Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two, From Natural Numbers to Real Numbers

José Grimm

Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two, From Natural Numbers to Real Numbers

JOSÉ GRIMM
Inria Sophia-Antipolis, Marelle Team
Email: Jose.Grimm@inria.fr

This paper describes a formalization of the first book of the series "Elements of Mathematics" by Nicolas Bourbaki, using the Coq proof assistant. In a first paper published in this journal, we presented the axioms and basic constructions (corresponding to a part of the first two chapters of book I, Theory of sets). We discuss here the set of integers (third chapter of book I, theory of set), the sets **Z** and **Q** (first chapter of book II, Algebra) and the set of real numbers (Chapter 4 of book III, General topology). We start with a comparison of the Bourbaki approach, the Coq standard library, and the Ssreflect library, then present our implementation.

1. INTRODUCTION

1.1 The Number Sets

This paper is the second in a series that explains our work on a project (named Gaia), whose aim is to formalize, on a computer, the fundamental notions of mathematics. We have chosen the "Elements of Mathematics" of Bourbaki as our guideline, the Coq proof assistant as tool, and ssreflect as proof language. As explained in the first paper [Gri10], our implementation relies on the theory of sets, rather than a theory of types.

In this paper we focus on the construction of some sets of numbers. Starting with the set of integers **N** (which is an additive monoid), one can define **Z** as its group of differences (this is a ring), then **Q** as its field of fractions (this is naturally a topological space), then **R**, the topological completion (that happens to be an Archimedean field); next comes the algebraic closure **C** of **R**. In this presentation, each set has a structure (group, ring, topology, order, etc), and the next set in the series is the unique (modulo isomorphism) extension satisfying some property.

There are many different implementations of these number sets in many proof assistants [BLM14]; we shall consider here only the Coq system, that uses a higher-order type theory and an intuitionist logic (the axioms of Excluded Middle and Choice are not taken for granted). We shall compare our implementation with two libraries: the Coq standard library and the Mathematical Components library[1] (see discussion in subsection 1.4). In Coq the set of natural integers comes for

[1]In short, MathComp; for details, see the web site http://math-comp.github.io/math-comp/

free; a typically untyped notion is that of an ordinal (hence that of a cardinal)[2]. In a theory of sets, the set of finite ordinals is the natural candidate for **N**. From **N** one gets **Z** and **Q** in a straightforward way (the only question being: should a fraction be in lowest terms[3]). The case of real numbers is more interesting: one can define the set of classical reals **R** as the set of classes of Cauchy sequences, for some equivalence relation; there are alternative presentations, in any case the Axiom of Choice is required. This set is however larger than the set of constructive real numbers. The set **R** is defined by an axiom in the standard library, not implemented in MathComp.

One possible implementation could be to follow Bourbaki (see discussion in subsection 1.3): Book I describes the theory of sets, introduces the order structure, and studies the set of natural numbers **N**; Book II defines and studies algebraic structures (monoid, group, ring, field, algebraic closure, etc) and defines **Z** and **Q**; Book III considers general topology and defines **R** and **C**. Implementing the whole of these three books is a challenge. Our ambition is more modest: provide a setting in which formulas of the form $\sum_{2^n \leq i < 2^{n+1}} 1/(s_i s_{i+1}) = 1$ can be expressed and proved correct [Here s_n is a sequence of integers defined by induction, see equation (9) below].

Our starting point is an implementation of Book I of Bourbaki (Theory of Sets, [Bou68]) with axioms designed by Carlos Simpson [Sim04a, Sim04b]; the implementation of Chapter 1 and 2 ("Description of formal mathematics", and "Theory of sets") has been presented in [Gri10], and a brief summary will be given below in subsection 1.5. Note that section 2 describes **N** and covers almost the whole of Chapter 3 ("Ordered sets, cardinals, integers") of Bourbaki. The next three sections describe **Z**, **Q** and **R** and some of their properties, using ideas from various sources. In the conclusion, we shall compare our implementation with the two Coqlibraries, and discuss further work on this subject.

1.2 Structures

We mentioned above that all the number sets had a structure; this means that, associated to the set, are operations with specific properties. For instance, in a set E, one can consider an addition $+$ and a comparison $\leq$; if some property of $+$ holds, then E is called a group, if some property of $\leq$ holds, then E is called an ordered set, if some additional property of $+$ and $\leq$ holds, then E is called an ordered group. Every ordered group is both a group and an ordered set (the converse may be false). Once a property has been proved for a given structure (say a group), it can be applied to all sets endowed with that structure. If two sets E, F are endowed with the same structure, a function $f : E \to F$ compatible with the operations is called a morphism (when bijective, it is an isomorphism). For instance the canonical injection $\mathbf{Z} \to \mathbf{Q}$ is a ring morphism. In the special case where $E \subset F$ and f is the canonical injection, then the structure of E is said to be induced by

[2] by the Burali-Forti paradox, there is no type for all the ordinals

[3] For instance 4/6 and 6/9 are *equivalent* for the equivalence that makes **Q** a quotient; in lowest terms they become *equal* to 2/3; in a theory of sets, the *classes* are equal and can be *identified* with their canonical representative 2/3

that of F (more simply one says that E is a subgroup of F); for instance, the image of the canonical injection $\mathbf{Z} \to \mathbf{Q}$ is a subring of $\mathbf{Q}$, ring-isomorphic to $\mathbf{Z}$. So, the question becomes: what is a structure, a sub-strucrure, a morphism, and how can these notions be implemented in computer.

Bourbaki has a whole Chapter of Book I that deals with structures. Its definitions are so far from the common use that nobody (even Bourbaki himself) uses them; an implementation of these ideas in Gaia has started. Carlos Simpson [Sim04a, Sim04b] (who designed the axioms used in Gaia) explains a possible design for structures. For instance an ordered group is a record[4] with three fields: the underlying set, the graph of the addition, the graph of the comparison, together with an axiom (that asserts some facts of the fields of the record). This idea seems nice and effective but only partially implemented in Gaia for lack of time.

In a higher-order type theory, like Coq, one can consider + as a notation for a function add of type $E \to E \to E$ (provided that the set E is type), hence define an ordered group structure as the record formed by: the set E, the function add, the function associated to $\leq$, the unit, and the axioms. The standard library of Coq provides some structures (ring, semi-ring, field), and a way to register instances of these structures, in order to be used by specialized tactics such as ring, field, nsatz, etc, (see the Coq reference manual).

In what follows, we consider the ssreflect library; an example of a commutative group is given in the tutorial [GM10]; details about effective implementation in ssreflect can be found in [GGMR09], for instance, it is explained how a commutative unit ring can be considered either as a commutative ring or as a unit ring (i.e., a ring in which some elements, called units, have a multiplicative inverse). Thanks to the mechanism of coercions and canonical structures, one can write expressions of the form '1:int_comRing' or '(1 / ((1+1):rat_Ring)) <= 1)%R'. In the second expression, the %R marker tells Coq to interpret some notations (here 1, +, /, $\leq$) in the ring scope. In both expressions, what follows the colon is not a type but an instance of a structure ($\mathbf{Z}$ as a commutative ring, or $\mathbf{Q}$ as a ring). The coercion mechanism now says that what precedes the colon has as type the carrier of the structure. The first example is equivalent to '1:int' and denotes $1_{\mathbf{Z}}$. The second example is more complicated. The denominator belongs to the carrier of the ring structure of $\mathbf{Q}$, the quotient belongs to the carrier of rat_unitRing, the unit ring structure of $\mathbf{Q}$ (this is the smallest structure in which it makes sense). The whole expression belongs to the carrier of rat_numDomainType (this is the smallest structure on $\mathbf{Q}$ that has a comparison). After that, Coq finds that each 1 is the unit of the underlying ring of the structure (i.e., something that simplifies to $1_{\mathbf{Q}}$). The important point here is that, not only Coq finds an interpretation of all notations, but the expression is trivially true.

A typical use of structures is the following: assume that E is a set, + a law of composition on E, that is associative and commutative, if a, b, c, d are four elements of E, then $(a + b) + (c + d) = (a + c) + (b + d)$. This is called addrACA

[4] The trick is that the record has type Set and not Type

in the case of a Z-module[5] (or a ring), mulrACA in the case of a commutative ring (when + is replaced by multiplication). Other examples are addnACA, minnACA, setUACA (addition on **N**, minimum on **N**, union of finite sets). It is transparent to the user whether these lemmas are an instance of the general lemma mulmACA or a particular case. (In Gaia, there is no generic structures, but we provides lemmas of this form in each particular case).

The library provides a generic sum $\sum_{i\in I} x_i$ where the index belongs to a finite set (a list, an interval I_n, or a more general interval $[n, m[$). The only assumption is that the operator has a unit (this means that one cannot take the minimum of a list of integers; but one can consider finite unions and intersections). This is a very powerful tool; again this is not implemented in Gaia. In the other hand, Gaia handles infinite sums (of cardinals, ordinals, etc), and one could define a sum over an infinite index, provided that all but a finite number of values are equal to the unit.

1.3 Constructing the real numbers in Bourbaki

We explain in this section how the sets **N**, **Z**, **Q**, and **R** with their algebraic structures are defined in Bourbaki.

1.3.1 *The set of natural numbers.* Bourbaki [Bou68] defines the cardinal $\mathrm{Card}(X)$ of a set X to be a representative of the equivalence class of X for the relation "there exists a bijection of X onto Y", denoted $\mathrm{Eq}(X, Y)$, and written "X and Y are equipotent". This equivalence class is not a set, so that the actual definition is $\tau_Z(\mathrm{Eq}(X, Z))$. Here τ is the Bourbaki's equivalent of Hilbert's epsilon operator; the definition implies that $\mathrm{Card}(X)$ is equipotent to X, and that $\mathrm{Eq}(X, Y)$ is equivalent to $\mathrm{Card}(X) = \mathrm{Card}(Y)$. These are the characteristic properties of the cardinal operator. The cardinals of the sets $\emptyset$, $\{\emptyset\}$ and $\{\emptyset, \{\emptyset\}\}$ are denoted by 0, 1 and 2. Note that $0 = \emptyset$, and this is the only case where the cardinal of a set can be explicitly computed.

The cardinal of the disjoint union S or cartesian product P of a family of cardinals $(A_i)_{i\in I}$ (i.e., the set of all (x, i), where $i \in I$ and $x \in A_i$, or the set of all functional graphs x with domain I such that $x_i \in A_i$) is called the cardinal sum or cardinal product of the family, denoted $\sum_{i\in I} A_i$ or $\prod_{i\in I} A_i$. Replacing A_i by an equipotent set replaces S or P by an equipotent set, thus leaves the cardinal unchanged. Addition and multiplication are associative and commutative; in particular, the sum and product of two cardinals is a cardinal, denoted $X + Y$ and $X \cdot Y$. The cardinal of the set Q of mappings $B \to A$ is denoted A^B (this is commonly called "A to the power of B", but for Bourbaki, the power of a set is its cardinal, so that "the power of the continuum" is the cardinal of **R**). Replacing A or B by equipotent sets replaces Q by an equipotent set, and leaves the cardinal unchanged.

The relation "X and Y are cardinals and X is equipotent to a subset of Y" (i.e., there is an injection $f : X \to Y$), denoted $X \leq Y$, is a well-ordering of the collection of cardinals, thanks to Zermelo's theorem. Every cardinal x has a successor x_+ such

[5] This is a commutative group, hence canonically isomorphic to a **Z**-module

that $x < x_+$, and for no y do we have $x < y < x_+$. Obviously $x+1 \leq x_+ \leq 2^x$. An integer is a cardinal x such that $x \neq x+1$. In this case, $x+1$ is the successor of x (the Generalized Continuum Hypothesis asserts that, if x is not an integer, then $x_+ = 2^x$).

There is a principle of induction: if p holds for zero and if for every integer n, $p(n) \Rightarrow p(n+1)$, then p holds for every integer. By the Axiom of Infinity, there is a set formed of all integers, denoted $\mathbf{N}$. This set is well-ordered by $\leq$, is stable by addition, multiplication; functions can be defined by induction on $\mathbf{N}$.

1.3.2 *Rational integers.* Bourbaki defines the set of rational integers in [Bou89a] (Algebra, Chapter I, section 2, paragraph 5) as the group of differences of $\mathbf{N}$. This is the least (in some sense) superset of $\mathbf{N}$ for which addition induces a group. More precisely: "The elements of $\mathbf{Z}$ are by definition the equivalence classes determined in $\mathbf{N} \times \mathbf{N}$ by the relation between (m_1, n_1) and (m_2, n_2) which is written $m_1 + n_2 = m_2 + n_1$; an element m of $\mathbf{N}$ is identified with the class consisting of the elements $(m+n, n)$, where $n \in \mathbf{N}$; it admits as negative in $\mathbf{Z}$ the class of elements $(n, m+n)$. Every element (p, q) of $\mathbf{N} \times \mathbf{N}$ may be written in the form $(m+n, n)$ if $p \geq q$ or in the form $(n, m+n)$ if $p \leq q$; it follows that $\mathbf{Z}$ is the union of $\mathbf{N}$ and the set of negatives of the elements of $\mathbf{N}$."

Let $\phi_+(m)$ be the first class mentioned above, and $\phi_-(m)$ the second, so that $\phi_-(m) = -\phi_+(m)$. The last statement of Bourbaki can be written as: $\mathbf{Z}$ is the union of the images of ϕ_+ and ϕ_-; moreover, these two functions are injective, and the intersection of the two images is the singleton $\{0\}$; the Coq standard library makes zero exceptional; the SSREFLECT library shifts ϕ_-, and Gaia disregards $\phi_-(0)$.

Addition is characterized by $\phi_+(a)+\phi_+(b) = \phi_+(a+b)$, $\phi_-(a)+\phi_-(b) = \phi_-(a+b)$, $\phi_-(a) + \phi_+(b) = \phi_+(b) + \phi_-(a)$ and $\phi_+(a) + \phi_-(b)$ is either $\phi_+(a-b)$ when $a \geq b$ or $\phi_-(b-a)$ otherwise. The relation $y - x \in \phi_+\langle \mathbf{N} \rangle$, denoted $x \leq y$, is a total order relation on $\mathbf{Z}$ which extends the order relation $\leq$ on $\mathbf{N}$ (i.e., ϕ_+ is an order isomorphism on its image). This order is invariant under translation (i.e., $x \leq y$ if and only if $x + z \leq y + z$).

Multiplication is defined in Bourbaki in paragraph 6. He says that, if E is a monoid, and $x \in E$, there is a unique homomorphism $f : \mathbf{N} \to E$ such that $f(1) = x$. This is denoted in SSREFLECT by `x *+ n` or `x ^+ n` in the case of a ring (that can be considered a monoid for both operations). Moreover, if x is invertible, there is a unique homomorphism $g : \mathbf{Z} \to E$ such that $g(1) = x$, it coincides with f on $\mathbf{N}$. The SSREFLECT library provides `x *- n` and `x ^- n` for the case of negative numbers, and `x *~ n` for rational integers. Multiplication on $\mathbf{Z}$ can be defined by taking $(\mathbf{Z}, +)$ for E.

1.3.3 *Rational numbers.* In section 8, paragraph 11 of [Bou89a], Bourbaki notices that $\mathbf{Z}$ is a principal ideal domain, thus defines gcd and lcm. In section 9, paragraph 4, he defines $\mathbf{Q}$ as the field of fractions of $\mathbf{Z}$. "Every rational number is thus of the form a/b where a and b are rational integers with $b \neq 0$ (and we may even take $b > 0$). $\mathbf{Q}_+$ is used to denote the set of rational numbers of the form a/b with $a \in \mathbf{N}$ and $b \in \mathbf{N}^*$." This set is stable by addition and multiplication; the

union of $\mathbf{Q}_+$ and the set of opposites of $\mathbf{Q}_+$ is $\mathbf{Q}$, the intersection being $\{0\}$; finally $\mathbf{Q}_+ \cap \mathbf{Z} = \mathbf{N}$ ("Obviously $\mathbf{N} \subset \mathbf{Z}$ and $\mathbf{N} \subset \mathbf{Q}_+$. Conversely, if x belongs to $\mathbf{Z} \cap \mathbf{Q}_+$, it is a rational integer; there exist two rational integers a and b with $a \geq 0$, $b > 0$ and $x = a/b$, whence $a = bx$; if $x \notin \mathbf{N}$, then $-x > 0$, whence $-a = b(-x) > 0$ and consequently $a < 0$ contrary to the hypothesis"). These relations are numbered from (1) to (5) by Bourbaki.

[Note: the proof of $\mathbf{Q}_+ \cap \mathbf{Z} = \mathbf{N}$ lacks the legendary rigor of Bourbaki. First it makes sense only if one identifies $\mathbf{N}$ as a subset of $\mathbf{Z}$ and $\mathbf{Z}$ as a subset of $\mathbf{Q}$. So, how should $x = a/b$ and $a = bx$ be interpreted, and why are these statements equivalent? Is a proof by contradiction needed, knowing that $b > 0$ and $bx \geq 0$ in $\mathbf{Z}$ implies $x \geq 0$?]

"Given two rational numbers x and y, we write $x \leq y$ if $y - x \in \mathbf{Q}_+$. It is easily deduced from (1), (3) and (4) that $x \leq y$ is a total ordering on $\mathbf{Q}$, from (5) that this relation induces the usual order relation on $\mathbf{Z}$. Finally from (1), it follows that the relations $x \leq y$ and $x' \leq y'$ imply $x + x' \leq y + y'$ and from (2) that the relation $x \leq y$ implies $xz \leq yz$ for all $z \geq 0$ and $xz \geq yz$ for all $z \leq 0$ (cf. VI, §2, no. 1)." (Chapter 6 of the Book of Algebra deals with ordered groups and fields). Bourbaki concludes this short presentation by defining the sign and absolute value of a rational number.

1.3.4 *Real numbers.* Real numbers are defined in Chapter 4 on the Book of Topology [Bou89b]. The first claim is that there is a unique topology on $\mathbf{Q}$, compatible with the additive structure such that a fundamental system of neighborhoods of zero is the set of intervals $]-a, a[$. This is a discrete Hausdorff topology; its completion is the set of real numbers, addition and multiplication being the continuous function extensions of those of $\mathbf{Q}$, the ordering is defined by the set of positive numbers (the adherence of the set of positive rational numbers).

This is a short introduction to real numbers; a whole book is dedicated to the study of functions of a real variable.

1.4 Real numbers in Coq

We shall explain how integers, rational and real numbers are implemented, both in the standard Coq library and in ssreflect[6]. To begin with, the natural numbers are defined by

```
Inductive nat : Set :=  O : nat | S : nat -> nat.
```

There is an induction principle, that allows one to define functions by induction, for instance addition, subtraction, multiplication, etc. We give the definition of comparison and maximum of the standard library (note that max is recursive in its first argument).

```
Inductive le (n:nat) : nat -> Prop :=
```

[6]Technically, MathComp is the name of the library, ssreflect the tactic language and Small Scale Reflection the principle behind the language and the library

```
  | le_n : n <= n
  | le_S : forall m:nat, n <= m -> n <= S m
  end
where "n <= m" := (le n m) : nat_scope.
Fixpoint max n m : nat :=
  match n, m with
    | O, _ => m
    | S n', O => n
    | S n', S m' => S (max n' m')
  end.
```

The SSREFLECT library provides a file *ssrnat.v*, that states arithmetic properties about natural numbers, using the same operations as above. However, boolean equality is preferred. For instance, we give the statement that a product of two naturals is equal to 1 if and only if both factors are equal to 1. Comparison is redefined by $m \leq n$ when $m - n$ is zero; this allows a simple definition of the maximum.

```
Lemma muln_eq1 m n : (m * n == 1) = (m == 1) && (n == 1).
Definition leq m n := m - n == 0.
Definition maxn m n := if m < n then n else m.
```

Note: most functions of the standard library with two arguments call them n and m; the SSREFLECT library uses alphabetic order (m and n) and is generally more systematic regarding naming conventions. The name of arguments is irrelevant, except for implicit arguments, when the function is not fully applied. For instance '`leq_trans`' takes 5 arguments, three natural numbers n, m, p, and proofs of $m \leq n$, $n \leq p$ and returns a proof of $m \leq p$. The first three arguments can be deduced from the type of arguments 4 and 5, hence are implicit. You can however say '`@leq_trans 3`'; this is the same as '`(leq_trans (n:=3))`'. Logically, n is the second argument; it is placed first in the list since m and p can be deduced from the conclusion $m \leq p$.

A `positive` number corresponds to a non-zero natural number written in base two: each digit is represented by `xO` or `xI`, but the leading digit one is represented by `xH`. A member of `N` is zero or positive. The SSREFLECT library does not use these two datatypes, but provides bijections between `nat` and `N`, and proofs that addition, multiplication, and exponentiation are morphisms.

The standard library defines `Z` as the disjoint union of two copies of `positive`, (sometimes called `pos` and `neg`), together with a zero, and defines all usual operations (we give opposite as an example). All properties of an ordered ring are satisfied. Moreover, COQ provides a nice concrete syntax (for instance, 17 can be parsed as a `nat`, a `positive`, an `N`, or a `Z`).

