Un Calcul des Constructions implicite avec sommes dépendantes et à inférence de type décidable.

Bruno Bernardo

THÈSE DE DOCTORAT
ÉCOLE POLYTECHNIQUE

Spécialité
Informatique Fondamentale

Présentée par
Bruno Bernardo

Pour obtenir le grade de
Docteur de l'École polytechnique

Sujet de la thèse
Un Calcul des Constructions implicite avec sommes dépendantes et à inférence de type décidable

Soutenue publiquement le 18 septembre 2015 devant le jury composé de

MM.	Bruno Barras	Co-directeur de thèse
	Gilles Barthe	Rapporteur
	Gilles Dowek	Co-directeur de thèse
	Alexandre Miquel	Rapporteur
	François Pottier	Examinateur
	Benjamin Werner	Examinateur

0855e84 2015-09-10 23:03:06 +0200

Sommaire

Introduction

L'informatique peut être définie comme la science du calcul. Parmi ses différentes branches, la théorie des langages de programmation étudie les systèmes formels permettant d'exprimer des processus calculatoires appelés algorithmes ou programmes. De nombreux langages de programmation sont basés sur un formalisme appelé λ-calcul.

Le λ-calcul, ses extensions et ses applications

Le λ-calcul a été inventé par le mathématicien américain Alonzo Church en 1932 [**Chu32**]. L'ambition première de Church était de proposer un système logique pouvant servir de fondation aux mathématiques et basé sur la notion de fonction plutôt que d'ensemble. Ce projet a échoué : l'incohérence du système de Church a été démontrée en 1935 (paradoxe de Kleene-Rosser [**KR35**]). Church a par contre montré ensuite que la partie purement fonctionnelle du λ-calcul (sans les règles logiques) est une représentation de toutes les fonctions calculables [**Chu36**]. Puis en 1940, il définit une version restreinte de son système de 1932, le λ-calcul simplement typé [**Chu40**]. Dans ce système est définie une notion de *type* qui permet de distinguer plusieurs classes de termes Ce système permet de définir un autre formalisme logique, la logique d'ordre supérieur, dont l'incohérence n'a pas été prouvée.

Ces travaux pionniers montrent la double vocation du λ-calcul comme langage de programmation (en tant que formalisme représentant des fonctions calculables) et comme système logique. Une autre étape importante a été la découverte par Curry [**CF58**] et Howard [**How80**] de la possibilité d'assimiler les types du λ-calcul simplement typé à des propositions de la logique minimale intuitionniste et les termes à des démonstrations. Ce principe reliant de manière profonde les notions d'algorithme et de démonstration est appelé *isomorphisme de Curry-Howard*.

La double interprétation du λ-calcul simplement typé a suscité la curiosité de logiciens et informaticiens et de nombreuses extensions du λ-calcul simplement typé ont été proposées : types polymorphes, types dépendants, ordre supérieur, univers, types inductifs… Ces extensions permettent d'avoir des systèmes logiques plus puissants : arithmétique du second ordre pour les types polymorphes, logique des prédicats pour les types dépendants, logique d'ordre supérieure… Elles permettent d'avoir des langages de programmation plus expressifs : les types polymorphes permettent de représenter des programmes paramétriques (par exemple, calculer la longueur d'une liste d'éléments d'un certain type quel que soit ce type), les types dépendants permettent d'avoir des spécifications plus fines (les types peuvent dépendre des termes),… Ainsi les extensions les plus riches, comme le Calcul des Constructions, qui a réussi à unifier le polymorphisme imprédicatif de Système F avec les types dépendants de la théorie des types de Martin-Löf prédicative, permettent de faire de la preuve formelle ou de la programmation certifiée. En effet, le système logique associé est suffisamment puissant pour exprimer et prouver des résultats mathématiques d'une part et, d'autre part, en tant que langage de programmation, le langage des types permet, d'après l'isomorphisme de Curry-Howard, de spécifier mathématiquement le comportement attendu d'un programme.

Problématique de la thèse

Notre thèse s'intéresse aux λ-calculs typés en tant que formalismes pour la programmation certifiée. Concevoir un programme vérifiant une spécification fine ne va pas sans inconvénients. En effet, un tel programme ne va pas être purement algorithmique : il va contenir également des informations permettant de garantir que la spécification est respectée. Nous distinguons alors la partie *calculatoire* de la partie *logique* d'un programme. La partie calculatoire consiste en l'algorithme proprement dit, c'est-à-dire le processus traitant les arguments d'entrée et retournant les arguments de sortie. La partie logique regroupe toutes les indications données afin que le terme vérifie sa spécification, c'est-à-dire qu'il soit d'un certain type. Ces dernières sont utiles statiquement pour garantir la sûreté du programme mais nuisibles dynamiquement car elles rendent l'exécution du programme plus lente et plus gourmande en mémoire.

Le but de nos travaux est de rendre possible la conception de programmes sûrs qui ne soient pas alourdis par des informations statiques.

L'assistant de preuves Coq. L'assistant de preuve Coq [**The15**] implémente un Calcul des Constructions Inductives (CIC pour *Calculus of Inductive Constructions*) [**PM96, Wer94, PPM90**]. La présence de types inductifs permet d'exprimer commodément les types de données utilisés dans les langages de programmation usuels. Coq a ainsi été utilisé pour certifier des programmes de grande envergure comme des compilateurs [**Ler06**] ainsi que pour prouver formellement des résultats mathématiques conséquents [**Gon07, GAA+13**]. La syntaxe de CIC est à la Church, [1] ce qui rend le typage décidable mais implique que les programmes de Coq vont contenir des indications logiques explicites. Il existe par contre un mécanisme d'extraction [**PM89, Let04, Glo12**] qui permet d'effacer une partie de ces informations. Ce mécanisme, externe au système, repose sur la distinction entre deux sortes du formalisme : Prop et Set. Il associe à tout terme Coq un programme d'un langage de programmation fonctionnel (OCaml, Haskell ou Scheme) en effaçant les parties typés par un type de type Prop. L'extraction est un mécanisme externe au système et sa correction repose sur des résultats théoriques de réalisabilité. Une limitation de ce mécanisme est que certaines indications logiques n'ont pas forcément un type de type Prop. Ainsi par exemple le type des entiers naturels `Nat` est de type Set. Or des entiers naturels peuvent être des arguments non-calculatoires : c'est le cas par exemple de la longueur des vecteurs (listes dépendantes).

Le Calcul Implicite de Miquel comme langage de programmation certifiée. Alexandre Miquel a défini dans sa thèse un Calcul des Constructions avec Univers enrichi d'un produit dépendant implicite. Le système de Miquel est un λ-calcul typé dont la syntaxe est à la Curry : l'abstraction n'est pas annotée et les termes qui ne sont pas des types sont en fait des termes du λ-calcul pur. La principale originalité de ce système est la présence d'un nouvel opérateur de type : le produit dépendant implicite. Ce produit, qui peut se comprendre comme un type intersection, n'a ni constructeur, ni eliminateur, ce qui permet de rendre implicites certains arguments. Cela permet à ce système, suffisamment riche pour la preuve formelle ou de la programmation certifiée, d'exprimer des termes purement calculatoires respectant une spécification fine : les annotations de types sont absentes et les arguments logiques comme les preuves peuvent être rendus implicites. Le système de Miquel est par contre très vraisemblablement indécidable, ce qui limite son utilisation : il n'existe pas de procédure qui reconnaisse tous les programmes valides.

Une approche hybride. La démarche que nous proposons cherche à combiner les avantages de ICC et de Coq. Ainsi le formalisme de Miquel avec un produit dépendant implicite nous semble être un bon cadre pour la programmation certifiée car il permet de rendre les parties logiques implicites.

1. Les termes ont des annotations de type.

Mais ce serait un meilleur formalisme si, comme Coq, son système de types contenait des types inductifs et si le typage était décidable.

Notre approche consiste à définir deux systèmes de types. Le premier est un système de types analogue à ICC (avec un système de types plus riche) où les parties statiques peuvent être implicites et le typage vraisemblablement indécidable. Le second est un système à la Church, où les parties considérées comme non-calculatoires sont annotées. Il est muni d'un mécanisme *interne* d'extraction qui associe à chaque terme un terme du premier système. Ce second système doit avoir deux caractéristiques. D'une part, son typage doit être décidable. D'autre part, il doit être une *représentation* du premier système : tout programme valide de l'un correspond à un programme valide de l'autre. Ainsi l'utilisateur va pouvoir élaborer son programme dans le système décidable et obtenir, après extraction, un programme du premier système où les parties statiques sont effacées.

Dans [**BB08**], nous avons proposé une représentation décidable du système de types d'Alexandre Miquel. Pour cela nous avons défini un calcul des constructions avec une syntaxe à la Church bicolore [2] et un mécanisme d'extraction qui transforme un terme en un terme du système de Miquel. Les parties qui seraient implicites dans le système de Miquel sont présentes explicitement - ce qui rend décidable le typage - mais marquées et ainsi effacées par extraction. Une caractéristique clé de notre système est sa règle de cumulativité qui compare des termes extraits et non des termes complets du langage. Nous montrons que notre système a un typage décidable et qu'il exprime exactement les mêmes termes que le système de Miquel. Il permet donc d'obtenir des programmes certifiés purement calculatoires : puisque le typage est décidable, il est possible d'implémenter un vérificateur de type qui permettra de certifier que les termes annotés respectent bien une spécification donnée ; ces termes sont annotés et contiennent des parties logiques mais leur extraction est purement calculatoire et respecte une spécification analogue.

Présentation de la thèse

Dans cette thèse nous étudions l'application de la démarche hybride présentée ci-dessus au calcul des constructions avec sommes dépendantes.

Cela est une première étape dans l'ajout des types inductifs. En effet il est possible de décomposer les types inductifs à partir d'éléments de base [3] [**ML85, Men88**] dont font partie les sommes dépendantes.

Cette thèse se divise en quatre parties.

(1) La première est consacrée à la présentation syntaxique du Calcul des Constructions implicite avec sommes dépendantes, système que nous noterons ICC_Σ. Ce système est l'extension du calcul implicite de Miquel. Le premier chapitre présente la syntaxe et les propriétés métathéoriques usuelles des PTS. Le deuxième chapitre traite de la préservation du typage, dont l'étude est sensiblement plus compliquée que dans les PTS habituels et même que dans ICC.

(2) La deuxième est consacrée à l'étude sémantique de ICC_Σ. Miquel avait proposé deux modèles pour ICC_Σ permettant respectivement de montrer la cohérence et la normalisation forte de ICC_Σ. Nous proposons une extension de son modèle de cohérence. Le modèle de

2. nous avons donc deux produits dépendants, deux abstractions et deux applications qui correspondent respectivement à une version explicite et implicite

3. Les sommes dépendantes, l'opérateur de point fixe, l'opérateur de récursion, l'égalité, les types singleton et vide chez Mendler, chez Martin-Löf les W-types remplacent les opérateurs de point fixe et l'opérateur de récursion

normalisation forte n'est pas traité. Cette partie contient trois chapitres. Le premier détaille un modèle abstrait de cohérence. Le deuxième présente des outils utilisés pour l'implémentation d'un modèle concret. Le dernier chapitre présente le modèle concret qui instancie le modèle abstrait, ce qui démontre finalement la cohérence de ICC_Σ.

(3) La troisième partie présente la représentation décidable de ICC_Σ. Ce système est appelé Calcul des Constructions implicite annoté avec sommes dépendantes. Nous le noterons $AICC_\Sigma$. Cette partie contient deux chapitres. Le premier décrit la syntaxe de ce système et prouve les propriétés métathéoriques usuelles des PTS. Le deuxième montre la propriété du relèvement : pour tout terme bien typé de ICC_Σ, il existe un terme bien typé de $AICC_\Sigma$ qui s'extrait vers lui. Cette propriété est très importante car elle prouve que tous les termes valides de ICC_Σ peuvent être représentés dans ICC_Σ, ce qui achève la démonstration que $AICC_\Sigma$ est bien une représentation de ICC_Σ.

(4) La quatrième partie démontre [4] que le typage dans $AICC_\Sigma$ est décidable. Elle contient deux chapitres. Le premier présente un algorithme d'inférence de type qui retourne un type non-annoté. Cet algorithme est défini sans aucun ajout au système et permet de définir un algorithme de vérification de type, où le type vérifié est annoté. Le deuxième chapitre présente un algorithme d'inférence de type qui retourne un type annoté de $AICC_\Sigma$. Définir un tel algorithme nécessite de rajouter des règles de réduction de termes annotés.

4. sous réserve de conjectures qui seront détaillées

Première partie

Syntaxe de ICC_{Σ}

Chapitre 1

Présentation syntaxique de ICC_Σ

ICC_Σ est une extension d'une version légèrement modifiée de, du calcul des constructions implicite (ICC) défini par Miquel dans sa thèse [**Miq01**]. Miquel définit un calcul des constructions ([**Coq85, CH88**]) avec, comme dans CC_ω [**Coq86**]), une hiérarchie d'univers emboîtés et une relation de cumulativité entre ces univers. Le système de Miquel se distingue toutefois nettement de CC_ω sur deux points : d'une part, sa syntaxe est à la Curry (pas d'annotation de type dans les abstractions), d'autre part il introduit une nouvelle construction primitive, le *produit dépendant implicite*, noté $\forall x : \mathrm{T}.\mathrm{U}$. Contrairement au produit dépendant habituel, le produit dépendant implicite n'a ni constructeur ni éliminateur. Intuitivement, il ressemble à l'opérateur de polymorphisme pour Système F ([**Gir72, Rey74**]) à la différence près que dans Système F, le polymorphisme n'agit que sur des variables de type, alors que dans ICC_Σ, le produit implicite agit sur n'importe quelle variable de terme. Ses règles d'introduction et d'élimination sont semblables à celles du polymorphisme imprédicatif d'une version à la Curry du Système F [**Lei83, MS82, Mit88**] où l'abstraction de type et l'application de type se font implicitement. Comme le remarquent Barthe et Coquand dans [**BC00**], le Calcul Implicite de Miquel combine la démarche des *Domain-Free Pure Type Systems* [**BS00**] et des Type assignment systems [**Bar91**].

ICC_Σ étend ICC en lui ajoutant les sommes dépendantes. Pour cela, nous allons enrichir le langage de termes (section 1), ajouter les règles de réduction liées aux sommes dépendantes et adapter la relation de cumulativité (section 2), compléter les règles de typage avec des règles de formation, introduction et élimination des sommes dépendantes (section 3) et enfin démontrer pour ICC_Σ quelques résultats métathéoriques simples déjà prouvés pour ICC (section 3.4).

1. Syntaxe

Nous introduisons dans cette section les différents termes du langage. Nous expliquerons plus en détail leur signification lors de la présentation des règles de typage (section 3).

1.1. Sortes et variables. De même que ICC, ICC_Σ se base sur une hiérarchie infinie dénombrable de sortes (ou univers dans la terminologie de la théorie des types de Martin-Löf).

1.1.1. DÉFINITION (Sortes). L'ensemble des sortes, noté Sort, est défini par :

$$\mathsf{Sort} = \{\mathsf{Prop}\} \cup \{\mathsf{Type}_i \mid i > 0\}$$

Nous avons une sorte imprédicative Prop et une infinité dénombrable de sortes prédicatives $(\mathsf{Type}_i)_{i>0}$. Notons l'absence dans ICC_Σ de la sorte imprédicative Set présente dans ICC. Cette sorte fait à nos yeux doublon avec Prop. Elle était cependant justifiée dans ICC_Σ par le fait que l'assistant de preuves Coq avait à l'époque une sorte Set prédicative. Ce n'est plus le cas aujourd'hui.

1.1.2. DÉFINITION (Variables). Nous considérons un ensemble infini dénombrable de variables, noté Var.

1.1.3. CONVENTION. Les expressions u, v, x, y, z désigneront des variables. Les expressions s, s', s_i (avec i entier naturel) désigneront des sortes.

1.2. Termes.

1.2.1. DÉFINITION (Termes). L'ensemble des termes de ICC_Σ est noté $\Lambda_{\mathrm{ICC}_\Sigma}$. Il est défini par la grammaire décrite dans la figure 1.

$\mathrm{M,N,T,U}$	::=	x	(Variable)
	|	s	(Sorte)
	|	$\Pi x:\mathrm{T}.\mathrm{U}$	(Produit)
	|	$\lambda x.\mathrm{M}$	(Abstraction)
	|	MN	(Application)
	|	$\forall x:\mathrm{T}.\mathrm{U}$	(Produit implicite)
	|	$\Sigma x:\mathrm{T}.\mathrm{U}$	(Somme dépendante)
	|	(M,N)	(Paire dépendante)
	|	$\mathsf{Elim}_\Sigma(xy.\mathrm{M},\mathrm{N})$	(Eliminateur de Somme dépendante)
	|	$\exists x:\mathrm{T}.\mathrm{U}$	(Existentielle)
	|	$(\diamond,\mathrm{M})$	(Paire existentielle)
	|	$\mathsf{Elim}_\exists(y.\mathrm{M},\mathrm{N})$	(Eliminateur d'existentielle)
	|	$\{x:\mathrm{T}\mid\mathrm{U}\}$	(Sous-ensemble)

FIGURE 1. Syntaxe des termes de ICC_Σ

En plus des variables et des sortes, le langage contient, comme dans le calcul des constructions, le produit dépendant $\Pi x:\mathrm{T}.\mathrm{U}$, ainsi que son constructeur, l'abstraction $\lambda x.\mathrm{M}$, et son éliminateur, l'application MN. Notons que dans l'abstraction, la variable abstraite n'a pas d'annotation de type. La nouveauté de ICC est le produit dépendant implicite $\forall x:\mathrm{T}.\mathrm{U}$, que nous avons déjà présenté.

La syntaxe de ICC_Σ étend celle de ICC en ajoutant les sommes dépendantes $\Sigma x:\mathrm{A}.\mathrm{B}$. Elles ont comme constructeurs les *paires dépendantes* (a,b) et comme éliminateurs les *éliminateurs de somme dépendante* $\mathsf{Elim}_\Sigma(xy.f,c)$. Dans la continuité de la syntaxe de ICC_Σ, qui est à la Curry, les paires n'ont pas d'annotation de type et les variables des éliminateurs ne sont pas non plus annotées. Nous ajoutons deux autres sommes dépendantes le type *sous-ensemble* $\{x:\mathrm{A}\mid\mathrm{B}\}$ et le *type existentiel* $\exists x:\mathrm{A}.\mathrm{B}$. Le type sous-ensemble n'a ni constructeur ni éliminateur. Le type existentiel a un constructeur $(\diamond,b)$ et un éliminateur $\mathsf{Elim}_\exists(y.f,c)$.

1.2.2. CONVENTION. L'abstraction de variables est une opération associative à gauche et l'application de termes est une opération associative à droite. Ainsi $\lambda x.\lambda y.\mathrm{M}=\lambda x.(\lambda y.\mathrm{M})$ et $\mathrm{MNR}=(\mathrm{MN})\,\mathrm{R}$. Par ailleurs, l'application de termes a la priorité sur l'abstraction de variable. Ainsi $\lambda x.\mathrm{MN}=\lambda x.(\mathrm{MN})$.

La classification suivante des termes de ICC_Σ sera utilisée tout au long de ce document.

1.2.3. DÉFINITION (Classification des termes). Soit M un terme de ICC_Σ.

- M est un *constructeur de type* s'il est un produit dépendant, un produit implicite, une somme dépendante, un type existentiel, un sous-ensemble ;
- M est un *constructeur* s'il est une abstraction, une paire dépendante ou une paire existentielle ;

- M est un *éliminateur* s'il est une application, un éliminateur de somme dépendante ou un éliminateur d'existentielle.

1.2.4. REMARQUE. Cette classification, usuelle dans la littérature, est liée aux règles de typage. Aussi sa signification et sa cohérence apparaîtront plus clairement dans la section 3.

1.2.5. NOTATION (Constructeur de type générique). Lorsque nous considérons un constructeur de type quelconque, nous utilisons le symbole $\square$ à la place du symbole qui caractérise un constructeur de type donné ($\Pi, \forall, \Sigma, \exists, \{\}$)). Ainsi $\square x : \mathrm{T}.\mathrm{U}$ désigne un constructeur de type qui peut-être l'un des constructeurs de type suivants : $\Pi x : \mathrm{T}.\mathrm{U}$, $\forall x : \mathrm{T}.\mathrm{U}$, $\Sigma x : \mathrm{T}.\mathrm{U}$, $\exists x : \mathrm{T}.\mathrm{U}$, $\{x : \mathrm{T} \mid \mathrm{U}\}$. Nous utilisons également des variantes de $\square$ ($\square', \square_i$ où $i \in \mathbb{N}$,...) si plusieurs constructeurs de type génériques rentrent en jeu.

1.2.6. DÉFINITION (Domaine et codomaine d'un constructeur de type). Soit $\square x : \mathrm{T}.\mathrm{U}$ un un constructeur de type de ICC_Σ. Les termes T et U s'appellent respectivement *domaine* et *codomaine* de $\square x : \mathrm{T}.\mathrm{U}$.

Variables libres et liées. Nous adaptons à la syntaxe de ICC_Σ les définitions usuelles.

1.2.7. DÉFINITION (Variables libres et liées). Les *variables libres* d'un terme M de ICC_Σ sont notées $\mathsf{FV}(\mathrm{M})$. Les *variables liées* d'un terme M de ICC_Σ sont notées $\mathsf{BV}(\mathrm{M})$. Elles sont définies inductivement à la figure 2.

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{FV}(x) & = & \{x\} & \mathsf{BV}(x) & = & \emptyset \\
\mathsf{FV}(s) & = & \emptyset & \mathsf{BV}(s) & = & \emptyset \\
\\
\mathsf{FV}(\square x : \mathrm{T}.\mathrm{U}) & = & \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{U}) \setminus \{x\}) & \mathsf{BV}(\square x : \mathrm{T}.\mathrm{U}) & = & \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{U}) \cup \{x\} \\
\\
\mathsf{FV}(\lambda x.\mathrm{M}) & = & \mathsf{FV}(\mathrm{M}) \setminus \{x\} & \mathsf{BV}(\lambda x.\mathrm{M}) & = & \mathsf{BV}(\mathrm{M}) \cup \{x\} \\
\mathsf{FV}(\mathrm{M}\,\mathrm{N}) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) & \mathsf{BV}(\mathrm{M}\,\mathrm{N}) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \\
\\
\mathsf{FV}((\mathrm{M}, \mathrm{N})) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) & \mathsf{BV}((\mathrm{M}, \mathrm{N})) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \\
\mathsf{FV}(\mathsf{Elim}_\Sigma(xy.f, c)) & = & \mathsf{FV}(c) \cup (\mathsf{FV}(f) \setminus \{x, y\}) & \mathsf{BV}(\mathsf{Elim}_\Sigma(xy.f, c)) & = & \mathsf{BV}(c) \cup \mathsf{BV}(f) \\
 & & & & & \cup \{x, y\} \\
\\
\mathsf{FV}((\diamond, b)) & = & \mathsf{FV}(b) & \mathsf{BV}((\diamond, b)) & = & \mathsf{BV}(b) \\
\mathsf{FV}(\mathsf{Elim}_\exists(y.f, c)) & = & \mathsf{FV}(c) \cup (\mathsf{FV}(f) \setminus \{y\}) & \mathsf{BV}(\mathsf{Elim}_\exists(y.f, c)) & = & \mathsf{BV}(c) \cup \mathsf{BV}(f) \cup \{y\}
\end{array}
$$

FIGURE 2. Variables libres et liées

1.2.8. CONVENTION. Les termes de la syntaxe sont définis à α-conversion près, c'est-à-dire au renommage près des variables liées.

1.2.9. NOTATION (Types non-dépendants). Si $x \notin \mathsf{FV}(\mathrm{U})$, nous notons $\mathrm{T} \to \mathrm{U} = \Pi x : \mathrm{T}.\mathrm{U}$, $[\mathrm{T}] \to \mathrm{U} = \forall x : \mathrm{T}.\mathrm{U}$, $\mathrm{T} \times \mathrm{U} = \Sigma x : \mathrm{T}.\mathrm{U}$, $[\mathrm{T}] \times \mathrm{U} = \exists x : \mathrm{T}.\mathrm{U}$ et $\mathrm{T} \times [\mathrm{U}] = \{x : \mathrm{T} \mid \mathrm{U}\}$.

Substitution de variables. Pour éviter la capture de variable libre par des lieurs lors de la substitution de termes, nous utilisons la convention habituelle due à Barendregt ([**Bar84**]).

1.2.10. CONVENTION (Convention de Barendregt). Si nous considérons un ensemble (fini) de termes $\mathrm{M}_1, \mathrm{M}_2, \ldots \mathrm{M}_n$ alors nous considérons, quitte à renommer par α-conversion, que les variables libres de tout terme sont distinctes des variables liées de tout autre terme. Autrement dit, pour $i, j \in [1 \ldots n]$, $\mathsf{FV}(\mathrm{M}_i) \cap \mathsf{BV}(\mathrm{M}_j) = \emptyset$.

1.2.11. DÉFINITION (Substitution). Soient M et R des termes de ICC_Σ et z une variable. Nous notons $\mathrm{M}[z/\mathrm{R}]$ le terme de ICC_Σ où les occurrences libres de z dans le terme M ont été remplacées par le terme R. Nous avons, d'après la convention de Barendregt, $\mathsf{FV}(\mathrm{R}) \cap \mathsf{BV}(\mathrm{M}) = \emptyset$, ce qui empêche les captures de variables. Nous avons également $z \notin \mathsf{BV}(\mathrm{M})$. Une définition inductive de la substitution est fournie par la figure 3.

$$
\begin{array}{rcl}
z[z/\mathrm{R}] & = & \mathrm{R} \\
x[z/\mathrm{R}] & = & x \quad (\text{avec } x \neq z) \\
s[z/\mathrm{R}] & = & s \\
\\
(\Box x : \mathrm{T} . \mathrm{U})[z/\mathrm{R}] & = & \Box x : (\mathrm{T}[z/\mathrm{R}]) . (\mathrm{U}[z/\mathrm{R}]) \\
\\
(\lambda x . \mathrm{M})[z/\mathrm{R}] & = & \lambda x . \mathrm{M}[z/\mathrm{R}] \\
(\mathrm{M}\,\mathrm{N})[z/\mathrm{R}] & = & \mathrm{M}[z/\mathrm{R}]\,\mathrm{N}[z/\mathrm{R}] \\
\\
(\mathrm{M}, \mathrm{N})[z/\mathrm{R}] & = & (\mathrm{M}[z/\mathrm{R}], \mathrm{N}[z/\mathrm{R}]) \\
(\mathsf{Elim}_\Sigma(xy.f, c))[z/\mathrm{R}] & = & \mathsf{Elim}_\Sigma(xy.f[z/\mathrm{R}], c[z/\mathrm{R}]) \\
\\
(\diamond, \mathrm{M})[z/\mathrm{R}] & = & (\diamond, \mathrm{M}[z/\mathrm{R}]) \\
(\mathsf{Elim}_\exists(y.f, c))[z/\mathrm{R}] & = & \mathsf{Elim}_\exists(y.f[z/\mathrm{R}], c[z/\mathrm{R}])
\end{array}
$$

FIGURE 3. Substitution

Le lemme suivant se démontre facilement par induction sur M.

