i, mutually disjoint and form a covering of E, [123].

pow x y, x^y, is the power function on the type nat, [389].

powerset x, $\mathfrak{P}(x)$, is the set of subsets of x.

$\mathrm{pr}_1 z$, $\mathrm{pr}_2 z$ stand for pr1 z and pr2 z. These are also denoted by P and Q. If z is the pair (x, y), these functions return x and y respectively.

pr_i f i, pr_it f i, $\mathrm{pr}_i f$, denotes a component of an element of a product.

pr_j f J, $\mathrm{pr}_J f$, is the function $(x_\iota)_{\iota\in I} \mapsto (x_\iota)_{\iota\in J}$

predecessor of x: is the greatest y such that $y < x$; In the case of ordinals is thee union, on the case of cardinals is cpred.

preorder r is a reflexive and transitive graph, [15].

preorder_r is a reflexive and transitive relation, [15].

prod_assoc_map is the function whose bijectivity is the "theorem of associativity of products".

prod_of_function u v, is the function $x \mapsto (u(x), v(x))$.

prod_of_products_canon F F', is the bijection between $\prod \mathrm{F}_\iota \times \prod \mathrm{F}'_\iota$ and $\prod(\mathrm{F}_\iota \times \mathrm{F}'_\iota)$.

prod_of_relation R R', $\mathrm{R} \times \mathrm{R}'$, is the product of two equivalences.

product A B, $\mathrm{A} \times \mathrm{B}$, is the set of all pairs (a, b) with $a \in \mathrm{A}$ and $b \in \mathrm{B}$

productt I X, product b g or productf I f, $\prod_{\iota\in I} \mathrm{X}_\iota$ is the product of a family of sets.

product1 x a is the product of the family defined on the singleton $\{a\}$ via value x.

product1_canon x a is the canonical application from x into product1 x a.

product2 x y is the product of the family defined on the doubleton $\{a, b\}$ via value x and y.

product2_canon x y is the canonical application from $x \times y$ into product2 x y.

product_compose, auxiliary function used for change of variables in a product.

product_order f g, product_order_r f g, is the order on the product $\prod \mathrm{X}_i$ induced by Γ_i (where f defines the family X_i and g defined the family Γ_i), [20].

pseudo_ordinal x says that x is a pseudo-ordinal, [60].

Q z, $\mathrm{pr}_2 z$ denotes y if z is the pair (x, y).

quotient R, E/R, is the set of equivalence classes of R

quotient_of_relations r s, R/S, is the quotient of two equivalences

quotient_order_r r s is the preorder relation induced in the quotient E/S by the preorder $\succ_\mathrm{R}$, where E is the common substrate of R and S, [264].

range f is the set of y for which there is an x with $(x, y) \in f$, it is $\mathrm{pr}_2\langle f\rangle$.

regular_cardinal x says that x is a regular cardinal, [207]

regular_ordinal x says that x is a regular ordinal, [**??**]

reflexive_r r x says that the relation r is reflexive in x.

reflexive_rr r says that the relation r is reflexive, [9].

related r x y is a short-hand for $(x, y) \in r$.

relation_on_quotient p r is the relation induced by $p(x)$ on passing to the quotient (with respect to x) with respect to R.

rep x is an element y such that $y \in x$, whenever x is not empty.

representative_system s f x means that, for all i, $s \cap \mathrm{X}_i$ is a singleton, where X_i is a partition of x associated to the function f.

representative_system_function g f x, means that g is an injection whose image is a system of representatives (see definition above).

restr x G is the restriction to x of the graph G.

rest_plus_interval a b, rest_minus_interval a b are the function $x \mapsto x + b$ and $x \mapsto x - b$ as bijections between $[0, a]$ and $[b, a + b]$, [109]

restricted_eq E is the relation "$x \in$ E and $y \in$ E and $x = y$".

restriction_function f x is like restr, but f and the restrictions are functions.

restriction2_axioms f x y is the condition: f is a function whose source contains x, whose target contains y, moreover $a \in x$ implies $f(a) \in y$.

restriction2 f x y, restriction2C f x y, restriction of f as a function $x \to y$.

restrictionC f H is the restriction to x of the function $f : a \to b$, where H proves $x \subset a$ implicitly.

restriction_product f j is the product of the restrictions of $\prod f$ to J.

restriction_to_image f is the restriction of the function f to its range, [74]

restriction_to_segment r x g, $g^{(x)}$, is the restriction of g to the segment S_x defined by the order r, [49]

restriction_to_segment_axiom r x g is the property for restriction_to_segment to be well-behaved, [49]

retraction: see is_left_inverse.

RHS is the right hand side of an equality.

right_directed r means that each doubleton is bounded above, [37].

right_inverseC, right inverse of a Coq function.