Note: notations can be overloaded, so that `0` may represent `O`, `N0`, `Z0`, or the zero of a ring; moreover, a function f (for instance the canonical injection $\mathbf{N} \to \mathbf{Z}$) may be declared as a coercion, so that $f(x)$ can be input as x. Each notation is defined in a scope (in the example above, `le` is defined together with its notation in the scope of natural numbers). Some notations may be ambiguous: for instance, if A and B are two types, with a coercion f, two operations g_A and g_B with the same

notation g, then $g(x) : B$ may stand for $f(g(x))$ or $g(f(x))$. For this reason there is a way to indicate in which scope should some notations be resolved.

```
Inductive positive : Set :=
  | xI : positive -> positive | xO : positive -> positive | xH : positive.
Inductive N : Set :=
  | N0 : N | Npos : positive -> N.
Inductive Z : Set :=
  | Z0 : Z | Zpos : positive -> Z | Zneg : positive -> Z.
Definition opp x :=
  match x with  | 0 => 0| pos x => neg x | neg x => pos x end.
```

The SSREFLECT library defines (in *ssrint.v*) a datatype int, formed of two copies of nat; one corresponds to the injection $\mathbf{N} \to \mathbf{Z}$, the second associates to n the opposite of $n+1$. We show here the definition of the opposite function. [Here %N says that the argument before it should be in parsed in the scope whose name is N; it is not necessary here as 'n.+1' is a non-overloaded notation for 'S n'. The trouble is that the standard and SSREFLECT libraries use the same N for different scopes.]

```
CoInductive int : Set :=  Posz : nat -> int | Negz : nat -> int.
Definition oppz m := nosimpl
  match m with
    | Posz n => if n is (n'.+1)%N then Negz n' else Posz 0
    | Negz n => Posz (n.+1)%N
  end.
```

Note that the datatype is CoInductive, thus lacks a principle of induction, but one is provided explicitly as: in order for $P(z)$ to hold for all $z \in \mathbf{Z}$, it suffices that $P(0)$ holds, and for every natural number n, $P(n)$ implies $P(n+1)$ and $P(-n)$ implies $P(-(n+1))$. This is for instance used to show associativity of addition. This datatype is embedded with a number of structures (Z-module, ring, real domain, etc). In particular there is an order and a norm, compatible with the ring operations, and satisfying the triangular inequality $|x+y| \leq |x| + |y|$. Every group is a Z-module (one can define x^n when x is in the group and $n \in \mathbf{Z}$, and this operation satisfies the usual properties). On a ring R, two operations are defined: $x.n$ and x^n, where x is in the ring and $n \in \mathbf{Z}$, such that $x.2 = x + x$ and $x^2 = x \cdot x$ (the second operation is most useful when x is a unit, so that $x \cdot x^{-1} = 1$; this is the case when x is a non-zero element of a field). On a real field, these operations satisfy some monotonicity properties (for instance: if $x \geq 0$, $y \geq 0$ and $n \neq 0$, then $x^n = y^n$ is equivalent to $x = y$). On every real domain, one can define a sign function $s(x)$ whose value is -1, 0, or 1 depending on whether $x < 0$, $x = 0$ and $x > 0$ (in fact, there are two such functions, with value in the domain or in $\mathbf{Z}$). Among other properties, $s(a \cdot b) = s(a) \cdot s(b)$. In the case of $\mathbf{Z}$, if $a(x)$ is the absolute value of x, we have $x = a(x) \cdot s(x)$ and $a(x) = x \cdot s(x)$.

In summary: the library defines many algebraic structures, some are classic like group, ring, field, other are a bit involved like real field or Archimedean field, and $\mathbf{Z}$ is an example of these structures.

The standard library defines $\mathbf{Q}$ and a relation Qeq denoted here $x \sim y$:

```
Record Q : Set := Qmake {Qnum : Z; Qden : positive}.
Notation QDen p := (Zpos (Qden p)).
Definition Qeq (p q : Q) := (Qnum p * QDen q)%Z = (Qnum q * QDen p)%Z.
```

The relation is an equivalence and **Q** is called a setoid. All operations, like addition, as well as comparison, are compatible with $\sim$. Addition is associative modulo $\sim$. The standard library has a bunch of functions that allow rewrite to replace $x + (y + z)$ by $(x + y) + z$ in situations where this is allowed.

```
Instance Qplus_comp : Proper (Qeq==>Qeq==>Qeq) Qplus.
Instance Qle_comp : Proper (Qeq==>Qeq==>iff) Qle.
Theorem Qplus_assoc : forall x y z, x+(y+z)==(x+y)+z.
```

On the other hand, one can reduce fractions x into $\bar{x}$ by extracting gcds (the definition here is strange, as it must cope with the fact that numerator and denominator have different types, notations are sometimes a bit folkloric), in such a way that $x \sim \bar{x}$, and $x \sim y$ implies $\bar{x} = \bar{y}$. Note that $x = \bar{x}$ if and only if the numerator and denominator of x are coprime; thus, if y is $\bar{x}$, we have $\bar{y} = y$.

```
Definition Qred (q:Q) :=
  let (q1,q2) := q in
  let (r1,r2) := snd (Z.ggcd q1 ('q2))
  in r1#(Z.to_pos r2).
Lemma Qred_correct : forall q, (Qred q) == q.
Lemma Qred_complete : forall p q,  p==q -> Qred p = Qred q.
Lemma Qred_involutive : forall q:Q, Qred (Qred q) = Qred q.
```

An extension to *QArith.v* is the following definition:

```
Record Qc : Set := Qcmake { this :> Q ; canon : Qred this = this }.
```

A element X of type Qc is a fraction X_v and a proof X_p of $\overline{X_v} = X_v$. The non-trivial point with this definition is to prove that $X_v \sim Y_v$ implies $X = Y$ (that $X_v = Y_v$ is obvious, so that Y_p becomes a second proof of $\overline{X_v} = X_v$, a non-trivial result says that two proofs of equality for a type A are equal if equality is decidable for this type, which is the case here). Since Qred is involutive, if x is a rational number, there is X (denote it $q(x)$) such that $X_v = \bar{x}$. Then $x, y \mapsto q(x+y)$ defines an addition on Qc. Subtraction, multiplication, division are similarly defined; the result is a ring and a field (in the sense of the ring and field tactics).

The SSREFLECT library defines (in file *rat.v*) the set of rational numbers similarly to Qc, where both numerator and denominator have the same type int; the condition "$\bar{x} = x$" is replaced by a boolean condition. Almost every SSREFLECT datatype is an instance of eqType (the mechanism of Canonical Structures makes this transparent to the user); this means that there is a boolean equality $x == y$ (not to be confused with the $==$ used above as notation for Qeq) such that $x == y \Leftrightarrow x = y$. The boolean equality of int obviously induces a boolean equality on rat.

```
Record rat : Set := Rat {
  valq : (int * int) ;
```

```
  _ : (0 < valq.2) && coprime `|valq.1| `|valq.2|
}.
Lemma rat_eqE x y : (x == y) = (numq x == numq y) && (denq x == denq y).
```

Assume $x \sim y$; this means $x = (a, b)$, $y = (c, d)$ and $ad = bc$. Since b is coprime with a, it divides d; similarly, d divides b. Since these two quantities are > 0, it follows $b = d$, thus $x = y$. This is expressed in the next lemma, using boolean equality. One deduces a function fracq (the equivalent of Qred) that converts a pair of integers into an element of $\mathbf{Q}$ by extracting gcds. This is the identity when applied to an element of $\mathbf{Q}$, and produces equal values when applied to equivalent pairs.

```
Lemma rat_eq x y : (x == y) = (numq x * denq y == numq y * denq x).
Fact valqK x : fracq (valq x) = x.
Fact fracq_eq x y : x.2 != 0 -> y.2 != 0 ->
  (fracq x == fracq y) = (x.1 * y.2 == y.1 * x.2).
```

All usual operations on $\mathbf{Q}$ are defined via fracq, and $\mathbf{Q}$ is a field. Now fracq becomes useless as 'fracq x' it is equal to $N(x)/D(x)$, where $N(x)$ and $D(x)$ are the numerator and denominator of x, coerced from $\mathbf{Z}$ to $\mathbf{Q}$. The set of all rational numbers with denominator 1 is canonically isomorphic to $\mathbf{Z}$ and is a subring; the set of these elements that are moreover positive is canonically isomorphic to $\mathbf{N}$ and is a sub-semiring. Define $a/b \leq c/d$ by $ad \leq bc$. This makes $\mathbf{Q}$ a real field. It is Archimedean (if $x = a/b$ and $n = |a| + 1$, then $x < n$ and $x \in \mathbf{N}$). For any number field R, there is a morphism $\mathbf{Q} \to R$ (it maps a/b to $f(a)/f(b)$ where $f : \mathbf{Z} \to R$ is the canonical morphism; note that $f(b)$ is not zero). Finally, rat is declared as a ring and a field for the ring and field tactics.

Consider now real numbers. The standard library defines them via a sequence of axioms. It is assumed that there is a set R, constants 0, 1, $+$, $-$, $*$, $<$ and u such that:

Rplus_comm: $\forall x, y,\ x + y = y + x$.
Rplus_assoc: $\forall x, y, z,\ (x + y) + z = x + (y + z)$.
Rplus_opp_r: $\forall x,\ x + (-x) = 0$.
Rplus_0_l: $\forall x,\ 0 + x = x$.
Rmult_comm: $\forall x, y,\ x * y = y * x$.
Rmult_assoc: $\forall x, y, z,\ (x * y) * z = x * (y * z)$.
Rinv_l: $\forall x$, if $x \neq 0$ then $(/x) * x = 1$.
Rmult_1_l: $\forall x,\ 1 * x = x$.
R1_neq_R0: $1 \neq 0$.
Rmult_plus_distr_l: $\forall x, y, z,\ x * (y + z) = x * y + x * z$.
total_order_T: $\forall x, y$, at least one of $x < y$, $x = y$ or $y > x$ holds.
Rlt_asym: $\forall x, y$, at least of one $x < y$ and $y < x$ is false.
Rlt_trans: $\forall x, y, z$, $x < y$ and $y < z$ implies $x < y$.
Rplus_lt_compat_l: $\forall x, y, z$, $x < y$ implies $z + x < z + y$.

Rmult_lt_compat_l: $\forall x, y, z$, $0 < z$ and $x < y$ implies $z * x < z * y$.
archimed: $\forall x$, $u(x) - 1 \leq x < u(x)$.
completeness: Every non empty bounded subset of R has a least upper bound.

Notes. That the ordering of $\mathbf{R}$ is total is expressed as: it is decidable that one of $x < y$, $x = y$ or $y > x$ holds; these conditions are mutually exclusive by antisymmetry. One can define by induction the canonical injection INR:$\mathbf{N} \to \mathbf{R}$, thus the canonical injection IZR:$\mathbf{Z} \to \mathbf{R}$. In the axiom archimed, the quantity $u(x)$ is in $\mathbf{Z}$, coerced to $\mathbf{R}$ (the axiom has $u(x) - x \leq 1$ instead of $u(x) - 1 \leq x$). This can be seen equivalent to the SSREFLECT definition: there is a natural number $v(x)$ such that $|x| < v(x)$. If n is the least natural satisfying this property, then $n - 1 \leq |x| < n$; this covers the case of positive numbers; defining $u(x)$ for negative numbers is easy.

The last axiom says the following. Let E be a real predicate (an object of type $\mathbf{R} \to$ Prop). An upper bound m is such that $E(x)$ implies $x \leq m$, and E is bounded if such an m exists. A least upper bound is an upper bound m such that for every upper bound b, we have $m \leq b$ (it is sometimes better to say: for every b such that $b < m$, there exists $x \in E$ such that $b < x$).

One deduces: every Cauchy sequence has a limit. In fact, if $(x_n)_{n \in \mathbf{N}}$ is Cauchy, it is bounded, so that one may consider $v_n = \sup_{n<k} x_k$. This sequence is decreasing, and has a lower bound, thus converges to v (in fact to $\inf v_n$). Clearly v is the limit of $(x_n)_n$. On the other hand, every real number x is the limit of a sequence of rational numbers: let $x_n = u(2^n x)/2^n$, by archimed we have $x_n - 1/2^n \leq x \leq u_n$, hence $|x - u_n| \leq 1/2^n$. The standard library provides many results about real numbers. For instance, the cosine function is continuous and has a sign change between $7/8$ and $7/4$. By the intermediate value theorem, there is a zero there; call it $\pi/2$. We have then the celebrated Machin Formula $\pi/4 = 4 \arctan(1/5) - \arctan(1/239)$.

```
Theorem R_complete :
  forall Un:nat -> R, Cauchy_crit Un -> { l:R | Un_cv Un l } .
Lemma Machin_4_5_239 : PI/4 = 4 * atan (/5) - atan(/239).
```

The SSREFLECT library, on the other hand, does not provide real numbers, but the MATHCOMP library[7] provides in *cauchyreals.v* a definition of real numbers as Cauchy sequences over some real field F. A real number x is a map $\mathbf{N} \to F$ satisfying the following property:

```
Lemma crealP (x : creal) : {asympt e : i j / `|x i - x j| < e}.
```

This has to be understood as: there is a function $m : F \to \mathbf{N}$ such that, if $\epsilon > 0$, for every i and j such that $m(\epsilon) \leq i$ and $m(\epsilon) \leq j$, it holds that $|x(i) - x(j)| \leq \epsilon$. Two real numbers x and y are called unequal if for some $\epsilon > 0$, we have $3\epsilon \leq |x(m_x(\epsilon)) - y(m_y(\epsilon))|$. This means: there is $\epsilon > 0$, such that, if i is big enough then $\epsilon \leq |x(i) - y(i)|$. If moreover $x(m_x(\epsilon)) + 3\epsilon \leq y(m_y(\epsilon))$, one writes $x < y$. This says that if i is big enough then $x(i) + \epsilon \leq y(i)$. If x and y are not unequal, then one

[7] In the *mathcomp/real_closed* directory on GitHub, accessible via the MATHCOMP web page

writes $x == y$; this is an equivalence relation (but not boolean); if $y < x$ is false, one writes $x \leq y$ (again this is not a boolean relation); it is a preorder, compatible with $x == y$. One has for instance:

```
Lemma eq_crealP (x y : creal) : {asympt e : i / `|x i - y i| < e} ->
  (x == y)%CR.
Lemma asympt_le (x y : creal) (le_xy : (x <= y)%CR) :
  {asympt e : i / x i < y i + e}.
```

The sum and product of two real numbers x and y are the sequences $i \mapsto x(i)+y(i)$ and $i \mapsto x(i) * y(i)$ (as every Cauchy sequence is bounded, the product of two real numbers is real). The operations (addition, multiplication, etc, including evaluation of polynomials with coefficients in F at a real number) are compatible with the equality.

This is a partial implementation of real numbers (for instance, associativity of addition is not shown), and could be completed; it it is however sufficient to implement the set of real algebraic numbers, as described in [Coh12].

1.5 The axioms of Gaia

Our work relies on an axiomatization of the theory of Zermelo-Fraenkel as described in [Gri10]; the axioms are the following:

```
Parameter Ro : forall x : Set, x -> Set.
Axiom R_inj : forall x : Set, injective (@Ro x).
Axiom extensionality : antisymmetric_r sub.
Parameter IM : forall (x : Set) (f : x -> Set), Set.
Axiom IM_P :  forall x f y, inc y (IM f) <-> exists a : x, f a = y.
Axiom p_or_not_p: forall P:Prop,  P \/ ~ P.
Parameter chooseT :  forall t : Type, (t -> Prop) -> inhabited t -> t.
Axiom chooseT_pr: forall (t : Type) (p : t -> Prop) (q : inhabited t),
   (exists x, p x) -> p (chooseT p q).
```

We first assume that there is some function Ro, such that, if x is a set, and a is of type x, then 'Ro x a' is a set; for any x, the function is injective; note that x is an implicit argument of Ro, since it is the type of a. The relation "there is some a such that 'Ro x a = t' " is denoted by 'inc t x' and interpreted as $t \in x$. The relation "for all t, $t \in x$ implies $t \in y$" is denoted 'sub x y' and interpreted as $x \subset y$. Our second axiom (Axiom of Extent) says: if $x \subset y$ and $y \subset x$ then $x = y$. We shall assume that, whenever x is a set, f a function on x (an object of type $x \to$ Set), there is some set I, such that $y \in I$ if and only if there is some a of type x such that $f(a) = y$. This implies the Axiom of Replacement: if f maps sets to sets and if X is a set, there is a set Y, denoted $\{f(x), x \in X\}$ such that $y \in Y$ if and only if there is $x \in X$ such that $y = f(x)$. It also implies a weaker form: if X is a set, P a predicate, there exists a set Y, denoted $\{x \in X, P(x)\}$ or 'Zo X P', such that $x \in Y$ if and only if $x \in X$ and $P(x)$ holds. If a, b, X are sets, we also deduce the existence of: the doubleton $\{a, b\}$ such that $x \in \{a, b\}$ if and only if $x = a$ or $x = b$, the union $\bigcup X$ such that $x \in \bigcup X$ if and only if there is y such that $x \in y$

and $y \in X$, and the power set $\mathfrak{P}(X)$, such that $x \in \mathfrak{P}(X)$ if and only if $x \subset X$. These sets are unique by the Axiom of Extent. In summary, our theory satisfies the axioms of Zermelo-Fraenkel.

Generally, in a theory of sets, one considers a collection of objects, called sets, such that an object is a set if and and only if it can be constructed via one of the five constructions given above. Since the empty collection satisfies the property, one assumes that there is at least one set, hence the empty set. Since V_ω (the collection of hereditarily finite sets) satisfies this property, one moreover assumes that there is at least one infinite set. Sometimes the Axiom of Foundation is assumed (but usual mathematics does not depend on it) or its converse (this requires a modification of the Axiom of Extent). Our approach is different: we assume that every Coq object of type Set is a set, and postulate the existence of some function Ro.

The Axiom of Infinity says: there is an infinite set. This can be interpreted as: there is a set X, and a function $f : X \to X$ which is injective and not surjective. For instance, nat is infinite: if $N(n)$ is '@Ro nat n', then for any n of type nat, we have $N(n) \in$ nat; if $g(n)$ is the pair $(N(n), N(n+1))$, then 'IM g' is the graph a function that maps $N(n)$ to $N(n+1)$; it is injective but not surjective[8]. Traditionally, the Axiom of Infinity says that there is a set X satisfying $I(X)$: "$\emptyset \in X$ and $\forall x, x \in X \Rightarrow x^+ \in X$" where x^+ is $x \cup \{x\}$ (One can show that the least subset Z of X that satisfies I is the set of all finite ordinals; the subset Y of X formed all ordinals of X is infinite as $x \mapsto x^+$ becomes injective). For instance, if f : nat $\to$ Set is defined by induction via $f(0) = \emptyset$ and $f(n+1) = f(n)^+$, then IMf satisfies I. One can define a set with an arbitrary cardinality n, for instance Empty_set is empty and bool has two elements. The current version of Gaia differs slightly from the description of [Gri10]: for instance emptyset is now a CoInductive variant of Empty_set, and the definition of the set with two elements (a copy of bool) has been removed. In fact, $\{x, y\}$ is now defined by 'IM (fun z => if z then x else y)' (note that z has type bool). Write 0 instead of $\emptyset$, 1 instead of $\{0, 0\}$ and 2 instead of $\{0, 1\}$. Then 2 is a set with exactly two elements, and is called "the canonical doubleton" (the Gaia names of these sets are C0, C1, C2).

We shall assume the Axiom of Choice in the form: whenever t is a type, q a proof that it is inhabited, and p a proposition over t, then there exists a set C such that, if p holds for some value, it holds for C. We shall almost exclusively consider the case where the type t is Set, inhabited by $\emptyset$. In this case, the axiom of choice says that there is a set $\tau_x(P)$, such that $(\exists x)P$ implies $P(\tau_x(P))$. In fact, Bourbaki uses $P(\tau_x(P))$ as the definition of $(\exists x)P$. Here $\tau_x(P)$ is some expression formed of P by binding the free variable x (thus is independent[9] of x) and $P(\tau_x(P))$ denotes the expression obtained from P by replacing every occurrence of x by $\tau_x(P)$; it is thus independent of x. The quantity $\tau_x(p(x))$ is denoted in Gaia by 'choose p', and

[8]For the formal definition of function and graph, see below; given a functional graph G and a superset X of its range, there is a unique function f with graph G and target X; in the example X is chosen as the domain of G, so that f has the same source and target; one can choose X as the range so that f becomes surjective.

[9]Bourbaki considers the notion of "assembly" which is a character string with links. Binding a variable consists in replacing it by a specific symbol and adding a link. There is no x in $\tau_x(P)$. This avoids the need for α-conversion, but an assembly cannot be represented in a computer.

the axiom of choice is: 'ex p → p(choose p)'. In the case where there is a unique element in some set E satisfying some property $p(x)$, this element is $\bigcup\{x \in E, p(x)\}$, denoted 'select p E', and the axiom of choice is not needed. Let E be a set, P be '$x \in E$', so that $(\exists x)P$ is equivalent to $E \neq \emptyset$; in this case, we denote by 'rep E' (in short $r(E)$) the quantity $\tau_x(P)$. We have then: $\forall E, E \neq \emptyset \Rightarrow r(E) \in E$. One easily deduces Zermelo's theorem: every set can be well-ordered (see [Gri10]).