1.2.12. Lemme (Variables libres ou liées dans une substitution)**.** *Soient* M *et* N *des termes de* ICC_Σ *et* x *une variable non-liée dans* M*. Nous avons alors :*

(1) $FV(\mathrm{M}[x/\mathrm{N}]) \subset (FV(\mathrm{M}) \setminus \{x\}) \cup FV(\mathrm{N})$

(2) $BV(\mathrm{M}[x/\mathrm{N}]) \subset (BV(\mathrm{M}) \setminus \{x\}) \cup BV(\mathrm{N})$

(3) et si $x \notin FV(\mathrm{M})$*, alors* $\mathrm{M}[x/\mathrm{N}] = \mathrm{M}$.

1.2.13. Lemme (Substitutions multiples)**.** *Soient* x, y *des variables distinctes,* $\mathrm{M}, \mathrm{N}, \mathrm{U}$ *des termes tels que* $y \notin FV(\mathrm{N})$*, alors*

$$\mathrm{U}[x/\mathrm{M}][y/\mathrm{N}] = \mathrm{U}[y/\mathrm{N}][x/\mathrm{M}[y/\mathrm{N}]].$$

DÉMONSTRATION. Par induction sur la structure de U. □

2. Calcul

2.1. Généralités.

2.1.1. NOTATION. Soit R une relation binaire dans $\Lambda_{\mathrm{ICC}_\Sigma}$ et soient M, N des termes de ICC_Σ.

Nous notons

- $\mathrm{M} \rhd_\mathrm{R} \mathrm{N}$ si et seulement si $(\mathrm{M}, \mathrm{N}) \in \mathrm{R}$;
- $\mathrm{M} \to_\mathrm{R} \mathrm{N}$ si et seulement si (M, N) appartient à la *clôture contextuelle* de R ;
- $\mathrm{M} \leftarrow_\mathrm{R} \mathrm{N}$ si et seulement si $\mathrm{N} \to_\mathrm{R} \mathrm{M}$;
- $\mathrm{M} \twoheadrightarrow_\mathrm{R} \mathrm{N}$ si et seulement si (M, N) appartient à la *clôture réflexive et transitive* de $\to_\mathrm{R}$;
- $\mathrm{M} \twoheadleftarrow_\mathrm{R} \mathrm{N}$ si et seulement si $\mathrm{N} \twoheadrightarrow_\mathrm{R} \mathrm{M}$;

- $M \cong_R N$ si et seulement si (M,N) appartient à la *clôture réflexive, symétrique et transitive* de $\rightarrow_R$.

2.1.2. DÉFINITION. Soit R une relation binaire dans ICC_Σ et M un terme de ICC_Σ.

- M est un R-*redex*[1] s'il existe un terme N de ICC_Σ tel que $M \triangleright_R N$;
- M est R-*réductible* s'il existe un terme N de ICC_Σ tel que $M \rightarrow_R N$; N est alors le terme R-*réduit* de M et M est le terme R-*expansé* de N ;
- M est *en forme* R-*normale* ou encore R-*irréductible*, s'il n'est pas R-*réductible* ;
- N est *une forme* R-*normale* de M si N est un terme en forme R-normale tel que $M \twoheadrightarrow_R N$.

2.1.3. REMARQUE. En l'absence d'ambiguité sur la relation R, nous utiliserons les expressions plus simples *redex*, *réductible*, *réduit*, *expansé*, *en forme normale* ou *irréductible*, *forme normale* en lieu et place des expressions de la définition précédente.

2.1.4. DÉFINITION (Confluence). Soit R une relation binaire sur les termes de ICC_Σ. R est *confluente* si pour tous termes M, M_1, M_2 de ICC_Σ tels que $M_1 \twoheadleftarrow_R M \twoheadrightarrow_R M_2$, il existe un terme M_3 tel que $M_1 \twoheadrightarrow_R M_3 \twoheadleftarrow_R M_2$.

2.1.5. Proposition (Unicité des formes normales)**.** *Soit* R *une relation binaire sur les termes de* ICC_Σ*. Si* R *est confluente, alors la forme* R*-normale d'un terme, si elle existe, est unique : si* M_1, M_2 *sont des formes* R*-normales de* M *alors* $M_1 = M_2$.

DÉMONSTRATION. Nous avons $N_1 \twoheadleftarrow_R M \twoheadrightarrow_R N_2$. Par confluence, il existe un terme M' tel que $N_1 \twoheadrightarrow_R M' \twoheadleftarrow_R N_2$. Puisque N_1 et N_2 sont en forme normale, nous avons $N_1 = M' = N_2$. □

L'unicité des formes normales justifie la notation suivante.

2.1.6. NOTATION (Forme normale). Soit R une relation binaire sur les termes de ICC_Σ. Si R est confluente, pour tout terme M, nous notons, si elle existe, $\mathsf{nf}_R(M)$ sa forme R-normale.

Un corollaire très utile de la confluence est la propriété de Church-Rosser qui indique que deux termes convertibles ont un réduit commun. Nous omettons la preuve de ce résultat classique.

2.1.7. Corollaire (Church-Rosser)**.** *Soient* R *une relation binaire sur les termes de* ICC_Σ *et* M, N *des termes de* ICC_Σ*. Si* R *est confluente et si* $M \cong_R N$ *alors il existe un terme* T *tel que* $M \twoheadrightarrow_R T$ *et* $N \twoheadrightarrow_R T$.

2.2. Réduction et conversion.

Réduction.

2.2.1. DÉFINITION (Réduction à la racine). Soient $x, y \in \mathbf{Var}$ et M, N, f, a, b des termes de ICC_Σ. Nous considérons quatre règles de réduction, définies par les relations binaires suivantes :

(1) la β-réduction :

$$(\lambda x.M)\,N \quad \triangleright_\beta \quad M[x/N]$$

(2) la ι_Σ-réduction :

$$\mathsf{Elim}_\Sigma(x\,y.f, (a, b)) \quad \triangleright_{\iota_\Sigma} \quad f[x/a][y/b]$$

(3) la $\iota_\exists$-réduction :

$$\mathsf{Elim}_\exists(y.f, (\diamond, b)) \quad \triangleright_{\iota_\exists} \quad f[y/b]$$

1. contraction de l'anglais *red*uctible *ex*pression

(4) et enfin la η-réduction :

$$\lambda x.\,\mathrm{M}\,x \quad \rhd_\eta \quad \mathrm{M} \quad (\text{si } x \notin \mathsf{FV}(\mathrm{M})).$$

Nous nous intéressons également à la réunion des règles β, ι_Σ et $\iota_\exists$, notée βι, et définie par :

$$\mathrm{M} \rhd_{\beta\iota} \mathrm{N} \quad \Leftrightarrow \quad \mathrm{M} \rhd_\beta \mathrm{N} \quad \text{ou} \quad \mathrm{M} \rhd_{\iota_\Sigma} \mathrm{N} \quad \text{ou} \quad \mathrm{M} \rhd_{\iota_\exists} \mathrm{N}$$

ainsi qu'à la réunion des quatre règles de réduction, notée βιη, et définie par :

$$\mathrm{M} \rhd_{\beta\iota\eta} \mathrm{N} \quad \Leftrightarrow \quad \mathrm{M} \rhd_{\beta\iota} \mathrm{N} \quad \text{ou} \quad \mathrm{M} \rhd_\eta \mathrm{N}$$

2.2.2. Lemme (Réduction et variables libres)**.** *Soient* M, N *des termes de ICC*$_\Sigma$ *tels que* $\mathrm{M} \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}$. *Alors* $FV(\mathrm{N}) \subset FV(\mathrm{M})$.

DÉMONSTRATION. Nous montrons trivialement que $\mathsf{FV}(\mathrm{N}) \subset \mathsf{FV}(\mathrm{M})$ si $\mathrm{M} \rhd_{\beta\iota\eta} \mathrm{N}$. Par induction sur la structure de M, nous montrons le résultat si $\mathrm{M} \to_{\beta\iota\eta} \mathrm{N}$. Nous concluons alors par récurrence sur le nombre d'étapes de réduction. □

2.2.3. Lemme (Réduction et substitution)**.** *Soient* x *une variable et* M, M′, N, N′ *des termes de ICC*$_\Sigma$.

(1) Si $\mathrm{M} \to_{\beta\iota\eta} \mathrm{M}'$, *alors* $\mathrm{M}[x/\mathrm{N}] \to_{\beta\iota\eta} \mathrm{M}'[x/\mathrm{N}]$.

(2) Si $\mathrm{N} \to_{\beta\iota\eta} \mathrm{N}'$ *alors* $\mathrm{M}[x/\mathrm{N}] \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}[x/\mathrm{N}']$ *et le nombre de réductions correspond au nombre d'occurrences libres de* x *dans* M.

(3) Si $\mathrm{M} \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}'$ *et* $\mathrm{N} \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}'$ *alors* $\mathrm{M}[x/\mathrm{N}] \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}'[x/\mathrm{N}']$.

DÉMONSTRATION.

(1) Par induction sur la structure de M. Les cas où la réduction a lieu dans un sous-terme strict de M sont évidents. Si la réduction a lieu à la racine, il faut distinguer quatre cas selon la règle de réduction concernée. Ces quatre cas se résolvent sans difficulté grâce au lemme des substitutions multiples (lemme 1.2.13).

(2) Par induction sur la structure de M.

(3) Nous généralisons par une récurrence immédiate le point (2) dans le cas où $\mathrm{N} \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}'$. Nous démontrons ensuite le résultat par récurrence sur le nombre de réductions de M.

□

Le lemme suivant nous indique les éventuels liens entre la structure d'un terme et celle de son réduit.

2.2.4. Lemme (Réduction et structure des termes)**.** *Soient* M, N *des termes de ICC*$_\Sigma$. *Si* $\mathrm{M} \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}$ *alors* M *est un éliminateur ou une abstraction ou bien* M *et* N *ont même structure.*

DÉMONSTRATION. Par récurrence sur le nombre de réductions.

- S'il n'y a aucune réduction, M = N et le résultat est évident.
- Si $\mathrm{M} \to_{\beta\iota\eta} \mathrm{M}_0 \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}$, montrons que M est un éliminateur ou une abstraction, ou bien M et N ont même structure. Nous avons deux cas :
  - la réduction $\mathrm{M} \to_{\beta\iota\eta} \mathrm{M}_0$ a lieu à la racine : M est un βιη-redex, donc un éliminateur ou une abstraction ;
  - sinon M et M_0 ont même structure ; nous avons alors par hypothèse de récurrence :
    - M_0 est un éliminateur ou une abstraction : M est alors également un éliminateur ou une abstraction ;
    - M_0 et N ont même structure : M et N ont alors également même structure.

□

2.2.5. Proposition (Confluence des βι- et βιη-réductions)**.** *Les relations* βι *et* βιη *sont confluentes.*

DÉMONSTRATION. Nous montrons la confluence de manière classique en utilisant des réductions parallèles. La preuve est détaillée dans l'annexe A. □

Conversion.

2.2.6. DÉFINITION (Conversion). La relation $\cong_{\beta\iota\eta}$ est également appelée *conversion*. Elle sera notée $\cong$ si ce n'est pas ambigu. Deux termes M, N de ICC_Σ sont *convertibles* si et seulement si $M \cong N$.

Puisque la βιη-réduction est confluente, nous en déduisons, d'après le corollaire 2.1.7 que la conversion vérifie la propriété de Church-Rosser.

2.2.7. Proposition (Church-Rosser)**.** *Soient* M_1, M_2 *des termes de* ICC_Σ *tels que* $M_1 \cong M_2$*. Alors il existe* M *tel que* $M_1 \twoheadrightarrow_{\beta\iota\eta} M \twoheadleftarrow_{\beta\iota\eta} M_2$.

Conséquences de Church-Rosser.

2.2.8. Lemme (Conversion et substitution)**.** *Soient* M, M', N, N' *des termes et* x *une variable. Si* $M \cong M'$ *et* $N \cong N'$*, alors* $M[x/N] \cong M'[x/N']$.

DÉMONSTRATION. Par application de la propriété de Church-Rosser et du point (3) du lemme 2.2.3. □

2.2.9. Lemme (Inversion de la conversion pour les sortes et les constructeurs de type)**.**

(1) Si s, s' *sont des sortes telles que* $s \cong s'$*, alors* $s = s'$.

(2) Si $\square x : T.U \cong \square' x : T'.U'$ *alors* $\square = \square'$ *(les constructeurs de type sont de même nature) et* $T \cong T'$ $U \cong U'$.

DÉMONSTRATION.

(1) Conséquence immédiate de Church-Rosser, puisque toute sorte est irréductible.

(2) D'après Church-Rosser, il existe un terme R tel que $\square x : T.U \twoheadrightarrow_{\beta\iota\eta} R \twoheadleftarrow_{\beta\iota\eta} \square' x : T'.U'$. D'après le lemme 2.2.4, et puisqu'un constructeur de type n'est pas un éliminateur ou une abstraction, $\square x : T.U$ et $\square' x : T'.U'$ sont de même nature. Nous montrons alors facilement que $T \cong T'$ et $U \cong U'$.

□

2.3. Cumulativité. Nous définissons une relation qui étend la hiérarchie des sortes aux constructeurs de type. Cette relation est sensiblement différente de celle de Miquel, qui ne tient compte que de la hiérarchie des sortes. Elle se rapproche de celle de Luo [**Luo94**].

2.3.1. DÉFINITION (Cumulativité). La relation de *cumulativité*, notée $\preceq$, est la relation engendrée par les règles d'inférence suivantes :

$$\frac{M \cong N}{M \preceq N}\ (\textsc{Cum-Refl})$$

$$\frac{i > 0}{\mathsf{Prop} \preceq \mathsf{Type}_i}\ (\textsc{Cum-Sort-1})$$

$$\frac{i \leq j}{\mathsf{Type}_i \preceq \mathsf{Type}_j}\ (\textsc{Cum-Sort-2})$$

$$\frac{\mathrm{T}_2 \preceq \mathrm{T}_1 \qquad \mathrm{U}_1 \preceq \mathrm{U}_2}{\Pi x:\mathrm{T}_1.\mathrm{U}_1 \preceq \Pi x:\mathrm{T}_2.\mathrm{U}_2}\ (\textsc{Cum-Prod})$$

$$\frac{\mathrm{T}_2 \preceq \mathrm{T}_1 \qquad \mathrm{U}_1 \preceq \mathrm{U}_2}{\forall x:\mathrm{T}_1.\mathrm{U}_1 \preceq \forall x:\mathrm{T}_2.\mathrm{U}_2}\ (\textsc{Cum-I-Prod})$$

$$\frac{\mathrm{A}_1 \preceq \mathrm{A}_2 \qquad \mathrm{B}_1 \preceq \mathrm{B}_2}{\Sigma x:\mathrm{A}_1.\mathrm{B}_1 \preceq \Sigma x:\mathrm{A}_2.\mathrm{B}_2}\ (\textsc{Cum-}\Sigma)$$

$$\frac{\mathrm{A}_1 \preceq \mathrm{A}_2 \qquad \mathrm{B}_1 \preceq \mathrm{B}_2}{\exists x:\mathrm{A}_1.\mathrm{B}_1 \preceq \exists x:\mathrm{A}_2.\mathrm{B}_2}\ (\textsc{Cum-U})$$

$$\frac{\mathrm{A}_1 \preceq \mathrm{A}_2 \qquad \mathrm{B}_1 \preceq \mathrm{B}_2}{\{x:\mathrm{A}_1 \mid \mathrm{B}_1\} \preceq \{x:\mathrm{A}_2 \mid \mathrm{B}_2\}}\ (\textsc{Cum-Sub})$$

$$\frac{\mathrm{M} \preceq \mathrm{N} \qquad \mathrm{N} \preceq \mathrm{U}}{\mathrm{M} \preceq \mathrm{U}}\ (\textsc{Cum-Trans})$$

Le résultat suivant montre que la substitution préserve la cumulativité.

2.3.2. Lemme (Cumulativité et substitution)**.** *Soient x une variable,* $\mathrm{M},\mathrm{M}',\mathrm{N}$ *des termes tels que* $\mathrm{M} \preceq \mathrm{M}'$. *Alors* $\mathrm{M}[x/\mathrm{N}] \preceq \mathrm{M}'[x/\mathrm{N}]$.

DÉMONSTRATION. Par induction sur la structure de la dérivation de $\mathrm{M} \preceq \mathrm{M}'$ en utilisant le lemme 2.2.8 pour la règle (CUM-REFL). □

2.3.3. REMARQUE. Du fait de la contra-variance des produits et de la variance pour les sommes, nous n'avons aucun lien entre $\mathrm{M}[x/\mathrm{N}]$ et $\mathrm{M}[x/\mathrm{N}']$ si $\mathrm{N} \preceq \mathrm{N}'$. Ainsi si $\mathrm{M} = x \to \mathsf{Prop}, \mathrm{N} = \mathsf{Prop}, \mathrm{N}' = \mathsf{Type}_1$, nous avons $\mathrm{M}[x/\mathrm{N}'] = \mathsf{Type}_1 \to \mathsf{Prop} \preceq \mathsf{Prop} \to \mathsf{Prop} = \mathrm{M}[x/\mathrm{N}]$. Et si $\mathrm{M} = x \times \mathsf{Prop}$, alors $\mathrm{M}[x/\mathrm{N}] = \mathsf{Prop} \times \mathsf{Prop} \preceq \mathsf{Type}_1 \times \mathsf{Prop} = \mathrm{M}[x/\mathrm{N}']$.

Dans le reste de la section, nous allons montrer des résultats d'inversion de la cumulativité similaires au corollaire 2.2.9 pour la conversion. Nous allons également démontrer que la cumulativité est une relation d'ordre pour la conversion.

Prouver des résultats d'inversion par induction sur la dérivation de $\mathrm{M} \preceq \mathrm{N}$ est délicat à cause de la règle (CUM-TRANS) qui introduit un terme intermédiaire dont nous ne savons rien. Pour contourner ce problème, nous définissons une deuxième relation, la *cumulativité restreinte*, sur laquelle il est plus facile de raisonner, et que nous allons relier à la cumulativité.

Cumulativité restreinte.

2.3.4. DÉFINITION (Cumulativité restreinte). La relation de *cumulativité restreinte*, notée $\leqslant$, est la relation engendrée par les règles d'inférence suivantes :

$$\frac{}{\mathrm{M} \leqslant \mathrm{M}}\ (\textsc{CumR-Eq})$$

$$\frac{i > 0}{\mathsf{Prop} \leqslant \mathsf{Type}_i}\ (\textsc{CumR-Sort-1})$$

$$\frac{i < j}{\mathsf{Type}_i \leqslant \mathsf{Type}_j}\ (\textsc{CumR-Sort-2})$$

$$\frac{\mathrm{T}_2 \leqslant \mathrm{T}_1 \qquad \mathrm{U}_1 \leqslant \mathrm{U}_2}{\Pi x:\mathrm{T}_1.\mathrm{U}_1 \leqslant \Pi x:\mathrm{T}_2.\mathrm{U}_2}\ (\textsc{CumR-Prod})$$

$$\frac{T_2 \leqslant T_1 \qquad U_1 \leqslant U_2}{\forall x : T_1 . U_1 \leqslant \forall x : T_2 . U_2} \text{ (CumR-I-Prod)}$$

$$\frac{A_1 \leqslant A_2 \qquad B_1 \leqslant B_2}{\Sigma x : A_1 . B_1 \leqslant \Sigma x : A_2 . B_2} \text{ (CumR-}\Sigma\text{)}$$

$$\frac{A_1 \leqslant A_2 \qquad B_1 \leqslant B_2}{\exists x : A_1 . B_1 \leqslant \exists x : A_2 . B_2} \text{ (CumR-U)}$$

$$\frac{A_1 \leqslant A_2 \qquad B_1 \leqslant B_2}{\{x : A_1 \mid B_1\} \leqslant \{x : A_2 \mid B_2\}} \text{ (CumR-Sub)}$$

Cette relation est plus simple que la cumulativité. D'une part, elle ne fait pas appel à la conversion mais à l'égalité syntaxique. D'autre part, nous n'avons pas de règle de transitivité. Le fait que les dérivations ne sont pas dirigées par la syntaxe (dû à (CumR-Eq)) n'est pas gênant.

Ainsi, il est aisé d'établir l'inversion de la cumulativité restreinte et de montrer que la cumulativité restreinte est une relation d'ordre.

2.3.5. Lemme (Inversion de la cumulativité restreinte)**.** *Si* $M \leqslant N$*, alors*

- *si* M *ou* N *n'est pas un constructeur de type ou une sorte,* $M = N$ *;*
- *sinon* M *et* N *sont tous deux des sortes ou tous deux des constructeurs de type ayant même structure.*

Démonstration. Immédiat d'après la définition des règles d'inférence de la cumulativité restreinte. □

2.3.6. Lemme. $\leqslant$ *est une relation d'ordre par rapport à l'égalité syntaxique.*

Démonstration. La réflexivité découle trivialement de (CumR-Eq). L'antisymétrie et la transitivité se prouvent sans difficulté par induction sur les dérivations et en utilisant le lemme précédent. □

Cumulativité restreinte et cumulativité. Afin de relier la cumulativité restreinte à la cumulativité, nous montrons deux lemmes auxiliaires.

Le premier résultat concerne la cumulativité restreinte et la conversion de termes.

2.3.7. Lemme (Cumulativité restreinte et conversion)**.** *Si* $M \leqslant N$ *et* $M \cong N$ *alors* $M = N$.

Démonstration. Par induction sur la dérivation de $M \leqslant N$. Évident pour la règle (CumR-Eq). Pour les autres règles, il suffit d'utiliser le point (2) du corollaire 2.2.9 (inversion de la conversion pour les constructeurs de type) pour pouvoir conclure grâce aux hypothèses d'induction. □

Montrons ensuite que la cumulativité restreinte est préservée par les réductions.

2.3.8. Lemme (Cumulativité restreinte et calcul)**.**

(1) Si $A \leqslant B$ *alors*

- *si* $A \to_{\beta\iota\eta} A'$*, il existe* B' *tel que* $B \to_{\beta\iota\eta} B'$ *et* $A' \leqslant B'$ *;*
- *si* $B \to_{\beta\iota\eta} B'$*, il existe* A' *tel que* $A \to_{\beta\iota\eta} A'$ *et* $A' \leqslant B'$*.*

(2) Si $A \leqslant B$ *alors*

- *si* $A \twoheadrightarrow_{\beta\iota\eta} A'$*, il existe* B' *tel que* $B \twoheadrightarrow_{\beta\iota\eta} B'$ *et* $A' \leqslant B'$ *;*
- *si* $B \twoheadrightarrow_{\beta\iota\eta} B'$*, il existe* A' *tel que* $A \twoheadrightarrow_{\beta\iota\eta} A'$ *et* $A' \leqslant B'$*.*

DÉMONSTRATION.

(1) Par induction sur la dérivation de $A \leqslant B$. La règle (CUMR-EQ) est évidente. Les autres règles se traitent, elles, de manière non-triviale, mais similaire. Détaillons, par exemple, le cas de la règle (CUMR-I-PROD) avec $A \to_{\beta\iota\eta} A'$. (Le cas où $B \to_{\beta\iota\eta} B'$ se traite similairement.)

- Nous avons
$$\frac{T_2 \leqslant T_1 \qquad U_1 \leqslant U_2}{\forall x : T_1 . U_1 \leqslant \forall x : T_2 . U_2} \text{ (CUMR-I-PROD)}$$
avec $A = \forall x : T_1 . U_1 \to_{\beta\iota\eta} A'$ et $B = \forall x : T_2 . U_2$. Montrons qu'il existe B' tel que $B \to_{\beta\iota\eta} B'$ et $A' \leqslant B'$.
- $\forall x : T_1 . U_1$ peut se réduire soit dans son domaine T_1, soit dans son codomaine U_1. Supposons, sans perte de généralité, que la réduction a lieu dans T_1. Il existe donc T_1' tel que $T_1 \to_{\beta\iota\eta} T_1'$ et $A' = \forall x : T_1' . U_1$.
- Par hypothèse d'induction sur $T_2 \leqslant T_1$, il existe T_2' tel que $T_2 \to_{\beta\iota\eta} T_2'$ et $T_2' \leqslant T_1'$. $B' = \forall x : T_2' . U_2$ convient donc bien.

(2) Immédiat par récurrence sur le nombre de réductions. Le cas de base est prouvé par le point précédent.

□

2.3.9. REMARQUE. Contrairement au point (2), nous procédons par induction mutuelle dans la démonstration du point (1). En effet, à cause de la contravariance du domaine dans les règles des produits, nous ne pouvons prouver séparément les cas $A \to_{\beta\iota\eta} A'$ et $B \to_{\beta\iota\eta} B'$.

Le lemme suivant fait le lien entre cumulativité et cumulativité restreinte. Il va permettre de montrer que la cumulativité est une relation antisymétrique.

2.3.10. Lemme (Cumulativité et cumulativité restreinte)**.**

(1) Si $M \leqslant N$*, alors* $M \preceq N$.

(2) Si $M \preceq N$ *alors il existe des termes* M_0, N_0 *tels que* $M \twoheadrightarrow_{\beta\iota\eta} M_0$, $N \twoheadrightarrow_{\beta\iota\eta} N_0$ *et* $M_0 \leqslant N_0$.

DÉMONSTRATION.

(1) Immédiat par induction sur $M \leqslant N$.

(2) Par induction sur la dérivation de $M \preceq N$.

- Pour la règle (CUM-REFL), nous concluons par Church-Rosser.
- Toutes les autres règles sauf (CUM-TRANS) sont évidentes.
- Traitons la règle (CUM-TRANS). Si
$$\frac{M_1 \preceq M_2 \qquad M_2 \preceq M_3}{M_1 \preceq M_3} \text{ (CUM-TRANS)}$$
D'après les hypothèses d'induction, nous avons N_1, N_2^1, N_2^2, N_3 tels que
$$\begin{array}{llll} N_1 \leqslant N_2^1 & M_1 \twoheadrightarrow_{\beta\iota\eta} N_1 & M_2 \twoheadrightarrow_{\beta\iota\eta} N_2^1 \\ N_2^2 \leqslant N_3 & M_2 \twoheadrightarrow_{\beta\iota\eta} N_2^2 & M_3 \twoheadrightarrow_{\beta\iota\eta} N_3. \end{array}$$
Par confluence, il existe N_2 tel que $N_2^1 \twoheadrightarrow_{\beta\iota\eta} N_2$ et $N_2^2 \twoheadrightarrow_{\beta\iota\eta} N_2$. D'après le point (2) du lemme 2.3.8 appliqué respectivement à $N_1 \leqslant N_2^1$ et $N_2^2 \leqslant N_3$, il existe N_1', N_3' tels que $N_1 \twoheadrightarrow_{\beta\iota\eta} N_1'$ et $N_1' \leqslant N_2$, $N_3 \twoheadrightarrow_{\beta\iota\eta} N_3'$ et $N_2 \leqslant N_3'$. Par transitivité, nous déduisons $N_1' \leqslant N_3'$ avec $M_1 \twoheadrightarrow_{\beta\iota\eta} N_1'$ et $M_3 \twoheadrightarrow_{\beta\iota\eta} N_3'$.

□

2.3.11. REMARQUE. Pour le point (2), nous avons de plus $M_0 = N_0$ si et seulement si $M \cong N$.

- $M_0 = N_0$ implique clairement $M \cong N$.
- Si $M \cong N$, alors $M_0 \cong N_0$ et nous concluons grâce au lemme 2.3.7.