Ro x or $\mathscr{R}x$ converts its argument x of type u to a set, which is an element of u.

saturated r x means: for every $y \in x$, the class of x for the relation r is a subset of x.

saturation_of r x is the saturation of x for r.

second_proj g is the function $x \mapsto \mathrm{pr}_2 x$ $(x \in g)$.

section: see is_right_inverse.

section_canon_proj R is the function from E/R into E induced by rep.

segment r x, S_x, is the interval $]\leftarrow, x[$, [45].

segment_c r x is the interval $]\leftarrow, x]$, [45].

Set or $\mathscr{E}$ is the type of sets.

set_for_equipotent_inf2_inf E psi is a set used when proving that E is equipotent to E × E when E is infinite, [144].

set_of_cardinals_le a, set_of_cardinals_lt a is the set of cardinals $\leq a$, or $\leq a$, [78].

set_of_CNF_lt e , set_of_CNFq b el, is the set of all CNF with degree less than e, or whose eponents are in the list el, [187], [198]

set_of_correspondences A B means the set of triples

set_of_cardinals_le x is the set of all cardinals $\leq x$, [78].

set_of_endomorphisms E, is the set of triples (G, E, E) associated to functions from E into E.

set_of_finite_subsets x is the set of finite subset sof x, [343].

set_of_functions E F, denoted $\mathscr{F}(E;F)$, is the set of triples (G,E,F) associated to functions from E into F.

set_of_functions_sum_le E n, set_of_functions_sum_eq E n is the set of functions $f : E \to [0,n]$ such that the sum $\sum f(i)$ is $\leq n$ or $= n$, [134].

set_of_graph_sum_le E n, set_of_graph_sum_eq E n are the sets of graphs of function $f : E \to [0,n]$ such that the sum $\sum f(i)$ is $\leq n$ or $= n$, [134].

set_of_graph_sum_le_int p n is the set of graphs of function $f : [0,p] \to [0,n]$ such that the sum $\sum f(i) \leq n$, [135].

set_of_gfunctions E F, denoted F^E, is the set of graphs of functions from E to F

set_of_incr_functions E F, set_of_strict_incr_functions E F, is the set of (strictly) increasing functions from E to F,[130]

set_of_increasing_functions_int p n is the set of increasing function $[0,p] \to [0,n]$, [135].

set_of_injections E F is the set of injective functions from E into F, [121].

set_of_majorants1 is used in an example.

set_of_partitions p E is the set of all partitions X_i of E, where each X_i has p_i elements, [123].

set_of_partition_set X is the set of all partitions of X, [13].

set_of_permutations E F, is the set of injective functions from E onto itself, [121].

set_of_segments r, set_of_segments_strict r, is the set of all segments of an ordered set (with possible exclusion of the whole set), [47].

set_of_sub_functions E F, denoted $\Phi(E;F)$ is the set of triples (G,A,F) associated to functions from $A \subset E$ into F.

set_of_graphs E F, is the set of functional graphs from E to F, [19].

set_of_preorders E, is the set of preorders on E, [20].

singleton x, $\{x\}$, is a set with one element.

single_list_prop A L Q says that all elements of the list L of type A satisfy the predicate Q, [401].

singular_ordinal x says that x is a singular ordinal, [**??**]

sof_value x y z converts three elements into a correspondence,.

small_set x means that x hast at most one element.

smallest_partition x is the singleton $\{x\}$.

source f contains (resp. is equal to) the domain of the graph of a correspondence f (resp. function f).

stationary_sequence f says that the restriction of f to some interval $[n, \to [$ is constant, [147].

strict_decreasing_map f r r', strict_decreasing_fun f r r' is a function such that $x <_r y$ implies $f(x) >_{r'} f(y)$, [22].

strict_increasing_map f r r', strict_increasing_fun f r r' is a function such that $x <_r y$ implies $f(x) <_{r'} f(y)$, [22].

strict_monotone_fun f r r' is a strictly increasing or strictly decreasing function, [22].

strict_sub x y, $x \subsetneq y$, means $x \subset y$ and $x \neq y$.

sub x y, $x \subset y$, means that x is a subset of y.

substrate r is the union of the domain and range.

subsets_with_p_elements p E, is the set of subsets of E having p elements, [127].

successor of x: is the least y such that $x < y$; In the case of ordinals is called succ_o, in the case of cardinals is succ.

succ x is $x+1$, [71].

succ_o x is $x \cup \{x\}$, [60].

succ_c x is the next infinite cardinal after x, [206].

sum_of_substrates g is the disjoint union of the substrates, [150].

sup r x y, $\sup(x, y)$, is the least upper bound of the pair $\{x, y\}$ (if it exists), [30].

supremum r X, $\sup_{\mathrm{E}} \mathrm{X}$, is the least upper bound of X (if it exists), [30].

sup_graph r f, $\sup_{x \in \mathrm{A}} f(x)$, is the least upper bound of the image of the graph f (if it exists), [33].