Bourbaki manipulates only two kinds of objects: terms and relations, this corresponds in Coq to Set and Prop, and he allows quantification only over terms (i.e., sets). There is no equivalent of 'fun x => x <> emptyset' or nonempty of type Set → Prop. However "$x \neq \emptyset$" is a relation with a free variable; let's call it P. Similarly $x \cup y$ is a term with two free variables; let's call it Q. One may replace x in P by Q, and then y by $\emptyset$. The result is denoted $(\emptyset|y)(Q|x)P$; if the order of substitutions is exchanged one gets $((\emptyset|y)Q|x)P$; this is traditionally denoted $P(Q(\emptyset))$, but this is ambiguous. For this reason, Bourbaki introduces the notation $T\|x\|$ (instead of a double bar, there is a specific symbol, not used by anyone else): "if A is an assembly and we are interested particularly in a letter x, or two distinct letters x and y (which may or may not appear in A), we shall often write $A\|x\|$ or $A\|x, y\|$. In this case we write $A\|B\|$ instead of $(B|x)A$. We denote by $A\|B, C\|$ the assembly obtained by simultaneously replacing x by B and y by C whenever they occur in A (note that x and y may occur in B and in C)." So, $P\|Q\|x, \emptyset\|\|$ means $((\emptyset|y)Q|x)P$. Note that, if x appears in A, and B is a relation, then one of $A\|x\|$ and $A\|B\|$ is ill-typed; for this reason Bourbaki uses $A\|B\|$ only when B is a term. The Axiom of Excluded Middle says that a relation R is either true or false. One has to be careful when the relation depends on a parameter: in fact Bourbaki notes: "Let R be a relation; it is the same whether we state the theorem R or the theorem $(\forall x)R$, or the metamathematical rule: if T is any term then $(T|x)R$ is a theorem." If P is the relation introduced above, then, obviously, $(\forall x)P$ and $(\forall x)\neg P$ are false, so neither P nor $\neg P$ are theorems.

Equality is defined by Bourbaki via two axiom schemes. Scheme S6 says that if U and V are terms and $R\|x\|$ is a relation, then $(U = V) \Rightarrow (R\|U\| \Leftrightarrow R\|V\|)$ is an axiom; scheme S7 says that if P and Q are relations, then $((\forall x)(P \Leftrightarrow Q)) \Rightarrow (\tau_x(P) = \tau_x(Q))$ is an axiom. Quoting Bourbaki, "Intuitively, the scheme S6 means that if two objects are equal, they have the same properties. Scheme S7 is more remote from everyday intuition". One consequence of S7 is reflexivity of equality, one deduces symmetry and transitivity. Another consequence is Criterion C44: $(T = U) \Rightarrow (V\|T\| = V\|U\|)$, whenever T, U and V are terms. Proof. Assume $T = U$; by symmetry we have $U = T$. Substitute this relation in $V\|x\| = V\|U\|$; we get $V\|U\| = V\|U\| \Leftrightarrow V\|T\| = V\|U\|$. The conclusion follows by reflexivity. This is exactly how rewrite works in Coq, while scheme S6 is eq_ind. For this reason we shall use Coq's equality in whatever follows. What about S7? Let P and Q be equivalent propositions. Assume that we know that there is a set E, such that $P(x)$ implies $x \in E$, or such that $\exists x \in E, P(x)$. Then $\tau_x(P)$ can be replaced by $r(\{x \in E, P\})$. By the axiom of extent, $\{x \in E, P\} = \{x \in E, Q\}$ and S7 holds (this argument explains why S7 is part of the definition of choiceType in ssreflect). On the other hand, consider $\mathrm{Card}(X) = \tau_Z(\mathrm{Eq}(X, Z))$ introduced in section 1.3.1. Let X' be a set equipotent to X, so that $\mathrm{Eq}(X, Z) \Leftrightarrow \mathrm{Eq}(X', Z)$. Scheme S7 says

$\mathrm{Card}(X) = \mathrm{Card}(X')$, but since there is no set containing all cardinals, this relation is unprovable in Gaia. Similarly, one cannot prove that two order isomorphic sets have the same order type. The first problem is solved by using a different definition of the cardinal of a set (see below); the second one by introducing an axiom (see section 4.1).

1.6 Order relations and ordered sets

A *pair* (x, y) is a set z from which x and y can be recovered (via $x = \mathrm{pr}_1 z$ and $y = \mathrm{pr}_2 z$). It is defined by an Axiom in Bourbaki [Bou68], and as $\{\{x\}, \{x, y\}\}$ in the 1970 French Edition [Bou70]. In Gaia, the constuctor is named J and the two projectors are P and Q.

A *graph* G is a set such that every element z of G is a pair, say (x, y); the set of all those x, namely $\mathrm{pr}_1\langle G\rangle$, is called the domain of the graph, and the set of all y, namely $\mathrm{pr}_2\langle G\rangle$, is called the range. A graph is *functional* if for any x in the domain, there is a unique y, such that $(x, y) \in G$; this quantity is called the value of G at x, and denoted $G(x)$ or G_x or 'Vg G x'. The quantity 'Lg E f' denotes the functional graph with domain E that maps x to $f(x)$. A *sequence* is a functional graph whose domain is $\mathbf{N}$. It is generally denoted by $(x_n)_n$, sometimes by $n \mapsto x_n$. A finite sequence is a functional graph whose domain is a finite subinterval of $\mathbf{N}$; it may be denoted by $(x_n)_{n<k}$ if the domain is the set of integers $< k$. A *correspondence* $\Gamma = (S, T, G)$ is a triple such that G is a graph whose domain is a subset of S (the source) and whose range is a subset of T (the target). A *function* is a correspondence such that its graph G is functional and its source is the domain of G. The value of Γ at x, denoted $\Gamma(x)$ or 'Vf Γ x', is $G(x)$. Functions can be injective, surjective, bijective. If $f : E \to F$ is a bijection, then E and F are said to be *equipotent*; this is an equivalence relation, since compositions of bijections are bijections, and bijections have an inverse for composition. The notations 'f \cf g' and 'f \co g' stand for the composition of two functional graphs or functions. The quantity 'Lf f E F' denotes the function with source E, target F that maps x to $f(x)$ (it makes sense if $f(x) \in F$ whenever $x \in E$). The notation $f\langle X\rangle$ denotes the set $\{f(x), x \in X\}$ (that exists by the Axiom of Replacement), when f is a functional term; it is extended to functional graphs, correspondences, and functions.

An *order* is a graph G such that the relation $(x, y) \in G$, written $x \leq y$, is an order relation (reflexive, antisymmetric, and transitive); here reflexive means that $x \leq y$ implies $x \leq x$ and $y \leq y$; this can also be restated as: if E is the substrate of G (the union of the domain and range of G), then $x \in E$ implies $x \leq x$. One says: G is an order on E instead of the longer: G is an order with substrate E. Finally, for Bourbaki, an *ordering* is a correspondence $\Gamma = (E, E, G)$, such that G is an order on E (by abuse of language, G itself is called an ordering). The relation $x \leq y$ is equivalent to $y \in \Gamma\langle x\rangle$ (an abuse of notations for $y \in \Gamma\langle\{x\}\rangle$). One says that E is ordered by Γ (or equivalently, by the relation $x \leq y$). An ordered set E is an abuse of language: it consists of writing E instead of G or Γ. For instance, Bourbaki says: "whenever $\mathbf{N}$ is considered as an ordered set, it is always the ordering (called the *usual* ordering) defined in §3, no. 2, that is under consideration, unless the contrary is expressly stated".

In Coq, we do not have the right to use Γ, G, $\leq$ and E as synonyms. The first simplification consists in ignoring Γ: for us an order is a graph G, it is an order on $\mathrm{pr}_1\langle G\rangle$. If $\leq$ is a relation then $\{(x,y) \in E \times E, x \leq y\}$, often denoted $E_{\leq}$, is an order on E in some cases. The relation $(x,y) \in E_{\leq}$ may be denoted by $x \leq_E y$; it is equivalent to "$x \in E$ and $y \in E$ and $x \leq y$". For instance, the relation defined by Bourbaki in §3, no. 2, (which is an order relation, but not an order, since it has no graph), denoted below by $\leq_c$, defines an order on $\mathbf{N}$, denoted Nat_order, associated to the relation $x \leq_{\mathbf{N}} y$, which is equivalent to "$x \in \mathbf{N}$ and $y \in \mathbf{N}$ and $x \leq_c y$". If G is an order on F, $E \subset F$, and $x \leq y$ is "$(x,y) \in G$", then $E_{\leq}$ is always on order on E, called the order induced by G on E. For instance, in section 4.1, we shall meet BQ_ordering, the usual order on $\mathbf{Q}$, and BQps_ordering, the order induced on $\mathbf{Q}_+^*$. We shall also meet BQ_int01_ordering, the order induced on the interval $]0,1[$; the associated relation $x \leq_I y$ is equivalent to $0 <_q x \leq_q y <_q 1$.

1.7 Cardinal and ordinal numbers

One can define a function by transfinite induction. Let E be a well-ordered set, and $\mathcal{T}$ a procedure that constructs a set given another set (technically a λ-term of type Set → Set). For any function f defined on E, let $f^{(x)}$ be the restriction of f to $]\leftarrow, x[$ (this is the set of elements y of E such that $y < x$[10]). There is a unique surjective function f defined on E such that $f(x) = \mathcal{T}(f^{(x)})$. This is Criterion C60 of Bourbaki and transfinite_defined in Gaia (see an example on page 37).

The quantity $\mathcal{T}(g)$ mentioned above has to be defined whatever g, but the only values that matter are when g is a function whose source is a subset of E. In some cases, there is a set X such that $\mathcal{T}(g)$ belongs to X; in this case, the target of f is a subset of X, and it suffices that $\mathcal{T}$ be defined on $X^{\mathfrak{P}(E)}$. In the general case, some axiom is required for the existence of f (in our case, the axiom of choice). Example: if $\mathcal{T}(g)$ is the target of g, the target of the function f so defined is called the *ordinal* of the well-ordered set E, and denoted $\mathrm{ord}(E)$. It happens that no set contains all ordinals. Note that two order isomorphic sets have the same ordinal.

We say that a set x is an *ordinal* if every transitive subset of x is x or a member of x (y is transitive if $a \in b$, $b \in y$ implies $a \in y$). The relation "x and y are ordinals such that $x \subset y$" (denoted $x \leq_o y$ or $x \leq y$ in what follows) is a well-order relation on the collection of ordinals. The intersection of a non-empty set of ordinals is an ordinal (this is the least element of the set); the union is also an ordinal (the supremum of the set); the collection of all ordinals has, however, no upper bound. For every ordinal x, the quantity $x \cup \{x\}$ (denoted x^+ as previously) is the successor of x (the least ordinal y such that $x < y$). Every element of an ordinal is an ordinal, and the relation "$a \in x$, $b \in x$, $a \subset b$" between a and b makes x a well-ordered set, that will be denoted by $o(x)$. For every well-ordered set E, $\mathrm{ord}(E)$ is an ordinal, and $o(\mathrm{ord}(E))$ is order isomorphic to E; conversely, if x is an ordinal, then $\mathrm{ord}(o(x)) = x$. (The definition of "ordinal" used here comes from an

[10] It is traditionally denoted $]-\infty, x[$; if E has a least element e, then $]\leftarrow, x[= [e, x[$; similarly $[x, \rightarrow[$ corresponds to $[x, +\infty[$

Exercice of Bourbaki, it is equivalent to the definition of von Neumann; for details of implementation, see [Gri09]).

Every set x is the substrate of a well-order E, so that there exists an ordinal $o(E)$ equipotent to x; the least such ordinal is called the *cardinal* of x and denoted by $\mathrm{card}(x)$. A cardinal is an ordinal x of the form $\mathrm{card}(y)$, i.e., it is such that every ordinal z equipotent to x satisfies $x \subset z$.

```
Definition cardinal x :=
  (least_ordinal (equipotent x) (ordinal (worder_of x))).
Definition cardinalp x:=
 ordinalp x /\ (forall z, ordinalp z -> x \Eq z -> sub x z).
```

By definition, $\mathrm{card}(X)$ is equipotent to X, and if X and Y are equipotent, they have the same cardinal. This means that our notion of cardinal behaves the same as that of Bourbaki. The cardinal of $\emptyset$ is $\emptyset$, denoted 0, the cardinal of any singleton is $\{0\}$, denoted 1. The sum and product of a family of cardinals is defined as the cardinal of the disjoint union or cartesian product, and x^y is the cardinal of the set of functional graphs $y \to x$ (this is also the cardinal of the set of functions $y \to x$).

The relation "x and y are cardinals such that $x \subset y$" (denoted $x \leq_c y$ or $x \leq y$) is the same as "x and y are cardinals such that $x \leq_o y$", thus is a well-order relation on the collection of cardinals, and every nonempty set of cardinals has a least element and a least upper bound. If E is a set, F is a subset of E, and f an injection $E \to F$, 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). One deduces the Cantor-Bernstein Theorem: if there is an injection $E \to F$ and an injection $F \to E$, then there is a bijection $E \to F$. Moreover, $\mathrm{card}(X) \leq \mathrm{card}(Y)$ if and only if there is an injection $X \to Y$. This means that $\leq_c$ coincides with the Bourbaki definition.

We say that an ordinal x is *finite* if it is not equipotent to x^+; we say that a set is finite if its cardinal is a finite ordinal. Note that zero (the empty set) is finite, and x is finite if and only if its successor is finite. If x is finite and $y \leq x$, then y is finite. The Cantor-Bernstein theorem says that if x is a finite ordinal, its successor is a cardinal (so that x itself is a cardinal). So, a cardinal is finite if and only if $x \neq x+1$. This is the Bourbaki definition of an *integer*. As mentioned above, nat is an infinite set, so that there exists at least one infinite ordinal, thus a least infinite ordinal ω.

```
Definition infinite_o u := u \Eq (osucc u).
Definition omega0 := least_ordinal infinite_o (cardinal nat).
```

It is immediate that $x <_o \omega$ if and only if x is a finite ordinal. Thus ω is the set of all finite ordinals. Since a finite ordinal is a cardinal, ω is also the set of all finite cardinals. It contains zero and is stable by successor; note that every ordinal, that contains zero and is stable by successor, must be infinite, thus is a superset of ω.

2. THE SET OF NATURAL NUMBERS N

In what follows, we rename ω as $\mathbf{N}$, and say that an element of $\mathbf{N}$ is a natural number. Remember that cardinal comparison coincides with ordinal comparison on $\mathbf{N}$, so that $x \leq_{\mathbf{N}} y$ (see above) makes $\mathbf{N}$ a well-ordered set. The quantity $\mathrm{card}(x^+)$, denoted '`csucc x`', is the successor of x for this ordering; in fact it is the ordinal successor of x when $x \in \mathbf{N}$, it is $x+1$ whenever x is a cardinal, thus is x when x is an infinite cardinal.

```
Definition Nat := omega0.
Definition natp x := inc x Nat.
```

Since $\mathbf{N}$ is an ordinal, for every non-empty subset X of $\mathbf{N}$, the quantity $\bigcap X$ is the least element of X. Since ω is the least ordinal that is neither zero nor a successor (i.e., is a limit ordinal), it follows that a natural number is zero or a successor (in this case, it is the successor of its *predecessor*, $x-1$) Assume that P is a property, satisfied by at least one integer; if the least integer satisfying P is not zero, it is the successor of some x such that P fails for x, and holds for $x+1$.

```
Lemma sub_natI_prop X (x := intersection X): sub X Nat ->
  [/\ natp x, forall t, t <c x -> ~ (inc t X) &
      nonempty X -> inc x X].
Lemma wleast_int_prop (prop:property):
  (exists2 x, natp x & prop x) ->
  prop \0c \/ (exists x, [/\ natp x, prop (csucc x) & ~ prop x]).
```

Let's state some notation conventions: we cannot use the same notation to denote the zero, addition, comparison of $\mathbf{N}$, $\mathbf{Z}$, etc, since these objects have the same type, so we add a suffix: `s` for set operations, `o` for ordinal, `c` for cardinal, `z`, `q`, `r`, for $\mathbf{Z}$, $\mathbf{Q}$ and $\mathbf{R}$. Thus, in the first lemma above, '`t <c x`' is cardinal comparison, in the second lemma `\0c` is the cardinal zero (i.e., the empty set). The quantities `\2o` and `\2c` denote 2 as an ordinal or a cardinal (these two quantities are equal to the canonical doubleton `C2`). The ordinal successor of x, '`osucc x`' is $x \cup \{x\}$, thus '`x +s1 x`'. The quantity '`Nat -s1 \0c`', denoted $\mathbf{N}^*$, is the set of all non-zero integers. Note that '`x %/c y`' and '`x %%z y`' are the quotient and remainder in the Euclidean division of x by y (on $\mathbf{N}$ and $\mathbf{Z}$ respectively), while '`x /q y`' is the quotient on $\mathbf{Q}$. Having notations that start with a minus sign or a slash is not really a good idea, as '`rewrite -cprodA /ratp`' has to be replaced by '`rewrite - cprodA / ratp`'.

Other naming conventions: we use the capital letters A, B, C, etc, in theorem names, similarly to SSREFLECT (A: associative, C: commutative, D: addition, B: subtraction, I: intersection, U: union, X: cartesian product, P: powerset). Lemmas of the form: $P \Leftrightarrow Q$ finish with a P, and B may stand for Bourbaki.

2.1 Induction properties

The most important difference between Bourbaki and Gaia concerns the principle of induction. Since Bourbaki uses first order logic, he cannot write: $\forall R, H \Rightarrow C$. So he says: in any theory (in which "integer" makes sense), for any formula R, if we

have a proof of H, then we can prove C. Since R may depend on some parameters, he adds some clues such as n is not a constant of the theory or u is a letter, or sometimes omits them.

C61. *Let $R\|n\|$ be a relation in a theory $\mathcal{T}$ (where n is not a constant of $\mathcal{T}$). Suppose that the relation*

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

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

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

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

The proof is by contradiction: if the conclusion were false, there would exist an integer not satisfying R, thus a least such one, either zero or a successor, both alternatives being absurd. We show here the Coq statement and its proof:

```
Lemma Nat_induction (r:property):
  (r \0c) -> (forall n, natp n -> r n -> r (csucc n)) ->
  (forall n, natp n -> r n).
Proof.
move=> r0 ri n nN.
case: (p_or_not_p (r n)) => // nrn.
have pa: (exists2 x, natp x & ~ r x) by exists n.
case:(wleast_int_prop pa) => // [] [x [xN nsr /excluded_middle rx]].
by case: nsr; apply: ri.
Qed.
```

C62. *Let u be a letter and let $T\|u\|$ be a term. Then there exists a set U and a mapping f of $\mathbf{N}$ onto U such that for each integer n we have $f(n) = T\|f^{(n)}\|$, where $f^{(n)}$ denotes the mapping of $[0,n[$ onto $f([0,n[)$ which agrees with f on $[0,n[$. Moreover the set U and the mapping f are uniquely determined by this condition.*

This criterion is the specialization to $\mathbf{N}$ of criterion C60 (introduced in section 1.7). It uses the interval $I_n = [0,n[$, while C60 considers $]\leftarrow,n[$. These sets are equal (because 0 is the least element of $\mathbf{N}$) to the set of all x such that $x <_{\mathbf{N}} n$. With the Gaia definition of integers, this is the set of all x such that $x <_o n$, hence n. Whenever possible, we use n instead of I_n in the code.

An example of a function f such that $f(n)$ depends on all values $f(i)$ for $i < n$ is EPfun_aux page 37. A more complicated example is the following. Let E be a set, and define a *formula* over E to be either $a = b$, $a \in b$ (where a and b are in E) or $\neg F$, $F \vee F'$, $\forall x, F$ (where F and F' are formulas, $x \in E$). This can be trivially defined in Coq via Inductive, and defining the set of free variables (not shadowed by a $\forall$) is trivial as well. In Bourbaki, one can define by induction a set X_n such that a formula of X_{n+1} is either a formula of X_n or satisfies the previous property (with F, F' in X_n), then consider the union of these X_n. Each formula has a depth (the least n such that the formula is in X_n), and one can show properties by induction on the depth. Defining a function (for instance the set of free variables) is more complicated (one defines by induction a function f_n whose source is X_n, then takes the common extension in order to get a function on the union).

Note: Bourbaki makes a difference between a "function" (possibly defined on $\mathbf{N}$) and a "mapping of $\mathbf{N}$ into U" (in French: "fonction" and "application de $\mathbf{N}$ dans U"); if "into" is replaced by "onto" (in French "dans" by "sur"), then the function is assumed surjective. The criterion could be simplified if the word "mapping" were replaced by "function", in the form: there exists a unique surjective function defined on $\mathbf{N}$ such that $f(n) = T\|f^{(n)}\|$. This formulation avoids the need to give a name to the target of f and claim its existence and uniqueness (idem for C63, where a different name has been used).

C63. *Let $S\|v\|$ and a be two terms. Then there exists a set V and a mapping f of $\mathbf{N}$ onto V such that $f(0) = a$ and $f(n) = S\|f(n-1)\|$ for each integer ≥ 1. Moreover, the set V and the mapping f are uniquely determined by these conditions.*

Usually, the second condition is written $f(n+1) = S\|f(n)\|$ for all n, and the existence part is what really matters. So we prove:

```
Lemma induction_defined_pr s a (f :=  induction_defined s a):
    [/\ source f = Nat,  surjection f, Vf f \0c  = a &
     forall n, natp n -> Vf f (csucc n) = s (Vf f n)].
```

One can also impose: $f(0) = a$ and $f(n+1) = s(n, f(n))$. The Bourbaki approach (using only induction_defined) is very complicated, but the result follows directly from C60. In the lemmas that follow, we consider a lambda term rather that an mapping.

```
Lemma induction_term0 s a: induction_term s a \0c = a.
Lemma induction_terms s a n:
  natp n -> induction_term s a (csucc n) = s n (induction_term s a n).
```

2.2 Arithmetic properties

Consider a family of sets X, i.e., a functional graph; it is characterized by its domain I and its evaluation map $i \mapsto X_i$. Note that every set X can be considered as a family by taking as domain the set of all i such that for at least one j we have $(i, j) \in X$ and as evaluation X_i some j satisfying this condition (an example is csum_Cn below). The disjoint union S of the family is the set of all (x, i), where $i \in I$ and $x \in X_i$. The cardinal of S are called the cardinal sum of the family and denoted 'csum X'; there is a variant csumb that takes two arguments: the domain and the evaluation function; what is traditionally written $\sum_{i\in I} X_i$ corresponds to 'csumb I (fun i => Vg X i)' or 'csumb I (Vg X)'.