2.3.12. Corollaire. *Si* $M \preceq N$ *et si* M *et* N *sont normalisables, alors* $\mathsf{nf}(M) \leqslant \mathsf{nf}(N)$.

DÉMONSTRATION. De $M \preceq N$ nous déduisons $\mathsf{nf}(M) \preceq \mathsf{nf}(N)$. Il suffit alors d'appliquer le point (2) du lemme précédent, puisque $\mathsf{nf}(M)$ et $\mathsf{nf}(N)$ sont irréductibles. □

2.3.13. Corollaire.

(1) si $M \preceq s$ *(resp.* $s \preceq M$*) alors il existe une sorte* s_M *telle que* $M \twoheadrightarrow_{\beta\iota\eta} s_M$ *et* $s_M \preceq s$ *(resp.* $s \preceq s_M$*);*

(2) si $M \preceq \Box x : T \,.\, U$ *(resp.* $\Box x : T \,.\, U \preceq M$*) alors il existe* T_M, U_M *tels que* $M \twoheadrightarrow_{\beta\iota\eta} \Box x : T_M \,.\, U_M$ *et* $\Box x : T_M \,.\, U_M \preceq \Box x : T \,.\, U$ *(resp.* $\Box x : T \,.\, U \preceq \Box x : T_M \,.\, U_M$*).*

DÉMONSTRATION. Les preuves des deux points sont similaires. Nous ne traitons que le deuxième car il est plus complexe. Par ailleurs, pour des raisons de symétrie, nous ne traitons que le cas $M \preceq \Box x : T \,.\, U$.

- D'après le point (2) du lemme 2.3.10 il existe des termes M_0 et R tels que $M_0 \leqslant R$ et $M \twoheadrightarrow_{\beta\iota\eta} M_0$ et $\Box x : T \,.\, U \twoheadrightarrow_{\beta\iota\eta} R$.
- D'après le lemme 2.2.4 appliqué à $\Box x : T \,.\, U \twoheadrightarrow_{\beta\iota\eta} R$, il existe des termes T_0 et U_0 tels que $R = \Box x : T_0 \,.\, U_0$. Nous avons donc $M_0 \leqslant \Box x : T_0 \,.\, U_0$.
- Par inversion de la cumulativité restreinte (lemme 2.3.5), M_0 est un constructeur de type de même nature que $\Box x : T_0 \,.\, U_0$; il existe donc des termes T_M et U_M tels que $M_0 = \Box x : T_M \,.\, U_M$. Nous avons bien $M \twoheadrightarrow_{\beta\iota\eta} \Box x : T_M \,.\, U_M$.
- Par ailleurs, nous déduisons facilement $\Box x : T_M \,.\, U_M \preceq \Box x : T \,.\, U$ de $M \preceq \Box x : T \,.\, U$.

□

2.3.14. Lemme (Relation d'ordre)**.** *La relation de cumulativité est une relation d'ordre par rapport à la conversion.*

DÉMONSTRATION.

- La réflexivité et la transitivité sont données respectivement par les règles (CUM-REFL) et (CUM-TRANS).
- Montrons que $\preceq$ est anti-symétrique : si $M \preceq N$ et $N \preceq M$, montrons que $M \cong N$.
  - D'après le point (2) du lemme 2.3.10, appliqué respectivement à $M \preceq N$ et $N \preceq M$, nous avons, M_1, N_1, M_2, N_2 tels que
    $$M_1 \leqslant N_1 \quad N_2 \leqslant M_2 \quad M_1 \twoheadleftarrow_{\beta\iota\eta} M \twoheadrightarrow_{\beta\iota\eta} M_2 \quad N_1 \twoheadleftarrow_{\beta\iota\eta} N \twoheadrightarrow_{\beta\iota\eta} N_2$$
  - Par confluence, il existe N_0 tel que $N_1 \twoheadrightarrow_{\beta\iota\eta} N_0 \twoheadleftarrow_{\beta\iota\eta} N_2$.
  - D'après le point (2) du lemme 2.3.8, il existe M_1', M_2' tels que $M_1 \twoheadrightarrow_{\beta\iota\eta} M_1'$ et $M_2 \twoheadrightarrow_{\beta\iota\eta} M_2'$, $M_1' \leqslant N_0$ et $N_0 \leqslant M_2'$.
  - Nous en déduisons $M_1' \leqslant M_2'$ et $M_1' \cong M_2'$, puis $M_1' = M_2'$ en appliquant le lemme 2.3.7.
  - Par anti-symétrie de $\leqslant$, nous avons $M_1' = N_0$. Nous avons donc $M \twoheadrightarrow_{\beta\iota\eta} N_0 \twoheadleftarrow_{\beta\iota\eta} N$, d'où $M \cong N$.

□

Inversion de la cumulativité.

2.3.15. Lemme. *Soient* M, M' *des termes de ICC*$_\Sigma$*. Si* $M \preceq M'$*, alors* M *et* M' *ont même structure ou au moins l'un d'entre eux est un éliminateur ou une abstraction.*

DÉMONSTRATION. Conséquence du point (2) du lemme 2.3.10, ainsi que du lemme 2.2.4 et de l'inversion de la cumulativité restreinte (lemme 2.3.5). □

Nous pouvons maintenant montrer les lemmes d'inversion de la cumulativité pour les sortes et constructeurs de type.

2.3.16. Lemme (Inversion de la cumulativité)**.**

- *(sortes) si* $s \preceq s'$ *alors* $s = \mathsf{Prop}$ *ou il existe des entiers* i, j *tels que* $i \leq j$ *et* $s = \mathsf{Type}_i$ *et* $s' = \mathsf{Type}_j$ *;*
- *(produits) si* $\Pi x : T . U \preceq \Pi x : T' . U'$ *ou* $\forall x : T . U \preceq \forall x : T' . U'$ *alors* $T' \preceq T$ *et* $U' \preceq U$ *;*
- *(sommes) si* $\Sigma x : A . B \preceq \Sigma x : A' . B'$ *ou* $\exists x : A . B \preceq \exists x : A' . B'$ *ou* $\{x : A \mid B\} \preceq \{x : A' \mid B'\}$ *alors* $A \preceq A'$ *et* $B \preceq B'$.

DÉMONSTRATION. Tous les cas sont similaires. Traitons par exemple celui du produit implicite. Si $\forall x : T . U \preceq \forall x : T' . U'$, montrons que $T' \preceq T$ et $U \preceq U'$.

- D'après le point (2) du lemme 2.3.10, il existe M, M' tels que $\forall x : T . U \twoheadrightarrow_{\beta\iota\eta} M$, $\forall x : T' . U' \twoheadrightarrow_{\beta\iota\eta} M'$ et $M \leqslant M'$.
- D'après le lemme 2.2.4, il existe T_0, U_0, T'_0, U'_0 tels que $M = \forall x : T_0 . U_0$ et $M' = \forall x : T'_0 . U'_0$. Nous avons donc
$$\forall x : T_0 . U_0 \cong \forall x : T . U \qquad \forall x : T'_0 . U'_0 \cong \forall x : T' . U' \qquad \forall x : T_0 . U_0 \leqslant \forall x : T'_0 . U'_0.$$
- Par inversion de la conversion (lemme 2.2.9) et de la cumulativité restreinte (lemme 2.3.5), nous avons :
$$T_0 \cong T \quad U \cong U_0 \quad T' \cong T'_0 \quad U'_0 \cong U' \quad T'_0 \leqslant T_0 \quad U_0 \leqslant U'_0.$$
- D'où, en appliquant soit la règle (CUM-REFL), soit le point (1) du lemme 2.3.10 :
$$T_0 \preceq T \quad U \preceq U_0 \quad T' \preceq T'_0 \quad U'_0 \preceq U' \quad T'_0 \preceq T_0 \quad U_0 \preceq U'_0.$$
- Par transitivité de la cumulativité, nous en déduisons $T' \preceq T$ et $U \preceq U'$.

□

3. Typage

Dans cette section, nous allons ajouter à notre langage des *règles de typage*. Ces règles permettent d'associer à un terme du langage un autre terme, son *type*. Ces règles sont une restriction du langage : certains termes ne peuvent être associés à aucun type. Mais ne considérer que les termes *typables*, c'est-à-dire les termes pour lesquels il est possible d'associer un type, présente des avantages. Ainsi, connaître le type d'un terme permet de lui donner une signification. Via l'isomorphisme de Curry-Howard [**CF58, How80**], nous pouvons associer une proposition mathématique à un type : un terme d'un certain type est alors interprété comme une preuve de la proposition associée à son type. Un deuxième avantage est de pouvoir garantir le comportement calculatoire des termes typables. Ainsi une propriété souhaitable des λ-calculs typés est le fait que tout terme bien typé ne peut se réduire indéfiniment. Cette propriété, appelée *normalisation forte*, signifie que tout programme (terme) bien typé retournera une valeur lorsqu'il est exécuté (réduit). [2]

2. cf. l'aphorisme de Robin Milner *"Well-typed programs can't go wrong."*

3.1. Jugements. Nous définissons deux nouvelles structures syntaxiques utilisées dans les règles de typage : les *jugements de bonne formation* et les *jugements de typage*. Ces jugements utilisent eux-mêmes une troisième structure : les contextes de typage.

Contextes de typage. Les contextes de typage permettent d'associer des types à un ensemble de variables.

3.1.1. DÉFINITION (Contextes). Un *contexte de typage* ou, plus brièvement, *contexte*, est une liste ordonnée, éventuellement vide, de *déclarations de type*. Une déclaration de type est un couple formé d'une variable x et d'un terme T, noté $x : \mathrm{T}$, qui permet ainsi d'assigner un type à une variable. Plus formellement, nous avons la définition inductive suivante :

$$\Gamma \quad ::= \quad \bullet \mid \Gamma; x : \mathrm{T}$$

Nous établissons maintenant quelques définitions et notations utiles pour manipuler les contextes.

3.1.2. DÉFINITION (Concaténation de contextes). Soient $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ et $\Delta = y_1 : \mathrm{U}_1; \ldots; y_m : \mathrm{U}_m$ deux contextes. Nous notons $\Gamma; \Delta = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n; y_1 : \mathrm{U}_1; \ldots; y_m : \mathrm{U}_m$ la *concaténation* de Γ et Δ.

3.1.3. REMARQUE. La concaténation de contextes est associative.

3.1.4. NOTATION. Soient x une variable, T un terme et Γ un contexte. Nous notons $(x : \mathrm{T}) \in \Gamma$ si Γ peut s'écrire $\Gamma = \Gamma_1; x : \mathrm{T}; \Gamma_2$, où Γ_1, Γ_2 sont des contextes.

3.1.5. DÉFINITION (Sous-contexte). Soient Γ, Γ' des contextes. Γ est un *sous-contexte* de Γ', et nous notons alors $\Gamma \subseteq \Gamma'$, si pour toute variable x et tout terme T, $(x : \mathrm{T}) \in \Gamma \Rightarrow (x : \mathrm{T}) \in \Gamma'$.

3.1.6. DÉFINITION. Si $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ est un contexte, nous définissons

- l'ensemble $\mathsf{DV}(\Gamma)$ des *variables déclarées* de Γ par $\mathsf{DV}(\Gamma) = \{x_1; \ldots; x_n\}$,
- l'ensemble $\mathsf{FV}(\Gamma)$ des *variables libres* de Γ par $\mathsf{FV}(\Gamma) = \mathsf{FV}(\mathrm{T}_1) \cup \cdots \cup \mathsf{FV}(\mathrm{T}_n)$.

Substitution et calcul dans un contexte. Nous étendons ici aux contextes ces deux notions définies préalablement pour les termes.

3.1.7. DÉFINITION (Substitution d'un contexte). Soient Γ un contexte, x une variable et N un terme. Nous définissons inductivement la *substitution d'un contexte* :

$$\begin{array}{rcl} \bullet\,[x/\mathrm{N}] & = & \bullet \\ (\Gamma; y : \mathrm{T})\,[x/\mathrm{N}] & = & \begin{cases} \Gamma\,[x/\mathrm{N}] & \text{si } x = y \\ \Gamma\,[x/\mathrm{N}]; y : \mathrm{T}\,[x/\mathrm{N}] & \text{sinon.} \end{cases} \end{array}$$

3.1.8. Lemme (Substitution d'un contexte et variables libres)**.** *Soient Γ un contexte, x une variable et* N *un terme. Si $x \notin \mathsf{DV}(\Gamma) \cup \mathsf{FV}(\Gamma)$, alors $\Gamma\,[x/\mathrm{N}] = \Gamma$.*

DÉMONSTRATION. Par induction sur Γ. Le cas $\Gamma = \bullet$ est immédiat. Si $\Gamma = \Gamma'; z : \mathrm{U}$, montrons que $\Gamma\,[x/\mathrm{N}] = \Gamma$. De $x \neq z$, nous déduisons $(\Gamma'; z : \mathrm{U})\,[x/\mathrm{N}] = \Gamma'\,[x/\mathrm{N}]; z : \mathrm{U}\,[x/\mathrm{N}]$. Par hypothèse d'induction sur Γ', nous avons $\Gamma'\,[x/\mathrm{N}] = \Gamma'$. Nous concluons en montrant que $\mathrm{U}\,[x/\mathrm{N}] = \mathrm{U}$, d'après le lemme 1.2.12, puisque $x \notin \mathsf{FV}(\mathrm{U}) \subset \mathsf{FV}(\Gamma)$. □

3.1.9. DÉFINITION (Contextes et calcul).

(1) La relation de conversion $\cong$ est étendue aux contextes en posant

$$x_1 : \mathrm{T}_1; x_2 : \mathrm{T}_2; \ldots; x_n : \mathrm{T}_n \cong x_1 : \mathrm{T}'_1; x_2 : \mathrm{T}'_2; \ldots; x_n : \mathrm{T}'_n$$

si pour tout i, $\mathrm{T}_i \cong \mathrm{T}'_i$.

(2) La relation de cumulativité $\preceq$ est étendue aux contextes en posant

$$x_1 : \mathrm{T}_1; x_2 : \mathrm{T}_2; \ldots; x_n : \mathrm{T}_n \preceq x_1 : \mathrm{T}'_1; x_2 : \mathrm{T}'_2; \ldots; x_n : \mathrm{T}'_n$$

si pour tout i, $\mathrm{T}_i \preceq \mathrm{T}'_i$.

Jugements. Nous pouvons maintenant définir les jugements utilisés dans les règles de typage.

3.1.10. DÉFINITION (Jugements). Soient Γ un contexte, M et T des termes de ICC_Σ. Nous définissons les jugements suivants :

- un *jugement de bonne formation* : $\Gamma \vdash$, qui signifie que le contexte Γ est bien formé
- un *jugement de typage* : $\Gamma \vdash \mathrm{M} : \mathrm{T}$, qui signifie que, sous le contexte Γ, le terme M est de type T. Nous disons alors que Γ est le *contexte du jugement*, M le *terme du jugement* et T le *type du jugement*.

Le jugement de typage permet d'assigner un type à un terme. La présence d'un contexte est nécessaire pour associer un type à chaque variable libre présente utilisée au cours de la dérivation (notamment les variables libres du terme et du type). Le deuxième jugement, le jugement de formation, énonce qu'un contexte est valide.

Nous définissons ici des expressions qui seront couramment utilisées.

3.1.11. DÉFINITION (Terme typable). Soient M, T des termes et Γ un contexte.

(1) Termes typables :
   - M est *typable dans le contexte* Γ s'il existe un terme T tel que $\Gamma \vdash \mathrm{M} : \mathrm{T}$ est dérivable.
   - M est *typable* s'il existe un contexte Γ tel que M est typable dans Γ.

(2) Termes typés :
   - M *est de type* T *dans* Γ si $\Gamma \vdash \mathrm{M} : \mathrm{T}$ est dérivable.
   - M *est de type* T s'il existe Γ tel que M est de type T dans Γ.

(3) Types :
   - T *est un type de* M *dans* Γ si $\Gamma \vdash \mathrm{M} : \mathrm{T}$ est dérivable.
   - T *est un type de* M s'il existe Γ tel que T est le type de M dans Γ.
   - T *est un type* s'il existe un terme M tel que T est un type de M.

3.1.12. REMARQUE. Si Γ est le contexte vide, nous écrivons respectivement $\vdash$ et $\vdash \mathrm{M} : \mathrm{T}$ plutôt que $\bullet \vdash$ et $\bullet \vdash \mathrm{M} : \mathrm{T}$.

3.1.13. CONVENTION. Nous étendons la convention de Barendregt aux jugements de typage. Quitte à renommer par α-conversion, nous supposons que dans un jugement de typage, les variables liées sont distinctes des variables libres et des variables déclarées. Plus précisément, si nous considérons le jugement de typage $\Gamma \vdash \mathrm{M} : \mathrm{T}$, avec $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$, et si nous posons $\mathsf{BV} = \bigcup_{i=1}^{n} \mathsf{BV}(\mathrm{T}_i) \cup \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{T})$ et $\mathsf{FV} = \bigcup_{i=1}^{n} \mathsf{FV}(\mathrm{T}_i) \cup \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{T})$, alors nous avons $\mathsf{BV} \cap (\mathsf{FV} \cup \mathsf{DV}(\Gamma)) = \emptyset$.

3.2. Paramètres. Nous définissons des paramètres semblables à ceux utilisés dans la présentation des PTS [**GN91, Bar91**]. Un PTS est habituellement paramétré par un ensemble de sortes, un ensemble d'axiomes et un ensemble de règles. Nous avons déjà présenté l'ensemble des sortes considérées. Nous décrivons ici les autres paramètres qui seront utilisés dans les règles de typage : un ensemble d'axiomes et deux ensembles de règles.

3.2.1. DÉFINITION (Ensembles d'axiomes et de règles).

$$\begin{aligned}
\mathbf{Axiom} &= \{(\mathsf{Prop}, \mathsf{Type}_1)\} \cup \{(\mathsf{Type}_i, \mathsf{Type}_{i+1}) \mid i > 0\} \\
\mathbf{Rule} &= \{(s, \mathsf{Prop}, \mathsf{Prop}) \mid s \in \mathbf{Sort}\} \cup \\
&\quad \{(\mathsf{Prop}, \mathsf{Type}_i, \mathsf{Type}_i), (\mathsf{Type}_i, \mathsf{Type}_j, \mathsf{Type}_{max(i,j)}) \mid i, j > 0\} \\
\mathbf{Rule}' &= \{(s, \mathsf{Prop}, s) \mid s \in \mathbf{Sort}\} \cup \\
&\quad \{(\mathsf{Prop}, \mathsf{Type}_i, \mathsf{Type}_i), (\mathsf{Type}_i, \mathsf{Type}_j, \mathsf{Type}_{max(i,j)}) \mid i, j > 0\}
\end{aligned}$$

Notons que nous avons deux systèmes de règles et non pas un seul comme habituellement dans les PTS. Cela est justifié pour des raisons de cohérence logique. Les types produits sont impédicatifs ($(s, \mathsf{Prop}, \mathsf{Prop}) \in \mathbf{Rule}$) alors que les sommes sont prédicatives ($(s, \mathsf{Prop}, s) \in \mathbf{Rule}$). Si les sommes étaient prédicatives, le système serait incohérent [**Luo90**].

La proposition suivante détaille les propriétés vérifiées par les paramètres des ICC_Σ. Elle se montre trivialement d'après les définitions de **Axiom**, **Rule** et **Rule**$'$.

3.2.2. Proposition (Propriétés des paramètres)**.**

(1) Les paramètres de ICC_Σ sont fonctionnels *:*

- *pour toutes sortes s, s', s'', si $(s, s'), (s, s'') \in \mathbf{Axiom}$ alors $s' = s''$;*
- *pour toutes sortes s_1, s_2, s, s', si $(s_1, s_2, s), (s_1, s_2, s') \in \mathbf{Rule}$ (resp. $\in \mathbf{Rule}'$) alors $s = s'$.*

(2) les ensembles de règles sont complets *: pour toutes sortes s_1, s_2 il existe une sorte s_3 telle que $(s_1, s_2, s_3) \in \mathbf{Rule}$ (resp. $(s_1, s_2, s_3) \in \mathbf{Rule}'$).*

(3) ICC_Σ n'a pas de sorte maximale *: pour toute sorte s, il existe une sorte s' telle que $(s, s') \in \mathbf{Axiom}$.*

(4) les paramètres sont compatibles avec la cumulativité *:*

- *si $(s_1, s_2), (s_1', s_2') \in \mathbf{Axiom}$ avec $s_1 \preceq s_1'$, alors $s_2 \preceq s_2'$*
- *si $(s_1, s_2, s_3), (s_1', s_2', s_3') \in \mathbf{Rule}$ avec $s_1 \preceq s_1'$ et $s_2 \preceq s_2'$, alors $s_3 \preceq s_3'$*
- *si $(s_1, s_2, s_3), (s_1', s_2', s_3') \in \mathbf{Rule}'$ avec $s_1 \preceq s_1'$ et $s_2 \preceq s_2'$, alors $s_3 \preceq s_3'$.*

3.3. Règles de typage.

3.3.1. DÉFINITION (Règles de typage). Les *règles de typage* sont les règles d'inférence décrites dans les figures 4 et 5.[3]

Classification globale des règles.
Nous pouvons regrouper les règles en différents groupes.

(1) (WF-E) et (WF-S) permettent de vérifier la bonne formation des contextes : un contexte est vide est bien formé ((WF-E)) et un contexte reste bien formé s'il est étendu par une variable fraîche associée à un type typé par une sorte.

(2) (VAR) et (SORT) permettent de typer une variable et une sorte.

(3) (CUM) est l'équivalent de la règle de conversion des PTS, la cumulativité remplaçant la βη-conversion habituelle.

(4) Pour chaque constructeur de type, nous avons, à la manière de la déduction naturelle [**Gen35, Pra65**] :

3. Ces règles, ainsi que les principaux systèmes d'inférence présentés dans ce manuscrit, sont récapitulés dans dans l'annexe D.

$$\frac{}{\bullet \vdash}(\text{WF-E}) \qquad \frac{\Gamma \vdash T : s \quad x \notin \mathsf{DV}(\Gamma)}{\Gamma; x : T \vdash}(\text{WF-S})$$

$$\frac{\Gamma \vdash \quad (x : T) \in \Gamma}{\Gamma \vdash x : T}(\text{VAR}) \qquad \frac{\Gamma \vdash \quad (s_1, s_2) \in \textbf{Axiom}}{\Gamma \vdash s_1 : s_2}(\text{SORT})$$

$$\frac{\Gamma \vdash T : s_1 \quad \Gamma; x : T \vdash U : s_2 \quad (s_1, s_2, s_3) \in \textbf{Rule}}{\Gamma \vdash \Pi x : T . U : s_3}(\text{E-PRD})$$

$$\frac{\Gamma; x : T \vdash M : U \quad \Gamma \vdash \Pi x : T . U : s}{\Gamma \vdash \lambda x . M : \Pi x : T . U}(\text{LAM})$$

$$\frac{\Gamma \vdash M : \Pi x : T . U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U\,[x/N]}(\text{APP})$$

$$\frac{\Gamma \vdash T : s_1 \quad \Gamma; x : T \vdash U : s_2 \quad (s_1, s_2, s_3) \in \textbf{Rule}}{\Gamma \vdash \forall x : T . U : s_3}(\text{I-PRD})$$

$$\frac{\Gamma; x : T \vdash M : U \quad \Gamma \vdash \forall x : T . U : s \quad x \notin \mathsf{FV}(M)}{\Gamma \vdash M : \forall x : T . U}(\text{GEN})$$

$$\frac{\Gamma \vdash M : \forall x : T . U \quad \Gamma \vdash N : T}{\Gamma \vdash M : U\,[x/N]}(\text{INST})$$

$$\frac{\Gamma \vdash M : T \quad \Gamma \vdash T' : s \quad T \preceq T'}{\Gamma \vdash M : T'}(\text{CUM})$$

FIGURE 4. Règles d'inférence de ICC$_\Sigma$ (calcul des constructions)

- une règle de *formation* qui type le constructeur de type : cette règle vérifie que le domaine et le codomaine sont typés par des sortes respectant les règles des paramètres **Rule** et **Rule**′ ;
- une règle d'*introduction* qui type le constructeur associé au constructeur de type considéré ;
- une règle d'*élimination* (deux règles pour le type sous-ensemble) qui type l'éliminateur associé au constructeur de type (s'il existe) ou qui, plus généralement, type un terme à partir d'un terme typé par le constructeur de type.

Les trois premiers groupes de règles ainsi que les règles du produit dépendant sont semblables à celles des PTS aux deux différences près déjà discutées : l'absence d'annotation dans les abstractions et l'utilisation de la relation de cumulativité au lieu de la conversion.

Étudions maintenant les règles des autres constructeurs de type.

Produit dépendant implicite.
Dans ICC, Miquel ajoute le produit dépendant implicite. Sa règle de formation est similaire à celle du produit dépendant usuel. L'abstraction, dans la règle d'introduction, l'application, dans la règle d'élimination, sont implicites, à la manière de l'abstraction et l'application de type dans Système F

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma; x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \Sigma x : A.B : s_3}\,(\Sigma)$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[x/a] \quad \Gamma \vdash \Sigma x : A.B : s}{\Gamma \vdash (a, b) : \Sigma x : A.B}\,(\Sigma\text{-I})$$

$$\frac{\Gamma \vdash P : \Sigma x : A.B \to s \quad \Gamma \vdash c : \Sigma x : A.B \quad \Gamma; x : A; y : B \vdash f : P\,(x, y)}{\Gamma \vdash \mathsf{Elim}_\Sigma(xy.f, c) : P\,c}\,(\Sigma\text{-E})$$

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma; x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \exists x : A.B : s_3}\,(\exists)$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[x/a] \quad \Gamma \vdash \exists x : A.B : s}{\Gamma \vdash (\diamond, b) : \exists x : A.B}\,(\exists\text{-I})$$

$$\frac{\Gamma \vdash P : \exists x : A.B \to s \quad \Gamma; x : A; y : B \vdash f : P\,(\diamond, y) \quad \Gamma \vdash c : \exists x : A.B \qquad x \notin \mathsf{FV}(f)}{\Gamma \vdash \mathsf{Elim}_\exists(y.f, c) : P\,c}\,(\exists\text{-E})$$

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma; x : A \vdash B : s_2}{\Gamma \vdash \{x : A \mid B\} : s_1}\,(\textsc{Sub})$$

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[x/a] \quad \Gamma \vdash \{x : A \mid B\} : s}{\Gamma \vdash a : \{x : A \mid B\}}\,(\textsc{Sub}\text{-I})$$

$$\frac{\Gamma \vdash a : \{x : A \mid B\}}{\Gamma \vdash a : A}\,(\textsc{Sub}\text{-E-1})$$

$$\frac{\Gamma \vdash P : s \qquad \Gamma; y : B[x/c] \vdash f : P \quad \Gamma \vdash c : \{x : A \mid B\} \quad y \notin \mathsf{FV}(f)}{\Gamma \vdash f : P}\,(\textsc{Sub}\text{-E-2})$$

FIGURE 5. Règles d'inférence de ICC_Σ (Types somme)

à la Curry [**BC00**]. Ainsi le produit implicite peut lui aussi, comme la quantification de type $\forall\alpha.A$ du Système F, se comprendre comme un type intersection, dans le sens où un terme de type $\forall x : T.U$ peut être de type $U[x/N]$ pour tout N typé par T (dans un contexte idoine).