surjective f, surjectiveC f, means that f is a surjection.

symmetric_r r says that the relation r is symmetric.

target f contains the range of the graph of a correspondence f.

the_CNF x, the_CNF_len, the_CNF_degree x is the CNF of cx, its length, its degree, [187].

the_greatest_element r, the_least_element_pr r denotes the greatest or least element of the ordering r, [25].

transf_axioms f A B says that for all $x \in \mathrm{A}$ we have $f(x) \in \mathrm{B}$, case where $\mathscr{L}_{\mathrm{A};\mathrm{B}} f$ is a function.

transfinite_def r p f, $\mathscr{I}(\mathrm{E}, p, f)$, says that f is defined by transfinite induction on the set E, well-ordered by r, via the property p, [50].

transfinite_defined r p is the function defined by the property p by transfinite induction on the well-ordered set r, [50].

transitive_r r says that the relation r is transitive.

transitive_set x says that if $a \in b$ and $b \in x$ then $a \in x$, [60].

trans_dec_set x says x is transitice and decent, [60].

total_order r means that r is a total order, [39].

two_points is the basic doubleton.

union X, $\bigcup \mathrm{X}$, is the union of a set of sets.

uniont I f, unionf x f, uniont g, $\bigcup_{\iota \in \mathrm{I}} \mathrm{X}_\iota$ is the set elements a with $a \in \mathrm{X}_\iota$ for some $\iota \in \mathrm{I}$.

union2 a b, $a \cup b$, is the union of two sets.

upper_bound r X x says that for all $y \in \mathrm{X}$, we have $y \leq_r x$, [29].

V x f, $\mathscr{V}(x, f)$ or $\mathscr{V}_f x$, is the value at the point x of the graph f.

variant, see Lvariant.

W x f, $\mathscr{W}_f x$, is the value at the point x of the function f.

well-ordered set, well-ordering relation, well-ordering: see worder.

worder r says that r is a well-ordering, [43]

worder_fam f says that f is a family of well-ordered sets, [48].

worder_r r says that r is a relation, that induces a well-ordering on each set where it is reflexive, [43].

Xo f y, $\mathscr{X}(f, y)$, this is $f(x)$ if $y = \mathscr{R} x$.

Yo P x y, $\mathscr{Y}(\mathrm{P}, x, y)$, is a function that associates to z the value x is P is true, and y if P is false.

Zo $\times$ R, $\mathscr{Z}(x,\mathrm{R})$, $\mathscr{E}_x(\mathrm{R})$ or $\{x,\mathrm{R}\}$: it is the set of all x that satisfy R.

Index

Bibliography

[1] Jon Barwise and Lawrence Moss. *Vicious Circles.* Number 60. CLSI Publications, 1996.

[2] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development.* Springer, 2004.

[3] N. Bourbaki. *Elements of Mathematics, Theory of Sets.* Springer, 1968.

[4] N. Bourbaki. *Éléments de mathématiques, Théorie des ensembles.* Diffusion CCLS, 1970.

[5] Georg Cantor. *Contributions to the Founding of the Theory of Transfinite Numbers.* Dover Publications Inc, 1897. Trans. P. Jourdain, 1955.

[6] Philip Carruth. Arithmetic of ordinals with applications to the theory of ordered abelian groups. *Bull. Amer. Math. Soc*, 48:262–271, 1942.

[7] José Grimm. Implementation of Bourbaki's Elements of Mathematics in Coq: Part One, Theory of Sets. Research Report RR-6999, INRIA, 2009. http://hal.inria.fr/inria-00408143/en/.

[8] Douglas Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid.* Basic Books, 1979.

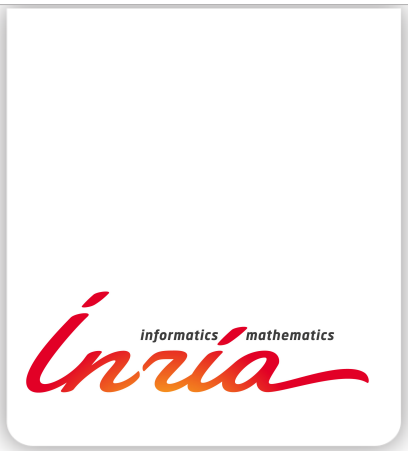
[9] Jean-Louis Krivine. *Théorie axiomatique des ensembles.* Presses Universitaires de France, 1972.

[10] Alfred Tarski. Quelques théorèmes sur les alephs. *Fundamenta Mathematicae*, 7:1–14, 1925.

[11] The Coq Development Team. The Coq reference manual. http://coq.inria.fr.

[12] Jean von Heijenoort, editor. *From Frege to Gödel: a source book in mathematical logic, 1879-1931.* Harvard University Press, 1967.

Contents



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399