The quantity $\sum_{i\in I} X_i$ remains unchanged if each X_i is replaced by an equipotent set (for instance its cardinal). Commutativity of addition means that if f is any bijection $J \to I$, then $\sum_{i\in I} a_i = \sum_{j\in J} a_{f(j)}$. Associativity of addition means: if $(I_k)_{k\in K}$ is a functional graph formed of mutually disjoint sets whose union is I and if $s_k = \sum_{i\in I_k} a_i$, then $\sum_{i\in I} a_i = \sum_{k\in K} s_k$. It is important (for the ease of proofs) to notice that we do not require a be a functional graph, nor a_i to be a cardinal.

```
Theorem csum_An f g:
  partition_w_fam g (domain f) ->
```

```
  csum f = csumb (domain g) (fun l => csumb (Vg g l) (Vg f)).
Theorem csum_Cn X f:
  target f = domain X -> bijection f ->
  csum X = csum (X \cf (graph f)).
```

Given two sets a, b, one can consider the family indexed by a set of two elements; the sum of this family is denoted $a + b$. Associativity and commutativity take a more familiar form:

```
Lemma csumC a b: a +c b = b +c a.
Lemma csumA a b c: a +c (b +c c) = (a +c b) +c c.
```

The cartesian product $\prod_{i\in I} X_i$ is the set of all functional graphs f defined on I such that $f(i) \in X_i$ whenever $i \in I$. Its cardinal is called the cardinal product of the family and is denoted here cprod or cprodb; in order to avoid ambiguities (for instance in equation (1), the product in the definition of I is the cartesian product, the two other products are cardinal products), Bourbaki uses a big P in Chapter 3, section 3, the usual notation later $\prod$ on. As in the case of a sum, the cardinal product remains unchanged if a factor X_i is replaced by an equipotent set, the factor need not be a cardinal. The product is associative, commutative; distributivity over sums is expressed as follows: let $((a_{j,i})_{i\in J_j})_{j\in L}$ be a double family, then

$$\prod_{j\in L} \Big(\sum_{i\in J_j} a_{j,i} \Big) = \sum_{f\in I} \Big(\prod_{j\in L} a_{j,f(j)} \Big) \qquad \text{where} \quad I = \prod_{j\in L} J_j. \tag{1}$$

As in the case of a sum, formulas are simpler when the family is reduced to two elements:

```
Theorem cprodDn f:
  cprodb (domain f) (fun l => csum (Vg f l)) =
  csumb (productf (domain f) (fun l => (domain (Vg f l))))
    (fun g => (cprodb (domain f) (fun l => Vg (Vg f l) (Vg g l)))).
Lemma cprodDr a b c:
  (b +c c) *c a = (b *c a) +c (c *c a).
Lemma cprodC a b: a *c b = b *c a.
Lemma csumA a b c:  a +c (b +c c) = (a +c b) +c c.
```

We define a^b as the cardinal of the set of functional graphs (or functions) $b \to a$. This is equal to the product of a family (indexed by b) of cardinals all equal to a, in the same manner as $a \cdot b$ is the sum of this family (moreover b is the sum of the family of cardinals all equal to one). A product is zero if and only if one factor is zero[11].

```
Lemma csum0r a: cardinalp a ->  a +c \0c = a.
Lemma cprod1r a: cardinalp a ->  a *c \1c = a.
Lemma cprod2_nz a b: a <> \0c  -> b <> \0c ->  a *c b <> \0c.
Lemma cprod0r a: a *c \0c = \0c.
Lemma cpow1x a: \1c ^c a = \1c.
```

[11] The expressions $a + 0$ and $a * 1$, defined whatever a are requal to card(a), hence to a when a is a cardinal

The operations defined above are compatible with comparison. First note that if $f : A \to B$ is injective, then $\mathrm{card}(A) \leq \mathrm{card}(B)$ (if $\mathrm{card}(B) \leq \mathrm{card}(A)$, we have an injection $B \to A$, and by the Cantor-Bernstein theorem A and B are equipotent). In particular, if A is a subset of B, we have $\mathrm{card}(A) \leq \mathrm{card}(B)$. To proceed further, we define $a - b$ to be the cardinal of the complement c of b in a. If $b \leq a$ (so that a and b are cardinals, $b \subset a$), then $\{b, c\}$ forms a partition of a; so that $a = b + (a - b)$. It follows that $b \leq a$ if and only if there is c such that $a = b + c$. Let's note that $\mathrm{card}(\mathfrak{P}(X)) = 2^X$ (for each $A \subset X$, we consider the characteristic function ϕ_A whose value is one in A and zero on the complement). There is no surjection $f : X \to \mathfrak{P}(X)$ (consider y such that $f(y)$ is the set of all z such that $f(z) \notin z$). One deduces $a < 2^a$, so that there is no set containing all cardinals.

```
Theorem card_setP X: cardinal (powerset X) = \2c ^c X.
Theorem cantor a: cardinalp a -> a <c (\2c ^c  a).
Lemma cantor_bis: ~ (exists a, forall x, cardinalp x -> inc x a).
```

Let's consider now the set $\mathbf{N}$ of finite cardinals. Since $x+1$ is the cardinal of x^+, it follows that a cardinal x is finite if and only if $x \neq x+1$. In this case $x+1 = x^+$. The relations $a + (b+1) = (a+b)+1$, $a \cdot (b+1) = a \cdot b + a$ and $a^{b+1} = a^b \cdot a$ show, by induction on b, that $\mathbf{N}$ is stable by addition, multiplication and exponentiation. Since $a - b \leq a$, it follows that $a - b$ is an integer whenever a is an integer. One can define (by induction on `nat`) a mapping $\mathtt{nat} \to \mathbf{N}$; it is injective (by induction on `nat`) and surjective (by induction on $\mathbf{N}$). Obviously, this function is compatible with the operations and ordering.

```
Fixpoint nat_to_B (n:nat) :=
  if n is m.+1 then csucc (nat_to_B m) else \0c.
Lemma nat_to_B_injective: injective nat_to_B.
Lemma nat_to_B_surjective x: natp x -> exists n, x = nat_to_B n.
```

As a consequence, every property valid on `nat`, and proved in the standard library of Coq, has a analogue on $\mathbf{N}$ with a trivial proof. For instance, if a, b, c are integers, $a < b$ implies $a + c < b + c$. Our proof is the following: since $a < b$ there is a non-zero integer d such that $b = a + d$ and d has the form $e + 1$; now $a + c < (a + c) + 1 \leq ((a + c) + 1) + e = b + c$. This property holds even when a and b are infinite (as $x + c = x$ whenever x is infinite and c finite), but fails if c is infinite and big enough (if c is infinite and $c \geq \max(a, b)$, then $a + c = b + c = c$).

If A and B are two finite sets with the same cardinal, then a function $f : A \to B$ is bijective if and only if it is injective (or surjective). The principle of induction on $\mathbf{N}$ can be extended as follows: let P be a property, true for the empty set, such that $P(A)$ implies $P(A \cup \{a\})$. Then P holds for every finite set. Assume that $P(\emptyset)$ is replaced by: P holds for every singleton; then P holds for every non-empty finite set. Example: let A be a totally ordered set; then every non-empty finite subset of A has a least and a greatest element. As a consequence, if A is finite, it is well-ordered.

2.3 Expansion to base b

If a and b are integers, b non-zero, there exists p such that $a < bp$. The least such integer is non-zero, thus of the form $q+1$. Moreover, there exists r such that $a = bq + r$, $0 \leq r < b$. The two quantities q and r, uniquely defined by this pair of relations, are called the quotient and remainder in the Euclidean division of a by b.

Let's notice that a finite sum (or product) of integers is an integer (proof by induction on the domain). In particular, if $f(k)$ is an integer for $k < n$, then $\sum_{k<n} f(k)$ and $\prod_{k<n} f(k)$ are integers. If S_n is the sum then $S_{n+1} = S_n + f(n)$; one could prove by induction on n via this formula that S_n is an integer. One also has $S_{n+1} = f(0) + \sum_{i<n} f(i+1)$. Consider for example

$$E_{nb}(a) = \sum_{i<n} a_i b^i$$

where $(a_i)_{i<n}$ is a sequence of integers and $b \geq 2$ [If $b = 0$, the sum is a_0 for $n > 0$, if $b = 1$, the sum is $\sum a_i$, we shall omit the index b in what follows]. We have

$$E_{n+1}(a) = E_n(a) + a_n b^n = a_0 + bE_n(a') \tag{2}$$

where a' is the sequence $i \mapsto a_{i+1}$. In what follows, we shall assume implicitly $a_i < b$ for each i. We deduce: a_n and $E_n(a)$ are the quotient and remainder in the division of $E_{n+1}(a)$ by b^n; $E_n(a')$ and a_0 are the quotient and remainder of the division of $E_{n+1}(a)$ by b.

It is easy to deduce from these considerations that, for every integer A, if $A < b^n$, there is a unique sequence $(a_i)_{i<n}$ such that $A = E_n(a)$; it is called the expansion to base b with n digits. If $n < m$ and A has two expansions with n and m digits, then $a_i = 0$ for $n \leq i$ for the second expansion; thus, there is a unique expansion (called normal) with $n = 0$ or $a_{n-1} \neq 0$. Let C be another number and c its expansion with m digits. We have now the following criterion for $A < C$: Either there is k such that $n \leq k < m$ and c_k is non-zero, or there are two indices k, p such $k < p \leq \min(n, m)$ and $a_k < c_k$, $a_i = c_i$ for $k < i < p$; moreover a_i and c_i are zero for $i \geq p$. In particular, if the expansions are normal, then $n < m$ implies $A < C$.

Note: assume $n = m$; the criterion simplifies to: there is an index k, such $k < n$, $a_k < c_k$, $a_i = c_i$ for $k < i < n$. Bourbaki states this in Proposition 8 of §5.7 as: "let E_k be the lexicographic product of the family $(J_h)_{0 \leq h \leq k-1}$ of intervals all identical with $[0, b-1]$, then f_k is an isomorphism of the ordered set E_k onto the interval $[0, b^k - 1]$." Here, all intervals (including $[0, h-1]$) are ordered by $\leq$; since the leading digits have to be compared first, this leading digit corresponds to $h = 0$; for this reason Bourbaki uses $f_k(r) = \sum_{h=0}^{k-1} r_h b^{k-h-1}$. The condition for the expansion to be normal simplifies to $r_0 \neq 0$. [Note that k has to be non-zero, so that zero is the only integer that has no expansion in the Bourbaki sense; our approach is a bit simpler and has no exception.] [Note also: in order to prove that $\mathbf{N}$ and $\mathbf{N} \times \mathbf{N}$ are equipotent, Bourbaki uses the fact that every integer can uniquely be written as an infinite sum $\sum_h a_h b^h$, where $a_h = 0$ for $h \geq k$.]

```
Lemma expansion_prop15 f g b n:
```

```
expansion f b n -> expansion g b n ->
( (expansion_value f b) <c  (expansion_value g b)
  <->  exists k,
       [/\ k <c n, (Vg f k) <c  (Vg g k) &
        (forall i, k <c i ->  i <c n -> Vg f i = Vg g i)]).
```

Let $C(A)$ be the sum $\sum_{i<n} a_i$ where $A = E_n(a)$. Assume that A is not a digit (so that $A > b$ and there is i such that $0 < i < n$ and $a_i \neq 0$). One gets $C(A) < A$. Let $C^k(A)$ be the k-th iteration of C. This sequence is eventually constant, let's denote the limit value by $C'(A)$. The two quantities A and $C(A)$ are equal modulo $b-1$ (since b^n is one modulo $b-1$). Thus A and $C'(A)$ are the same modulo $b-1$. One can be more precise: either $A = 0$ case where $C'(A) = A$, or $A \neq 0$, case where $C'(A)$ is between 1 and $b-1$; if y is the remainder in the division of A by $b-1$, then either $y = 0$ (division is exact) and $C'(A) = b-1$, or $C'(A) = y$. We show here what happens when $b = 10$.

```
Lemma divisibiliy_by_nine f k: expansion_ten f k ->
  let g:= (Lg (domain f) (fun i=> (Vg f i) *c (\10c ^c i))) in
  eqmod \9c (csum g) (csum f).
Lemma eqmod_contraction_rep9 a
   (x := contraction_rep \10c a) (y := a %%c \9c) :
   natp a ->
   [/\ eqmod \9c a x,
    (a = \0c -> x = \0c) &
    (a <> \0c -> (y = \0c -> x = \9c) /\ (y <> \0c -> x = y))].
```

A special case is expansion to base two. Here $a_i < b$ says $a_i = 0$ or $a_i = 1$. The number of digits in a normal expansion of n is called the base two logarithm of n. If n is non-zero, this is a non-zero integer such that $2^{\ln(n)-1} \leq n < 2^{\ln(n)}$. Moreover $a_{\ln(n)-1} = 1$. We define the base-two reverse $r(n)$ of n to be $E_l(a')$, where the sequence a is the base two expansion of n, l its length, and $a'_i = a_{l-1-i}$. The relation $a_{l-1} = 1$ says $a'_0 = 1$, so that $r(n)$ is odd (unless $n = 0$). We have $r(r(n)) = n$ when n is odd. This can be restated as: $r(r(r(n))) = r(n)$, whatever n.

```
Definition base_two_reverse n :=
   let F := the_expansion \2c n in
   let p := cardinal (domain F) in
   expansion_value (Lg p (fun z => (Vg F (p -c (csucc z))))) \2c.
Lemma base2r_oddK_bis n (r := base_two_reverse) :
  natp n -> r (r (r n)) = r n.
```

2.4 Combinatorial Analysis

A key relation is (shepherd's principle in France): if $f : E \to F$ is any mapping such that the inverse image of any singleton has the same cardinal c, then the cardinal of E is c times the cardinal of F. [Note: For some reason, Bourbaki assumes f surjective; if x is in F but not in the image, the cardinal of the inverse image of $\{x\}$ is zero, thus $c = 0$, thus no element is in the image, thus E is empty and the result is still true].

A first consequence is: the number A_{nm} of injections of a set with m elements into a set of n elements satisfies $A_{n,m+1} = A_{n,m} \cdot (n-m)$ thus $A_{nm} = n!/(n-m)!$, where $n!$ is $\prod_{i<n}(i+1)$. Taking $n = m$ says that $n!$ is the number of permutations of a set with n elements. [Proof: if E has $m+1$ elements, a is one of those, $E' = E - \{a\}$, a function $f : E \to F$ is characterized by its value at a, and the restriction f' to E'. In order for the function f to be injective, we need f' to be injective and the value $f(a)$ not in the range of the restriction. The set in which $f(a)$ can be chosen depends on f', but not its cardinal, which is $n-m$].

Let $(p_i)_{i<h}$ be a finite sequence of integers, $n = \sum p_i$, E a set with n elements; the number of ways to write E as the union of disjoint sets E_i with cardinal p_i is $n!/(\prod_{i<h} p_i!)$. Proof. Let $I = [0, n[$, and consider a partition I_k of I formed of sets with p_i elements (I_k contains the p_k smallest elements not in I_j for $j < k$). Let $f : I \to E$ be a bijection, g a permutation of I, E_k the image by $f \circ g$ of I_k, $G(g)$ the sequence of these E_k. The family $G(g)$ is a solution, every solution has this form; the relation $G(g) = G(g')$ is equivalent to say that $g^{-1} \circ g'$, restricted to each I_k, is a permutation (size of the proof: 450 lines of script).

```
Definition partition_with_pi_elements p E f :=
  [/\ domain f = domain p,
  (forall i, inc i (domain p) -> cardinal (Vg f i) = Vg p i) &
    partition_w_fam f E].

Definition partitions_pi p E :=
  Zo (gfunctions (domain p) (powerset E)) (partition_with_pi_elements p E).

Theorem number_of_partitions_bis p E:
  finite_int_fam p -> csum p = cardinal E ->
  cardinal (partitions_pi p E) =
  (factorial (cardinal E)) %/c
           (cprodb (domain p) (fun z => factorial (Vg p z))).
```

Using the same techniques, one deduces the following (250 lines of script).

$$\sum_{p_1+p_2+\cdots+p_k=n} \frac{n!}{p_1!\cdots p_k!} a_1^{p_1} \ldots a_k^{p_k} = (a_1 + a_2 + \cdots + a_k)^n. \tag{3}$$

```
Lemma sum_of_gen_binom0 E n a:
  natp n ->  cardinal E = n -> finite_int_fam a ->
  (csum a) ^c n =
     csumb (graphs_sum_eq (domain a) n)
       (fun p =>
         (cardinal (partitions_pi p E)) *c
            (cprodb (domain a) (fun i=> ((Vg a i) ^c (Vg p i))))).
```

Application: the number of subsets with p elements in a set with n elements is $n!/(p!(n-p)!) = \binom{n}{p}$. One deduces that, if $p \leq n$, then $p!(n-p)!$ divides $n!$, and the quotient could be used to define the binomial coefficient. We give here an inductive definition and a proof of correctness, as an example of how to proceed when the induction principle (induction_defined or induction_term) cannot

be used directly. So, we define a functional graph f_n with domain $\mathbf{N}$ such that $f_{n+1}(m) = f_n(m) + f_n(m-1)$, and apply it to m. The anonymous function has an unused parameter (the value of n), the second parameter being f_n. The expression '`variant a b c d`', short for '`Yo (d = a) b c`', means: if $d = a$ then b else c, it is used to say that $f_0(0) = 1$ and $f_{n+1}(0) = 0$.

```
Definition binom n m :=
  Vg (induction_term
      (fun _ T: Set => Lg Nat (fun z => variant \0c \1c
          (Vg T z +c Vg T (cpred z)) z))
       (Lg Nat (variant \0c \1c \0c))
       n) m.
Lemma binom_alt_pr n m: natp n -> natp m ->
  (binom n m) *c  (factorial m) *c (factorial (n -c m)) =
  Yo (m <=c n) (factorial n) \0c.
```

Our library contains many other formulas, using similar or different proof techniques. For instance (3) holds in any ring (the only condition being that $a_i a_j = a_j a_i$) and can be shown by induction[12]. The number of strictly increasing functions $E \to F$ (where both sets are finite and totally ordered, of cardinal n and m) is $\binom{m}{n}$ (the function is uniquely determined by its image, a subset of n elements of F). One deduces the number of increasing functions: we may reduce the case to $E = I_n$ and $F = I_m$ (the ordered set of integers $< n$ or $< m$); if $g(i) = f(i) + i$, then f is increasing if and only if g is strictly increasing, and the range of g is I_{n+m-1}. Let $h(i) = f(i+1) - f(i)$; then $f(i)$ is the sum of the $h(j)$. So counting the number of (strictly) increasing functions is the same as counting the number of functions h such that $\sum h_i = m$ (or $\sum h_i \leq m$). This is the number of monomials with n variables of total degree m or $\leq m$; these two numbers are related and can be computed by induction on n.

Many of the formulas can be expressed and proved using the structure of finite sets provided by the SSREFLECT library (thus do not require any axiom). For instance the file *binomial.v* provides three lemmas that give the number of injections, the number of increasing functions (in fact, sorted lists) and the number of functions such that $\sum h_i \leq m$.

```
Lemma card_inj_ffuns D T :
  #|[set f : {ffun D -> T} | injectiveb f]| = #|T| ^_ #|D|.
Lemma card_sorted_tuples m n :
  #|[set t : m.-tuple 'I_n.+1 | sorted leq (map val t)]| = 'C(m + n, m).
Lemma card_partial_ord_partitions m n :
  #|[set t : m.-tuple 'I_n.+1 | \sum_(i <- t) i <= n]| = 'C(m + n, m).
```

One of our objectives is to solve in COQ all exercices of Bourbaki, and if possible, directly in SSREFLECT. This is sometimes a challenge: for instance, let $S_{n,p}$ be the number of surjections $I_n \to I_p$. There is an explicit formula, but it requires negative numbers. On the other hand, there is an implicit formula relating $S_{n,i}$, the binomial

[12] rings are not implemented in Gaia.

coefficients and the number of functions $I_n \to I_p$ (every function is a surjection on its image). There is also a simple recurrence relation, $S_{n,p} = p(S_{n-1,p} + S_{n-1,p-1})$ (proved in Gaia), and if $P_{n,p}$ is the number of partitions of I_n into p parts, then $S_{n,p} = p!P_{n,p}$. The quantity P is known as Stirling numbers of the second kind; we have defined it in SSREFLECT, and shown that it is the number of partitions, but we have not yet managed to compute the number of surjections (work in progress).