Notons la condition de bord $x \notin \mathsf{FV}(M)$ dans la règle d'introduction qui est extrêmement importante car elle interdit d'abstraire implicitement une variable qui apparaît calculatoirement dans le terme.

Sommes dépendantes.
ICC_Σ enrichit ICC avec des sommes dépendantes. Les sommes dépendantes ont été introduites par Martin-Löf dans sa théorie des types [**ML75, MLS84**]. Elles permettent de typer les paires dépendantes (a, b), où le type de b dépend de a.

Nous introduisons trois sommes dépendantes dans ICC_Σ.

- La première somme dépendante est analogue à la somme dépendante prédicative de la théorie des types de Martin-Löf. La seule différence est que, puisque la syntaxe est à la Curry, les paires et les variables des éliminateurs ne sont pas annotées. Pour l'élimination, nous choisissons d'utiliser un éliminateur unique $\mathsf{Elim}_\Sigma(xy.f, c)$ plutôt que la formulation avec les deux projections π_1, π_2 et la règle de réduction $(\pi_1(c), \pi_2(c)) \to c$ (*surjective pairing*).
- Nous ajoutons deux autres sommes dépendantes : le type existentiel et le type sous-ensemble. Elles jouent pour les paires un rôle analogue au produit dépendant implicite pour les abstractions. Ainsi leur raison d'être est de permettre de typer les paires (a, b) lorsque a (type existentiel) ou b (type sous-ensemble) dont le contenu n'est pas considéré calculatoire.

Le type existentiel est semi-implicite.
Le type existentiel n'est pas malheureusement pas complètement un type implicite. Idéalement, nous aurions voulu avoir les règles d'introduction et d'élimination suivantes [**Ber09**] :

$$\frac{\Gamma \vdash a : \mathrm{A} \qquad \Gamma \vdash b : \mathrm{B}\,[x/a] \qquad \Gamma \vdash \exists x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash b : \exists x : \mathrm{A}.\mathrm{B}}$$

$$\frac{\begin{array}{cc}\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s & \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,y \\ \Gamma \vdash c : \exists x : \mathrm{A}.\mathrm{B} & x \notin \mathsf{FV}(f)\end{array}}{\Gamma \vdash f\,[y/c] : \mathrm{P}\,c}$$

Malheureusement, ces règles invalident la propriété de préservation du typage. En effet nous pouvons reproduire dans ICC$_\Sigma$ le contre-exemple donné dans [**BDd95**] et repris dans [**SU98**] et [**Tat07**]. Posons $\mathrm{M}_1 = z\,((\lambda t.\,t)\,x\,y)\,((\lambda t.\,t)\,x\,y)$ et $\mathrm{M}_2 = z\,(x\,y)\,((\lambda t.\,t)\,x\,y)$ et $\Gamma = \mathrm{B} : s_\mathrm{B}; \mathrm{C} : s_\mathrm{C}; \mathrm{D} : s_\mathrm{D}; x : \mathrm{B} \to \exists \mathrm{X} : \mathrm{D}.(\mathrm{X} \to \mathrm{X}); y : \mathrm{B}; z : \Pi \mathrm{X} : \mathrm{D}.((\mathrm{X} \to \mathrm{X}) \to (\mathrm{X} \to \mathrm{X}) \to \mathrm{C})$ où $s_\mathrm{B}, s_\mathrm{C}, s_\mathrm{D}$ sont des sortes quelconques. Nous avons $\Gamma \vdash \mathrm{M}_1 : \mathrm{C}$ et $\mathrm{M}_1 \to_\beta \mathrm{M}_2$ mais $\Gamma \not\vdash \mathrm{M}_2 : \mathrm{C}$.

Nous avons donc modifié les règles en adaptant à ICC$_\Sigma$ la solution utilisée par [**Tat07**]. Nous avons ainsi introduit un constructeur pour le type existentiel, la paire existentielle, qui est une paire où le membre de gauche est anonyme (mais sa présence est nécessaire pour la distinguer des autres termes). Nous ajoutons également un éliminateur explicite qui ressemble à celui de la somme dépendante : la seule différence est qu'une seule variable est liée à la branche f. La présence d'un constructeur et d'un éliminateur conduisent naturellement à l'ajout d'une règle de réduction : il s'agit de la règle $\iota_\exists$. Au final, le type existentiel a toutes les caractéristiques d'un type explicite (constructeur, éliminateur, règle de réduction) même si les parties logiques n'apparaissent pas.

Type sous-ensemble. Ce type est bien un type implicite : il n'a ni constructeur, ni éliminateur, ni règle de réduction associée. Notons la présence de deux règles d'élimination au lieu d'une seule pour les deux autres types somme. Cela est un choix effectué pour des raisons de commodité. En effet, nous aurions pu choisir la règle suivante en lieu et place [4] des deux règles d'élimination :

$$\frac{\Gamma \vdash \mathrm{P} : \{x : \mathrm{A} \mid \mathrm{B}\} \to s \qquad \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,x \qquad y \notin \mathsf{FV}(f)}{\Gamma \vdash f\,[x/c] : \mathrm{P}\,c}\ (\textsc{Sub-E})$$

Mais cela aurait rendu la preuve de préservation du typage (cf. chapitre 2) sensiblement plus compliquée, du fait de la présence d'une substitution dans le terme typé par la conclusion.

4. Nous comparons dans l'annexe B ICC$_\Sigma$ avec un système de types où les règles (Sub-E-1) et (Sub-E-2) sont remplacées par la règle (Sub-E). Nous montrons qu'un tel système est inclus dans ICC$_\Sigma$, et que sous certaines conditions, la réciproque est vraie.

Règles de ICC omises dans ICC$_\Sigma$. Dans ICC$_\Sigma$, nous omettons deux règles présentes dans ICC.

La première de ces deux règles est la règle de l'extensionnalité.

$$\frac{\Gamma \vdash \lambda x.\,\mathrm{M}\,x : \mathrm{T} \qquad x \notin \mathsf{FV}(\mathrm{M})}{\Gamma \vdash \mathrm{M} : \mathrm{T}} \text{ (Ext)}$$

Nous n'avons pas retenu cette règle car cela aurait fortement compliqué l'étude du système (en particulier la préservation du typage et le renforcement). Mais avoir cette règle dans ICC$_\Sigma$ nous aurait permis d'avoir la préservation du typage par η-réduction dans ICC$_\Sigma$, ce qui ne semble pas être le cas dans le système actuel (cf. chapitre suivant).

La deuxième règle que nous n'avons pas retenue est la règle de renforcement.

$$\frac{\Gamma; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{U} \qquad x \notin \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{U})}{\Gamma \vdash \mathrm{M} : \mathrm{U}} \text{ (Str)}$$

Nous ne souhaitons pas avoir cette règle car, comme le remarque Miquel, elle revient à assimiler le produit implicite non-dépendant $[\mathrm{T}] \to \mathrm{U}$ et son codomaine U. Cela nous semble incompatible avec le fait d'avoir un systèmes de type pour la programmation certifiée : un programme qui vérifierait la spécification $[\mathrm{T}] \to \mathrm{U}$ vérifierait automatiquement la spécification U même si T n'est pas habité. [5]

3.3.2. Définition (Jugement valide dans ICC$_\Sigma$). Un jugement $\Gamma \vdash \mathrm{M} : \mathrm{T}$ est *valide* ou *dérivable* dans ICC$_\Sigma$ s'il existe une dérivation utilisant les règles de typage de ICC$_\Sigma$ dont il est la racine.

3.4. Propriétés.

Propriétés métathéoriques de base.

3.4.1. Lemme (Bonne formation des contextes)**.** *Soient* Γ, Δ *des contextes et* M, T *des termes.*

(1) Toute dérivation de $\Gamma; \Delta \vdash$ *inclut une dérivation de* $\Gamma \vdash$.

(2) Toute dérivation de $\Gamma; \Delta \vdash \mathrm{M} : \mathrm{T}$ *inclut une dérivation de* $\Gamma \vdash$.

Démonstration. Par induction mutuelle sur les structures des dérivations des jugements $\Gamma; \Delta \vdash$ et $\Gamma; \Delta \vdash \mathrm{M} : \mathrm{T}$ en considérant la dernière règle appliquée. Le cas de (Wf-E) est trivial. Pour les autres règles, si $\Gamma; \Delta$ est le contexte de la conclusion, toutes les prémisses ont un contexte de la forme $\Gamma; \Delta; \Delta'$. Nous pouvons alors immédiatement conclure par induction. □

En prenant $\Delta = \bullet$, nous démontrons ce corollaire qui sera utilisé plus souvent.

3.4.2. Corollaire (Inclusion de dérivation)**.** *Si* $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *alors* $\Gamma \vdash$. *De plus la dérivation de* $\Gamma \vdash$ *est incluse dans celle de* $\Gamma \vdash \mathrm{M} : \mathrm{T}$.

3.4.3. Lemme (Déclaration des variables libres)**.** *Soit* $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ *un contexte*

(1) Si $\Gamma \vdash$, *alors* $FV(\mathrm{T}_i) \subset \{x_1, \ldots, x_{i-1}\}$ *pour tout* $i \in [1 \ldots n]$.

(2) Si $\Gamma \vdash \mathrm{M} : \mathrm{T}$, *alors* $FV(\mathrm{M}) \subset DV(\Gamma)$, $FV(\mathrm{T}) \subset DV(\Gamma)$ *et* $FV(\mathrm{T}_i) \subset \{x_1, \ldots, x_{i-1}\}$ *pour tout* $i \in [1 \ldots n]$.

Démonstration. Par induction mutuelle sur la structure des dérivations en considérant la dernière règle appliquée. Toutes les règles se traitent aisément compte-tenu de la définition des variables libres d'un terme (définition 1.2.7 du chapitre 1) et des variables déclarées d'un contexte (définition 3.1.6). Pour les règles (App) et (Inst), nous utilisons le lemme 1.2.12. □

3.4.4. Remarque. Si $\Gamma \vdash$, alors $\mathsf{FV}(\Gamma) \subset \mathsf{DV}(\Gamma)$.

5. Notons que ICC muni de la règle de renforcement est cohérent. Cela ne l'invalide donc pas en tant que système logique.

Affaiblissement.

3.4.5. Lemme (Affaiblissement généralisé)**.** *Soient* Γ, Γ', Δ *des contextes tels que* $\Gamma \subseteq \Gamma'$, $\Gamma' \vdash$ *et* $DV(\Gamma') \cap DV(\Delta) = \emptyset$. *Alors*

(1)

$$\Gamma; \Delta \vdash \quad \Rightarrow \quad \Gamma'; \Delta \vdash$$

(2)

$$\Gamma; \Delta \vdash M : T \quad \Rightarrow \quad \Gamma'; \Delta \vdash M : T$$

DÉMONSTRATION. Par induction mutuelle sur les dérivations des jugements $\Gamma; \Delta \vdash$ et $\Gamma; \Delta \vdash M : T$ en considérant la dernière règle appliquée. Toutes les règles sont immédiates. □

En prenant $\Delta = \bullet$, nous avons le résultat classique d'affaiblissement.

3.4.6. Corollaire (Affaiblissement)**.** *Si* $\Gamma \subseteq \Gamma'$ *et* $\Gamma' \vdash$ *alors*

$$\Gamma \vdash M : T \quad \Rightarrow \quad \Gamma' \vdash M : T$$

3.4.7. REMARQUE. Il est délicat de démontrer directement le corollaire, à cause des règles dont une des prémisses a un contexte $\Gamma; \Delta$ (Δ contient une ou deux déclarations de variables) qui étend le contexte Γ de la conclusion. En effet, pour pouvoir appliquer par induction le lemme à la prémisse dont le contexte est $\Gamma; \Delta$ il faut montrer $\Gamma'; \Delta \vdash$. Puisque Δ contient une ou deux déclarations de variables, cela est faisable par inversion de la règle (WF-S) et en utilisant la bonne formation des contextes, mais cela est un peu technique.

Substitutivité.

3.4.8. Lemme (Substitutivité)**.** *Si* $\Gamma \vdash N : U$ *alors :*

(1)

$$\Gamma; x : U; \Delta \vdash \quad \Rightarrow \quad \Gamma; \Delta[x/N] \vdash$$

(2)

$$\Gamma; x : U; \Delta \vdash M : T \quad \Rightarrow \quad \Gamma; \Delta[x/N] \vdash M[x/N] : T[x/N]$$

DÉMONSTRATION. Nous procédons par induction mutuelle sur les dérivations des jugements $\Gamma; x : U; \Delta \vdash$ et $\Gamma; x : U; \Delta \vdash M : T$ en considérant la dernière règle appliquée.

- (WF-E) ne peut être la dernière règle appliquée.
- Les cas des règles (WF-S) et (SORT), des règles de formation ((E-PRD), (I-PRD), (Σ), (∃), (SUB)), de (LAM) et (Σ-E), (SUB-E-1) se résolvent immédiatement par induction.
- Pour les règles (GEN), (∃-E) et (SUB-E-2), qui ont des conditions de bord, il suffit d'utiliser, en plus des hypothèses d'induction, le lemme 1.2.12.
- (VAR) La dernière règle appliquée est :

$$\frac{\Gamma; x : U; \Delta \vdash \qquad (y : A) \in \Gamma; x : U; \Delta}{\Gamma; x : U; \Delta \vdash y : A}\ (\text{VAR})$$

  Si $y \neq x$, les hypothèses d'induction permettent de conclure. Si $y = x$, par induction $\Gamma; \Delta[x/N] \vdash$, nous pouvons donc affaiblir $\Gamma \vdash N : U$ et dériver $\Gamma; \Delta[x/N] \vdash N : U$.
- Les règles (APP), (INST), (Σ-I), (∃-I), (SUB-I) contiennent des substitutions. Elles se traitent par induction en utilisant le lemme des substitutions multiples (lemme 1.2.13).
- Pour la règle (CUM), l'appel aux hypothèses d'induction est complété par le lemme 2.3.2 de la page 14 liant cumulativité et substitution.

□

Cumulativité de contextes et jugements.

3.4.9. Lemme. *Soient* Γ', Γ *des contextes tels que* $\Gamma' \preceq \Gamma$ *et* $\Gamma' \vdash$*, alors :*

(1)

$$\Gamma; \Delta \vdash \quad \Rightarrow \quad \Gamma'; \Delta \vdash$$

(2)

$$\Gamma; \Delta \vdash M : T \quad \Rightarrow \quad \Gamma'; \Delta \vdash M : T$$

DÉMONSTRATION. Par induction mutuelle sur les dérivations des jugements $\Gamma; \Delta \vdash$ et $\Gamma; \Delta \vdash M : T$ en considérant la dernière règle appliquée. Toutes les règles sont immédiates sauf (VAR) dans le cas où la variable est une variable de Γ et non de Δ.

Si $\Gamma; \Delta \vdash x : T$ avec $(x : T) \in \Gamma$, montrons que $\Gamma'; \Delta \vdash x : T$. Par hypothèse, nous avons $\Gamma'; \Delta \vdash$ et T' tel que $T' \preceq T$ et $(x : T') \in \Gamma'$. En appliquant (VAR), nous dérivons $\Gamma'; \Delta \vdash x : T'$.

Montrons qu'il existe une sorte s telle que $\Gamma'; \Delta \vdash T : s$. Par application de (CUM), nous déduirons $\Gamma'; \Delta \vdash x : T$. Posons $\Gamma = \Gamma_1; x : T; \Gamma_2$ et $\Gamma' = \Gamma'_1; x : T; \Gamma'_2$. D'après le lemme 3.4.1, la dérivation de $\Gamma; \Delta \vdash x : T$ inclut une dérivation de $\Gamma_1; x : T \vdash$. Par inversion de (WF-S), il existe une sorte s telle qu'une dérivation de $\Gamma_1 \vdash T : s$ est incluse dans celle de $\Gamma; \Delta \vdash x : T$. Par induction, nous dérivons $\Gamma'_1 \vdash T : s$, puis, par affaiblissement, $\Gamma'; \Delta \vdash T : s$. □

De même que pour l'affaiblissement, la présence de Δ est nécessaire car certaines prémisses des règles de typage ont un contexte qui étend le contexte de la conclusion. En choisissant $\Delta = \bullet$, nous obtenons ce résultat plus pratique.

3.4.10. Corollaire (Cumulativité et changement de contexte)**.** *Si* $\Gamma \vdash M : T$, $\Gamma' \preceq \Gamma$ *et* $\Gamma' \vdash$*, alors* $\Gamma' \vdash M : T$.

Échange de variables. L'énoncé habituel du lemme d'échange de variables est le suivant : la dérivabilité de $\Gamma; x : A; y : B; \Delta \vdash M : T$, avec $x \notin \mathsf{FV}(B)$, implique celle de $\Gamma; y : B; x : A; \Delta \vdash M : T$. Cet énoncé est valable dans ICC, grâce à la présence de la règle de renforcement, mais ne l'est pas a priori dans ICC_Σ. Intuitivement, le fait que $x \notin \mathsf{FV}(B)$ ne permet pas de prouver que $\Gamma; x : A; y : B \vdash$ car x peut être utilisé dans la dérivation prouvant que B est typé par une sorte même s'il n'apparaît pas dans B. [6]

3.4.11. Lemme (Échange de variables)**.**

(1) Si $\Gamma; x : A; y : B; \Delta \vdash$ *et* $\Gamma \vdash B : s$ *alors* $\Gamma; y : B; x : A; \Delta \vdash$.

(2) Si $\Gamma; x : A; y : B; \Delta \vdash M : T$ *et* $\Gamma \vdash B : s$ *alors* $\Gamma; y : B; x : A; \Delta \vdash M : T$.

DÉMONSTRATION. Les points (1) et (2) se prouvent simultanément par récurrence sur la longueur du contexte Δ.

- Cas de base : $\Delta = \bullet$.
  (1) D'après le lemme 3.4.1 (bonne formation des contextes), nous avons $\Gamma; x : A \vdash$. Par inversion de la règle (WF-S), il existe une sorte s' telle que $\Gamma \vdash A : s'$. En appliquant (WF-S), nous dérivons $\Gamma; y : B \vdash$. Par affaiblissement, $\Gamma; y : B \vdash A : s$. En appliquant (WF-S), nous dérivons $\Gamma; y : B; x : A \vdash$.
  (2) Nous avons $\Gamma; x : A; y : B \vdash$ (lemme 3.4.1). D'après (1), $\Gamma; y : B; x : A \vdash$. Nous concluons par affaiblissement (lemme 3.4.6) puisque $\Gamma; x : A; y : B \subseteq \Gamma; x : A; y : B$.

6. Prendre par exemple $\Gamma = A : \mathsf{Prop}; B : [A] \to \mathsf{Prop}$.

- Cas récursif.
  (1) Par inversion de (WF-S) et par hypothèse de récurrence.
  (2) Similaire au cas de base.

□

Lemmes d'inversion faible. Les lemmes d'inversion sont beaucoup plus compliqués à mettre en place dans ICC_Σ que dans les PTS à la Church, car les règles de typage sont faiblement dirigées par la syntaxe.[7] En effet, en plus de la règle de cumulativité, les règles d'introduction et d'élimination des deux constructeurs de type purement implicites (le produit dépendant implicite et le type sous-ensemble), soit cinq règles, modifient la structure du type d'un jugement sans modifier celle du terme. Ainsi une abstraction peut certes être typée par un produit dépendant, mais elle peut l'être également par un produit dépendant implicite ou un type sous-ensemble…

Pour contourner ce problème, nous introduisons la notion de types dérivés. Son but est de réfléter les transformations qui peuvent avoir lieu sur la structure du type d'un jugement l'application des règles d'introduction et d'élimination des produits dépendants implicites et des types sous-ensemble.

3.4.12. DÉFINITION (Types dérivés). Soit T un terme de ICC_Σ. L'ensemble des *types dérivés* de T, noté $\mathsf{Deriv}(\mathrm{T})$, est défini inductivement par :

$$\frac{}{\mathrm{T} \in \mathsf{Deriv}(\mathrm{T})}\ (\textsc{Deriv-Base})$$

$$\frac{\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})}{\forall x : \mathrm{A}.\,\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})}\ (\textsc{Deriv-}\forall)$$

$$\frac{\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})}{\{x : \mathrm{R} \mid \mathrm{B}\} \in \mathsf{Deriv}(\mathrm{T})}\ (\textsc{Deriv-Sub})$$

$$\frac{\mathrm{R} \in \mathsf{Deriv}(\mathrm{T}) \qquad \mathrm{R} \preceq \mathrm{P}}{\mathrm{P} \in \mathsf{Deriv}(\mathrm{T})}\ (\textsc{Deriv-Cum})$$

3.4.13. Lemme. *Si* $\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})$ *alors* $\mathrm{T} \preceq \mathrm{R}$ *ou il existe* A, B *tels que* $\forall x : \mathrm{A}.\,\mathrm{B} \preceq \mathrm{R}$ *ou* $\{x : \mathrm{A} \mid \mathrm{B}\} \preceq \mathrm{R}$.

DÉMONSTRATION. Par induction sur la dérivation de $\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})$ en considérant la dernière règle appliquée. Les règles (DERIV-BASE), (DERIV-∀) et (DERIV-SUB) sont immédiates. Pour (DERIV-CUM), l'hypothèse d'induction permet de conclure sans difficulté du fait de la transitivité de $\preceq$. □

Nous montrons ici que la notion de types dérivés est conservée par substitution. Cela est rendu nécessaire par la règle (INST).

3.4.14. Lemme (Types dérivés et substitution)**.** *Si* $\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})$ *et si* $z \notin BV(\mathrm{R})$ *alors* $\mathrm{R}[z/\mathrm{N}] \in \mathsf{Deriv}(\mathrm{T}[z/\mathrm{N}])$.

DÉMONSTRATION. Par induction sur la dérivation de $\mathrm{R} \in \mathsf{Deriv}(\mathrm{T})$. Nous considérons la dernière règle appliquée.

- (DERIV-BASE) : Si $\mathrm{R} = \mathrm{T}$, nous avons bien $\mathrm{T}[z/\mathrm{N}] \in \mathsf{Deriv}(\mathrm{T}[z/\mathrm{N}])$.
- (DERIV-∀) : Par hypothèse d'induction, $\mathrm{R}[z/\mathrm{N}] \in \mathsf{Deriv}(\mathrm{T}[z/\mathrm{N}])$ d'où $(\forall x : \mathrm{A}.\,\mathrm{R})[z/\mathrm{N}] = \forall x : \mathrm{A}[z/\mathrm{N}].\,\mathrm{R}[z/\mathrm{N}] \in \mathsf{Deriv}(\mathrm{T}[z/\mathrm{N}])$.
- (DERIV-SUB) : similaire au cas précédent.

□

7. Nous y reviendrons à la section 2 du chapitre 2

3.4.15. Lemme.

(1) Si $R \in \mathsf{Deriv}(T)$ *et si* $R \preceq \forall x : A.B$ *alors* $T \preceq \forall x : A.B$ *ou* $B \in \mathsf{Deriv}(T)$.

(2) Si $R \in \mathsf{Deriv}(T)$ *et si* $R \preceq \{x : A \mid B\}$ *alors* $T \preceq \{x : A \mid B\}$ *ou* $A \in \mathsf{Deriv}(T)$.

DÉMONSTRATION. Les deux points se prouvent de manière similaire. Traitons le cas du produit implicite. Nous procédons par induction sur la dérivation de $R \in \mathsf{Deriv}(T)$ en considérant la dernière règle appliquée.

- (DERIV-BASE) : Nous avons $R = T \preceq \forall x : A.B$.
- (DERIV-∀) : Si $R = \forall x : A_R.B_R \preceq \forall x : A.B$ avec $B_R \in \mathsf{Deriv}(T)$. Par inversion de la cumulativité, $B_R \preceq B$. En appliquant (DERIV-CUM) à $B_R \preceq B$, nous avons bien $B \in \mathsf{Deriv}(T)$.
- (DERIV-SUB) : cas impossible car, par inversion de la cumulativité, nous ne pouvons avoir $R = \{x : A_R \mid B_R\} \preceq \forall x : A.B$.
- (DERIV-CUM) : immédiat par hypothèse d'induction car $P \preceq R \preceq \forall x : A.B$.

□

Nous pouvons maintenant exprimer des résultats d'inversion pour ICC_Σ. Notons l'absence de lemme pour la variable et une plus faible caractérisation pour les éliminateurs, car il n'est pas possible de connaître la structure de leur type naturel. [8]

3.4.16. Lemme (Lemmes d'inversion faible)**.** *Les lemmes d'inversion sont présentés à la figure 6.*

DÉMONSTRATION. Nous ne traitons que le cas de l'abstraction. Les autres cas sont similaires (sortes, constructeurs de type et autres constructeurs) ou plus simples (éliminateurs).

Par induction sur la dérivation du jugement $\Gamma \vdash \lambda x.M : R$ en considérant la dernière règle de la dérivation. Sept cas sont possibles.

- (LAM) Cas de base : évident.
- (GEN) Si

$$\frac{\Gamma; z : T \vdash \lambda x.M : U \qquad \Gamma \vdash \forall z : T.U : s_0 \qquad z \notin \mathsf{FV}(\lambda x.M)}{\Gamma \vdash \lambda x.M : \forall z : T.U}$$

  montrons qu'il existe des termes A, B, une sorte s et un contexte Δ tels que $\Gamma; \Delta; x : A \vdash M : B$, $\Gamma; \Delta \vdash \Pi x : A.B : s$ et $\forall z : T.U \in \mathsf{Deriv}(\Pi x : A.B)$.
  - Par hypothèses d'induction sur $\Gamma; z : T \vdash \lambda x.M : U$, il existe A, B, s et Δ_0 tels que $\Gamma; z : T; \Delta_0; x : A \vdash M : B$, $\Gamma; z : T; \Delta_0 \vdash \Pi x : A.B : s$ et $U \in \mathsf{Deriv}(\Pi x : A.B)$.
  - En appliquant (DERIV-∀) à $U \in \mathsf{Deriv}(\Pi x : A.B)$, nous obtenons $\forall z : T.U \in \mathsf{Deriv}(\Pi x : A.B)$. A, B, s et $\Delta = z : T; \Delta_0$ conviennent donc.
- (INST) Si

$$\frac{\Gamma \vdash \lambda x.M : \forall z : T.U \qquad \Gamma \vdash N : T}{\Gamma \vdash \lambda x.M : U\,[z/N]}$$

  montrons qu'il existe des termes A, B, une sorte s et un contexte Δ tels que $\Gamma; \Delta; x : A \vdash M : B$, $\Gamma; \Delta \vdash \Pi x : A.B : s$ et $U\,[z/N] \in \mathsf{Deriv}(\Pi x : A.B)$.
  - Par hypothèses d'induction sur $\Gamma \vdash \lambda x.M : \forall z : T.U$, il existe A_0, B_0, s et Δ_0 tels que $\Gamma; \Delta_0; x : A_0 \vdash M : B_0$, $\Gamma; \Delta_0 \vdash \Pi x : A_0.B_0 : s$ et $\forall z : T.U \in \mathsf{Deriv}(\Pi x : A_0.B_0)$.
  - Posons $A = A_0\,[z/N]$, $B = B_0\,[z/N]$ et $\Delta = \Delta_0\,[z/N]$.