3. THE SET OF RATIONAL INTEGERS Z

We define $\mathbf{Z}$ as the disjoint union of $\mathbf{N}^*$ and $\mathbf{N}$. An element of this union is a pair (a, b) where either $b = \mathtt{C0}$ and $a \in \mathbf{N}^*$, or $b = \mathtt{C1}$ and $a \in \mathbf{N}$. We denote by $\phi_+(a)$ the pair $(a, \mathtt{C1})$, and by $\phi_-(a)$ the pair $(a, \mathtt{C0})$ when $a \neq 0$, extended by $\phi_-(0) = \phi_+(0)$. The image of ϕ_+ will be denoted by $\mathbf{Z}_+$, the image of ϕ_- will be denoted by $\mathbf{Z}_-$ (these two sets contain zero). Removing zero yields $\mathbf{Z}_+^*$ and $\mathbf{Z}_-^*$.

```
Definition Nats:= Nat -s1 \0c.
Definition BZ := canonical_du2 Nats Nat.
Definition BZ_of_nat x := J x C1.
Definition BZm_of_nat x := Yo (x = \0c) \0z (J x C0).
Notation BZ_val := P (only parsing).
Notation BZ_sg := Q (only parsing).
Definition intp x := inc x BZ.
```

We show here simple functions: absolute value, sign and opposite. Note that if x is the pair (a, b), then the absolute value is a (considered in $\mathbf{Z}$) and the sign is b (converted into $+1$ or -1; except that the sign of zero is zero).

```
Definition BZabs x := BZ_of_nat (BZ_val x).
Definition BZsign x:= Yo (BZ_val x = \0c) \0z (Yo (BZ_sg x = C1) \1z \1mz).
Definition BZopp x:=
  Yo (BZ_sg x = C0) (BZ_of_nat (BZ_val x)) (BZm_of_nat (BZ_val x)).
```

Consider the following three functions, with range $\mathbf{N}^2$, $\mathbf{Z}$ and $\mathbf{N}^2$ respectively:

$$f : x \in \mathbf{Z} \mapsto \begin{cases} (0, |x|) & \text{if } x \in \mathbf{Z}_+ \\ (|x|, 0) & \text{otherwise,} \end{cases}$$

$$g : (x, y) \in \mathbf{N}^2 \mapsto \begin{cases} \phi_+(y - x) & \text{if } x \leq y \\ \phi_-(x - y) & \text{otherwise,} \end{cases}$$

$$h : ((x, y), (x', y')) \in \mathbf{N}^2 \times \mathbf{N}^2 \mapsto (x + x', y + y').$$

Note that g is surjective as $g(f(x)) = x$, but not injective; however $g(x, y) = g(x', y')$ is equivalent to $x + y' = x' + y$. The operation $x +_z y = g(h(f(x), f(y)))$ is clearly associative, and makes $\mathbf{Z}$ the group of differences of $\mathbf{N}$.

```
Definition Bzsum x y :=
  let f := fun x => Yo (inc x BZp) (J \0c (BZ_val x)) (J (BZ_val x) \0c) in
  let g := fun x => Yo ((P x) <=c (Q x))
```

```
              (BZ_of_nat((Q x) -c (P x))) (BZm_of_nat ((P x) -c (Q x))) in
  let h := fun x y => J ( (P x) +c (P y)) ( (Q x) +c (Q y)) in
  g (h (f x) (f y)).
```

Let x and y be two numbers, with absolute value a and b; then $x+y$ is $g(z)$, where z is one of $(0, a+b)$, (a,b), (b,a) or $(a+b,0)$, depending on the signs of the numbers. In the first and last cases, the sum is $\phi_+(a+b)$ and $\phi_-(a+b)$. This leads to the following equivalent (longer, but more natural) definition:

```
Definition BZsum x y:=
  let abs_sum :=  (BZ_val x) +c (BZ_val y) in
    let abs_diff1 :=  (BZ_val x) -c (BZ_val y) in
      let abs_diff2 :=  (BZ_val y) -c (BZ_val x) in
        Yo (inc x BZp /\ inc y BZp) (BZ_of_nat abs_sum)
        (Yo ( ~ inc x BZp /\ ~ inc y BZp) (BZm_of_nat abs_sum)
          (Yo (inc x BZp /\ ~ inc y BZp)
            (Yo ( (BZ_val y) <=c (BZ_val x))
              (BZ_of_nat abs_diff1) (BZm_of_nat abs_diff2))
            (Yo ( (BZ_val x) <=c (BZ_val y))
              (BZ_of_nat abs_diff2) (BZm_of_nat abs_diff1)))).
Lemma BZsum_alt x y: intp x -> intp y -> BZsum x y = Bzsum x y.
```

Addition on $\mathbf{Z}$ is compatible with addition on $\mathbf{N}$ and makes it a commutative group; the successor and predecessor functions (that map z to $z+1$ or $z-1$) are compatible with those of $\mathbf{N}$.

```
Lemma BZsum_cN x y: natp x -> natp y ->
  BZ_of_nat x +z BZ_of_nat y = BZ_of_nat (x +c y).
Lemma BZsucc_N x: natp x ->  BZsucc (BZ_of_nat x) = BZ_of_nat (csucc x).
Lemma BZprec_N x: inc x Nats -> BZpred (BZ_of_nat x) = BZ_of_nat (cpred x).
```

Consider the ordinal sum (see section 4.1 for the definition) of the opposite of the order of $\mathbf{N}^*$ and the natural order of $\mathbf{N}$. This is a total order on $\mathbf{Z}$, as each component is totally ordered. The order is compatible with addition (so that $\mathbf{Z}$ is an ordered group). Note that ϕ_+ is strictly increasing and $z \mapsto -z$ is strictly decreasing (it is an order isomorphism of $\mathbf{Z}$ onto the opposite order of $\mathbf{Z}$). Since $\mathbf{N}$ is well-ordered, there is a unique order isomorphism $\mathbf{N} \to \mathbf{N}$; here we have: for any $x \in \mathbf{Z}$, there is a unique order isomorphism $f : \mathbf{Z} \to \mathbf{Z}$ such that $f(0) = x$; it is defined by $f(z) = z + x$; and all order isomorphisms are of this form.

```
Definition BZ_ordering:=
  order_sum2 (opp_order (induced_order Nat_order Nats)) Nat_order.
Lemma zle_cN a b: natp a -> natp b ->
    (a <=c b <-> BZ_of_nat a <=z BZ_of_nat b).
Lemma zlt_opp x y: x <z y -> (BZopp y) <z (BZopp x).
Lemma zle_opp_iso:
  order_isomorphism (Lf BZopp BZ BZ) BZ_ordering (opp_order BZ_ordering).
Lemma zlt_succ1P a b: intp a -> intp b ->
  (a <z (BZsucc b) <-> a <=z b).
Lemma BZ_order_isomorphism_P f:
```

```
(order_isomorphism f BZ_ordering BZ_ordering) <->
(exists2 u, intp u & f = Lf (fun z => z +z u) BZ BZ).
```

We show here that every non-empty, totally ordered, complete and connected set is order isomorphic to $\mathbf{Z}$. Let's say that x and y are consecutive when $x < y$, and there is no z such that $x < z < y$. In this case, we say that y is a successor of x and that x is a predecessor of y. We say that the set is complete when each x has a successor and a predecessor. In a totally ordered set, these quantities are unique. We say that a set is connected when no trivial subset is stable by neighborhood (successor and predecessor).

Primo, it is obvious that $z+1$ and $z-1$ are the successor and predecessor of z in $\mathbf{Z}$. Secondo, if P is a property, $a \in \mathbf{Z}$, in order for P to hold on $\mathbf{Z}$, it suffices that $P(a)$ holds, that $P(x)$ implies $P(x+1)$ for $x \geq a$, and that $P(x)$ implies $P(x-1)$ for $x \leq a$ [Note: this is a generalization of the induction principle of SSREFLECT, which holds when $a = 0$]. Thus, every subset of $\mathbf{Z}$ stable by neighborhood that contains a contains all elements, and $\mathbf{Z}$ is connected. Tertio, if E is an ordered set, if $f : \mathbf{Z} \to E$ is a mapping such that $f(z) < f(z+1)$, then f is strictly increasing. Assume that E is totally ordered and complete, and $a \in E$. Define, by induction on $\mathbf{N}$, two sequences x and y, such that $x_0 = y_0 = a$, x_{n+1} is the successor of x_n and y_{n+1} is the predecessor of y_n. We may consider this as a function $f : \mathbf{Z} \to E$, that satisfies: for every z, $f(z)$ and $f(z+1)$ are consecutive. Thus, f is an order isomorphism on its image, which is clearly stable by neighborhood. So, if E is connected, f is surjective.

```
Definition consecutive r x y :=
   glt r x y /\ forall z, inc z (substrate r) -> ~(glt r x z /\ glt r z y).
Definition or_succ r x := select (fun z => consecutive r x z) (substrate r).
Definition or_pred r x := select (fun z => consecutive r z x) (substrate r).
Definition or_complete r := forall x, inc x (substrate r) ->
   (exists y, consecutive r x y) /\ (exists y, consecutive r y x).
Definition or_stable r E:= forall x, inc x E ->
   inc (or_succ r x) E /\ inc  (or_pred r x) E.
Definition or_connected r:= forall E, sub E (substrate r) -> or_stable r E
  -> E = emptyset \/ E = substrate r.
Definition or_likeZ r := [/\ total_order r, or_complete r & or_connected r].

Lemma BZ_order_props: or_likeZ BZ_ordering.
Lemma BZ_order_props_bis r: nonempty (substrate r) -> or_likeZ r ->
  r \Is BZ_ordering.
```

Defining multiplication, and proving the usual properties is easy. Note that the only invertible elements are 1 and -1.

```
Definition BZprod x y :=
  let aux := BZ_of_nat ((BZ_val x) *c (BZ_val y)) in
     (Yo (BZ_sg x = BZ_sg y) aux (BZopp aux)).
Lemma BZprod_1_inversion_l x y : intp x -> intp y -> x *z y = \1z ->
   (x = y /\ (x = \1z \/ x = \1mz)).
```

If a and b are in $\mathbf{Z}$, $b \neq 0$, there is a unique solution in $\mathbf{Z}$ of $a = bq+r$, $0 \leq r < |b|$. We give here the definition and characteristic property.

```
Definition BZdivision_prop a b q r :=
  [/\ a = (b *z q) +z r, r <z (BZabs b) & inc r BZp].

Definition BZquo a b :=
  let  q:= BZ_of_nat ((BZ_val a) %/c (BZ_val b)) in
    Yo (b = \0z) \0z
    (Yo (BZ_sg a = C1) (Yo (BZ_sg b = C1) q (BZopp q))
      (Yo ((BZ_val a %%c BZ_val b) = \0c)  (Yo (BZ_sg b = C1) (BZopp q) q)
        (Yo (BZ_sg b = C1) (BZopp (BZsucc q)) (BZsucc q)))).
Lemma BZdvd_correct a b: intp a -> intp b -> b <> \0z ->
  [/\ inc (a %/z b) BZ, inc (a %%z b) BZp &
  (BZdivision_prop a b (a %/z b) (a %%z b))].
Lemma BZdvd_unique1 a b q r: intp a -> intp b ->
  intp q -> intp r -> b <> \0z ->
  BZdivision_prop a b q r -> (q = a %/z b /\ r  = a %%z b).
```

Denote by $I(a, b)$ the set of all elements of the form $au + bv$, where u and v are in $\mathbf{Z}$. This is an ideal: it is stable by addition, and by multiplication by an element of $\mathbf{Z}$. In particular, if $x \in I$, then $-x \in I$. As a consequence, if I contains a non-zero element, it contains a strictly positive element. The set of strictly positive elements of I can be considered as a subset of $\mathbf{N}$, thus has a least element, say c. If $x \in I$, and $x = cq + r$, then $r \in I$. But if r is the remainder of x by c, we get $0 \leq r < |c|$, thus $r = 0$. It follows that $I(a, b) = I(c, c)$. The quantity c is called the gcd of a and b, and ab/c is called the lcm.

```
Definition BZgcd a b := select (fun z => BZ_ideal1 z = BZ_ideal2 a b) BZp.
Definition BZlcm a b := (a *z b) %/z (BZgcd a b).
```

Let's say that a divides b if for some q we have $b = aq$. This defines an order on $\mathbf{Z}_+^*$, and makes it a distributive lattice each pair x, y has a supremum and an infimum (here gcd and lcm), and there are some distributivity properties, for instance $\gcd(a, \mathrm{lcm}(b, c)) = \mathrm{lcm}(\gcd(a, b), \gcd(a, c))$. (Exercice 1.16 of Bourbaki [Bou68] lists other properties, in particular, the previous relation remains true if gcd and lcm are exchanged).

```
Definition BZdvdordering := graph_on BZdivides BZps.

Lemma BZdvd_lattice_aux x y: inc x BZps -> inc y BZps ->
  (least_upper_bound BZdvdordering (doubleton x y) (BZlcm x y)
   /\ (greatest_lower_bound BZdvdordering (doubleton x y) (BZgcd x y))).
Lemma  BZdvd_latticeD: distributive_lattice1 BZdvdordering.
```

We give here an alternate characterization of the gcd as the greatest common divisor; this works when a and b are possibly negative (in a general ring, the gcd is defined modulo an invertible element).

```
Definition BZgcd_prop a b p :=
```

```
   [/\ p %|z a, p %|z b  & forall t, t %|z a  -> t %|z b  -> t %|z p].
Lemma BZgcd_prop3 a b: intp a -> intp b ->
   (BZgcd_prop a b (BZgcd a b)
       /\ forall g,  BZgcd_prop a b g -> (BZgcd a b) = BZabs g).
```

We say that a and b are coprime and write $a \perp b$ if their gcd is one; this is obviously equivalent to: there exists u and v such that the Bezout relation $au+bv = 1$ holds. If g is the gcd of a and b, then a/g and b/g are coprime. We shall need the important property: if a and b are coprime, then $\gcd(a, bc) = \gcd(a, c)$.

```
Definition BZcoprime a b :=  BZgcd a b = \1z.
Definition Bezout_rel a b u v :=  (a *z u) +z (b *z v) = \1z.
Definition BZBezout a b :=
  exists u v, [/\ intp u, intp v & Bezout_rel a b u v].
Lemma BZ_Bezout_cofactors a b: intp a -> intp b ->
 (a<> \0z \/ b <> \0z) ->
  BZBezout (a %/z (BZgcd a b)) (b %/z (BZgcd a b)).
Lemma BZgcd_simp a b c: intp a -> intp b -> intp c ->
   BZcoprime a b -> BZgcd a (b *z c) = BZgcd a c.
```

The Bezout relation is not unique: for every q, $u' = u + qb$, $v' = v - qa$ yields another solution. Assume $a \neq 0$, and take for q the quotient of v by a, it follows: there is a unique solution to $au + bv = 1$ with $0 \leq v < |a|$.

Assume $a \in \mathbf{N}$, $b \in \mathbf{N}$. We can coerce a and b into $\mathbf{Z}$, compute the gcd, and put it back on $\mathbf{N}$. It satisfies the expected properties. Unless a and b are trivial, the Bezout relation does not hold on $\mathbf{N}$. However the variant $au = 1 + bv$ holds. Of course, this equation does not have a unique solution. One may assume $u < b$ when $b > 1$. (Note: SSREFLECT gives an algorithm, that returns such a pair, with $v < a$, when $a \neq 0$).

```
Definition Ngcd_prop a b p :=
   [/\ natp p, p %|c a, p %|c b  &
    forall t, natp t  ->  t %|c a  -> t %|c b  -> t %|c p].
Lemma Ngcd_P a b: natp a -> natp b ->
   (Ngcd_prop a b (Ngcd a b)
       /\ forall g,  Ngcd_prop a b g -> (Ngcd a b) = g).
Lemma Nbezout a b: natp a -> natp b -> a <> \0c -> Ncoprime a b ->
   exists u v, [/\ natp u, natp v, a *c u = \1c +c b *c v&
      (b <=c \1c \/ u <c b)].
```

Application: $\gcd(F_n, F_m) = F_{\gcd(n,m)}$, where F_n is the n-th Fibonacci number. In particular, two consecutive Fibonacci numbers are coprime. We give here a formula for F_{n-m}; if m is $n-1$ or $n-2$, we get a Bezout relation between F_n and F_{n+1}, of the form $au = 1 + bv$, $u < b$.

```
Lemma Ngcd_fib n m: natp n -> natp m ->
  Ngcd (Fib n) (Fib m) = Fib (Ngcd n m).
Lemma Fib_sub n m: natp n -> natp m -> m <=c n->
  Fib (n -c m) = Yo (evenp m)
```

```
    (Fib n *c Fib (csucc m) -c Fib (csucc n) *c Fib m)
    (Fib (csucc n) *c Fib m -c Fib n *c Fib (csucc m)).
```

We give here an isomorphism between $\mathbf{Z}$ and the data structure int of SSREFLECT; we show for instance that gcd and coprime are the same notions in both cases.

```
Definition BZ_of_Z (n:int) :=
  match n with
    | Posz p => BZ_of_nat (nat_to_B p)
    | Negz p => BZm_of_nat (nat_to_B p.+1)
end.

Lemma BZ_of_Z_surjective x : intp x -> exists y, BZ_of_Z y = x.
Lemma BZ_of_Z_injective x y :  BZ_of_Z x = BZ_of_Z y ->  x = y.
Lemma BZ_of_gcd (n m: nat) :
  BZgcd (BZ_of_Z n) (BZ_of_Z m) = BZ_of_Z (gcdn n m).
Lemma BZ_of_coprime (n m: nat) :
  BZcoprime (BZ_of_Z n) (BZ_of_Z m) <-> (coprime n m).
```

4. THE SET OF RATIONAL NUMBERS Q

We define $\mathbf{Q}$ as the set of pairs (a, b), such that $a \in \mathbf{Z}$, $b \in \mathbf{Z}_+^*$, a and b being coprime. The two quantities a and b are called the numerator and denominator, and denoted $N(x)$ and $D(x)$. Define $\psi(a, b) = (a/g, b/g)$ where g is the gcd of a and b. If $a \in \mathbf{Z}$ and $b \in \mathbf{Z}_+^*$, then $\psi(a, b) \in \mathbf{Q}$. We have $\psi(a, b) = \psi(a', b')$ when $ab' = a'b$.

```
Definition BQ := Zo (BZ \times BZps) (fun z => BZcoprime (Qnum z) (Qden z)).
Definition BQ_of_pair a b :=
   J (a %/z (BZgcd a b)) (b %/z (BZgcd a b)).
Lemma BQ_of_pair_prop4 a b:
  intp a -> inc b BZps -> inc (BQ_of_pair a b) BQ.
Lemma BQ_of_pair_prop5 a b c d:
  intp a -> inc b BZps -> intp c -> inc d BZps ->
   (a *z d = b *z c <-> BQ_of_pair a b = BQ_of_pair c d).
```

The relation "$x \in \mathbf{Q}$, $y \in \mathbf{Q}$, $N(x)D(y) \leq D(x)N(y)$", denoted by $x \leq_q y$, is a total order on $\mathbf{Q}$. Defining addition, subtraction, multiplication and proving that $\mathbf{Q}$ is an ordered ring is easy.

```
Definition BQle_aux x y := (Qnum x) *z (Qden y) <=z  (Qden x) *z (Qnum y).
Definition BQ_le x y := [/\ ratp x, ratp y & BQle_aux x y].
Notation "x <=q y" := (BQ_le x y) (at level 60).
Lemma BQle_P x y: gle BQ_ordering  x y <-> x <=q y.
Lemma BQor_tor: total_order BQ_ordering.
```

Define the inverse x^{-1} of a non-zero rational number x as the pair $(D(x), N(x))$ when $N(x) > 0$ and $(-D(x), -N(x))$ otherwise; then $x^{-1} \cdot x = 1$. For completeness, we define the inverse of zero to be zero, so that $(x^{-1})^{-1} = x$ in any case. Since x has the same sign as its inverse, it follows easily that $\mathbf{Q}$ is an ordered field.

```
Definition BQinv x :=
  Yo (inc (Qnum x) BZps) (J (Qden x) (Qnum x))
    (Yo (Qnum x = \0z) \0q (J (BZopp (Qden x)) (BZopp (Qnum x)))).
Lemma BQprod_inv1 x : ratp x -> x <> \0q -> (x *q  (BQinv x)) = \1q.
Lemma BQinv_K x: ratp x -> BQinv (BQinv x)  = x.
Lemma BQinv_sign x: ratp x -> BQsign (BQinv x) = BQsign x.
```

4.1 The order type of $\mathbf{Q}$

Let η be "the ordinal type of the aggregate R of all rational numbers which are greater than 0 and smaller than 1, in their natural order of precedence" as defined by Cantor in [Can97, first article, §9].

Cantor notices that one can well-order R, the set of all p/q, where p and q are coprime and $0 < p < q$, by comparing first $p + q$, and then p/q. In particular, he deduces that R has cardinal $\aleph_0$. On the other hand, for the natural ordering, R has no least element (consider $x/2$) and no greatest element (consider $(x + 1)/2$), and R is everywhere dense, in the sense that for every $x \in R$ and $y \in R$, if $x < y$, there is $z \in R$ such that $x < z < y$ (consider $(x + y)/2$). Let's say that M is η-like if M is totally ordered, infinite countable, and satisfies these three conditions. Cantor proves, in §9, that an η-like set has the same order type as R.

We shall give the proof below, under the form "M and R are order isomorphic". Let's assume that there is a way to assign to each ordered set M some quantity $\overline{M}$ (the order type, or ordinal type for Cantor); the theorem takes then the form $\overline{M} = \eta$, where $\eta = \overline{R}$. According to Cantor, $\overline{M}$ is the "general concept which results from M if [...]. Thus $\overline{M}$ is itself an ordered aggregate [...]". The second sentence can be restated in modern words as: $\overline{M}$ is an ordered set, order isomorphic to M; a trivial consequence of the condition of the first sentence is that two ordered sets that are order isomorphic have the same order type.

There is no obvious definition of $\overline{M}$. In the case of a well-ordered set, one could use von Neumann ordinals. Bourbaki uses $\tau_Y(X \operatorname{Is} Y)$, where "Is" means order isomorphic. In the four following lemmas, we shall admit the existence of an order type and its characteristic properties.

```
Parameter order_type_of: Set -> Set.
Axiom order_type_exists:
  forall x, order x -> x \Is (order_type_of x).
Axiom order_type_unique:
  forall x y, x \Is y -> (order_type_of x = order_type_of y).
```

Let E and F be two ordered sets, and G the ordinal sum (this means that G is the order on the disjoint union of E and F, where each element of E comes before each element of F). Now, $\overline{G}$ depends only on $\overline{E}$ and $\overline{F}$ and is called the sum of the two order types. Similarly, one can define the product of two order types by considering the lexicographic product of F and E (this product is non-commutative). An easy consequence of Cantor's theorem is then: $\eta + \eta = \eta$, $\eta \cdot \eta = \eta$; there are other formulas, for instance $\eta + 1 + \eta = \eta$ and: the opposite of η is η.

```
Lemma Cantor_eta_pr4: Cantor_eta  +t Cantor_eta = Cantor_eta.
Lemma Cantor_eta_pr5: Cantor_eta  +t \1t +t Cantor_eta = Cantor_eta.
Lemma Cantor_eta_pr6: Cantor_eta  *t Cantor_eta = Cantor_eta.
Lemma Cantor_eta_pr7: OT_opposite Cantor_eta = Cantor_eta.
```

It is easy to show that $\mathbf{Q}$ and $\mathbf{Q}_+^*$ are η-like, thus are order isomorphic to R. We give a direct proof here. Let $f(z) = z/(1+z)$. This function is defined when $z \neq -1$ and is strictly increasing (on either side of the cut). It is an isomorphism $\mathbf{Q}^+ \to]0,1[$. Define $g(z)$ by: if $z < 0$, then $1/(1-z)$, otherwise $z+1$. Both pieces are strictly increasing, the image of the first piece is $]0,1[$, the image of the second piece is $[1,\to[$, so that g is an order isomorphism $\mathbf{Q} \to \mathbf{Q}^+$.

```
Lemma BQ_iso1: order_isomorphism
   (Lf (fun z => z /q (\1q +q z)) BQps BQ_int01)
    BQps_ordering  BQ_int01_ordering.
Lemma BQ_iso2: order_isomorphism
   (Lf (fun z => Yo (z <q \0q) (\1q /q (\1q -q z)) (z +q \1q)) BQ BQps)
    BQ_ordering BQps_ordering.
```

The isomorphisms given above are not unique: in fact, the number of order isomorphisms $\mathbf{Q} \to \mathbf{Q}$ is the cardinal of $\mathbf{R}$, thus $2^{\mathfrak{c}}$, where $\mathfrak{c} = \aleph_0$ is the cardinal of $\mathbf{N}$ and also of $\mathbf{Q}$. First note that an isomorphism is a function, and the number of functions is $\mathfrak{c}^{\mathfrak{c}}$, this is $2^{\mathfrak{c}}$ since $\mathfrak{c}$ is an infinite cardinal. Obviously, $z \mapsto az + b$ is an order isomorphism when $a > 0$.

Let f be an element of $2^{\mathfrak{c}}$; it can be considered as a function $\mathbf{N} \to \{0,1\}$, so that we can define, by induction, $g(n+1) = g(n) + f(n) + 1$, $g(0) = 0$. Obviously $f \mapsto g$ is injective, g is strictly increasing and unbounded. There is a function h, defined on $\mathbf{Q}^+$, such that, on each interval $[n, n+1]$, h has the form $z \mapsto az + b$, with $h(n) = g(n)$ and $h(n+1) = g(n+1)$. Since g is strictly increasing and unbounded, so is h. Extend h by $h(z) = z$ for $z < 0$. Then h is an order isomorphism, and $g \mapsto h$ is obviously injective.

```
Lemma BQ_iso3 a b: inc a BQps -> inc b BQ ->
    order_isomorphism (Lf (fun z => a*q z +q b) BQ BQ)
      BQ_ordering  BQ_ordering.
Lemma BQ_iso4 (E := Zo (permutations BQ)
       (fun f => order_isomorphism f BQ_ordering  BQ_ordering)):
   cardinal E = \2c ^c aleph0.
```

A consequence of $\eta + \eta = \eta$ is that $\mathbf{Q}^*$ and $\mathbf{Q}$ are order isomorphic (here $\mathbf{Q}^*$ is the disjoint union of $\mathbf{Q}_-^*$ and $\mathbf{Q}_+^*$, $\mathbf{Q}_-^*$ is isomorphic to $\mathbf{Q}_+^*$ via $x \mapsto -1/x$, so that $\mathbf{Q}^*$ is order-isomorphic to the ordinal sum of two copies $\mathbf{Q}$). As mentioned above, the number of order isomorphisms $f : \mathbf{Q}^* \to \mathbf{Q}$ is the cardinal of $\mathbf{R}$. This is not a coincidence.

If A and B are the sets of all $f(x)$ such that $x < 0$ and $x > 0$ respectively, then A and B are disjoint, the union is $\mathbf{Q}$, every element of A is less than every element of B. The pair (A, B) will be called an irrational cut in section 5.1. Consider the two sequences $a_i = f(-1/(i+1))$ and $b_i = f(1/(i+1))$, indexed by $i \in \mathbf{N}$. They are

called *adjacent* because the first sequence is strictly increasing, the second sequence is strictly decreasing, $a_i < b_j$, and the infimum of $b_i - a_i$ is zero. Given adjacent sequences, we can define two disjoint sets, A, the set of all x such that $x < a_i$ for some i and B, the set of all x such that $b_i < x$ for some i. If B has an infimum q, then A is the set of numbers $< q$ while B is the set of numbers $> q$; otherwise $A \cup B = \mathbf{Q}$ and (A, B) is an irrational cut.

We show here that each pair of adjacent sequences (with an irrational cut) yields an order isomorphism $f : \mathbf{Q}^* \to \mathbf{Q}$. The idea is to impose$f(-1/(i+1)) = a_i$ and $f(1/(i+1)) = b_i$. On the intervals $[1/(n+2), 1/(n+1)]$ and $[-1/(n+1), -1/(n+2)]$, f has the form $z \mapsto \alpha z + \beta$, on the intervals $]\leftarrow, -1]$ and $[1, \rightarrow[$ it has the form $z \mapsto z + \beta$.

```
Lemma multiple_interpolation_prop3 f1 f2
   (Zb := Zo BQ (fun z => exists2 n, natp n & f1 n <q z))
   (Za := Zo BQ (fun z => exists2 n, natp n & z <q f2 n)):
  (forall n, natp n -> f1 (csucc n) <q f1 n) ->
  (forall n, natp n -> f2 n <q f2 (csucc n)) ->
  (disjoint Za Zb) -> (Za \cup Zb = BQ) ->
  exists2 g, order_isomorphism g
    (induced_order BQ_ordering (BQ -s1 \0q)) BQ_ordering &
   image_by_fun g BQps = Zb.
```

We shall prove later on that for every irrational cut (A, B), there is a pair of adjacent sequences (a, b) that defines the cut. Thus, there is an order isomorphism $f : \mathbf{Q}^* \to \mathbf{Q}$ such that $f(\mathbf{Q}_+^*) = B$. As there is at least one irrational number, there is at least one order isomorphism $f : \mathbf{Q}^* \to \mathbf{Q}$. In the two lemmas that follow (that should be placed at the end of section 5.1.), r and r' are the orders on $\mathbf{Q}$ and $\mathbf{Q}^*$.

```
Lemma BQ_order_isomorphisms_spec x:  irrationalp x ->
  exists2 f, order_isomorphism f r' r & Vfs f BQps  = x.
Lemma BQ_order_isomorphisms_spec2: exists f, order_isomorphism f r' r.
```

Let's prove Cantor's theorem. We first assume that we has two η-like ordered sets, E and F; as the sets are infinite countable, we consider two bijections $f : \mathbf{N} \to E$ and $g : \mathbf{N} \to F$. These sets are totally ordered by r1 and r2, satisfying the conditions stated above (no least element, no greatest element, no empty interval). We pretend that there is an order isomorphism $\psi : E \to F$; let $h = g^{-1} \circ \psi \circ f$, so that $\psi = g \circ h \circ f^{-1}$. The objective is to construct by induction a bijection $h : \mathbf{N} \to \mathbf{N}$ such that $f(i) < f(j)$ is equivalent to $g(h(i)) < g(h(j))$. At any stage, we choose the least integer compatible with the values chosen previously; thus $h(0) = 0$. Let's explain on an example how to compute $h(3)$. Since E is totally ordered, the three elements $f(0)$, $f(1)$ and $f(2)$ can be arranged in increasing order. This means that there is a permutation σ of I_3 such that $i \mapsto f(\sigma(i))$ is strictly increasing, and a necessary condition for h is that $i \mapsto g(h(\sigma(i)))$ is also strictly increasing. Assume for instance $\sigma(0) = 2$, $\sigma(1) = 0$ and $\sigma(2) = 1$. This means $f(2) < f(0) < f(1)$ and $g(h(2)) < g(h(0)) < g(h(1))$. Let k be the position of $x = f(3)$ relative to $f(0)$, $f(1)$ and $f(2)$. So, $k = 0$ means $x < f(2)$, $k = 1$ means $f(2) < x < f(0)$, $k = 2$

means $f(0) < x < f(1)$ and $k = 3$ means $f(1) < x$. Let $y_i = g(h(i))$; the position of $y = g(h(3))$ relative to y_0, y_1 and y_2 is exactly k. Whatever k, there is always at least one y satisfying this condition, since there are elements a, b, c and d such that $a < y_2 < b < y_0 < c < y_2 < d$. We define $h(3)$ to be the least integer $g^{-1}(y)$ such that y satisfies this condition.

The first definition given here expresses that $i \mapsto f(\sigma(i))$ is strictly increasing; the last defines $s = S_k(\sigma)$ by $s(i) = \sigma(i)$ for $i < k$, $s(i) = \sigma(i-1)$ for $i > k$, and $s(k) = n$.

```
Definition EPperm_M s n:=
   inc s (perm_int n) /\
   forall i j, i<c j -> j <c n -> glt r1 (Vf f (Vf s i)) (Vf f (Vf s j)).

Definition EPperm_extend_aux n k s :=
  fun z => Yo (z <c k) (Vf s z) (Yo (z = k) n (Vf s (cpred z))).
Definition EPperm_extend n k s :=
  Lf (EPperm_extend_aux n k s) (csucc n) (csucc n).
```

The first definition here, write it $C(k, x)$, says that x is at position k (given an order r, a size n and a function h that plays the role of $f \circ \sigma$). We first specialize it to E, and say: if r is a total order, x not in the image of h, there is a unique k such that $C(k, x)$ holds, it will be denoted $k_C(s, n)$. For completeness $k_C(s, 0) = 0$.

```
Definition EPperm_compat0 r h n k x :=
   [/\ k <=c n,
    (k = \0c -> glt r x  (h \0c)),
    (k = n -> k <>\0c -> glt r (h (cpred n)) x) &
    (k <c n -> k <> \0c -> (glt r (h (cpred k)) x) /\ glt r x (h k))].
Definition EPperm_compat s n k :=
   EPperm_compat0 r1 (fun i =>  (Vf f (Vf s i))) n k (Vf f n).
Definition EPperm_next_index  n s :=
  Yo (n= \0c) \0c (select (EPperm_compat s n) Nat).
```

We define by induction a sequence $(s_n)_n$ of permutations of I_n, satisfying the conditions stated above, via $s_{n+1} = S_k(s_n)$, where $k = k_C(s_n, n)$. Note that s_0 and s_1 are both defined to be the identity function on I_0 and I_1 respectively.

```
Definition EPperm_rec :=
   induction_term (fun n s =>
       (Yo (n = \0c) (identity \1c)
          (EPperm_extend n (EPperm_next_index n s) s))) empty_function.
Lemma EPperm_recs_mon n: natp n -> EPperm_M (EPperm_rec n) n.
```

We assume here that we know the restriction h_1 of our function to I_n. Let s_n be as above, $k = k_C(s_n, n)$; define $h(i) = g(h_1(s_n(i)))$. Let K be the set of all j such that $C(k, g(j))$ holds. Since h is strictly increasing, with values in the η-like set F, and since g is surjective, this set is not empty, thus has a least element (which is the intersection). Note that, if h_1 is replaced by h_2 that takes the same value, we get the same result.

```
Definition EPpermi_fct h n k :=
    intersection (Zo Nat (fun j => EPperm_compat0 r2 h n k (Vf g j))).
```

We denote by $v(h_1)$ the quantity defined above. By transfinite induction, we define a function h, such that $h(n) = v(h_n)$ where h_n is the restriction of h to I_n. Since $n = 0$ is special, we impose $h(0) = 0$. It is clear (but the proof is a bit long, 140 lines) that $g \circ h \circ f^{-1}$ is an order isomorphism.

```
Definition EPpermi_next ph n (s:= EPperm_rec n)
   (h := fun i => Vf g (Vf ph (Vf s i)))
   (k := (EPperm_next_index n s)) :=
  Yo (n = \0c) \0c (EPpermi_fct h n k).
Definition EPfun_aux :=
  transfinite_defined Nat_order (fun u => (EPpermi_next u (source u))).
Lemma EPfun_aux_pr1 (h := EPfun_aux) :
  [/\ surjection h, source h = Nat, sub (target h) Nat,
      Vf h \0c = \0c &
     forall n, natp n -> Vf h n = EPpermi_next (restriction1 h n) n].
```

We prove surjectivity of h by induction as follows. Assume that every integer $< n$ is in the range of h, and let's show that n is in the range. The assumption is that, for some m, if K is the set of all $h(i)$ for $i < m$, if $i < n$, then $i \in K$. If $n \in K$, there is nothing to do. Otherwise, we may consider the position q of $g(n)$ compared to the $g(j)$, $j \in K$. There is a least k_1 such that q is the position of $f(k_1)$ compared to the $g(j)$, $j < m$. This is expressed by the following definition:

```
Definition EPperm_2pos m k1 q n
   (s :=  EPperm_rec m)   (f1 := fun i => Vf f (Vf s i))
   (h:= fun i => Vf g (Vf EPfun_aux (Vf s i)))
   (P1 := fun i => EPperm_compat0 r1 f1 m q (Vf f i)) :=
   (P1 k1 /\ (forall i, natp i -> P1 i -> k1 <=c i))
  /\ EPperm_compat0 r2 h m q (Vf g n).
```

We pretend $h(k_1) = n$. Consider what happens when m is incremented. Let $k = k_C(s_m, m)$. If $k = k_1$, the result is clear. Otherwise, we insert a new element in the sequence. The new q will be unchanged if $q < k$, incremented by one otherwise, but k_1 is left unchanged (the proof of this fact required 100 lines of script). We cannot always be in this situation, since $m < k_1$ (200 more lines of scripts are required). We finally have:

```
Definition EP_fun := g \co (EPfun_aux) \co  (inverse_fun f).
Lemma EP_fun_pr: order_isomorphism EP_fun r1 r2.
```

The main result is then the following (the proof script of this fact, including definitions) is a bit over 1000 lines.

```
Lemma Cantor_eta_pr r1 r2:
  eta_like r1 -> eta_like r2 -> r1 \Is r2.
```

4.2 The Stern Brocot sequence

We know that $\mathbf{Q}$ is countable, but is there an explicit bijection $\mathbf{N} \to \mathbf{Q}$? A positive answer was given by Moritz Stern in 1858 [Ste58] and independently by Achille Brocot in 1861. If s_n denotes the sequence (listed under number A002487 in the On-line Encyclopedia of Integer Sequences) formed of all numbers that appear in the basic Stern diatomic sequence (see [Gri14] for details), then the following conditions hold: s_k and s_{k+1} are coprime, and every rational number is uniquely a quotient s_k/s_{k+1}. The Stern-Brocot tree is a tree representation of the sequence, it also provides an algorithm for the inverse function $\mathbf{Q} \to \mathbf{N}$; it can be used for an implementation of $\mathbf{Q}$ in Coq, see [Niq07, NB04].

The sequence was rediscovered in 1976 by Dijkstra, under the name "fusc" [Dij76a, Dij76b], with the following definition:

$$s_1 = 1, \quad s_{2n} = s_n, \quad s_{2n+1} = s_n + s_{n+1}. \tag{4}$$

In the first paper Dijkstra writes: if $f_1 = \mathrm{fusc}(n_1)$ and $f_2 = \mathrm{fusc}(n_2)$, "if there exists an N such that $n_1 + n_2 = 2^N$, then f_1 and f_2 are relatively prime" and "if f_1 and f_2 are relatively prime, then there exists an n_1, an n_2, and an N such that $n_1 + n_2 = 2^N$. In the above recursive definition, this is no longer obvious, at least not to me; hence its name."

There is an implementation in pure ssreflect of the definitions and theorems given in this section, see [Gri14]; we show the definition and the two claims of Dijkstra.

```
Definition fusc n :=
  let fix loop  n k :=
    if k is k'.+1 then
      if n<=1 then n else if (odd n) then (loop n./2 k') + loop (uphalf n) k'
      else loop n./2 k'
    else 0
  in loop n n.
Lemma fusc_fusc3 n m p: n + m = 2^p -> coprime (fusc n) (fusc m).
Lemma fusc_fusc2 a b: coprime a b ->
   exists n m p, [/\ n + m = 2^p, a = fusc n & b = fusc m].
```

The second paper of Dijkstra starts with two properties.

$$s(n) = s(i(n)), \qquad s(n) = s(r(n)). \tag{5}$$

Here $r(n)$ is the base two reverse of n and $i(n)$ the number obtained by inverting in the binary representation of n all inner digits (inner digits are those that follow the initial 1 and are before the last 1; if n is a power of two there is no inner digit and $i(n) = n$). For instance $19 = 10011_2$, $r(19) = 11001_2 = 25$ and $i(19) = 11101_2 = 29$. If n is odd then $i(r(n)) = r(i(n))$. Moreover, unless n is a power of two, then $n + i(n) = 3p$ where p is the greatest power of two such that $p \leq n$ (example: $19 + i(19) = 3 \cdot 2^4$). From $r(2k+1) = 2^{\log_2(k)} + r(k)$, it follows the magic formula

$$r(n-1) + i(r(n)) = 2^{\log_2 n} \qquad (n > 1, n \text{ odd}). \tag{6}$$

The first equality of (5), together with $n + i(n) = 3p$, gives the palindrome condition: if $a \geq p$, $b \geq p$ and $a + b = 3p$, then $s(a) = s(b)$. A consequence is: for every $k > 1$ there is k' such that $s(k) = s(k' + 1)$ and $s(k + 1) = s(k')$; moreover $k + k'$ is odd.

We pretend that, for every function s satisfying (5), the claims of the first paper of Dijkstra are equivalent to the Stern properties: whatever k, $s(k) \perp s(k + 1)$; if $a \perp b$ there is k such that $a = s(k)$ and $b = s(k+1)$. The fusc function satisfies the Stern properties, but we doubt that Dijkstra used the reasoning explained here (he qualifies the second equality of (5) as "more surprising") and we have no idea how to prove directly the claims of the first paper.