8. C'est-à-dire le type du jugement où apparaît pour la première fois dans la dérivation le terme considéré.

$$\Gamma \vdash s : \mathrm{R} \Longrightarrow \exists s' \left\{ \begin{array}{l} (s, s') \in \mathbf{Axiom} \\ \mathrm{R} \in \mathsf{Deriv}(s') \end{array} \right.$$

$$\Gamma \vdash \Box x : \mathrm{T} . \mathrm{U} : \mathrm{R} \Longrightarrow \exists s_1, s_2, s_3, \Delta \left\{ \begin{array}{l} \Gamma; \Delta \vdash \mathrm{T} : s_1 \\ \Gamma; \Delta; x : \mathrm{T} \vdash \mathrm{U} : s_2 \\ (s_1, s_2, s_3) \in \mathbf{Rule}_\Box \\ \mathrm{R} \in \mathsf{Deriv}(s_3) \end{array} \right.$$

$$\Gamma \vdash \lambda x . \mathrm{M} : \mathrm{R} \Longrightarrow \exists \mathrm{T}, \mathrm{U}, s, \Delta \left\{ \begin{array}{l} \Gamma; \Delta; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{U} \\ \Gamma; \Delta \vdash \Pi x : \mathrm{T} . \mathrm{U} : s \\ \mathrm{R} \in \mathsf{Deriv}(\Pi x : \mathrm{T} . \mathrm{U}) \end{array} \right.$$

$$\Gamma \vdash \mathrm{M}\,\mathrm{N} : \mathrm{R} \Longrightarrow \exists x, \mathrm{T}, \mathrm{U}, \Delta \left\{ \begin{array}{l} \Gamma; \Delta \vdash \mathrm{M} : \Pi x : \mathrm{T} . \mathrm{U} \\ \Gamma; \Delta \vdash \mathrm{N} : \mathrm{T} \end{array} \right.$$

$$\Gamma \vdash (a, b) : \mathrm{R} \Longrightarrow \exists x, \mathrm{A}, \mathrm{B}, \Delta \left\{ \begin{array}{l} \Gamma; \Delta \vdash a : \mathrm{A} \\ \Gamma; \Delta \vdash b : \mathrm{B}[x/a] \\ \mathrm{R} \in \mathsf{Deriv}(\Sigma x : \mathrm{A} . \mathrm{B}) \end{array} \right.$$

$$\Gamma \vdash \mathsf{Elim}_\Sigma(xy.f, c) : \mathrm{R} \Longrightarrow \exists \mathrm{P}, \mathrm{A}, \mathrm{B}, s, \Delta \left\{ \begin{array}{l} \Gamma; \Delta \vdash \mathrm{P} : \Sigma x : \mathrm{A} . \mathrm{B} \to s \\ \Gamma; \Delta \vdash c : \Sigma x : \mathrm{A} . \mathrm{B} \\ \Gamma; \Delta; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}(x, y) \end{array} \right.$$

$$\Gamma \vdash (\diamond, b) : \mathrm{R} \Longrightarrow \exists x, a, \mathrm{A}, \mathrm{B}, \Delta \left\{ \begin{array}{l} \Gamma; \Delta \vdash a : \mathrm{A} \\ \Gamma; \Delta \vdash b : \mathrm{B}[x/a] \\ \mathrm{R} \in \mathsf{Deriv}(\exists x : \mathrm{A} . \mathrm{B}) \end{array} \right.$$

$$\Gamma \vdash \mathsf{Elim}_\exists(y.f, c) : \mathrm{R} \Longrightarrow \exists x, \mathrm{P}, \mathrm{A}, \mathrm{B}, s, \Delta \left\{ \begin{array}{l} \Gamma; \Delta \vdash \mathrm{P} : \exists x : \mathrm{A} . \mathrm{B} \to s \\ \Gamma; \Delta \vdash c : \exists x : \mathrm{A} . \mathrm{B} \\ \Gamma; \Delta; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}(\diamond, y) \\ x \notin \mathsf{FV}(f) \end{array} \right.$$

FIGURE 6. Lemmes d'inversion faible pour ICC_Σ

- D'après le point (1) du lemme 3.4.15 appliqué à $\forall z : \mathrm{T} . \mathrm{U} \in \mathsf{Deriv}(\Pi x : \mathrm{A}_0 . \mathrm{B}_0)$, nous avons $\mathrm{U} \in \mathsf{Deriv}(\Pi x : \mathrm{A}_0 . \mathrm{B}_0)$, puisque $\Pi x : \mathrm{A}_0 . \mathrm{B}_0$ n'est pas comparable à un produit implicite. D'après le lemme 3.4.14, $\mathrm{U}[z/\mathrm{N}] \in \mathsf{Deriv}(\Pi x : \mathrm{A} . \mathrm{B})$.
- Puisque $\Gamma \vdash \mathrm{N} : \mathrm{T}$, et que $z \notin \mathsf{FV}(\mathrm{M})$, nous avons, par substitutivité sur $\Gamma; \Delta_0; x : \mathrm{A}_0 \vdash \mathrm{M} : \mathrm{B}_0$ et $\Gamma; \Delta_0 \vdash \Pi x : \mathrm{A}_0 . \mathrm{B}_0 : s$, $\Gamma; \Delta; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$ et $\Gamma; \Delta \vdash \Pi x : \mathrm{A} . \mathrm{B} : s$.

- (Cum) Par hypothèses d'induction et en appliquant (Deriv-Cum).
- (Sub-I) Si

$$\frac{\Gamma \vdash \lambda x . \mathrm{M} : \mathrm{T} \qquad \Gamma \vdash b : \mathrm{U}[z/\lambda x . \mathrm{M}] \qquad \Gamma \vdash \{z : \mathrm{T} \mid \mathrm{U}\} : s_0}{\Gamma \vdash \lambda x . \mathrm{M} : \{z : \mathrm{T} \mid \mathrm{U}\}}$$

  montrons qu'il existe $\mathrm{A}, \mathrm{B}, s$ et Δ tels que $\Gamma; \Delta; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$, $\Gamma; \Delta \vdash \Pi x : \mathrm{A} . \mathrm{B} : s$ et $\{z : \mathrm{T} \mid \mathrm{U}\} \in \mathsf{Deriv}(\Pi x : \mathrm{A} . \mathrm{B})$.
  - Par hypothèses d'induction sur $\Gamma \vdash \lambda x . \mathrm{M} : \mathrm{T}$, il existe $\mathrm{A}, \mathrm{B}, s$ et Δ tels que $\Gamma; \Delta; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$, $\Gamma; \Delta \vdash \Pi x : \mathrm{A} . \mathrm{B} : s$ et $\mathrm{T} \in \mathsf{Deriv}(\Pi x : \mathrm{A} . \mathrm{B})$.
  - Nous concluons en appliquant (Deriv-Sub) à $\mathrm{T} \in \mathsf{Deriv}(\Pi x : \mathrm{A} . \mathrm{B})$.
- (Sub-E-1) Si

$$\frac{\Gamma \vdash \lambda x.\mathrm{M} : \{z : \mathrm{T} \mid \mathrm{U}\}}{\Gamma \vdash \lambda x.\mathrm{M} : \mathrm{T}} \text{ (Sub-E-1)}$$

montrons qu'il existe $\mathrm{A}, \mathrm{B}, s$ et Δ tels que $\Gamma; \Delta; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$, $\Gamma; \Delta \vdash \Pi x : \mathrm{A}.\mathrm{B} : s$ et $\mathrm{T} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$.

- Par hypothèses d'induction sur $\Gamma \vdash \lambda x.\mathrm{M} : \{z : \mathrm{T} \mid \mathrm{U}\}$, il existe $\mathrm{A}, \mathrm{B}, s$ et Δ tels que $\Gamma; \Delta; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$, $\Gamma; \Delta \vdash \Pi x : \mathrm{A}.\mathrm{B} : s$ et $\{z : \mathrm{T} \mid \mathrm{U}\} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$.
- D'après le point (2) du lemme 3.4.15 appliqué à $\{z : \mathrm{T} \mid \mathrm{U}\} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$ nous avons $\mathrm{T} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$, puisque $\Pi x : \mathrm{A}.\mathrm{B}$ n'est pas comparable à un type sous-ensemble.

- (Sub-E-2) Si

$$\frac{\Gamma \vdash \mathrm{P} : s_0 \qquad \Gamma \vdash c : \{z : \mathrm{T} \mid \mathrm{U}\} \qquad \Gamma; y : \mathrm{U}\,[z/c] \vdash \lambda x.\mathrm{M} : \mathrm{P} \qquad y \notin \mathsf{FV}(\lambda x.\mathrm{M})}{\Gamma \vdash \lambda x.\mathrm{M} : \mathrm{P}} \text{ (Sub-E-2)}$$

montrons qu'il existe $\mathrm{A}, \mathrm{B}, s$ et Δ tels que $\Gamma; \Delta; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$, $\Gamma; \Delta \vdash \Pi x : \mathrm{A}.\mathrm{B} : s$ et $\mathrm{P} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$.

- Par hypothèses d'induction sur $\Gamma; y : \mathrm{U}\,[z/c] \vdash \lambda x.\mathrm{M} : \mathrm{P}$, il existe $\mathrm{A}, \mathrm{B}, s$ et Δ_0 tels que $\Gamma; y : \mathrm{U}\,[z/c]; \Delta_0; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$, $\Gamma; y : \mathrm{U}\,[z/c]; \Delta_0 \vdash \Pi x : \mathrm{A}.\mathrm{B} : s$ et $\mathrm{P} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$.
- Nous concluons en posant $\Delta = y : \mathrm{U}\,[z/c]; \Delta_0$.

□

3.4.17. Remarque. Il n'est pas possible d'inverser similairement la variable. Ainsi l'énoncé suivant n'est pas valable.

$$\Gamma \vdash x : \mathrm{R} \Longrightarrow \exists \mathrm{T} \begin{cases} (x : \mathrm{T}) \in \Gamma \\ \mathrm{R} \in \mathsf{Deriv}(\mathrm{T}) \end{cases}$$

En effet, nous pouvons construire un contre-exemple. Posons $\Gamma = x : [\mathsf{Prop}] \to \mathsf{Prop}; y : \mathsf{Prop}$. Nous avons $\Gamma \vdash x : \mathsf{Prop}$ et $(x : [\mathsf{Prop}] \to \mathsf{Prop}) \in \Gamma$, mais $\mathsf{Prop} \notin \mathsf{Deriv}([\mathsf{Prop}] \to \mathsf{Prop})$ car cela contredirait le lemme 3.4.13.

Nous remarquons par ailleurs que la preuve précédente échouerait dans les cas (Inst) et (Sub-E-1) à cause de l'utilisation du lemme 3.4.15 car T peut a priori être un produit implicite ou un type sous-ensemble.

C'est également pour cette raison que les lemmes concernant les éliminateurs sont plus faibles : ainsi pour l'application, nous ne pouvons pas prouver que $\mathrm{R} \in \mathsf{Deriv}(\mathrm{U}\,[x/\mathrm{N}])$.

Le corollaire suivant sera utile pour les algorithmes d'inférence de type de la dernière partie et pour prouver le report de la η-réduction (cf. annexe C).

3.4.18. Corollaire. *Soient* Γ *un contexte,* s *une sorte,* x *une variable et* $\mathrm{M}, \mathrm{T}, \mathrm{U}$ *des termes de* ICC_Σ*. Les jugements* $\Gamma \vdash \lambda x.\mathrm{M} : s$ $\Gamma \vdash \lambda x.\mathrm{M} : \Sigma x : \mathrm{T}.\mathrm{U}$ $\Gamma \vdash \lambda x.\mathrm{M} : \exists x : \mathrm{T}.\mathrm{U}$ *ne sont pas dérivables.*

Démonstration. Nous ne traitons que le cas de la sorte, les autres points sont similaires.

Si $\Gamma \vdash \lambda x.\mathrm{M} : s$ est dérivable, alors d'après le lemme précédent, il existe $\mathrm{A}, \mathrm{B}, \mathrm{R}$ tels que $\mathrm{R} \preceq s$ et $\mathrm{R} \in \mathsf{Deriv}(\Pi x : \mathrm{A}.\mathrm{B})$. De $\mathrm{R} \preceq s$, nous déduisons (lemme 2.3.15 page 18) que D'après le lemme 3.4.13, $\Box x : \mathrm{T}.\mathrm{U} \preceq \mathrm{R}$ avec $\Box \in \{\Pi, \forall, \{\}\}$, d'où $\Box x : \mathrm{T}.\mathrm{U} \preceq s$. Cela contredit l'inversion de la cumulativité (lemme 2.3.15 page 18). □

Indécidabilité de l'inférence de type. Le problème de l'inférence de type dans un système de types peut s'énoncer ainsi : existe-t-il un algorithme qui prend en entrée un contexte Γ supposé bien formé et un terme M, et qui retourne un terme T tel que $\Gamma \vdash M : T$ est dérivable ? Si la réponse est positive, nous disons que le typage est *décidable*, sinon nous disons que le typage est *indécidable*.

Nous ne prouvons pas de résultat concernant la décidabilité du typage dans ICC_Σ, mais il nous semble extrêmement probable que le typage soit indécidable. En effet, nous avons des résultats d'indécidabilité du typage pour des versions à la Curry des systèmes suivants : Système F [**Wel99**], le $\lambda\Pi$-calcul [**Dow93**], Système F_ω [**Urz93, Urz97**] qui sont inclus dans ICC_Σ.[9] Un autre résultat pertinent, montré dans [**GHRDR93**], est que la décidabilité du typage dans un calcul des constructions à la Curry se ramène à celle dans F_ω.

9. Bien sûr, l'inclusion d'un langage dans un autre ne permet pas de conclure la décidabilité du typage de l'un de celle de l'autre.

Chapitre 2

Préservation du typage par réduction

La propriété de préservation du typage ou auto-réduction, exprime le fait que le réduit d'un terme bien typé est un terme bien typé et de même type. Cette propriété caractérise la sûreté d'un langage de programmation, car elle est équivalente au fait qu'un programme continue de respecter sa spécification lors de son exécution.

Tous les systèmes de types usuels respectent cette propriété. Ce chapitre examine ce qu'il en est pour ICC_Σ. Nous verrons tout d'abord que la η-réduction ne préserve probablement pas le typage (section 1). Puis, après une discussion rapide des difficultés à établir la préservation du typage pour la βι-réduction (section 2), nous introduisons deux outils syntaxiques permettant de surmonter ces difficultés : les substitutions multiples (section 3) et les jugements de structure (section 4). Nous pouvons alors prouver des lemmes d'inversion des types (section 5) et des termes (section 6), avant de finalement prouver la propriété de préservation du typage (section 7).

1. η-réduction et préservation du typage

Nous présentons tout d'abord un exemple de η-redex bien typé dont le réduit semble mal typé.

Considérons le contexte $\Gamma = \mathrm{T} : \mathsf{Prop}; x : [\mathrm{T}] \to \mathrm{T} \to \mathsf{Prop}$. La figure 7 montre que nous pouvons dériver $\Gamma \vdash \lambda y\,.\,x\,y : \mathrm{T} \to \mathsf{Prop}$. De plus, nous avons bien $y \notin \mathsf{FV}(x) = \{x\}$. Par contre nous ne pouvons a priori pas dériver $\Gamma \vdash x : \mathrm{T} \to \mathsf{Prop}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Gamma; y : \mathrm{T} \vdash}}{\Gamma; y : \mathrm{T} \vdash x : [\mathrm{T}] \to \mathrm{T} \to \mathsf{Prop}}\ (\textsc{Var}) \quad \dfrac{\dfrac{\vdots}{\Gamma; y : \mathrm{T} \vdash}}{\Gamma; y : \mathrm{T} \vdash y : \mathrm{T}}\ (\textsc{Var})}{\Gamma; y : \mathrm{T} \vdash x : \mathrm{T} \to \mathsf{Prop}}\ (\textsc{Inst}) \quad \dfrac{\dfrac{\vdots}{\Gamma; y : \mathrm{T} \vdash}}{\Gamma; y : \mathrm{T} \vdash y : \mathrm{T}}\ (\textsc{Var})}{\dfrac{\Gamma; y : \mathrm{T} \vdash x\,y : \mathsf{Prop}}{\Gamma \vdash \lambda y\,.\,x\,y : \mathrm{T} \to \mathsf{Prop}}\ (\textsc{Lam})}$$

FIGURE 7. Dérivation du η-redex

Pour prouver en toute rigueur que $\mathrm{T} \to \mathsf{Prop}$ n'est pas un des types possibles pour x dans Γ, il faut *a priori* établir un lemme d'inversion de la variable. Nous n'établissons pas ce lemme ici : la non-préservation du typage par η reste donc une conjecture, mais une conjecture raisonnable.

Il est intéressant de remarquer que dans les systèmes de types usuels, la η-réduction préserve le typage. Pour les PTS (à la Church) la preuve de préservation de typage pour η est faite en utilisant le lemme de *renforcement* : si $\Gamma; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{R}$ et $x \notin \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{R})$, alors $\Gamma \vdash \mathrm{M} : \mathrm{R}$ ([**Geu92**]).

Dans le système de Miquel, la préservation du typage par η est prouvée par induction en utilisant la règle de typage (Ext) pour le cas du redex :[1]

1. Comme le note Miquel, cela est aussi le cas dans le système F à la Curry de Joe B. Wells [**Wel96**]

$$\frac{\Gamma \vdash \lambda x.\,M\,x : T \qquad x \notin \mathsf{FV}(M)}{\Gamma \vdash M : T}\ (\textsc{Ext})$$

Notons également que le lemme de renforcement n'est pas prouvé dans le système de Miquel mais affirmé par une règle de typage :

$$\frac{\Gamma; x : T \vdash M : U \qquad x \notin \mathsf{FV}(M) \cup \mathsf{FV}(U)}{\Gamma \vdash M : U}\ (\textsc{Str})$$

D'après Miquel, la raison de la non-admissibilité du renforcement est la « présence du *produit implicite non-dépendant* »,[2] et, plus précisément le fait que $[T] \to U$ peut être habité alors que U ne l'est pas. Notons que le contre-exemple donné ci-dessus joue sur le même ressort.

2. Difficultés à établir la préservation du typage par βι-réduction

Dans les PTS usuels, la preuve de préservation de typage pour la β-réduction repose sur l'inversion de la conversion (qui repose sur la confluence) et des jugements de typage [**Geu92**].

Puisque nous avons déjà inversé la cumulativité, il nous suffit a priori d'établir les lemmes d'inversion pour pouvoir montrer la préservation du typage par βι-réduction.

Inverser un jugement $\Gamma \vdash M : R$, où M est un terme de structure connue, consiste à déduire des jugements typant les sous-termes de M ainsi qu'à relier R à un type canonique. Ainsi, dans un PTS $\Gamma \vdash M\,N : R$ implique l'existence de T, U tels que $\Gamma \vdash M : \Pi x : T.\,U$ et $\Gamma \vdash N : T$ avec $R \cong U\,[x/N]$.

Les lemmes d'inversion de jugement se déduisent en général assez facilement des règles de typage en raisonnant sur la dernière règle appliquée dans la dérivation de $\Gamma \vdash M : R$. Celle-ci est soit la règle caractéristique de la structure de M, soit la règle de cumulativité (ou de cumulativité selon le système considéré). Il suffit alors de remonter la dérivation tant que la dernière règle appliquée est la règle de cumulativité.

Comme nous l'avons déjà remarqué au chapitre précédent, établir des lemmes d'inversion est beaucoup plus complexe pour ICC_Σ, car la dernière règle de la dérivation d'un jugement $\Gamma \vdash M : R$ peut être choisie parmi non pas deux mais sept règles : la règle caractéristique de la structure de M, la règle de cumulativité, les deux règles d'introduction et d'élimination du produit implicite et les trois règles d'introduction et d'élimination du type sous-ensemble. R n'est donc plus comparable par cumulativité au type naturel de M ; il peut également être encoquillé dans des produits implicites ou des types sous-ensemble. Ainsi nous pouvons dériver $T : \mathsf{Type}_0; U : \mathsf{Type}_0 \vdash \mathsf{Prop} : \forall x : T.\,\{z : [U] \to \mathsf{Type}_0 \mid \forall y : T \to \mathsf{Prop}.\,y\,x\}$.

Miquel a également rencontré un problème analogue puisque, dans ICC, la dernière règle peut également être la règle d'introduction ou d'élimination du produit implicite. Pour tenir compte de la présence de produits implicites entourant le type naturel, un contexte de typage reprenant les quantifications des produits implicites éventuellement présents a été ajouté dans l'énoncé des lemmes d'inversion. Ainsi de $\Gamma \vdash \Pi z : T.\,U : R$ nous pouvons déduire qu'il existe un triplet de sortes $(s_1, s_2, s_3) \in$ **Rule** *et un contexte* Δ tels que

- $\Gamma; \Delta \vdash T : s_1$
- $\Gamma; \Delta; z : T \vdash U : s_2$
- $\forall \Delta.s_3 \cong R$

(avec si $\Delta = x_1 : T_1; \ldots; x_n : T_n$, $\forall \Delta.M = \forall (x_1 : T_1) \ldots \forall (x_n : T_n).M$.)

Malheureusement, nos tentatives pour généraliser la solution de Miquel dans ICC_Σ ont seulement abouti aux lemmes d'inversion présentés au chapitre précédent. La différence cruciale entre

2. cf. p : 65 dans [**Miq01**]

ces lemmes et les lemmes d'inversion de Miquel est qu'il est possible dans ICC de relier R à Δ. Nous ne sommes pas parvenus à le faire pour ICC_Σ. Les problèmes sont principalement venus de la règle (SUB-E-2), et en particulier de la prémisse $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$: la variable y interfère avec le contexte Δ dans $\Gamma; y : \mathrm{B}; \Delta$ sans que nous puissions échanger y et Δ.

Notre solution est plus complexe et nécéssite l'introduction de deux outils. Le premier, les *substitutions typées*, formalise la substitution de plusieurs variables par des termes. Le second est un nouveau jugement, appelé *jugement de structure*, qui relie, via une pile de termes, un type à un constructeur de type encoquillé en lui.

3. Substitutions

Le but de cette section est d'établir un outil permettant de réaliser la substitution simultanée de plusieurs variables dans un terme ou un jugement. Nous souhaitons de plus que cette substitution préserve le typage, c'est-à-dire que la substitution d'un jugement dérivable soit un jugement dérivable.

Nous présentons dans une première sous-section une substitution qui permet de substituer plusieurs variables simultanément, sans nécessairement préserver le typage. Puis, dans une deuxième sous-section, nous définissons des substitutions typées par un contexte et montrons qu'elles préservent le typage.

3.1. Substitutions non-typées.

Substitutions et termes.

3.1.1. DÉFINITION (Substitution). Une *substitution* σ est une fonction de $\mathbf{Var}$ vers Λ_{ICC} qui est égale à l'identité sauf pour un nombre fini de variables. L'ensemble des ces variables, appelé son *domaine*, est noté $\mathsf{Dom}(\sigma)$.

La substitution dont le domaine est vide est notée id.

Si $\mathsf{Dom}(\sigma) = \{x_1, \ldots, x_n\}$, les valeurs $\mathrm{M}_1, \ldots, \mathrm{M}_n$ prises par σ en $x_1, \ldots, x_n$, définissent parfaitement la substitution. Nous notons alors $\sigma = \{x_1 \mapsto \mathrm{M}_1; \ldots; x_n \mapsto \mathrm{M}_n\}$. L'ensemble des *variables libres* de σ, noté $\mathsf{FV}(\sigma)$, est défini par : $\mathsf{FV}(\sigma) = \mathsf{FV}(\mathrm{M}_1) \cup \cdots \cup \mathsf{FV}(\mathrm{M}_n)$.

3.1.2. DÉFINITION (Substitution d'un terme). Soit σ une substitution.

Nous étendons σ à tous les termes de ICC_Σ en définissant inductivement la fonction $\hat{\sigma} : \Lambda_{\text{ICC}} \to \Lambda_{\text{ICC}}$:

$$
\begin{array}{rcl}
\hat{\sigma}(x) & = & \sigma(x) \\
\hat{\sigma}(s) & = & s \\
\hat{\sigma}(\Pi x : \mathrm{T}.\mathrm{U}) & = & \Pi x : \hat{\sigma}(\mathrm{T}).\hat{\sigma}(\mathrm{U}) \\
\hat{\sigma}(\lambda x.\mathrm{M}) & = & \lambda x.\hat{\sigma}(\mathrm{M}) \\
\hat{\sigma}(\mathrm{M}\,\mathrm{N}) & = & \hat{\sigma}(\mathrm{M})\,\hat{\sigma}(\mathrm{N}) \\
\hat{\sigma}(\forall x : \mathrm{T}.\mathrm{U}) & = & \forall x : \hat{\sigma}(\mathrm{T}).\hat{\sigma}(\mathrm{U}) \\
\hat{\sigma}(\{x : \mathrm{A} \mid \mathrm{B}\}) & = & \{x : \hat{\sigma}(\mathrm{A}) \mid \hat{\sigma}(\mathrm{B})\} \\
\hat{\sigma}(\exists x : \mathrm{A}.\mathrm{B}) & = & \exists x : \hat{\sigma}(\mathrm{A}).\hat{\sigma}(\mathrm{B}) \\
\hat{\sigma}((\diamond, b)) & = & (\diamond, \hat{\sigma}(b)) \\
\hat{\sigma}(\mathsf{Elim}_\exists(y.f, c)) & = & \mathsf{Elim}_\exists(y.\hat{\sigma}(f), \hat{\sigma}(c)) \\
\hat{\sigma}(\Sigma x : \mathrm{A}.\mathrm{B}) & = & \Sigma x : \hat{\sigma}(\mathrm{A}).\hat{\sigma}(\mathrm{B}) \\
\hat{\sigma}((\mathrm{M}, \mathrm{N})) & = & (\hat{\sigma}(\mathrm{M}), \hat{\sigma}(\mathrm{N})) \\
\hat{\sigma}(\mathsf{Elim}_\Sigma(xy.f, c)) & = & \mathsf{Elim}_\Sigma(xy.\hat{\sigma}(f), \hat{\sigma}(c))
\end{array}
$$

3.1.3. REMARQUES. (1) Dans la suite nous assimilerons σ et $\hat{\sigma}$ en notant $\sigma(M)$ le terme $\hat{\sigma}(M)$.

(2) Nous substituons les variables simultanément et non pas séquentiellement. Ainsi, si $\sigma = \{x \mapsto y; y \mapsto x\}$, alors $\sigma(x\,y) = y\,x$, tandis que $(x\,y)[x/y][y/x] = x\,x$ et $(x\,y)[y/x][x/y] = y\,y$.

(3) Si $\sigma = \{x \mapsto N\}$, alors $\sigma(M) = M[x/N]$.

(4) Pour tout terme M, $\mathsf{id}(M) = M$.

3.1.4. CONVENTION. Dans le prolongement des conventions précédentes, afin d'éviter à la fois la capture de variables libres et la substitution de variables liées, nous renommons par α-conversion les variables liées d'un terme M auquel est appliqué une substition σ afin d'avoir $\mathsf{BV}(M) \cap (\mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma)) = \emptyset$.

3.1.5. Lemme. *Soient σ_1, σ_2 des substitutions et M un terme. Si, pour tout $x \in FV(M)$, $\sigma_1(x) = \sigma_2(x)$, alors $\sigma_1(M) = \sigma_2(M)$.*

DÉMONSTRATION. Immédiat par induction sur la structure de M. □

3.1.6. Lemme (Substitution et variables libres)**.** *Si M est un terme, on a*

$$\begin{aligned} FV(\sigma(M)) &= \bigcup_{x \in FV(M)} FV(\sigma(x)) \\ &= (FV(M) \setminus \mathsf{Dom}(\sigma)) \cup \bigcup_{x \in \mathsf{Dom}(\sigma) \cap FV(M)} FV(\sigma(x)) \\ &\subset FV(M) \cup FV(\sigma) \end{aligned}$$

DÉMONSTRATION. La première égalité se montre par induction sur la structure de M. La deuxième égalité est claire car $\sigma(x) = x$ si $x \notin \mathsf{Dom}(\sigma)$. Enfin, l'inclusion est immédiate par définition de $\mathsf{FV}(\sigma)$. □

3.1.7. Lemme (Substitution et substitution de variables)**.** *Si $x \notin \mathsf{Dom}(\sigma) \cup FV(\sigma)$ alors $\sigma(M[x/N]) = \sigma(M)[x/\sigma(N)]$.*

DÉMONSTRATION. Par induction sur la structure de M. En appliquant la convention de Barendregt et la convention 3.1.4, tous les cas sont évidents sauf celui de la variable.

Si $M = x \notin \mathsf{Dom}(\sigma)$, alors $\sigma(M) = x$ d'où $\sigma(M)[x/\sigma(N)] = \sigma(N) = \sigma(M[x/N])$.

Si $M = y \neq x$, alors $\sigma(M[x/N]) = \sigma(y)$ et $\sigma(M)[x/\sigma(N)] = \sigma(y)[x/\sigma(N)]$. Or $x \notin \mathsf{FV}(\sigma(y))$ car $\mathsf{FV}(\sigma(y)) \subset \mathsf{FV}(\sigma)$ par définition. Donc $\sigma(M)[x/\sigma(N)] = \sigma(y) = \sigma(M[x/N])$. □

3.1.8. REMARQUE. Le résultat est faux si $x \in \mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma)$. Soient x, y, z des variables deux à deux distinctes.

- Considérons $\sigma = \{x \mapsto y\}$, $M = x$ et $N = z$. Alors $x \in \mathsf{Dom}(\sigma) \setminus \mathsf{FV}(\sigma)$, $\sigma(M[x/N]) = z$ et $\sigma(M)[x/\sigma(N)] = y$.
- Considérons $\sigma = \{y \mapsto x\}$, $M = y$ et $N = z$. Alors $x \in \mathsf{FV}(\sigma) \setminus \mathsf{Dom}(\sigma)$, $\sigma(M[x/N]) = x$ et $\sigma(M)[x/\sigma(N)] = z$.

3.1.9. Corollaire (Substitution, réduction, conversion et cumulativité)**.** *Soient T, T' des termes et σ une substitution.*

$$\begin{array}{lrcl} (1) & T \to_{\beta\iota\eta} T' & \Rightarrow & \sigma(T) \to_{\beta\iota\eta} \sigma(T') \\ (2) & T \twoheadrightarrow_{\beta\iota\eta} T' & \Rightarrow & \sigma(T) \to_{\beta\iota\eta} \sigma(T') \\ (3) & T \cong T' & \Rightarrow & \sigma(T) \cong \sigma(T') \\ (4) & T \preceq T' & \Rightarrow & \sigma(T) \preceq \sigma(T') \end{array}$$

DÉMONSTRATION. (2) se déduit de (1) par une récurrence immédiate sur le nombre de réductions. (3) est une conséquence de Church-Rosser et (2). (4) se montre immédiatement par induction sur la dérivation $T \preceq T'$. Nous appliquons (3) pour le cas de la règle (CUM-REFL). Les autres règles se traitent immédiatement car toute substitution conserve les sortes et la structure d'un terme.

Pour montrer (1) nous devons distinguer deux cas selon que la réduction a lieu à la racine ou dans un sous-terme.

- Réduction à la racine : Il suffit de montrer les résultats suivants

$$\begin{aligned}
\sigma((\lambda x.M)\,N) &\rhd_\beta \sigma(M[x/N]) \\
\sigma(\mathsf{Elim}_\exists(y.f,(\diamond,b))) &\rhd_\iota \sigma(f[y/b]) \\
\sigma(\mathsf{Elim}_\Sigma(xy.f,(a,b))) &\rhd_\iota \sigma(f[x/a][y/b]) \\
\sigma(\lambda x.M\,x) &\rhd_\eta \sigma(M) \text{ si } x \notin \mathsf{FV}(M).
\end{aligned}$$

  - Les trois premiers cas se démontrent facilement en utilisant la définition de la substitution et le lemme 3.1.7. [3]
  - Montrons le cas de la η-réduction. Nous avons $x \notin \mathsf{FV}(M)$. De plus, d'après la convention 3.1.4, $x \notin \mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma)$: nous avons donc $\sigma(\lambda x.M\,x) = \lambda x.\sigma(M)x$. Il suffit donc de montrer que $x \notin \mathsf{FV}(\sigma(M))$. Cela est immédiat car, d'après le lemme 3.1.6, $\mathsf{FV}(\sigma(M)) \subset \mathsf{FV}(M) \cup \mathsf{FV}(\sigma)$.
- Réduction dans un sous-terme : par induction sur la structure de T. La preuve est immédiate puisque la définition 3.1.2 suit la structure du terme.

□

Substitutions et contextes. Nous prolongeons également la notion de substitution aux contextes.

3.1.10. DÉFINITION (Substitution d'un contexte). Soit σ une fonction de substitution. Nous notons $\sigma(\Gamma)$ le contexte défini par induction sur Γ :

$$\begin{aligned}
\sigma(\bullet) &= \bullet \\
\sigma(\Gamma; x:T) &= \begin{cases} \sigma(\Gamma) & \text{si } x \in \mathsf{Dom}(\sigma) \\ \sigma(\Gamma); x:\sigma(T) & \text{sinon.} \end{cases}
\end{aligned}$$

Appliquer une substitution à un contexte consiste donc à effacer les déclarations de variables appartenant au domaine et à appliquer la substitution aux types des variables qui n'apparaissent pas dans le domaine de la substitution.

3.1.11. REMARQUES.

(1) Nous avons donc $\mathsf{DV}(\sigma(\Gamma)) = \mathsf{DV}(\Gamma) \setminus \mathsf{Dom}(\sigma) \subset \mathsf{DV}(\Gamma)$.

(2) Si σ est une substitution et si $\Gamma \vdash M : T$ est un jugement dérivable, nous n'avons pas en général $\sigma(\Gamma) \vdash \sigma(M) : \sigma(T)$. Considérons par exemple la substitution $\sigma = \{x \mapsto \mathsf{Prop}\}$ et le jugement dérivable $T : \mathsf{Prop}; x : T \vdash x : T$. Le jugement substitué est $T : \mathsf{Prop} \vdash \mathsf{Prop} : T$, qui n'est pas dérivable a priori. [4]

(3) La substitution vide ne modifie pas le contexte : pour tout contexte Γ, $\mathsf{id}(\Gamma) = \Gamma$.

3.1.12. Lemme. *Soit* Γ *un contexte bien formé et* σ_1, σ_2 *des substitutions telles que pour tout* $x \in DV(\Gamma)$, $\sigma_1(x) = \sigma_2(x)$. *Alors* $\sigma_1(\Gamma) = \sigma_2(\Gamma)$.

3. Le respect de la convention 3.1.4 implique que la condition du lemme est vérifiée.

4. Pour une preuve rigoureuse, il faudrait un lemme d'inversion pour les sortes. Nous ne l'établirons pas dans ce manuscrit.

DÉMONSTRATION. Par induction sur la structure de Γ. Le cas de base est évident. Supposons que $\Gamma; z:\mathrm{T} \vdash$ et que pour tout $x \in \mathsf{DV}(\Gamma; z:\mathrm{T})$, $\sigma_1(x) = \sigma_2(x)$. Montrons que $\sigma_1(\Gamma; z:\mathrm{T}) = \sigma_2(\Gamma; z:\mathrm{T})$. Par hypothèse d'induction (puisque $\Gamma \vdash$), nous avons $\sigma_1(\Gamma) = \sigma_2(\Gamma)$. Par ailleurs, $\sigma_1(z) = \sigma_2(z)$, donc $z \in \mathsf{Dom}(\sigma_1)$ si et seulement si $z \in \mathsf{Dom}(\sigma_2)$. D'après la définition 3.1.10, il suffit de montrer que $\sigma_1(\mathrm{T}) = \sigma_2(\mathrm{T})$. Ceci est une conséquence du lemme 3.1.5 puisque $\mathsf{FV}(\mathrm{T}) \subset \mathsf{DV}(\Gamma)$ et que σ_1 et σ_2 sont égales sur $\mathsf{DV}(\Gamma)$. □

3.1.13. REMARQUE. Le résultat n'est pas valable si le contexte n'est pas bien formé. $\Gamma = x:\mathrm{T}$, $\sigma_1 = \{\mathrm{T} \to \mathsf{Prop}\}$, $\sigma_2 = \{\mathrm{T} \to \mathsf{Type}_0\}$ est un contre-exemple.

3.2. Substitutions typées. Dans cette sous-section, nous définissons une classe restreinte de substitutions, les *substitutions typées.*[5]

Ces substitutions sont associées à un contexte de typage et vérifient certaines propriétés, notamment la préservation du typage.

3.2.1. DÉFINITION (Substitution typée). Soient Γ un contexte et σ une substitution. Nous considérons le *jugement de typage de substitution*, noté $\_ \vdash \sigma \div \Gamma$, qui indique que *la substitution* σ *est bien typée dans ICC*$_\Sigma$ *sous le contexte* Γ.

Les jugements dérivables sont définis à partir des trois règles d'inférence suivantes :

$$\frac{}{\_ \vdash \mathsf{id} \div \bullet}\ (\textsc{Subst-Id})$$

$$\frac{\_ \vdash \sigma \div \Gamma \qquad \Gamma; x:\mathrm{T} \vdash}{\_ \vdash \sigma \div \Gamma; x:\mathrm{T}}\ (\textsc{Subst-Gen})$$

$$\frac{\_ \vdash \sigma \div \Gamma \qquad \sigma(\Gamma) \vdash \mathrm{N} : \sigma(\mathrm{T}) \qquad \Gamma; x:\mathrm{T} \vdash}{\_ \vdash \sigma \cup \{x \mapsto \mathrm{N}\} \div \Gamma; x:\mathrm{T}}\ (\textsc{Subst-Inst})$$

Nous montrons facilement les résultats suivants :

3.2.2. Lemme (Propriétés immédiates)**.** *Si* $\_ \vdash \sigma \div \Gamma$*, alors*

(1) $\Gamma \vdash$

(2) $\mathsf{Dom}(\sigma) \cup \mathit{FV}(\sigma) \subset \mathit{DV}(\Gamma)$.

(3) (Affaiblissement) si $\_ \vdash \sigma \div \Gamma$ *et si* $\Gamma; \Delta \vdash$ *alors* $\_ \vdash \sigma \div \Gamma; \Delta$.

DÉMONSTRATION.

(1) se montre immédiatement par induction sur la structure de la dérivation de $\_ \vdash \sigma \div \Gamma$.

(2) se montre également par induction sur la structure de la dérivation de $\_ \vdash \sigma \div \Gamma$.

- (Subst-Id) est immédiat.
- (Subst-Gen) se montre facilement à partir des hypothèses d'induction.
- (Subst-Inst) est un peu moins immédiat. Si
$$\frac{\_ \vdash \sigma \div \Gamma \qquad \sigma(\Gamma) \vdash \mathrm{N} : \sigma(\mathrm{T}) \qquad \Gamma; x:\mathrm{T} \vdash}{\_ \vdash \sigma \cup \{x \mapsto \mathrm{N}\} \div \Gamma; x:\mathrm{T}}\ (\textsc{Subst-Inst})$$
Montrons que $\mathsf{Dom}(\sigma \cup \{x \mapsto \mathrm{N}\}) \cup \mathsf{FV}(\sigma \cup \{x \mapsto \mathrm{N}\}) \subset \mathsf{DV}(\Gamma; x:\mathrm{T})$. Par hypothèse d'induction, nous avons $\mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma) \subset \mathsf{DV}(\Gamma)$. Il suffit donc de montrer que $\mathsf{FV}(\mathrm{N}) \subset \mathsf{DV}(\Gamma)$. En appliquant le le point (2) du lemme 3.4.3 page 25 à la deuxième prémisse, nous avons $\mathsf{FV}(\mathrm{N}) \subset \mathsf{DV}(\sigma(\Gamma))$. Nous concluons alors grâce au point (1) de la remarque 3.1.11.

(3) se montre par une récurrence immédiate sur la taille de Δ en appliquant la règle (Subst-Gen) pour le cas de récurrence.

□

5. Notre définition est différente des substitutions typées usuelles (cf. par exemple [**Hof97**] p.20). Nous conjecturons que notre présentation aurait également été possible avec la définition habituelle.

Représentation ordonnée d'une substitution typée. Nous avons vu que les variables de $\mathsf{Dom}(\sigma)$ sont des variables du contexte de typage Γ. Nous pouvons donc les ordonner suivant l'ordre des variables de Γ.

3.2.3. DÉFINITION (Représentation d'une substitution ordonnée par son contexte). Si $\_ \vdash \sigma \div \Gamma$ avec $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ et $\sigma \neq \emptyset$, alors sa *représentation ordonnée par son contexte de typage* est

$$\sigma = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_k} \mapsto \mathrm{N}_k\}$$

où

- $k \leq n$ est le cardinal de $\mathsf{Dom}(\sigma)$,
- $i_1, \ldots, i_k$ sont des entiers tels que $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ et $\mathsf{Dom}(\sigma) = \{x_{i_1}; \ldots; x_{i_k}\}$,
- $\mathrm{N}_j = \sigma(x_{i_j})$ pour tout j tel que $1 \leq j \leq k$.

Remarquons que cette représentation est unique.

Nous pouvons maintenant exprimer le résultat suivant d'inversion de $\_ \vdash \sigma \div \Gamma$ qui caractérise les sous-substitutions d'une substitution typée.

3.2.4. Lemme (Inversion de la dérivation d'une substitution bien typée)**.** *Si* $\_ \vdash \sigma \div \Gamma$ *avec* $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ *et* $\sigma = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_k} \mapsto \mathrm{N}_k\}$ *la représentation ordonnée de* σ.

Pour j *compris entre 0 et* $k-1$, *nous notons* Γ_j *le contexte contenant toutes les déclarations précédant* $x_{i_{j+1}} : \mathrm{T}_{i_{j+1}}$, *soit*

$$\Gamma_j = x_1 : \mathrm{T}_1; \ldots; x_{i_j} : \mathrm{T}_{i_j}; x_{i_j+1} : \mathrm{T}_{i_j+1}; \ldots; x_{i_{j+1}-1} : \mathrm{T}_{i_{j+1}-1}$$

(avec $\Gamma_0 = \bullet$ *si* $i_1 = 1$*).*

Par ailleurs, nous notons $\sigma_0 = \mathsf{id}$ *et pour tout* j *compris entre 1 et* k,

$$\sigma_j = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_j} \mapsto \mathrm{N}_j\}.$$

Nous avons alors, pour tout $1 \leq j \leq k$ *:*

$$\_ \vdash \sigma_j \div \Gamma_{j-1}; x_{i_j} : \mathrm{T}_{i_j} \qquad \sigma_{j-1}(\Gamma_{j-1}) \vdash \mathrm{N}_j : \sigma_{j-1}(\mathrm{T}_{i_j})$$

DÉMONSTRATION. Par induction sur la structure de la dérivation de $\_ \vdash \sigma \div \Gamma$.

- (SUBST-ID) est sans objet.
- (SUBST-GEN) Si

$$\frac{\_ \vdash \sigma \div \Gamma' \qquad \Gamma'; z : \mathrm{U} \vdash}{\_ \vdash \sigma \div \Gamma'; z : \mathrm{U}} \text{ (SUBST-GEN)}$$

  avec $\Gamma = \Gamma'; z : \mathrm{U}$. En appliquant par induction le lemme à la prémisse $\_ \vdash \sigma \div \Gamma'$, nous avons, pour tout $1 \leq j \leq k$:

$$\_ \vdash \sigma_j \div \Gamma'_{j-1}; x_{i_j} : \mathrm{T}_{i_j} \quad \sigma_{j-1}(\Gamma'_{j-1}) \vdash \mathrm{N}_j : \sigma_{j-1}(\mathrm{T}_{i_j})$$

  Nous concluons en remarquant que $\Gamma'_j = \Gamma_j$ pour tout $0 \leq j \leq k-1$.

- (SUBST-INST) Si

$$\frac{\_ \vdash \sigma_{k-1} \div \Gamma' \qquad \sigma_{k-1}(\Gamma') \vdash \mathrm{N}_k : \sigma_{k-1}(\mathrm{T}_{i_k}) \qquad \Gamma'; x_{i_k} : \mathrm{T}_{i_k} \vdash}{\_ \vdash \sigma_{k-1} \cup \{x_{i_k} \mapsto \mathrm{N}_k\} \div \Gamma'; x_{i_k} : \mathrm{T}_{i_k}} \text{ (SUBST-INST)}$$

  avec $\Gamma = \Gamma' : x_{i_k} : \mathrm{T}_{i_k}$ et $\sigma = \sigma_{k-1} \cup \{x_{i_k} \mapsto \mathrm{N}_k\}$.
  Montrons pour tout $1 \leq j \leq k$ que $\_ \vdash \sigma_j \div \Gamma_{j-1}; x_{i_j} : \mathrm{T}_{i_j}$ et $\sigma_{j-1}(\Gamma_{j-1}) \vdash \mathrm{N}_j : \sigma_{j-1}(\mathrm{T}_{i_j})$.
  - Si $0 \leq j \leq k-1$, nous avons $\Gamma_j = \Gamma'_j$, donc en appliquant par induction le lemme à la prémisse $\_ \vdash \sigma_{k-1} \div \Gamma'$, nous prouvons les résultats souhaités pour $1 \leq j \leq k-1$.

- Si $j = k$, $\_ \vdash \sigma_k \div \Gamma_{k-1}; x_{i_k} : \mathrm{T}_{i_k}$ équivaut à $\_ \vdash \sigma \div \Gamma$ qui est une hypothèse de départ, et $\sigma_{k-1}(\Gamma_{k-1}) \vdash \mathrm{N}_k : \sigma_{k-1}(\mathrm{T}_{i_k})$ est la deuxième prémisse puisque $\Gamma' = \Gamma_{k-1}$.

□

Substitutions séquentielles. Une propriété intéressante des substitutions typées est que, contrairement aux substitutions non-typées, nous pouvons contrôler finement la manière dont sont substituées les variables. Nous avons remarqué plus haut (point (2) de la remarque 3.1.3) que les substitutions de variables pour une substitution quelconque se font de manière simultanée. Nous montrons que si la substitution est typée par un contexte, cette opération peut être vue comme une suite de substitutions d'une variable (au sens de la définition 1.2.11 page 10), ordonnée par l'ordre des variables dans le contexte de typage. Ainsi les substitutions typées peuvent être comprises comme des généralisations à plusieurs variables de la substitution $\mathrm{M}[x/\mathrm{N}]$ d'une variable dans un terme.

3.2.5. Lemme. *Avec les notations du lemme 3.2.4, nous avons, pour tout j tel que $0 \leq j \leq k-1$*

(1) pour tout terme M

$$\sigma_{j+1}(\mathrm{M}) = \sigma_j(\mathrm{M})\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$$

(2) pour tout contexte Δ

$$\sigma_{j+1}(\Delta) = \sigma_j(\Delta)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$$

DÉMONSTRATION.

(1) Par induction sur M. Les cas inductifs sont évidents. Seul le cas de la variable n'est pas immédiat. Posons $\mathrm{M} = x$. Nous avons $\mathsf{Dom}(\sigma_j) = \{x_{i_1}; \ldots; x_{i_j}\}$ et $\mathsf{Dom}(\sigma_{j+1}) = \mathsf{Dom}(\sigma_j) \cup \{x_{i_{j+1}}\}$. Trois cas se présentent :
   - Si $x \notin \mathsf{Dom}(\sigma_{j+1})$, alors $x \notin \mathsf{Dom}(\sigma_j)$, d'où $\sigma_{j+1}(x) = \sigma_j(x) = x$. Comme, de plus, $x \neq x_{i_{j+1}}$, nous avons $x\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] = x$.
   - Si $x = x_{i_{j+1}}$, alors $\sigma_{j+1}(x) = \mathrm{N}_{j+1}$ et $\sigma_j(x) = x$, d'où $\sigma_j(x)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] = \mathrm{N}_{j+1}$.
   - Enfin si $x \in \mathsf{Dom}(\sigma_j)$, il existe $j_0 \leq j$ tel que $x = x_{i_{j_0}}$, d'où $\sigma_{j+1}(x) = \sigma_j(x) = \mathrm{N}_{j_0}$. Montrons que $x_{i_{j+1}} \notin \mathsf{FV}(\mathrm{N}_{j_0})$, nous en déduirons que $\mathrm{N}_{j_0}\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] = \mathrm{N}_{j_0}$. D'après le lemme 3.2.4, nous avons $\sigma_{j_0-1}(\Gamma_{j_0-1}) \vdash \mathrm{N}_{j_0} : \sigma_{j_0-1}(\mathrm{T}_{i_{j_0}})$, d'où $\mathsf{FV}(\mathrm{N}_{j_0}) \subset \mathsf{DV}(\sigma_{j_0-1}(\Gamma_{j_0-1})) \subset \mathsf{DV}(\Gamma_{j_0-1})$. Nous concluons alors immédiatement puisque $x_{i_{j+1}} \notin \mathsf{DV}(\Gamma_{j_0-1})$.

(2) Par induction sur Δ. Le cas $\Delta = \bullet$ est évident. Si $\Delta = \Delta'; z : \mathrm{U}$, nous avons par hypothèse d'induction $\sigma_{j+1}(\Delta') = \sigma_j(\Delta')\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$. Nous distinguons alors comme dans (1) les trois cas suivants :
   - si $z \notin \mathsf{Dom}(\sigma_{j+1})$ alors

$$\begin{aligned}\sigma_{j+1}(\Delta) &= \sigma_{j+1}(\Delta'; z : \mathrm{U}) \\ &= \sigma_{j+1}(\Delta'); z : \sigma_{j+1}(\mathrm{U})\end{aligned}$$

     Par hypothèse d'induction et d'après le point (1), nous déduisons

$$\begin{aligned}\sigma_{j+1}(\Delta) &= \sigma_j(\Delta')\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]; z : \sigma_j(\mathrm{U})\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] \\ &= \sigma_j(\Delta'; z : \mathrm{U})\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] \\ &= \sigma_j(\Delta)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}].\end{aligned}$$

   - si $z = x_{i_{j+1}} \notin \mathsf{Dom}(\sigma_j)$, alors $\Delta = \Delta'; x_{i_{j+1}} : \mathrm{U}$. Montrons que $\sigma_{j+1}(\Delta) = \sigma_j(\Delta)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$.
     - Par définition de la substitution d'un contexte (définition 3.1.10), et par hypothèse d'induction sur Δ', nous avons $\sigma_{j+1}(\Delta) = \sigma_j(\Delta')\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$.
     - Montrons alors que $\sigma_j(\Delta)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] = \sigma_j(\Delta')\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$.
       — Puisque $x_{i_{j+1}} \notin \mathsf{Dom}(\sigma_j)$, nous avons $\sigma_j(\Delta) = \sigma_j(\Delta'); x_{i_{j+1}} \sigma_j(\mathrm{U})$ (définition 3.1.10).

— Nous en déduisons, $\sigma_j(\Delta)[x_{i_{j+1}}/\mathrm{N}_{j+1}] = \sigma_j(\Delta')[x_{i_{j+1}}/\mathrm{N}_{j+1}]$, d'après la définition 3.1.7 page 19.

- si $z \in \mathsf{Dom}(\sigma_j) \subset \mathsf{Dom}(\sigma_j)$ alors d'après la définition 3.1.10, $\sigma_{j+1}(\Delta) = \sigma_{j+1}(\Delta')$ et $\sigma_j(\Delta) = \sigma_j(\Delta')$. Nous concluons alors par hypothèse d'induction.

□

Une récurrence immédiate sur k prouve le corollaire suivant.

3.2.6. Corollaire (Substitutions séquentielles)**.** *Si* $\_ \vdash \sigma \div \Gamma$ *avec* $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ *et* $\sigma = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_k} \mapsto \mathrm{N}_k\}$ *la représentation ordonnée de* σ.

Alors pour tout entier $j < k$*, tout terme* M *et tout contexte* Δ*, nous avons*

$$\sigma(\mathrm{M}) = \sigma_j(\mathrm{M})[x_{i_{j+1}}/\mathrm{N}_{j+1}]\ldots[x_{i_k}/\mathrm{N}_k]$$
$$\sigma(\Delta) = \sigma_j(\Delta)[x_{i_{j+1}}/\mathrm{N}_{j+1}]\ldots[x_{i_k}/\mathrm{N}_k]$$

En particulier, pour $j = 0$ *:*

$$\sigma(\mathrm{M}) = \mathrm{M}[x_{i_1}/\mathrm{N}_1]\ldots[x_{i_k}/\mathrm{N}_k]$$
$$\sigma(\Delta) = \Delta[x_{i_1}/\mathrm{N}_1]\ldots[x_{i_k}/\mathrm{N}_k]$$

et pour $j = k-1$ *:*

$$\sigma(\mathrm{M}) = \sigma_{k-1}(\mathrm{M})[x_{i_k}/\mathrm{N}_k]$$
$$\sigma(\Delta) = \sigma_{k-1}(\Delta)[x_{i_k}/\mathrm{N}_k]$$

.

Préservation du typage. Nous pouvons maintenant montrer la propriété qui a motivé l'introduction des substitutions typées.

3.2.7. Proposition (Préservation du typage par substitution)**.** *Si* $\_ \vdash \sigma \div \Gamma$ *et si* $\Gamma \vdash \mathrm{M} : \mathrm{R}$*, alors* $\sigma(\Gamma) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{R})$.

DÉMONSTRATION. Par récurrence sur le cardinal k de $\mathsf{Dom}(\sigma)$.

- Si $k = 0$ alors σ est l'identité et le résultat est immédiat.
- Si $k \geq 1$:
  - En utilisant les notations de la définition 3.2.3 :
    - — il existe un contexte Δ tel que
    $$\Gamma = \Gamma_{k-1}; x_{i_k} : \mathrm{T}_{i_k}; \Delta$$
    - — d'après le lemme 3.2.4 appliqué avec $j = k-1$, $\_ \vdash \sigma_{k-1} \div \Gamma_{k-2}; x_{i_{k-1}} : \mathrm{T}_{i_{k-1}}$, d'où par affaiblissement
    $$\_ \vdash \sigma_{k-1} \div \Gamma$$
    - — d'après le lemme 3.2.4 appliqué avec $j = k$,
    $$\sigma_{k-1}(\Gamma_{k-1}) \vdash \mathrm{N}_k : \sigma_{k-1}(\mathrm{T}_{i_k})$$
  - Montrons que $\sigma(\Gamma) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{R})$. Puisque $\sigma = \sigma_k$ et d'après le corollaire 3.2.6, il suffit de montrer que
  $$\sigma_{k-1}(\Gamma)[x_{i_k}/\mathrm{N}_k] \vdash \sigma_{k-1}(\mathrm{M})[x_{i_k}/\mathrm{N}_k] : \sigma_{k-1}(\mathrm{R})[x_{i_k}/\mathrm{N}_k]$$
    - — Montrons que $\sigma_{k-1}(\Gamma_{k-1}); x_{i_k} : \sigma_{k-1}(\mathrm{T}_{i_k}); \sigma_{k-1}(\Delta) \vdash \sigma_{k-1}(\mathrm{M}) : \sigma_{k-1}(\mathrm{R})$.
    Par hypothèse de récurrence, nous avons $\sigma_{k-1}(\Gamma) \vdash \sigma_{k-1}(\mathrm{M}) : \sigma_{k-1}(\mathrm{R})$. Nous concluons en remarquant que $\Gamma = \Gamma_{k-1}; x_{i_k} : \mathrm{T}_{i_k}; \Delta$.