Let's consider the following context:

```
Section Dijkstra.
Variable s: nat -> nat.

Definition Dijkstra_prop1:= forall n m p,
  n + m = 2^p -> coprime (s n) (s m).
Definition Dijkstra_prop2 := forall a b,
  coprime a b ->  exists n m p, [/\ n + m = 2^p, a = s n & b = s m].
Definition Dijkstra_prop1bis :=
  forall k,  coprime (s k) (s k.+1).
Definition Dijkstra_prop2bis := forall a b,
  coprime a b.+1 -> exists k, a = s k /\ b.+1 = s k.+1.

Hypothesis DH_FR: forall n, s (base2rev n) = s n.
Hypothesis DH_BI: forall n, s (bin_invert n) = s n.
Hypothesis DH_1: s 1 = 1.
```

The key relation of the proofs is the following: assume $n+m = 2^p$, $n \leq m$ and m odd. Take $m' = r(i(m))$; the magic relation says $n = r(m' - 1)$. So $s(m) = s(m')$ and $s(n) = s(m' - 1)$. The second relation implies the palindrome condition, and its consequence. Hence, there is m'' such that $s(m) = s(m'' - 1)$ and $s(n) = s(m'')$. This means that we can get rid of the condition $n \leq m$. The first relation says $s(2k) = s(k)$ (so that we can get rid of the condition m odd) [size of the proof: 200 lines].

```
Lemma DH_prop1_from_bis: Dijkstra_prop1bis -> Dijkstra_prop1.
Lemma DH_prop2_from_bis: s 0 = 0 -> Dijkstra_prop2bis ->  Dijkstra_prop2.
Lemma DH_prop1_to_bis: Dijkstra_prop1 -> Dijkstra_prop1bis.
Lemma DH_prop2_to_bis: s 0 = 0 -> Dijkstra_prop2 -> Dijkstra_prop2bis.
End Dijkstra.
```

The Gaia definition is by transfinite induction: $s(n) = V(s_n)$, where s_n is the restriction of s to the set of integers $< n$, and V some function compatible with equation (4). Our function satisfies (4) because all arguments to s_n in V are less than n (in the same manner, the loop above never terminates with the "else" part, if the argument in non-zero). Note that (4) has unique solution, and setting $s_0 = 0$ causes no harm.

```
Definition fusc_next F n:=
  Yo (n = \0c) \0c (Yo (n = \1c) \1c (Yo (evenp n) (Vf F (chalf n))
   ((Vf F (chalf n)) +c (Vf F (csucc (chalf n))))))).

Definition fusc :=
  Vf (transfinite_defined Nat_order (fun u => (fusc_next u (source u)))).
```

Let $x_n = s_n/s_{n+1}$, this is the solution of:

$$x_{2n} = \frac{1}{1+1/x_n} (\text{or } \frac{1}{x_{2n}} = \frac{1}{x_n} + 1), \qquad x_{2n+1} = x_n + 1 \qquad x_0 = 0. \tag{7}$$

The function $n \mapsto x_n$ is a bijection $\mathbf{N} \to \mathbf{Q}^+$. First note that, by induction, the two quantities s_n and s_{n+1} are coprime. Let's show (by induction) that the function is injective. Assume $x_n = x_m$. Then n and m have the same parity (since $x_{2k} < 1$ and $x_{2k+1} \geq 1$). Assume for instance $n = 2p$ and $m = 2q$. We have $s_p = s_q$ and $s_p + s_{p+1} = s_q + s_{q+1}$, thus $s_{p+1} = s_{q+1}$, and we conclude by induction; the other case is similar. Let's show (by induction on $p + q$) that $x = p/q$ is of the form x_k. If $x \geq 1$, then $(p - q)/q = x_n$ for some n, so that $x = x_{2n+1}$; if $0 < x < 1$, then $(q - p)/p = x_n$ for some n, so that $x = x_{2n}$; the case $x = 0$ is trivial.

```
Definition fusc_quo n:= BQ_of_nat (fusc n) /q  BQ_of_nat (fusc (csucc n)).
Lemma fusc_quo_bijection:
   bijection_prop (Lf fusc_quo1 Nat BQp) Nat BQp.
```

There is a simple relation between x_n and x_{n+1}.

$$F(x_n) = x_{n+1}, \qquad F(x) = \frac{1}{1 + 2\lfloor x \rfloor - x}. \tag{8}$$

```
Definition rat_iterator x :=
   BQinv (\1q +q (BQdouble (BQ_of_Z(BQfloor x))) -q x).
Lemma rat_fusc n: natp n -> rat_iterator (fusc_quo n) = fusc_quo (csucc n).
```

We may consider the sequence $(s_n)_{n>0}$ as an infinite array, by considering $A_{ij} = s_{2^i+j}$. Thus row i is formed of values with index beween 2^i and 2^{i+1}. A priori, 2^{i+1} is excluded, but recall that $s_n = 1$ when n is a power of two. Each row is a palindrome. Each column is in arithmetic progression, the common difference being a Stern number. This is expressed as $A_{i+1,j} = A_{i,j} + s_j$, or $s_{2p+j} = s_{p+j} + s_j$ (whenever p is a power of two).

```
Lemma fusc_palindrome n a b (p := \2c ^c n): natp n ->
   p <=c a -> a <=c cdouble p -> p <=c b -> b <=c cdouble p ->
   a +c b = \3c *c p -> fusc a = fusc b.
Lemma fusc_col_progression i j: natp i -> natp j -> j <=c \2c ^c i ->
  fusc(\2c ^c (csucc i) +c j) = fusc(\2c ^c i +c j) +c fusc j.
```

A remarkable relation is that the maximum of each row is a Fibonacci number (that $k \leq 2^n$ implies $s_k \leq F_{n+1}$ is rather obvious by induction on n: write $s_{2m+1} = s_m + s_{m+1}$ and notice that one of m, $m + 1$ is even). On the other hand, since two consecutive Fibonacci numbers are coprime, there is m such that $s_m = F_n$

and $s_{m+1} = F_{n+1}$, thus $s_{2m+1} = F_{n+2}$; that $2m+1$ is on row n is less obvious. It happens that the position of the first maximum of the row (there is a second maximum by symmetry) satisfies the relation $c_{n+2} = c_{n+1} + 2c_n$. There is an explicit solution $c_n = (4.2^n - (-1)^n)/3$.

```
Definition Fib_fusc_rec :=
 induction_term  (fun _ v => (J (csucc (cdouble (Q v))) (P v +c Q v)))
   (J \0c \0c).
Definition Fib_fusc n := P ( Fib_fusc_rec n).

Lemma fusc_bound1 n k: natp n -> k <=c \2c ^c n -> fusc k <=c Fib (csucc n).
Lemma Fib_fusc_val n: natp n -> (fusc (Fib_fusc n)) = Fib n.
Lemma Fib_fusc_bound n (p := \2c ^c n) (v := Fib_fusc (csucc (csucc n))):
   natp n -> (p <=c v /\ v <=c (cdouble p)).
```

We have shown above that every integer n has a unique expansion to base 2, i.e., can be written as $n = \sum_{i<k} a_i 2^i$, with $a_i < 2$, and a_{k-1} is non-zero. What happens if $a_i < 2$ is replaced by $a_i \leq 2$? The relation a_{k-1} non-zero says $n \geq 2^{k-1}$, thus $k < n$. Thus the sequence $(a_i)_{i<k}$ belongs to some finite set X_n, whose cardinal is s_{n+1}. We have shown the same property in pure SSREFLECT; the difficulty is the following: we can only compute the cardinal of a subset X_n of a finite type T. We cannot say: X_n is a subset of the set of functional graphs whose domain is a subset of $\mathbf{N}$ with values in I_3 (the Gaia definition), nor that X_n is a subset of the set of sequences of length $\leq \log_2 n$ with values in I_3, because this finite set is not defined in SSREFLECT (although defining it should not be too complicated). For this reason, we replace the condition a_{k-1} non-zero by: the sequence has a fixed length. So, X_n is a subset of the set of sequences of length k with values in I_3. Now X_n depends on k, say becomes $X_{n,k}$ but as explained above, if n has two representations of length k_1 and k_2, then the digits, starting with $\max(k_1, k_2)$, are zero (a non-zero digit is at position $\leq \log_2 n$). As a consequence, if $k \geq \log_2 n$, the cardinal of $X_{n,k}$ becomes independent of k. The pure SSREFLECT case is the following:

```
Fixpoint wbase2 (l: seq 'I_3) :=
  if l is a :: l' then a + (wbase2 l').*2 else 0.
Definition card_wbase2' k n := #|[set t :k.-tuple 'I_3 | wbase2 t == n ]|.
Definition card_wbase2 n := card_wbase2' (log2 n) n.
Lemma card_wbase2_val n: card_wbase2 n = fusc n.+1.
```

Defining X_n and its cardinal $C(n)$ in Gaia is a bit easier. In both cases, we reason as follows. Consider $a \in X_n$, and let b be the sequence with the first element removed. Then $n = a_0 + 2m$ and $b \in X_m$. If n is odd, then $a_0 = 1$; it follows that $C(2m+1) = C(m)$. On the other hand, if n is even, then a_0 can be 0 or 2, so $n = 2m$ or $n = 2m+2$, it follows $C(2m+2) = C(m+1) + C(m)$ thus the result:

```
Lemma expe_fusc n: natp n -> Nbexp n = fusc (csucc n).
```

We consider here three sums where the index ranges beween 2^n and $2^{n+1}-1$:

$$\sum_i s_i = 3^n, \qquad \sum_i 1/(s_i s_{i+1}) = 1, \qquad \sum_i s_i/s_{i+1} = (3\cdot 2^n - 1)/2. \tag{9}$$

Let w_i be the generic term of the sum. In the first case, w_i is an integer and the sum is over the interval $[2^n, 2^{n+1}[$ (below is a sum over 'Nintc n', this is the interval $[0,n]$). In the two other cases, w_i is a rational number, and we compute $\sum_{i<p} w_{p+i}$, where $p=2^n$ and $\sum_p$ is defined by induction over p. There is no generic sum in Gaia, so the properties of $\sum$ used here are proved only for $\mathbf{Q}$.

In the first case, $w_i = s_i$. Write $T_n = \sum w_i$ as $T_n = \sum w_{2i} + \sum w_{2i+1}$. The first sum is T_{n-1}, and the second sum is $\sum w_i + \sum w_{i+1} = 2T_{n-1}$; so $T_n = 3T_{n-1}$ hence $T_n = 3^n$. In the second case, $w_i = 1/(s_i s_{i+1})$. The sum of the i first terms is s_i/s_{p+i} provided $i \le q$, (where $p=2^n$, $q=p/2$). Since q is a power of two, the partial sum up to q is $s_q/s_{3q} = 1/2$. The palindrome condition says that the other part has the same sum, so that $T_n = 1$. We finally assume $w_i = s_i/s_{i+1}$. As previously, we separate the cases, where i is even, and the case i is odd. Let $a=s_i$, $b=s_{i+1}$. The first sum is $\sum a/(a+b)$, the second sum is $\sum (a+b)/b = q + \sum a/b$, and it can be evaluated by induction. We split the first sum in two parts $i<q$ and $q\le i$, and use the palindrome condition for the second half; this has as effect to exchange a and b, so we get $\sum b/(a+b)$; adding these things together gives $q/2$. All in all, the result is $(3p-1)/2$ (unless $n=0$, and $p=1$, this is not an integer).

```
Definition fsimpl_sum p i :=
  qsum(fun j => BQinv (BQ_of_nat (fusc (p +c j) *c fusc (p +c (csucc j))))) i.

Lemma csum_fusc_row n: natp n ->
  csumb (interval_co Nat_order (\2c ^c n)  (\2c ^c (csucc n))) fusc
  = \3c ^c n.
Lemma fusc_sum_simpl n (p := \2c ^c  n):
  natp n -> fsimpl_sum p p = \1q.
Lemma qsum_fusc n (p := \2c ^c  n) : natp n ->
 qsum (fun i => fusc_quo (p +c i)) p =
   BQhalf (BQ_of_nat (cpred (\3c *c p))).
```

Let's notice the following remarkable result. We consider the quantities $\binom{i}{j}$ where $i+j=n$. If we sum these quantities, we get the Fibonacci sequence (obvious by induction). If we reduce modulo 2, then sum, we get s_{n+1}. The trick is that $b(2n,2k+1)$ is even, so that $b(2n,2k)$, $b(2n+1,2k+1)$, $b(2n,2k+1)$ and $b(2n+1,2k)$ are $b(n,k)$ modulo 2 (here b is the binomial coefficients, proofs are by induction).

```
Lemma sum_diag_pascal n: natp n ->
  csumb (Nintc n) (fun k => binom (n -c k) k)  = Fib (csucc n).
Definition sum_diag_pascal2 n :=
  csumb (Nintc n) (fun k => (binom (n -c k) k) %%c \2c).
Lemma sum_diag_pascal_prop n: natp n ->
   (sum_diag_pascal2 n) = fusc (csucc n).
```

5. THE SET OF REAL NUMBERS

5.1 Definition

A lattice is an ordered set such that each pair of elements has a least upper bound and a greatest lower bound. In a complete lattice, all subsets (even empty ones) have this property. In Exercise 1.15, Bourbaki asks to prove the existence of a "completion"; this is some subset $\tilde{X}$ of $\mathfrak{P}(X)$, which is a complete lattice for set inclusion, and there is a canonical injection $X \to \tilde{X}$ which is strictly increasing. If X is a totally ordered set, then an element of the completion is a set A such that $t \in A$ and $x \leq t$ implies $x \in A$. The complement B satisfies: if $t \in B$ and $t \leq x$ then $x \in B$. The pair (A, B) is called a *Dedekind cut*.

We shall consider here the ordered set $\mathbf{Q}$, and the set of Dedekind cuts, represented by the B part. This is sometimes called the "extended real number line". It has a least element $-\infty$ (when A is empty) and a greatest element $+\infty$ (when B is empty). Excluding these two elements gives the real line. Let's denote by C_x and C'_x the intervals $]x, \to[$ and $[x, \to[$ (the set of all rational numbers that are respectively $> x$ and $\geq x$). The function C defines the injection $\mathbf{Q} \to \mathbf{R}$. Note that if B has a least element y, it is of the form C'_y; we shall identify this interval with C_y. This means that we shall only consider cuts where B has no least element; if B has an infimum, it is of the form C_x, and is called a rational cut; otherwise, its complement has no supremum and B is called an *irrational* cut.

```
Definition real_dedekind B :=
   [/\ sub B  BQ, nonempty B, B <> BQ,
      (forall x y, inc x B -> x <q y -> inc y B) &
      (forall x, inc x B -> exists2 y, inc y B & y <q x)].
Definition irrationalp B :=  real_dedekind B /\
  (forall x, inc x (BQ -s B) -> exists2 y, inc y (BQ -s B) & x<q y).
Definition rationalp x := real_dedekind x /\ ~ (irrationalp x).
Definition BR_of_Q x := Zo BQ (fun z => x <q z).
```

We define here the notion of adjacent sequences (a, b) (cf. section 4.1) and the associated cut (A, B). For instance, B is the set of all i such that $b_i < x$ for some i, so is the union of the intervals $]b_i, \to[$. Some authors assume the sequences monotone instead of strictly monotone. There are two cases: if the sequence $(b_i)_i$ is not eventually constant, we can extract a strictly decreasing sequence that defines the same cut. On the other hand, if $b_i = b$ for large i, then B is the interval $]b, \to[$, the cut is rational, and we may replace the sequence by $i \mapsto b + 1/(i+1)$ which is a strictly decreasing sequence that gives the same cut.

```
Definition BQpair_aux C :=
   [/\ fgraph C, domain C = Nat,
       forall n, natp n -> inc (Vg C n) (BQ \times BQ),
       forall n, natp n -> P (Vg C n) <q P (Vg C (csucc n))&
       forall n, natp n -> Q (Vg C (csucc n)) <q Q (Vg C n)].

Definition BQpair_aux2a C :=
   forall n, natp n -> P (Vg C n) <q Q (Vg C n).
```

```
Definition BQpair_aux2b C :=
   forall n m, natp n -> natp m -> P (Vg C n) <q Q (Vg C m).
Definition BQpair C  :=  BQpair_aux C /\ BQpair_aux2b C.

Definition BQpairL C :=
   Zo BQ (fun x => exists2 n, natp n & x <q P (Vg C n)).
Definition BQpairR C :=
   Zo BQ (fun x => exists2 n, natp n & Q (Vg C n) <q x).
```

Let (A, B) be the cut associated to the pair of sequences (a, b). If u and v are two rational numbers with $u < v$ then either v is in B, or u is in A, or else, for all i we have $v - u \leq b_i - a_i$. This last case contradicts the condition that the infimum of $b_i - a_i$ is zero. So either $A \cup B$ is $\mathbf{Q}$ (and B is irrational) or it is $\mathbf{Q}$ minus a singleton x (and $B = C_x$). The converse holds: if x is a real number, there is a pair of adjacent sequences such that $B = x$; if x is rational, $a_i = x - 1/(i+1)$, $b_i = x + 1/(i+1)$ is a possibility, $\mathbf{Q}$ is the disjoint union of A, B and $\{x\}$; if x is irrational the axiom of choice is needed and $\mathbf{Q}$ is the disjoint union of A and B. [Note: this requires approximatively 400 lines of script].

```
Definition BQpair_aux3 C :=
   singletonp (BQ -s ((BQpairL C \cup BQpairR C))).

Lemma BQpair_irrational3 x: irrationalp x ->
  exists C, [/\  BQpairR C = x,  BQpair C  & BQpairL C \cup BQpairR C = BQ].
Lemma BQpair_rational2 x: ratp x ->
  exists C, [/\ BQpairR C = BR_of_Q x, BQpair C & BQpair_aux3 C].
```

We show here: if $\sqrt{2}$ (denoted B) is the set of x such that $2 < x^2$, then $\sqrt{2}$ is irrational in the sense given above. Firstly $x^2 = 2$ has no solution since if $x = a/b$ where a and b are coprime, we have $x^2 = a^2/b^2$ where a^2 and b^2 are coprime. We get $a^2 = 2$ and $b^2 = 1$, absurd. Secondly, B has no infimum. Since $(x - \epsilon)^2 - 2 \geq x^2 - 2 - 2\epsilon x$, whenever $x > 0$, it is possible to choose ϵ such that $x - \epsilon \in B$. If we choose ϵ by $x^2 - 2 - 2\epsilon x = 0$ we are led to: if $f(x) = (x^2+2)/(2x)$, then $x \in B$ implies $f(x) \leq x$ and $f(x) \in B$. Note that $f(x) = x$ says $x^2 = 2$, which is impossible. Thirdly, the complement of B has no supremum; it suffices to show that the set A' of all $x > 0$ such that $x^2 < 2$ has no supremum. Our proof is as follows. Take $x = a/b$ in A' and $y = (a+1/4a)/b$; we have $x < y$ and $(by)^2 \leq a^2+1$; since a is an integer, $a^2 < 2b^2$ says $a^2 + 1 \leq 2b^2$, so that we get $y^2 < 2$.

Notes. Let $g(x) = (f(x) + x)/2$. If x is large enough, $x \in A'$ says $g(x) \in A'$ and $x \leq g(x)$. The functions f and g can be used to create a pair of adjacent sequences for $\sqrt{2}$. Since A' is the set of all $2/x$ for x in B, A has a supremum if and only if B has an infimum; the equivalent of f is $h(t) = 4t/(2 + t^2)$. Obviously, $t \leq h(t)$ is equivalent to $t^2 \leq 2$. Moreover $h(t) = \sqrt{2} - \sqrt{2}(t - \sqrt{2})^2/(t^2+2)$, so that $h(t) \leq \sqrt{2}$ (of course, this formula holds on $\mathbf{R}$, but not on $\mathbf{Q}$).

```
Definition BRsqrt2 := (Zo BQps (fun z => \2q <q z *q z)).
Lemma sqrt2_irrational: irrationalp BRsqrt2.
```

5.2 Arithmetic properties

We define $\mathbf{R}$ as the set of all Dedekind cuts, and introduce the canonical inclusion $\mathbf{Q} \to \mathbf{R}$ and some constants, like zero, one, etc. The relation $x \supset y$ makes it a totally ordered set; it is not a complete lattice, but every non-empty bounded set has a supremum or an infimum (intersection and union; note that the intersection could be of the form C'_x, case where C_x has to be used instead). There is no greatest element; in fact, for each real number x, then is an integer n such that $x \leq n$ (take for n the successor of the integer part of any element of x). We say that a real number x is positive if it is a subset of $\mathbf{Q}_+$. This is the same as $0 \leq x$. Since x has no least element, this is also $x \subset \mathbf{Q}_+^*$. Note that $\mathbf{Q}_+^* = 0$. We shall often use the following property: for every real number x, every rational number $\epsilon > 0$, there is y in x such that $y - \epsilon$ is not in x. It implies that $\mathbf{Q}$ is dense in $\mathbf{R}$ (if $x \in \mathbf{R}$, $\epsilon > 0$, there is y in $\mathbf{Q}$ such that $|y - x| < \epsilon$).

```
Definition BR := Zo (powerset BQ) real_dedekind.
Definition BR_zero := BR_of_Q \0q.
Notation "\0r" := BR_zero.
Definition BRp := Zo BR (fun z => sub z BQp).

Definition BR_order := opp_order (sub_order BR).
Definition BR_le x y := [/\ realp x, realp y  & sub y x].
Definition BR_lt x y :=  BR_le x y /\ x <> y.

Lemma rle0xP x: \0r <=r x <-> inc x BRp.
Theorem BR_archimedean  x: realp x ->
    exists2 n, natp n & x <r (BR_of_Q (BQ_of_nat n)).
Lemma BR_sup_exists X: sub X BR -> nonempty X ->
  bounded_above BR_order X -> has_supremum BR_order X.
```

One can define addition, subtraction, multiplication, division, etc, and show the usual properties. The opposite of a real number x is the set of the opposites of the complement of x. If y is rational, and $x = C_y$, this set is C'_{-y}, and C_{-y} has to be used instead.

```
Definition BRopp x := Yo (rationalp x)
   (BR_of_Q (BQopp (BQ_of_R x)))  (fun_image (BQ -s x) BQopp).
Lemma BRopp_Q x: ratp x -> BRopp (BR_of_Q x) = BR_of_Q (BQopp x)
Lemma BRopp_irrational x: irrationalp x ->
   BRopp x = (fun_image (BQ -s x) BQopp).
```

The sum $x + y$ of two real numbers is the set of all $a + b$, where $a \in x$, $b \in y$. Similarly, $x \cdot y$ is the set of all $a \cdot b$, where $a \in x$, $b \in y$, at least when a and b are > 0. The inverse is defined like the opposite (at least for positive arguments). Equipped with these operations, $\mathbf{R}$ is an ordered field. We show here some properties: if $x > 0$, then $x < 1$ is the same as $1 < x^{-1}$, if x and y are non-zero, then $1/x - 1/y = (y - x)/(x \cdot y)$.

```
Definition BRinv x (aux:= fun z => BQps -s fun_image z BQinv) :=
   Yo (rationalp x)
```

```
   (BR_of_Q (BQinv (BQ_of_R x)))
    (Yo (inc x BRps) (aux x) (BRopp (aux (BRopp x)))).

Lemma BRinv_Q x: ratp x -> BRinv (BR_of_Q x) = BR_of_Q (BQinv x).
Lemma BRinv_irrational x (aux:= fun z => BQps -s fun_image z BQinv):
   irrationalp x ->
   BRinv x = (Yo (inc x BRps) (aux x) (BRopp (aux (BRopp x)))).
Lemma BQ_ltinv1 x: inc x BRps ->
   (x <r \1r <-> \1r <r BRinv x).
Lemma BRinv_diff x y: realp x -> realp y ->x <> \0r -> y <> \0r ->
 (BRinv x -r BRinv y) = (y -r x) /r (x *r y).
```