— Par substitutivité (lemme 3.4.8 page 26), puisque $\sigma_{k-1}(\Gamma_{k-1}) \vdash N_k : \sigma_{k-1}(T_{i_k})$, nous avons

$$\sigma_{k-1}(\Gamma_{k-1}); \sigma_{k-1}(\Delta)\,[x_{i_k}/N_k] \vdash \sigma_{k-1}(M)\,[x_{i_k}/N_k] : \sigma_{k-1}(R)\,[x_{i_k}/N_k]$$

— Il suffit donc pour conclure de montrer que

$$\sigma_{k-1}(\Gamma)\,[x_{i_k}/N_k] = \sigma_{k_1}(\Gamma_{k-1});\ x_{i_k} : \sigma_{k-1}(T_{i_k});\ \sigma_{k-1}(\Delta)\,[x_{i_k}/N_k].$$

— Nous avons, puisque $\Gamma = \Gamma_{k-1}; x_{i_k} : T_{i_k}; \Delta$:

$$\sigma_{k-1}(\Gamma)\,[x_{i_k}/N_k] = \sigma_{k-1}(\Gamma_{k-1})\,[x_{i_k}/N_k];\ x_{i_k} : \sigma_{k-1}(T_{i_k})\,[x_{i_k}/N_k];\ \sigma_{k-1}(\Delta)\,[x_{i_k}/N_k]$$

— Montrons que $\sigma_{k-1}(\Gamma_{k-1})\,[x_{i_k}/N_k] = \sigma_{k-1}(\Gamma_{k-1})$. D'après le lemme 3.1.8 page 19, il suffit de montrer que $x_{i_k} \notin \mathsf{DV}(\sigma_{k-1}(\Gamma_{k-1})) \cup \mathsf{FV}(\sigma_{k-1}(\Gamma_{k-1}))$. Puisque $\sigma_{k-1}(\Gamma_{k-1}) \vdash$, nous avons $\mathsf{FV}(\sigma_{k-1}(\Gamma_{k-1})) \subset \mathsf{DV}(\sigma_{k-1}(\Gamma_{k-1}))$. Il suffit donc de montrer que $x_{i_k} \notin \mathsf{DV}(\sigma_{k-1}(\Gamma_{k-1}))$, ce qui est immédiat car $x_{i_k} \notin \mathsf{DV}(\Gamma_{k-1}) \supset \mathsf{DV}(\sigma_{k-1}(\Gamma_{k-1}))$.

— Montrons que $\sigma_{k-1}(T_{i_k})\,[x_{i_k}/N_k] = \sigma_{k-1}(T_{i_k})$. D'après le lemme 1.2.12 page 10, il suffit de montrer que $x_{i_k} \notin \mathsf{FV}(\sigma_{k-1}(T_{i_k}))$. De $\sigma_{k-1}(\Gamma_{k-1}) \vdash N_k : \sigma_{k-1}(T_{i_k})$, nous déduisons $\mathsf{FV}(\sigma_{k-1}(T_{i_k})) \subset \mathsf{DV}(\sigma_{k-1}(\Gamma_{k-1}))$. Nous avons montré ci-dessus que $x \notin \mathsf{DV}(\sigma_{k-1}(\Gamma_{k-1}))$, ce qui nous permet de conclure.

□

3.2.8. Corollaire (Préservation de la bonne formation des contextes par substitution)**.** *Si* $\_ \vdash \sigma \div \Gamma$ *alors* $\sigma(\Gamma) \vdash$.

DÉMONSTRATION. Par induction sur la dérivation de $\_ \vdash \sigma \div \Gamma$.

- (SUBST-ID) est immédiat.
- Si

$$\frac{\_ \vdash \sigma \div \Gamma' \qquad \Gamma'; x : T \vdash}{\_ \vdash \sigma \div \Gamma'; x : T}\ (\text{SUBST-GEN})$$

avec $\Gamma = \Gamma'; x : T$, nous avons $x \notin \mathsf{Dom}(\sigma)$ et devons donc montrer que $\sigma(\Gamma'); x : \sigma(T) \vdash$. Nous avons $\_ \vdash \sigma \div \Gamma'$ (prémisse) et $\Gamma' \vdash T : s$ (inversion de la prémisse $\Gamma'; x : T \vdash$). D'après la proposition 3.2.7, nous avons $\sigma(\Gamma') \vdash \sigma(T) : s$. Nous concluons en appliquant (WF-S).

- Si

$$\frac{\_ \vdash \sigma' \div \Gamma' \qquad \sigma'(\Gamma') \vdash N : \sigma'(T) \qquad \Gamma'; x : T \vdash}{\_ \vdash \sigma' \cup \{x \mapsto N\} \div \Gamma'; x : T}\ (\text{SUBST-INST})$$

avec $\Gamma = \Gamma'; x : T$ et $\sigma = \sigma' \cup \{x \mapsto N\}$, montrons que $\sigma(\Gamma) \vdash$. Nous avons, par hypothèse d'induction, $\sigma'(\Gamma') \vdash$. Il suffit donc de montrer que $\sigma(\Gamma) = \sigma'(\Gamma')$. Puisque $x \in \mathsf{Dom}(\sigma)$, $\sigma(\Gamma) = \sigma(\Gamma')$. Puisque $\Gamma' \vdash$ et que $x \notin \mathsf{DV}(\Gamma')$, le lemme 3.1.12 nous indique que $\sigma(\Gamma') = \sigma'(\Gamma')$.

□

4. Jugements de structure

Nous définissons et étudions dans cette section un nouveau jugement, le *jugement de structure*, qui sera utilisé dans les lemmes d'inversion.

4.1. Définitions.

4.1.1. DÉFINITION (Pile de termes). Une *pile de termes* est une liste ordonnée de termes définie inductivement ainsi :

$$\pi \quad ::= \quad \circ \mid \mathsf{push}(\mathrm{N}, \pi)$$

4.1.2. DÉFINITION (Jugement de structure). Un *jugement de structure* est un jugement de la forme $\Gamma \vdash \pi : \mathrm{R} \rhd \mathrm{X}$ où Γ est un contexte, π une pile de termes, R et X des termes. X est appelée la *base* du jugement.

Ce jugement doit se comprendre ainsi. Γ et R sont respectivement le contexte et le type du jugement de typage que nous cherchons à inverser. X est une sorte ou un constructeur de type explicite encoquillé dans le terme R suite à l'application des règles (CUM), (GEN), (INST), (SUB-I), (SUB-E-1) et (SUB-E-2). La pile π contient des termes fournis lors de l'application de la règle (INST).

Ainsi si nous considérons le jugement donné plus haut en exemple

$$\mathrm{T} : \mathsf{Type}_0; \mathrm{U} : \mathsf{Type}_0 \vdash \mathsf{Prop} : \forall x : \mathrm{T}. \{z : [\mathrm{U}] \to \mathsf{Type}_0 \mid \forall y : \mathrm{T} \to \mathsf{Prop}. y\, x\}$$

un jugement de structure associé sera de la forme

$$\mathrm{T} : \mathsf{Type}_0; \mathrm{U} : \mathsf{Type}_0 \vdash \mathsf{push}(\mathrm{N_T}, \mathsf{push}(\mathrm{N_U}, \circ)) : \forall x : \mathrm{T}. \{z : [\mathrm{U}] \to \mathsf{Type}_0 \mid \forall y : \mathrm{T} \to \mathsf{Prop}. y\, x\} \rhd \mathsf{Type}_0$$

où $\mathrm{N_T}$ est lié à la quantification implicite dépendante $\forall x : \mathrm{T}. (\ldots)$ et $\mathrm{N_U}$ est lié à la quantification implicite non-dépendante $[\mathrm{T}] \to (\ldots)$. Notons que la quantification $\forall y : \mathrm{T} \to \mathsf{Prop}. (\ldots)$ qui se trouve dans le domaine du type sous-ensemble n'est pas prise en compte dans la pile.

4.1.3. DÉFINITION (Règles de dérivation des jugements de structure). Les jugements de structure valides sont les jugements dérivés à partir des règles suivantes :

$$\frac{\Gamma \vdash}{\Gamma \vdash \circ : s \rhd s} \text{ (STRUCT-BASE-S)}$$

$$\frac{\Gamma \vdash \Pi x : \mathrm{T}. \mathrm{U} : s}{\Gamma \vdash \circ : \Pi x : \mathrm{T}. \mathrm{U} \rhd \Pi x : \mathrm{T}. \mathrm{U}} \text{ (STRUCT-BASE-}\Pi\text{)}$$

$$\frac{\Gamma \vdash \exists x : \mathrm{A}. \mathrm{B} : s}{\Gamma \vdash \circ : \exists x : \mathrm{A}. \mathrm{B} \rhd \exists x : \mathrm{A}. \mathrm{B}} \text{ (STRUCT-BASE-}\exists\text{)}$$

$$\frac{\Gamma \vdash \Sigma x : \mathrm{A}. \mathrm{B} : s}{\Gamma \vdash \circ : \Sigma x : \mathrm{A}. \mathrm{B} \rhd \Sigma x : \mathrm{A}. \mathrm{B}} \text{ (STRUCT-BASE-}\Sigma\text{)}$$

$$\frac{\Gamma \vdash \pi : \mathrm{R} \rhd \mathrm{X} \qquad \mathrm{R}' \preceq \mathrm{R}}{\Gamma \vdash \pi : \mathrm{R}' \rhd \mathrm{X}} \text{ (STRUCT-IND-CONV)}$$

$$\frac{\Gamma \vdash \pi : \mathrm{R} \rhd \mathrm{X}}{\Gamma \vdash \pi : \{x : \mathrm{R} \mid \mathrm{B}\} \rhd \mathrm{X}} \text{ (STRUCT-IND-}\{\}\text{)}$$

$$\frac{\Gamma \vdash \pi : \mathrm{R}[x/\mathrm{N}] \rhd \mathrm{X} \qquad \Gamma \vdash \mathrm{N} : \mathrm{A}}{\Gamma \vdash \mathsf{push}(\mathrm{N}, \pi) : \forall x : \mathrm{A}. \mathrm{R} \rhd \mathrm{X}} \text{ (STRUCT-IND-}\forall\text{)}$$

Les règles de dérivation peuvent se diviser en deux groupes. Le premier regroupe les règles de base qui initialisent la pile : R est égal au type X, il n'y a pas de quantification implicite dans R et la pile est donc vide. Il y a quatre règles de base, relatives aux sortes et aux trois constructeurs de type explicites, qui sont les types naturels des termes que nous allons inverser (constructeurs de type, abstraction, paire existentielle, paire dépendante).

Le second groupe, constitué des trois dernières règles, correspond aux modifications que peut subir R par les règles (CUM), (INST) et (SUB-E-1). L'application de la règle (INST) provoque l'empilement

de l'argument implicite en haut de la pile. La pile ne change pas dans le cas de la cumulativité ou de l'application de (SUB-E-1).

Ces règles seront utilisées dans les démonstrations par induction des lemmes d'inversion. Elles permettront de *remonter* la dérivation, c'est-à-dire de montrer que les hypothèses de départ sur le type R sont également valides pour les prémisses, et de pouvoir ainsi appliquer par induction le lemme d'inversion aux prémisses. (STRUCT-IND-CONV) sera utilisée pour la règle (CUM), (STRUCT-IND-{}) pour (SUB-E-1) et (STRUCT-IND-∀) pour (INST). Cela explique notamment que dans (STRUCT-IND-CONV), la condition est $R' \preceq R$ et non $R \preceq R'$.

Notons également que le jugement de structure n'impose pas à R d'être bien typé. En pratique cependant, cela sera le cas, car R correspondra au type d'un jugement de typage.

4.2. Résultats utiles. Les deux lemmes suivants se démontrent immédiatement par induction sur la dérivation du jugement de structure.

4.2.1. Lemme. *Si* $\Gamma \vdash \pi : R \rhd X$ *alors il existe une sorte* s *telle que* $\Gamma \vdash X : s$. *Si de plus* R *est une sorte ou un constructeur de type explicite, alors* $R \preceq X$ *et* $\pi = \circ$.

4.2.2. REMARQUE. Le lemme de déclaration des variables libres (lemme 3.4.3 page 25) nous permet de déduire que nous avons $\mathsf{FV}(X) \subset \mathsf{DV}(\Gamma)$ si $\Gamma \vdash \pi : R \rhd X$.

4.2.3. Lemme (Affaiblissement du jugement de structure)**.** *Si* $\Gamma \vdash \pi : R \rhd X$, $\Gamma \subseteq \Gamma'$ *et* $\Gamma' \vdash$ *alors* $\Gamma' \vdash \pi : R \rhd X$.

4.2.4. REMARQUE. Ce lemme sera utile pour le cas de la règle (SUB-E-2) dans les différentes inductions des lemmes d'inversion du reste du chapitre.

Le lemme suivant permet d'inverser les jugements de structure. Il est utile pour le cas des règles (GEN) et (SUB-I) dans les preuves des lemmes d'inversion.

4.2.5. Lemme (Inversion du jugement de structure)**.**

(1) Si $\Gamma \vdash \pi : R \rhd X$ *avec* $\forall x : T.\, U \preceq R$ *et* $\Gamma; x : T \vdash$ *alors il existe un terme* N *et une pile* π' *tels que*

$$\pi = \mathsf{push}(N, \pi') \quad \Gamma \vdash N : T \quad \Gamma \vdash \pi' : U\,[x/N] \rhd X$$

(2) Si $\Gamma \vdash \pi : R \rhd X$ *avec* $\{x : A \mid B\} \preceq R$ *alors* $\Gamma \vdash \pi : A \rhd X$.

DÉMONSTRATION. Les deux preuves sont similaires. Nous ne détaillons que la première, légèrement plus complexe.

- Nous procédons par induction sur la dérivation du jugement de structure. D'après le lemme 2.3.15 page 18, $\forall x : T.\, U \preceq R$ implique que R n'est ni une sorte, ni un produit dépendant, ni un type existentiel, ni une somme dépendante. La dernière règle appliquée est donc (STRUCT-IND-CONV) ou (STRUCT-IND-∀).
  - Dans le cas de (STRUCT-IND-CONV), nous pouvons appliquer par induction le lemme à la prémisse et conclure immédiatement.
  - Dans le cas de (STRUCT-IND-∀), nous avons des termes A, B, N et une pile de termes π' tels que :

    $$R = \forall x : A.\, B \quad \pi = \mathsf{push}(N, \pi') \quad \Gamma \vdash N : A \quad \Gamma \vdash \pi' : B\,[x/N] \rhd X$$

    Montrons que $\Gamma \vdash N : T$ et $\Gamma \vdash \pi' : U\,[x/N] \rhd X$.
    - Nous avons $\forall x : T.\, U \preceq \forall x : A.\, B$. Par inversion (lemme 2.3.16 page 18), nous savons que $A \preceq T$ et $U \preceq B$. D'après le lemme 2.3.2 page 14, nous avons également $U\,[x/N] \preceq B\,[x/N]$.

— Montrons que $\Gamma \vdash N : T$. Par inversion de $\Gamma; x : T \vdash$, il existe une sorte s telle que $\Gamma \vdash T : s$. Puisque $A \preceq T$, nous dérivons $\Gamma \vdash N : T$ par application de la règle (CUM) au jugement $\Gamma \vdash N : A$.

— Montrons que $\Gamma \vdash \pi' : U[x/N] \rhd X$. Il suffit d'appliquer (STRUCT-IND-CONV) à $\Gamma \vdash \pi' : B[x/N] \rhd X$ puisque $U[x/N] \preceq B[x/N]$.

□

4.2.6. REMARQUE. La condition $\Gamma; x : T \vdash$ est nécessaire dans le cas du produit car il faut convertir le type de N. Cela n'est pas le cas pour le type sous-ensemble car il n'y a pas d'instanciation de variable.

5. Inversion des constructeurs de types

5.1. Explications informelles. Nous avons maintenant les outils syntaxiques pour énoncer et démontrer des lemmes d'inversion.

Contrairement aux lemmes d'inversion des PTS, et même de ICC, nous allons supposer plutôt que montrer un lien entre le type du jugement inversé et son type naturel. Ainsi, si le jugement considéré est $\Gamma \vdash M : R$, alors nous allons supposer $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : R \rhd X$, où X est le type naturel du terme M (sorte si M est un constructeur de type, constructeur de type associé si M est un constructeur). Intuitivement la substitution substitue les variables du contexte qui peuvent être généralisées dans la *suite* de la dérivation, tandis que la pile contient les généralisations déjà effectuées auparavant. Les résultats que nous allons montrer ne sont donc pas très généraux, mais malgré tout suffisamment forts pour démontrer la proposition du type des types et la préservation du typage, car ils sont toujours utilisés dans des cas où R est de la forme naturelle.

Les preuves se font par induction mutuelle sur la dérivation de $\Gamma \vdash M : R$ en considérant la dernière règle appliquée. Sept cas sont possibles : la règle de formation ou d'introduction liée à M (cas de base), la règle de cumulativité (CUM), les deux règles d'introduction et d'élimination du produit implicite, les trois règles d'introduction et d'élimination du type sous-ensemble.

Le cas de base et la règle (SUB-E-2) étant à part, la preuve va consister pour chaque autre règle à (1) montrer que les hypothèses du lemme sont vérifiées pour la prémisse typant M, (2) appliquer par induction le lemme à la prémisse, (3) montrer que les hypothèses d'induction sont bien les conclusions recherchées. (CUM) et les règles d'élimination (INST) et (SUB-E-1) se traitent facilement en appliquant les règles inductives de dérivation de jugements de structure. Les règles d'introduction (GEN) et (SUB-I) vont nécessiter d'inverser les jugements de structure (lemme 4.2.5). (SUB-E-2) se traite différemment : il faut appliquer aux hypothèses d'induction (SUB-E-2) pour pouvoir conclure.

Cette section est consacrée à l'inversion des constructeurs de type et à la démonstration de la proposition du type des types. La section suivante étudiera l'inversion des constructeurs et la préservation du typage par $\beta\iota$-réduction.

5.2. Résultats.

5.2.1. Lemme (Inversion des constructeurs de type)**.** *Si* $\Gamma \vdash \Box z : T.U : R$ *et soient* s *une sorte,* σ *une substitution et* π *une pile de termes telles que*

$$\_ \vdash \sigma \div \Gamma \quad \sigma(\Gamma) \vdash \pi : \sigma(R) \rhd s$$

Alors il existe des sortes s_1, s_2, s_3 *telles que*

$$\sigma(\Gamma) \vdash \sigma(T) : s_1 \quad \sigma(\Gamma); z : \sigma(T) \vdash \sigma(U) : s_2 \quad (s_1, s_2, s_3) \in \mathbf{Rule}_\Box \quad s_3 \preceq s.$$

DÉMONSTRATION. Nous procédons par induction sur la dérivation du jugement de typage. La dernière règle de la dérivation de $\Gamma \vdash \Box z : \mathrm{T} \,.\, \mathrm{U} : \mathrm{R}$ est parmi les règles (□-FORM), (GEN), (INST), (CUM), (SUB-I), (SUB-E-1), (SUB-E-2).

- (□-FORM) : Si $\sigma(\Gamma) \vdash \pi : s_3 \rhd s$ et

$$\frac{\Gamma \vdash \mathrm{T} : s_1 \qquad \Gamma; z : \mathrm{T} \vdash \mathrm{U} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\Box}{\Gamma \vdash \Box z : \mathrm{T} \,.\, \mathrm{U} : s_3}$$

  - D'après le lemme 4.2.1, nous avons $s_3 \preceq s$.
  - Puisque $\_ \vdash \sigma \div \Gamma$, nous obtenons $\sigma(\Gamma) \vdash \sigma(\mathrm{T}) : s_1$ en substituant $\Gamma \vdash \mathrm{T} : s_1$ par σ (lemme 3.2.7).
  - En appliquant (SUBST-GEN), nous avons $\_ \vdash \sigma \div \Gamma; z : \mathrm{T}$. Nous pouvons alors substituer par σ dans $\Gamma; z : \mathrm{T} \vdash \mathrm{U} : s_2$ et obtenir $\sigma(\Gamma; z : \mathrm{T}) \vdash \sigma(\mathrm{U}) : s_2$. Puisque $z \notin \mathsf{Dom}(\sigma)$, nous avons $\sigma(\Gamma; z : \mathrm{T}) = \sigma(\Gamma); z : \sigma(\mathrm{T})$ et pouvons conclure.

- (GEN) : Si

$$\frac{\Gamma; x : \mathrm{A} \vdash \Box z : \mathrm{T} \,.\, \mathrm{U} : \mathrm{R} \qquad x \notin \mathsf{FV}(\Pi z : \mathrm{T} \,.\, \mathrm{U})}{\Gamma \vdash \Box z : \mathrm{T} \,.\, \mathrm{U} : \forall x : \mathrm{A} \,.\, \mathrm{R}}\ (\text{GEN})$$

  avec s, σ et π telles que

  - $\_ \vdash \sigma \div \Gamma$ et
  - $\sigma(\Gamma) \vdash \pi : \forall x : \sigma(\mathrm{A}) \,.\, \sigma(\mathrm{R}) \rhd s$.

  Montrons qu'il existe des sortes s_1, s_2, s_3 telles que

  - $\sigma(\Gamma) \vdash \sigma(\mathrm{T}) : s_1$
  - $\sigma(\Gamma); z : \sigma(\mathrm{T}) \vdash \sigma(\mathrm{U}) : s_2$
  - $(s_1, s_2, s_3) \in \mathbf{Rule}_\Box$
  - $s_3 \preceq s$.

  Pour cela nous voulons appliquer par induction le lemme à la prémisse $\Gamma; x : \mathrm{A} \vdash \Box z : \mathrm{T} \,.\, \mathrm{U} : \mathrm{R}$. Nous procédons en trois étapes.

  (1) Montrons qu'il existe une substitution σ_0 et une pile de termes π_0 tels que $\_ \vdash \sigma_0 \div \Gamma; x : \mathrm{A}$ et $\sigma_0(\Gamma; x : \mathrm{A}) \vdash \pi_0 : \sigma_0(\mathrm{R}) \rhd s$.

  - Montrons que $\sigma(\Gamma); x : \sigma(\mathrm{A}) \vdash$.
    - De $\_ \vdash \sigma \div \Gamma$ et $\Gamma; x : \mathrm{A} \vdash$ nous déduisons, par affaiblissement, $\_ \vdash \sigma \div \Gamma; x : \mathrm{A}$.
    - Nous concluons alors grâce au corollaire 3.2.8, puisque $x \notin \mathsf{Dom}(\sigma)$.
  - Puisque $\sigma(\Gamma); x : \sigma(\mathrm{A}) \vdash$, nous pouvons appliquer le point (1) du lemme 4.2.5 et inverser $\sigma(\Gamma) \vdash \pi : \forall x : \sigma(\mathrm{A}) \,.\, \sigma(\mathrm{R}) \rhd s$. Il existe donc un terme N et une pile π_0 tels que
    - $\pi = \mathsf{push}(\mathrm{N}, \pi_0)$
    - $\sigma(\Gamma) \vdash \mathrm{N} : \sigma(\mathrm{A})$
    - $\sigma(\Gamma) \vdash \pi_0 : \sigma(\mathrm{R})\,[x/\mathrm{N}] \rhd s$.
  - Nous pouvons dériver

$$\frac{\_ \vdash \sigma \div \Gamma \qquad \sigma(\Gamma) \vdash \mathrm{N} : \sigma(\mathrm{A}) \qquad \Gamma; x : \mathrm{A} \vdash}{\_ \vdash \sigma \cup \{x \mapsto \mathrm{N}\} \div \Gamma; x : \mathrm{A}}\ (\text{SUBST-INST})$$

    d'où, en posant $\sigma_0 = \sigma \cup \{x \mapsto \mathrm{N}\}$, $\_ \vdash \sigma_0 \div \Gamma; x : \mathrm{A}$.

- Montrons que $\sigma_0(\Gamma; x : A) \vdash \pi_0 : \sigma_0(R) \rhd s$. Puisque $\sigma(\Gamma) \vdash \pi_0 : \sigma(R)[x/N] \rhd s$, il suffit de montrer que $\sigma_0(R) = \sigma(R)[x/N]$ et $\sigma_0(\Gamma; x : A) = \sigma(\Gamma)$. Puisque $\sigma(\Gamma) \vdash \pi : \forall x : \sigma(A).\sigma(R) \rhd s$, il suffit de montrer que
  - — $\sigma_0(R) = \sigma(R)[x/N]$ découle du corollaire 3.2.6.
  - — Par définition de la substitution d'un contexte, nous avons $\sigma_0(\Gamma; x : A) = \sigma_0(\Gamma)$. Comme σ et σ_0 sont identiques sur $\mathsf{DV}(\Gamma)$, nous avons $\sigma_0(\Gamma) = \sigma(\Gamma)$ (lemme 3.1.12), d'où $\sigma_0(\Gamma; x : A) = \sigma(\Gamma)$.

(2) Nous pouvons donc appliquer par induction le lemme à $\Gamma; x : A \vdash \square z : T.U : R$ et en déduisons l'existence de sortes s_1, s_2, s_3 telles que
- $\sigma_0(\Gamma; x : A) \vdash \sigma_0(T) : s_1$
- $\sigma_0(\Gamma; x : A); z : \sigma_0(T) \vdash \sigma_0(U) : s_2$
- $(s_1, s_2, s_3) \in \mathbf{Rule}$
- $s_3 \preceq s$.

(3) Puisque $\sigma_0(\Gamma; x : A) = \sigma(\Gamma)$, il suffit pour conclure de montrer que $\sigma_0(T) = \sigma(T)$ et $\sigma_0(U) = \sigma(U)$.
- De $\Gamma \vdash \square z : T.U : \forall x : A.R$, nous déduisons $\mathsf{FV}(T) \subset \mathsf{DV}(\Gamma)$ et $\mathsf{FV}(U) \subset \mathsf{DV}(\Gamma) \cup \{z\}$. D'après la convention 3.1.13 page 20 appliquée au jugement $\Gamma; x : A \vdash \square z : T.U : R$, nous avons $z \neq x$ et $z \notin \mathsf{DV}(\Gamma)$, d'où $z \notin \mathsf{Dom}(\sigma') = \mathsf{Dom}(\sigma) \cup \{x\}$.
- σ et σ' sont donc identiques sur $\mathsf{FV}(T)$ et $\mathsf{FV}(U)$, d'où, d'après le lemme 3.1.5, $\sigma'(U) = \sigma(U)$ et $\sigma'(T) = \sigma(T)$.