Application: the square of $\sqrt{2}$ is 2. An element of the square of $\sqrt{2}$ is the product of two numbers u and v of $\sqrt{2}$; this product is obviously > 2; conversely, assume $t > 2$; write $t = a/b$, $d = 2a+1$, let n be the integer such that $n^2 \leq abd^2 < (n+1)^2$ and c be n considered in $\mathbf{Q}$. A little computation shows then $u = c/(bd)$ and $v = (ad)/c$ are in $\sqrt{2}$, and the product is t.

```
Lemma BRsqrt2_prop: inc  BRsqrt2 BRps /\ BRsquare  BRsqrt2 = \2r.
```

5.3 Cauchy sequences

We say that a sequence x (with values in $\mathbf{R}$ or $\mathbf{Q}$) has a limit l if for every rational number $\epsilon > 0$, there is an integer N such that if $N \leq n$, then $|x_n - l| < \epsilon$. We say that the sequence is Cauchy, if for every every rational number $\epsilon > 0$, there is an integer N such that if $N \leq n$, and $N \leq m$, then $|x_n - x_m| < \epsilon$. If a limit exists it is unique and the sequence is Cauchy.

```
Definition BR_seq x := [/\ fgraph x, domain x = Nat & sub (range x) BR].
Definition CauchyR x := BR_seq x /\
      forall e, inc e BQps -> exists2 N, natp N &
         forall n m, natp n -> natp m -> N <=c n -> N <=c m ->
            BRabs ((Vg x n) -r (Vg x m)) <r (BR_of_Q e).
Definition limitR x v:=
   forall e, inc e BQps -> exists2 N, natp N &
   forall n, natp n ->  N <=c n -> BRabs ((Vg x n) -r v) <r (BR_of_Q e).

Lemma limitR_unique x v1 v2: BR_seq x -> realp v1 -> realp v2 ->
   limitR x v1 -> limitR x v2 -> v1 = v2.
Lemma CauchyR_when_limit x:
  BR_seq x -> (exists2 y, realp y & limitR x y) -> CauchyR x.
```

An important property satisfied by $\mathbf{R}$ and not $\mathbf{Q}$: every Cauchy sequence has a limit. Consider first a Cauchy sequence of rational numbers y. Let B be the set of all $t \in \mathbf{Q}$ such that, for some N, $y_n < t$ when $n \geq N$. If B has a least element b, then b is the limit of the sequence y. Otherwise B is a real number (note that B is non-empty since the sequence y is bounded), and is the limit of y. Let x be a real Cauchy sequence. We can find (via the axiom of choice) a sequence of rational numbers y such that $|x_n - y_n| < 1/(n+1)$. This sequence is then Cauchy, and its limit is the limit of x. [The proof requires 400 lines of script].

```
Lemma BR_complete s: CauchyR s ->
    exists2 x, realp x & limitR s x.
```

5.4 Continuity

We say that f is continuous at x if the usual epsilon-delta condition is satisfied. This is a local condition. Addition, subtraction, multiplication, division are continuous (since the inverse of zero is zero, this function has a discontinuity there; so that division is continuous in its second argument only when it is non-zero). A non-trivial property is that the product of two continuous functions is continuous; in particular $x \mapsto x^2$ is continuous.

```
Definition BR_near x e y:= realp y /\  BRabs (x -r y) <=r e.
Definition continuous_at f x:=
   forall e, inc e BRps -> exists2 d, inc d BRps &
   forall y,  BR_near x d y ->  BR_near (f x) e (f y).
Definition continuous f:= forall x, realp x -> continuous_at f x.

Lemma continuous_comp f g x:
  continuous_at f x ->  continuous_at g (f x) ->
  continuous_at (g \o f) x.
Lemma continuous2_div x y: realp x -> realp y ->
    (continuous_at (BRdiv ^~y) x) /\ (y <> \0r -> continuous_at (BRdiv x) y).
Lemma continuous_square: continuous BRsquare.
```

Let f be a continuous function on $[x, y]$ (it is semi-continuous at x and y); if v is between $f(x)$ and $f(y)$, then there is z between x and y such that $f(z) = v$. Proof. We may assume $f(x) \leq f(y)$ and $v = 0$. Let E be the set of all t in the interval $I = [x, y]$ such that $f(t) \geq 0$. This is a non-empty bounded set thus has an infimum z in I. Assume $f(z) > 0$. This says $z \neq x$, so that there exists t (near z) such that $x < t < z$ with $f(t) > 0$. This says $t \in E$, absurd. Assume $f(z) < 0$; this says $z \neq y$. There is $\epsilon > 0$ such that $z + \epsilon < y$ and that f is negative on $[z, z + \epsilon]$. No element of this interval is in E, so that $z + \epsilon$ is a lower bound of E, contradicting the fact that z is the greatest lower bound. It follows $f(z) = 0$.

```
Definition Bolzano_hyp f x y:=
    [/\ continuous_right f x, continuous_left f y &
    (forall z, x <r z -> z <r y -> continuous_at f z)].
Lemma Bolzano2 f x y v:  x <=r y -> Bolzano_hyp f x y ->
    (BR_between v (f x) (f y)) ->
    exists2 z, (x <=r z /\ z <=r y) & f z = v.
```

Application: each positive real number x has a square root. We have $(-y)^2 = y^2$ but the square root is unique in $\mathbf{R}_+$. By uniqueness, the square root of 2 is equal to the quantity $\sqrt{2}$ introduced above.

```
Lemma BRsqrt_exists x: inc x BRp -> exists2 y, inc y BRp & x = BRsquare y.
Definition BRsqrt x := select (fun z => x = BRsquare z) BRp.
Lemma sqrt2_prop : BRsqrt2 = BRsqrt \2r.
```

Assume that f is a function such that $a \leq t$ implies $a \leq f(t) \leq t$. Fix $x_0 \geq a$ and define x_n by $x_{n+1} = f(x_n)$. Then the sequence $(x_n)_{n\in\mathbf{N}}$ is decreasing hence has a limit x (namely $\inf_i(x_i)$) such that $a \leq x$. If f is continuous at x, then $f(x) = x$. [Note: by assumption $f(a) = a$ so that the limit could be a; if $f(t) = 1/(1+(t-1))^2$, the limit is $b = 1$ when $b \leq x_0 \leq 2$, and $a = 0$ otherwise.]

```
Lemma decreasing_bounded_limit a xn (x := infimum BR_order (range xn)):
  BR_seq xn ->
  (forall n, natp n -> a <=r (Vg xn n)) ->
  (forall n, natp n -> Vg xn (csucc n) <=r  Vg xn n) ->
  (a <=r x /\ limitR xn x).
Lemma decreasing_limit_bounded_fix a x0 f
  (seq:= induction_defined f x0)  (xn := Lg Nat (Vf seq))
  (x := infimum BR_order (range xn)):
  (forall x, a <=r x ->  a <=r f x /\  f x <=r x) -> a <=r x0 ->
  (continuous_at f x) ->
  [/\ a <=r x, f x = x & limitR xn x].
```

Application. We define here a pair of adjacent sequences defining $\sqrt{a}$. In fact we assume a real positive, so that the members of the sequence are real, but if a were rational, then the sequence would be rational as well. The first sequence corresponds to the Newton Scheme for solving $x^2 = a$. Let $f(x) = (x^2 + a)/(2x)$. Let B be the set of all positive x such that $x^2 \geq a$. We have $x - f(x) = (x^2 - a)/(2x)$, so that, if $x \in B$ we have $f(x) \leq x$. We have $f(x)^2 - a = (x^2 - a)^2/(4x^2)$, so that, if $x > 0$, then $f(x) \in B$. If $x_{n+1} = f(x_n)$, where x_0 is big enough (say $\geq 1 + a$) then x_n converges to some x (the infimum of the range). If the limit is zero then a must be zero; otherwise f is continuous at x, so that $f(x) = x$ and $x^2 = a$.

Let now $g(t) = (f(t) + t)/2$ and $y_{n+1} = g(y_n)$, Let A be the set of all positive t such that $a/9 \leq t^2 \leq a$. We simplify our reasoning by assuming that a has a square root b, and define A as the set of all t such that $b/3 \leq t \leq b$. We have $t - g(t) = (t - f(t))/2$, so that $g(t) \leq t$ when $t \in B$, thus $t \leq g(t)$ when $t \in A$. We have $g(t) - b = (t - b)(3t - b)/4t$. This says that A is invariant by g. Thus if $y_0 \in A$, the sequence $(y_n)_n$ is increasing in A; its supremum y is a fixed point of g, thus of f, thus is b. [Size of the proof script: 280 lines.]

```
Lemma square_root_cv1 a b (f := fun z => (BRsquare z +r a) /r (\2r *r z))
  (seq:= induction_defined f b)  (xn := Lg Nat (Vf seq))
  (x := infimum BR_order (range xn)):
  inc a BRp -> \1r +r a <=r b ->
  [/\ inc x BRp, limitR xn x & BRsquare x = a].
Lemma square_root_cv3 a b (f := fun z => (BRsquare z +r a) /r (\2r *r z))
  (g := fun z => BRhalf ((f z) +r z)) (s := BRsqrt a)
  (seq:= induction_defined g b)  (xn := Lg Nat (Vf seq))
  (x := supremum BR_order (range xn)):
  inc a BRp ->  (s /r \3r) <=r b -> b <=r s ->
  [/\ inc x BRp, limitR xn x & x = s].
```

6. CONCLUSION

Notes on the Implementation

The work presented here corresponds to 8 files (*sset7*, *sset8*, *sset9*, *sset10*, *ssetz*, *ssetq1*, *ssetq2*, *ssetr*), 360 definitions, 71 notations, 2700 lemmas and 34,000 lines of code (16,000 lines for **N** and 18,000 for the other data structures). It implements the basic properties of **N**, **Z** and **Q**, presents the arithmetic properties of **R** and some of its analytic properties, with a focus on the properties of order. It represents 20% of the Gaia library. The code is available on the Web, under http://www-sop.inria.fr/marelle/gaia.

In [Gri10] we wrote "Coq was much slower on **Q** than on **Z**"; in particular the two tactics 'auto with fprops' and 'autorewrite with aw' were sometimes very slow. The first remedy was to use a second data base and add tactics 'auto with qprops' and 'autorewrite with qw'. We then removed the most costly calls to these tactics, and in the current version, we removed the second database, and the implementation of **Z**, **Q** and **R** adds no hint to the first database. This is inspired from the ssreflect library, which very rarely uses auto.

We also noticed a slowdown when replacing one definition of addition on **Z** by a variant; the exact reason is unclear: is it due to a limited number of lemmas that became much slower, or a great number of lemmas that became just a bit slower? For instance, in the proof of $(a+b)^2 = 4ab + (a-b)^2$ on **Q**, which is rather straightforward, replacing trivial by reflexivity reduces compilation time from 2.5 seconds to 0.5 second. On the other hand, a previous version of Gaia used the 'Module Type' trick of ssreflect in order to make the cardinal function completely opaque. This is no more needed.

Comparison with the Coq library

As shown in the text, defining a function $f : \mathsf{nat} \to \mathbf{N}$, and showing that it is bijective is quite easy; this obviously preserves the structures (addition, multiplication, comparison). Using the axiom of choice (for the non-empty type nat), one can introduce the inverse of this function (but we never needed it in our development). Similarly, we have defined a bijection $g : \mathsf{int} \to \mathbf{Z}$, and proved that it respects the structures; in particular, it respects divisibility and gcd, so that defining $h : \mathsf{rat} \to \mathbf{Q}$ should be straightforward as well (work in progress).

The case of real numbers is more challenging: if $x : \mathsf{R}$ is a real number (according to the standard library of Coq), it is the limit of a sequence of rational numbers x_i. One can cast this into a Cauchy sequence in **Q**, and take its limit $\bar{x} \in \mathbf{R}$; one has to show that this number is independent of the chosen sequence. Conversely, given $\bar{x} \in \mathbf{R}$, associated to a Dedekind cut (A, B), it should be possible to consider A as a "bounded subset" of R and use the completeness axiom to get a real x. One has to prove that this is the inverse function of the preceding one, and that it respects the operations. [This is a possible extension of our library].

All the code described in section 4.2 has been implemented both in Gaia and directly in ssreflect, using the int and rat type, see [Gri14]. Some comments:

—When dealing with a non-trivial recurrence relation of the form (4), the Gaia definition is trivial, and one has to prove *a posteriori* that it makes sense, on the other hand Coq requires a measure (an *a priori* proof of termination) or a bound on the number of operations, and a proof that the bound is big enough.

—In Coq, we can use the list data structure, instead of a functional graph on I_n; for instance, the expansion to base b is a list a and $\sum a_i b^i$ is defined by the Horner Scheme; the second relation of (2), namely $E_{n+1}(a) = a_0 + bE_n(a')$ becomes trivial, and the first is more complicated. Defining the base two reverse is easier: it suffices to revert the list.

—Assume that we want to count something, for instance, the number of integers $< n$ such that the binary representation is formed of some digits 1, followed by some digits 0; the number of integers $< n$ whose base ten expansion contains only even digits; the number of ways to express an integer as a sum of distinct Fibonacci numbers; the number of ways to write n in base two, using 0, 1 and 2 as digits; the number of surjective functions $I_n \to I_m$. In each case, the objective is to compute $c_n = \mathrm{card}(E_n)$ for some finite set E_n. The ssreflect library provides a theory of finite sets, in which these cardinals can be computed. The difficulty is that the theory requires a finite type T such that $E_n \subset T$. In the first two cases, one can take $T = I_n$; but this choice makes proofs by induction impossible; so one takes $T = I_m$ for some $m \geq n$, this gives a cardinal $c_{n,m}$, and an additional task: prove that is it independent of m. The other cases are similar, a bit more complicated.

—Some exercises of section 5 of [Bou68] ask to prove some formulas on $\mathbf{Z}$ by using different ways to compute the cardinal of a same set; for instance $\sum_k (-1)^k \binom{n}{k} = 0$ says that there are as many subsets of I_n with even and odd cardinal. Showing this in ssreflect should be easy (there are of course alternate proofs). In the current version of Gaia, we cannot even express the formula; so one of our priorities is to develop a generic sum on $\mathbf{Z}$, in order to solve more such exercises.

Comparison with Bourbaki

A fondamental difference between Bourbaki and Coq is the type system: there are two types in Bourbaki: set and relation, and one can only quantify over sets, while Coq has an arbitrary number of types. This means that Criterion C61 and C62 are written in English and not in mathematical language of the form: $\forall r, r(0) \wedge H(r) \implies C(r)$ (for some H and C). Note that r is a relation, and replacing the free variable n by 0 gives the relation $r(0)$. It is forbidden to replace n by a relation. In Gaia we make the distinction between Prop (relations without free variables) and Set→ Prop (relations with one free variable) and we allow Prop→ Prop (relations with one free variable that is a relation). We make a distinction between $x = x$ (a relation with a free parameter) and $\forall x, x = x$ (a theorem).

Our current implementation of cardinals uses von Neumann ordinals, rather than the Axiom of Choice. This allowed us to weaken our Axiom of Choice (Scheme S7 is not an axiom in Gaia; in other terms, $\forall x, P \iff Q$ does not imply $\tau_x P = \tau_x Q$). [On occasions, Bourbaki is a bit conservative, for instance, the only English edition of the Theory of Sets, translated in 1968, uses an axiom for the ordered pair,

although Kuratowski introduced its definition in 1921]. Bourbaki introduces his Axiom of Infinity (hence the set of natural integers) in section 6, page 183, while "integer" is defined on page 166 and the principle of induction (criterion C61) on page 168. As the Axiom of Infinity is implicit in Gaia, we introduced **N** much earlier. There is no other fundamental difference between the Bourbaki integers and those of Gaia.

Implementing **Z** as explained by Bourbaki is more challenging. It imposes to define the notion of: monoid, group, homomorphism, group of differences, etc (20 pages of the Book of Algebra). This is technically possible, and corresponds to a part of the *ssralg.v* file of SSREFLECT. To go further (establish some properties of **Z** and define **Q**), one can either implement the whole Chapter 1 of the Book of Algebra (120 pages) or select some specific topics. In any case, it requires to implement the equivalent of *ssralg.v*, including canonical structures, *bigop.v*, etc. Now, some properties of **Q** (for instance that it is an ordered field) is defined in Chapter 6 (still more things to implement, for instance, the equivalent of *ssrnum.v*). The fact that every rational number can unique be written as a/b where a and b are coprime is explained in another Chapter. Implementing the whole Book of Algebra (or just the part needed for **Q**) using an untyped theory like Bourbaki makes little sense, as it can never be as simple or efficient as the SSREFLECT library.

The case of real numbers is a bit different. Bourbaki introduces **R** as a topological field (obtained by completion of **Q**) and studies it in the Book of Topology as a topological object; many other properties are found in the Book "Functions of a Real Variable". There is so much material in these books that implementing them would take forever; it is much wiser to start with a simple definition (for instance, that of the standard library of COQ), and extend it, little by little, with independent modules.

ACKNOWLEDGMENT

We wish to thank the anonymous referee for his valuable suggestions.

References

[BLM14] Sylvie Boldo, Catherine Lelay, and Guillaume Melquiond. Formalization of Real Analysis: A Survey of Proof Assistants and Libraries. *Mathematical Structures in Computer Science*, page 38, 2014. https://hal.inria.fr/hal-00806920.

[Bou68] N. Bourbaki. *Elements of Mathematics, Theory of Sets*. Springer, 1968.

[Bou70] N. Bourbaki. *Éléments de mathématiques, Théorie des ensembles*. Diffusion CCLS, 1970.

[Bou89a] N. Bourbaki. *Elements of Mathematics, Algebra I*. Springer, 1989.

[Bou89b] N. Bourbaki. *Elements of Mathematics, General Topology*. Springer, 1989. Translation of the 1966 French edition.

[Can97] Georg Cantor. *Contributions to the Founding of the Theory of Transfinite Numbers*. Dover Publications Inc, 1897. Trans. P. Jourdain, 1955.

[Coh12] Cyril Cohen. Construction of real algebraic numbers in Coq. In Lennart Beringer and Amy Felty, editors, *ITP - 3rd International Conference on Interactive Theorem Proving - 2012*, Princeton, United States, August 2012. Springer. https://hal.inria.fr/hal-00671809.

[Dij76a] Edsger W. Dijkstra. An exercise for Dr. R.M.Burstall. Available at http://www.cs.utexas.edu/users/EWD/ewd05xx/EWD570.PDF, May 1976.

[Dij76b] Edsger W. Dijkstra. More about the function "fusc" (a sequel to EWD570). Available at http://www.cs.utexas.edu/users/EWD/ewd05xx/EWD578.PDF, August 1976.

[GGMR09] François Garillot, Georges Gonthier, Assia Mahboubi, and Laurence Rideau. Packaging mathematical structures. In *Theorem Proving in Higher Order Logics, Lecture Notes in Computer Science 5674*, 2009. https://hal.inria.fr/inria-00368403.

[GM10] Georges Gonthier and Assia Mahboubi. An introduction to small scale reflection in Coq. *Journal of Formalized Reasoning*, 3(2):95–152, 2010. https://jfr.unibo.it/article/view/1979.

[Gri09] José Grimm. Implementation of Bourbaki's Elements of Mathematics in Coq: Part Two; Ordered Sets, Cardinals, Integers. Research Report RR-7150, INRIA, 2009. http://hal.inria.fr/inria-00440786/en/.

[Gri10] José Grimm. Implementation of Bourbaki's Elements of Mathematics in Coq, part one theory of sets. *Journal of Formalized Reasoning*, 3(1):79–126, 2010.

[Gri14] José Grimm. Fibonacci numbers and the Stern-Brocot tree in Coq. Research Report RR-8654, Inria Sophia Antipolis, December 2014.

[NB04] Milad Niqui and Yves Bertot. Qarith: Coq formalisation of lazy rational arithmetic. In Stefano Berardi, Mario Coppo, and Ferruccio Damiani, editors, *Types for Proofs and Programs*, volume 3085 of *Lecture Notes in Computer Science*, pages 309–323. Springer Berlin Heidelberg, 2004.

[Niq07] Milad Niqui. Exact arithmetic on the Stern-Brocot tree. *Journal of Discrete Algorithms*, 5(2), 2007.

[Sim04a] Carlos Simpson. Computer theorem proving in math. Technical report, CNRS, Laboratoire J.A. Dieudonne, 2004. arXiv:math/0311260v2.

[Sim04b] Carlos Simpson. Set-theorical mathematics in Coq. Technical report, CNRS, Laboratoire J.A. Dieudonne, 2004. arXiv:math/0402336v1.

[Ste58] Moritz Stern. Ueber eine zahlentheorische Funktion. *Journal für die reine und angewandte Mathematik*, 55:193–220, 1858.