- (Inst) : Si

$$\frac{\Gamma \vdash \square z : T.U : \forall x : A.R \qquad \Gamma \vdash N : A}{\Gamma \vdash \square z : T.U : R[x/N]}\ (\text{Gen})$$

avec s, σ et π telles que $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \sigma(R[x/N]) \rhd s$. Montrons qu'il existe des sortes s_1, s_2, s_3 telles que
- $\sigma(\Gamma) \vdash \sigma(T) : s_1$
- $\sigma(\Gamma); z : \sigma(T) \vdash \sigma(U) : s_2$
- $(s_1, s_2, s_3) \in \mathbf{Rule}_\square$
- $s_3 \preceq s$.

Pour cela, montrons que $\sigma(\Gamma) \vdash \mathsf{push}(\sigma(N), \pi) : \forall x : \sigma(A).\sigma(R) \rhd s$. Nous conclurons alors en appliquant par induction le lemme à la prémisse $\Gamma \vdash \Pi z : T.U : \forall x : A.R$.

Nous souhaitons dériver $\sigma(\Gamma) \vdash \mathsf{push}(\sigma(N), \pi) : \forall x : \sigma(A).\sigma(R) \rhd s$ en utilisant la règle (Struct-Ind-∀) :

$$\frac{\sigma(\Gamma) \vdash \pi : \sigma(R)[x/\sigma(N)] \rhd s \qquad \sigma(\Gamma) \vdash \sigma(N) : \sigma(A)}{\sigma(\Gamma) \vdash \mathsf{push}(\sigma(N), \pi) : \forall x : \sigma(A).\sigma(R) \rhd s}\ (\text{Struct-Ind-}\forall)$$

Pour cela, il suffit de montrer les deux prémisses.
- Montrons $\sigma(\Gamma) \vdash \pi : \sigma(R)[x/\sigma(N)] \rhd s$. Puisque $\sigma(\Gamma) \vdash \pi : \sigma(R[x/N]) \rhd s$, il suffit de montrer que $\sigma(R[x/N]) = \sigma(R)[x/\sigma(N)]$. D'après le corollaire 3.1.7, il suffit de montrer que $x \notin \mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma)$, ce qui, quitte à renommer par α-conversion les variables liées dans $\forall x : A.R$, est immédiat.
- Montrons $\sigma(\Gamma) \vdash \sigma(N) : \sigma(A)$. Puisque $\_ \vdash \sigma \div \Gamma$, il suffit d'appliquer la proposition 3.2.7 (préservation du typage) au jugement $\Gamma \vdash N : A$.

- (Cum) : Si

$$\frac{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}' \qquad \Gamma \vdash \mathrm{R} : s \qquad \mathrm{R}' \preceq \mathrm{R}}{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}} \text{ (Cum)}$$

avec $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd s$.

Montrons que $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}') \rhd s$. Nous en déduirons le résultat souhaité en appliquant par induction le lemme à la prémisse $\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}'$.

D'après le lemme 3.1.9 appliqué à $\mathrm{R}' \preceq \mathrm{R}$, nous avons $\sigma(\mathrm{R}') \preceq \sigma(\mathrm{R})$. En appliquant (Struct-Ind-Conv), nous dérivons $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}') \rhd \mathrm{X}$.

- (Sub-I) : Si

$$\frac{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}}{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \{x : \mathrm{R} \mid \mathrm{B}\}} \text{ (Sub-I)}$$

avec $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \{x : \sigma(\mathrm{R}) \mid \sigma(\mathrm{B})\} \rhd s$.

Par inversion de $\sigma(\Gamma) \vdash \pi : \{x : \sigma(\mathrm{R}) \mid \sigma(\mathrm{B})\} \rhd s$ (point (2) du lemme 4.2.5), nous avons $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd s$. Nous déduisons alors le résultat souhaité en appliquant par induction le lemme à la prémisse $\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}$.

- (Sub-E-1) : Si

$$\frac{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \{x : \mathrm{R} \mid \mathrm{B}\}}{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}} \text{ (Sub-E-1)}$$

avec $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd s$.

  - Nous dérivons $\sigma(\Gamma) \vdash \pi : \sigma(\{x : \mathrm{R} \mid \mathrm{B}\}) \rhd s$ en appliquant à $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd s$ la règle (Struct-Ind-{}).
  - Nous déduisons alors le résultat souhaité en appliquant par induction le lemme à la prémisse $\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \{x : \mathrm{R} \mid \mathrm{B}\}$.

- (Sub-E-2) : Si

$$\frac{\Gamma; y : \mathrm{B}[x/c] \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R} \qquad \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} \qquad y \notin \mathsf{FV}(\Box z : \mathrm{T}.\mathrm{U})}{\Gamma \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}} \text{ (Sub-E-2)}$$

avec $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd s$.

Montrons qu'il existe des sortes s_1, s_2, s_3 telles que

  - $\sigma(\Gamma) \vdash \sigma(\mathrm{T}) : s_1$
  - $\sigma(\Gamma); z : \sigma(\mathrm{T}) \vdash \sigma(\mathrm{U}) : s_2$
  - $(s_1, s_2, s_3) \in \mathbf{Rule}_\Box$
  - $s_3 \preceq s$.

Nous procédons en trois étapes.

(1) La première étape consiste à appliquer par induction le lemme à la prémisse $\Gamma; y : \mathrm{B}[x/c] \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}$.

  - Montrons $\_ \vdash \sigma \div \Gamma; y : \mathrm{B}[x/c]$.
    - De $\Gamma; y : \mathrm{B}[x/c] \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{R}$ nous déduisons $\Gamma; y : \mathrm{B}[x/c] \vdash$.
    - Nous concluons en appliquant (Subst-Gen) à $\_ \vdash \sigma \div \Gamma$ et à $\Gamma; y : \mathrm{B}[x/c] \vdash$.
  - Montrons que $\sigma(\Gamma; y : \mathrm{B}[x/c]) = \sigma(\Gamma); y : \sigma(\mathrm{B})[x/\sigma(c)]$.
    - Puisque $\_ \vdash \sigma \div \Gamma$, nous avons $y \notin \mathsf{Dom}(\sigma)$, d'où $\sigma(\Gamma; y : \mathrm{B}[x/c]) = \sigma(\Gamma); y : \sigma(\mathrm{B}[x/c])$.
    - Il suffit donc de montrer que $\sigma(\mathrm{B}[x/c]) = \sigma(\mathrm{B})[x/\sigma(c)]$.
      - De $\_ \vdash \sigma \div \Gamma$, nous déduisons $\mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma) \subset \mathsf{DV}(\Gamma)$. De plus, d'après la convention 3.1.13 page 20 appliquée au jugement $\Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\}$, $x \notin \mathsf{DV}(\Gamma)$, d'où $x \notin \mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma)$.

— Nous concluons alors en appliquant le lemme 3.1.7 (substitution et substitution de variables) à σ et $B[x/c]$.

- Montrons $\sigma(\Gamma; y : B[x/c]) \vdash \pi : \sigma(R) \rhd s$ par affaiblissement de $\sigma(\Gamma) \vdash \pi : \sigma(R) \rhd s$.
  — D'après le corollaire 3.2.8 appliqué à $\_ \vdash \sigma \div \Gamma; y : B[x/c]$, nous avons $\sigma(\Gamma; y : B[x/c]) \vdash$.
  — Nous avons vu que $\sigma(\Gamma; y : B[x/c]) = \sigma(\Gamma); y : \sigma(B)[x/\sigma(c)]$, d'où $\sigma(\Gamma); y : \sigma(B)[x/\sigma(c)] \vdash$.
  — Nous concluons alors bien par affaiblissement de $\sigma(\Gamma) \vdash \pi : \sigma(R) \rhd s$ (lemme 4.2.3).
- Les hypothèses d'induction nous indiquent alors qu'il existe des sortes s_1, s_2, s_3 telles que
  — $\sigma(\Gamma; y : B[x/c]) \vdash \sigma(T) : s_1$ soit $\sigma(\Gamma); y : \sigma(B)[x/\sigma(c)] \vdash \sigma(T) : s_1$,
  — $\sigma(\Gamma; y : B[x/c]); z : \sigma(T) \vdash \sigma(U) : s_2$ soit $\sigma(\Gamma); y : \sigma(B)[x/\sigma(c)] z : \sigma(T) \vdash \sigma(U) : s_2$,
  — $(s_1, s_2, s_3) \in \mathbf{Rule}_{\square}$
  — $s_3 \preceq s$.

(2) La deuxième étape est de dériver $\sigma(\Gamma) \vdash \sigma(T) : s_1$. Pour cela nous voulons appliquer la règle de typage (Sub-E-2) :

$$\frac{\begin{cases} \sigma(\Gamma) \vdash s_1 : s_1' \\ \sigma(\Gamma) \vdash \sigma(c) : \{x : \sigma(A) \mid \sigma(B)\} \end{cases} \qquad \begin{cases} \sigma(\Gamma); y : \sigma(B)[x/\sigma(c)] \vdash \sigma(T) : s_1 \\ y \notin \mathsf{FV}(\sigma(T)) \end{cases}}{\sigma(\Gamma) \vdash \sigma(T) : s_1} \text{(Sub-E-2)}$$

La prémisse typant $\sigma(T)$ a déjà été prouvée. Montrons les deux autres prémisses et la condition de bord.

- Montrons qu'il existe une sorte s_1' telle que $\sigma(\Gamma) \vdash s_1 : s_1'$. Par absence de sorte maximale de **Axiom**, il existe une sorte s_1' telle que $(s_1, s_1') \in \mathbf{Axiom}$. Par ailleurs, $\_ \vdash \sigma \div \Gamma$ implique $\sigma(\Gamma) \vdash$. Nous concluons alors en appliquant (Sort).
- Montrons que $\sigma(\Gamma) \vdash \sigma(c) : \{x : \sigma(A) \mid \sigma(B)\}$ est dérivable : puisque $\_ \vdash \sigma \div \Gamma$, nous pouvons substituer le jugement $\Gamma \vdash c : \{x : A \mid B\}$ par σ.
- Montrons enfin que $y \notin \mathsf{FV}(\sigma(T))$. Puisque $y \notin \mathsf{DV}(\Gamma)$, il suffit de montrer que $\mathsf{FV}(\sigma(T)) \subset \mathsf{DV}(\Gamma)$.
  — Nous avons $\mathsf{FV}(\sigma(T)) \subset \mathsf{FV}(T) \cup \mathsf{FV}(\sigma)$ (lemme 3.1.6).
  — $\mathsf{FV}(\sigma) \subset \mathsf{DV}(\Gamma)$ puisque $\_ \vdash \sigma \div \Gamma$ (lemme 3.2.2).
  — Montrons que $\mathsf{FV}(T) \subset \mathsf{DV}(\Gamma)$.
    — $\mathsf{FV}(T) \subset \mathsf{FV}(\square z : T.U)$, par définition des variables libres d'un terme et puisque $z \notin \mathsf{FV}(T)$ (convention de Barendregt).
    — Puisque $\Gamma \vdash \square z : T.U : R$, nous avons $\mathsf{FV}(\square z : T.U) \subset \mathsf{DV}(\Gamma)$.

(3) La troisième et dernière étape est de dériver $\sigma(\Gamma); z : \sigma(T) \vdash \sigma(U) : s_2$.
De même que pour l'étape précédente, nous souhaitons utiliser la règle (Sub-E-2) :

$$\frac{\begin{cases} \sigma(\Gamma); z : \sigma(T); y : \sigma(B)[x/\sigma(c)] \vdash \sigma(U) : s_2 \\ \sigma(\Gamma); z : \sigma(T) \vdash \sigma(c) : \{x : \sigma(A) \mid \sigma(B)\} \end{cases} \qquad \begin{cases} \sigma(\Gamma); z : \sigma(T) \vdash s_2 : s_2' \\ y \notin \mathsf{FV}(\sigma(U)) \end{cases}}{\sigma(\Gamma); z : \sigma(T) \vdash \sigma(U) : s_2} \text{(Sub-E-2)}$$

Montrons les trois prémisses et la condition de bord.

- Montrons que $\sigma(\Gamma); z : \sigma(T); y : \sigma(B)[x/\sigma(c)] \vdash \sigma(U) : s_2$ est dérivable.

— Par hypothèse d'induction Nous avons montré $\sigma(\Gamma); y : \sigma(B)\,[x/\sigma(c)]; z : \sigma(T) \vdash \sigma(U) : s_2$ (hypothèses d'induction) et $\sigma(\Gamma) \vdash \sigma(T) : s_1$ (cf. deuxième étape).

— Nous pouvons donc conclure en appliquant le lemme d'échange de variables (lemme 3.4.11 page 27) à $\sigma(\Gamma); y : \sigma(B)\,[x/\sigma(c)]; z : \sigma(T) \vdash \sigma(U) : s_2$.

- Montrons que $\sigma(\Gamma); z : \sigma(T) \vdash \sigma(c) : \{x : \sigma(A) \mid \sigma(B)\}$ est dérivable. Nous avons montré à la deuxième étape la dérivabilité $\sigma(\Gamma) \vdash \sigma(c) : \{x : \sigma(A) \mid \sigma(B)\}$. Nous concluons alors par affaiblissement du jugement puisque $\sigma(\Gamma); z : \sigma(T) \vdash$.
- Par absence de sorte maximale de **Axiom**, il existe une sorte s_2' telle que $(s_2, s_2') \in$ **Axiom**. Nous concluons en appliquant (Sort) $\sigma(\Gamma); z : \sigma(T) \vdash$.
- Montrons que $y \notin \mathsf{FV}(\sigma(U))$. Nous procédons comme dans la preuve de $y \notin \mathsf{FV}(\sigma(T))$, en renommant éventuellement par α-conversion $\Box z : T\,.\,U$ afin d'avoir $y \neq z$.

□

5.2.2. REMARQUE. Dans la preuve précédente, les cas des règles (Gen), (Inst), (Cum), (Sub-I) et (Sub-E-1) se résolvent sans dépendre de la forme du terme à inverser. Les preuves des autres lemmes d'inversion seront donc identiques pour les cas de ces règles. Le cas de base dépend naturellement du terme à inverser. Enfin la règle (Sub-E-2) se traite au cas par cas du fait de l'utilisation du lemme d'échange des variables.

Nous avons besoin dans des preuves ultérieures d'un résultat moins général et plus simple à énoncer que celui du lemme précédent. Le point (1) du corollaire suivant exprime précisément ce qui est nécessaire pour prouver le lemme du type des types (ci-après dans cette section) et les lemmes d'inversion de la section 6. Le point (2) sera utilisé ultérieurement dans la preuve de la préservation du typage (section 7).

5.2.3. Corollaire (Inversion des constructeurs de type)**.** *Si* $\Gamma \vdash \Box x : T\,.\,U : s$ *alors :*

(1) il existe des sortes s_1, s_2, s_3 *telles que*

- $\Gamma \vdash T : s_1$
- $\Gamma; x : T \vdash U : s_2$
- $(s_1, s_2, s_3) \in \mathbf{Rule}_\Box$
- $s_3 \preceq s$.

(2) $\Gamma \vdash U\,[x/N] : s_2$ *pour tout terme* N *tel que* $\Gamma \vdash N : T$.

DÉMONSTRATION. (1) Il suffit d'appliquer le lemme précédent avec $\sigma = \emptyset$ et $\pi = \circ$, puisque nous avons $\_ \vdash \emptyset \div \Gamma$ et $\Gamma \vdash \bullet : s \rhd s$.

(2) Conséquence de (1) : par application du lemme de substitutivité (lemme 3.4.8 page 26) en substituant x par N dans $\Gamma; x : T \vdash U : s_2$.

□

La proposition suivante indique que tout terme typant un autre terme dans un contexte est typable par une sorte dans ce même contexte. Cette propriété est souhaitable pour elle-même dans tout PTS. Elle est également utile pour montrer d'autres résultats, par exemple dans ce chapitre, les lemmes d'inversions des constructeurs et la préservation du typage par $\beta\iota$-réduction.

5.2.4. Proposition (Type des types)**.** *Si* $\Gamma \vdash M : T$ *alors il existe une sorte* $s \in \mathcal{S}$ *telle que* $\Gamma \vdash T : s$.

DÉMONSTRATION. Nous procédons par induction en considérant la dernière règle appliquée dans la dérivation du jugement $\Gamma \vdash M : T$. La plupart des règles ne pose aucune difficulté. Seules les règles d'élimination ((App), (Inst), (Sub-E-1)) nécessitent de faire appel au corollaire 5.2.3.

- Pour les règles d'introduction ((Lam), (Gen), (Σ-I), (∃-I), (Sub-I)), la règle de cumulativité (Cum) et (Sub-E-2), le jugement dont nous souhaitons montrer la dérivation, $\Gamma \vdash T : s$, est l'une des prémisses.
- Pour la règle (Sort) et les règles de formation ((E-Prd), (I-Prd), (Σ), (∃), (Sub)), le terme T est une sorte. Par absence de sorte maximale dans l'ensemble **Axiom**, il existe une sorte telle que $(T, s) \in \mathbf{Axiom}$. Le lemme de bonne formation des contextes (lemme 3.4.1 page 25) appliqué à $\Gamma \vdash M : T$ nous indique que $\Gamma \vdash$. Nous pouvons alors, en appliquant la règle (Wf-S), dériver $\Gamma \vdash T : s$.
- (Σ-E), (∃-E) : il suffit d'appliquer la règle (App) aux deux prémisses typant P et c.
- (Var) : il existe Γ_1 et Γ_2 tels que $\Gamma = \Gamma_1; x : T; \Gamma_2$. En inversant $\Gamma_1; x : T \vdash$, nous dérivons $\Gamma_1 \vdash T : s$, puis, par affaiblissement, le jugement $\Gamma \vdash T : s$.
- (App), (Inst), (Sub-E-1) : Par hypothèse d'induction, il existe une sorte s telle que, respectivement, $\Gamma \vdash \Pi x : T . U : s$, $\Gamma \vdash \forall x : T . U : s$ ou $\Gamma \vdash \{x : A \mid B\} : s$ sont dérivables. Nous concluons alors grâce au corollaire 5.2.3 (point (2) pour les produits, point (1) pour le type sous-ensemble).

□

6. Inversion des constructeurs

Dans cette section, nous présentons des résultats d'inversion de l'abstraction, de la paire dépendante, et de la paire existentielle. Ces résultats vont permettre de traiter les cas de réduction de tête dans la preuve de la préservation du typage.

L'inversion de l'abstraction et de la paire dépendantes sont analogues. Leurs démonstrations ressemblent à celle de l'inversion des constructeurs de type. Seuls le cas de base et celui de la règle (Sub-E-2) sont plus compliqués du fait que le type naturel n'est plus une sorte mais un produit ou une somme dépendante. Le cas de la paire existentielle est plus délicat ; nous prouvons un résultat plus faible mais néanmoins suffisant pour la preuve de la préservation du typage.

6.1. Inversion de l'abstraction.

6.1.1. Lemme (Inversion de l'abstraction)**.** *Soient* $\Gamma \vdash \lambda z . M : R$ *un jugement,* σ *une substitution, et* π *une pile de termes,* T, U *des termes tels que :*

- $\_ \vdash \sigma \div \Gamma$
- $\sigma(\Gamma) \vdash \pi : \sigma(R) \rhd \Pi z : T . U$

Alors $\sigma(\Gamma); z : T \vdash \sigma(M) : U$.

Démonstration. Par induction sur la dérivation du jugement. La dernière règle de la dérivation de $\Gamma \vdash \lambda z . M : R$ est parmi les règles (Lam), (Gen), (Inst), (Cum), (Sub-I), (Sub-E-1), (Sub-E-2). Nous ne nous intéressons qu'au cas de base (règle (Lam)) et à celui de la règle (Sub-E-2). Les autres cas se traitent comme dans la démonstration de l'inversion des types (cf. remarque 5.2.2).

- (Lam) : Si

$$\frac{\Gamma; z : A \vdash M : B}{\Gamma \vdash \lambda z . M : \Pi z : A . B}\ (\text{Lam})$$

avec σ, π, T, U tels que :

- $\_ \vdash \sigma \div \Gamma$
- $\sigma(\Gamma) \vdash \pi : \Pi z : \sigma(A) . \sigma(B) \rhd \Pi z : T . U$

Montrons que $\sigma(\Gamma); z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \mathrm{U}$. Pour cela, montrons d'abord que $\sigma(\Gamma); z : \sigma(\mathrm{A}) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{B})$. Nous montrerons ensuite qu'il est possible de convertir le contexte $\sigma(\Gamma); z : \sigma(\mathrm{A})$ en $\sigma(\Gamma); z : \mathrm{T}$ puis le type $\sigma(\mathrm{B})$ en U.

- Montrons que $\sigma(\Gamma); z : \sigma(\mathrm{A}) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{B})$.
  - — Par affaiblissement de $\_ \vdash \sigma \div \Gamma$, nous avons $\_ \vdash \sigma \div \Gamma; z : \mathrm{A}$.
  - — Nous pouvons donc appliquer la proposition 3.2.7 au jugement $\Gamma; z : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$ et à la substitution σ et obtenir $\sigma(\Gamma; z : \mathrm{A}) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{B})$.
  - — Puisque $z \notin \mathsf{Dom}(\sigma)$, nous avons $\sigma(\Gamma; z : \mathrm{A}) = \sigma(\Gamma); z : \sigma(\mathrm{A})$, d'où le résultat souhaité.
- D'après le lemme 4.2.1 appliqué à $\sigma(\Gamma) \vdash \pi : \Pi z : \sigma(\mathrm{A}) . \sigma(\mathrm{B}) \rhd \Pi z : \mathrm{T} . \mathrm{U}$:
  - — $\Pi z : \sigma(\mathrm{A}) . \sigma(\mathrm{B}) \preceq \Pi z : \mathrm{T} . \mathrm{U}$
  - — il existe une sorte s telle que $\sigma(\Gamma) \vdash \Pi z : \mathrm{T} . \mathrm{U} : s$.

  Par inversion de la cumulativité (lemme 2.3.16 page 18), nous avons $\mathrm{T} \preceq \sigma(\mathrm{A})$ et $\sigma(\mathrm{B}) \preceq \mathrm{U}$. Par inversion du produit (corollaire 5.2.3), il existe s' telle que $\sigma(\Gamma); z : \mathrm{T} \vdash \mathrm{U} : s'$.
- Montrons que $\sigma(\Gamma); z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{B})$.
  - — De $\sigma(\Gamma); z : \mathrm{T} \vdash \mathrm{U} : s'$, nous déduisons $\sigma(\Gamma); z : \mathrm{T} \vdash$.
  - — De $\mathrm{T} \preceq \sigma(\mathrm{A})$, nous avons $\sigma(\Gamma); z : \mathrm{T} \preceq \sigma(\Gamma); z : \sigma(\mathrm{A})$.
  - — Nous pouvons donc appliquer le corollaire 3.4.10 page 27 (cumulativité et changement de contexte) à $\sigma(\Gamma); z : \sigma(\mathrm{A}) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{B})$ et obtenir le résultat voulu.
- Montrons enfin que $\sigma(\Gamma); z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \mathrm{U}$: puisque $\sigma(\mathrm{B}) \preceq \mathrm{U}$ et $\sigma(\Gamma); z : \mathrm{T} \vdash \mathrm{U} : s'$ nous pouvons appliquer (Cum) à $\sigma(\Gamma); z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{B})$ et conclure.

- (Sub-E-2) : La preuve est similaire à celle de l'inversion des constructeurs de type. Une différence importante est l'utilisation de l'inversion du produit explicite (corollaire 5.2.3). Si

$$\frac{\Gamma; y : \mathrm{B}\,[x/c] \vdash \lambda z . \mathrm{M} : \mathrm{R} \qquad \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} \qquad y \notin \mathsf{FV}(\lambda z . \mathrm{M})}{\Gamma \vdash \lambda z . \mathrm{M} : \mathrm{R}}\ (\text{Sub-E-2})$$

avec $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \mathrm{R} \rhd \Pi z : \mathrm{T} . \mathrm{U}$, montrons que $\sigma(\Gamma); z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \mathrm{U}$.

- Nous montrons d'abord, comme dans la première étape de l'inversion des types, que $\sigma(\Gamma); y : \sigma(\mathrm{B})\,[x/\sigma(c)]; z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \mathrm{U}$ d'après les hypothèses d'induction.
- Ensuite, d'après le lemme 4.2.1, il existe une sorte s telle que $\sigma(\Gamma) \vdash \Pi z : \mathrm{T} . \mathrm{U} : s$. Par inversion (corollaire 5.2.3), il existe une sorte s' telle que $\sigma(\Gamma) \vdash \mathrm{T} : s'$.
- Nous pouvons alors appliquer le lemme d'échange de variables (lemme 3.4.11) à $\sigma(\Gamma); y : \sigma(\mathrm{B})\,[x/\sigma(c)]); z : \mathrm{T} \vdash \sigma(\mathrm{M}) : \mathrm{U}$ et en déduire $\sigma(\Gamma); z : \mathrm{T}; y : \sigma(\mathrm{B})\,[x/\sigma(c)]) \vdash \sigma(\mathrm{M}) : \mathrm{U}$.
- Nous concluons alors, comme dans la troisième étape de la preuve de l'inversion des constructeurs de type, en appliquant au jugement $\sigma(\Gamma); z : \mathrm{T}; y : \sigma(\mathrm{B})\,[x/\sigma(c)]) \vdash \sigma(\mathrm{M}) : \mathrm{U}$ la règle (Sub-E-2).

□

6.2. Inversion de la paire dépendante.

6.2.1. Lemme (Inversion de la paire dépendante)**.** *Soient* $\Gamma \vdash (a\,,\, b) : \mathrm{R}$ *un jugement,* σ *une substitution, et* π *une pile de termes,* A, B *des termes tels que :*

- $\_ \vdash \sigma \div \Gamma$
- $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd \Sigma z : \mathrm{A} . \mathrm{B}$

Alors $\sigma(\Gamma) \vdash \sigma(a) : \mathrm{A}$ *et* $\sigma(\Gamma) \vdash \sigma(b) : \mathrm{B}\,[z/\sigma(a)]$.

DÉMONSTRATION. La preuve est similaire au lemme d'inversion de l'abstraction. Nous ne détaillons que le cas de base, le seul qui varie sensiblement. Si

$$\frac{\Gamma \vdash a : \mathrm{T} \qquad \Gamma \vdash b : \mathrm{U}\,[z/a]}{\Gamma \vdash (a, b) : \Sigma z : \mathrm{T}\,.\,\mathrm{U}}\ (\Sigma\text{-I})$$

avec σ et π telles que $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \Sigma z : \sigma(\mathrm{T})\,.\,\sigma(\mathrm{U}) \rhd \Sigma z : \mathrm{A}\,.\,\mathrm{B}$. Montrons que $\sigma(\Gamma) \vdash \sigma(a) : \mathrm{A}$ et $\sigma(\Gamma) \vdash \sigma(b) : \mathrm{B}\,[z/\sigma(a)]$.

- En appliquant la substitution σ aux jugements de la dérivation (lemme 3.2.7), nous avons :
  - $\sigma(\Gamma) \vdash \sigma(a) : \sigma(\mathrm{T})$
  - $\sigma(\Gamma) \vdash \sigma(b) : \sigma(\mathrm{U})\,[z/\sigma(a)]$
- De même que pour l'abstraction, en utilisant le lemme 4.2.1 puis par inversion de la cumulativité, nous montrons que $\sigma(\mathrm{T}) \preceq \mathrm{A}$ et $\sigma(\mathrm{U}) \preceq \mathrm{B}$ et qu'il existe une sorte s telle que $\sigma(\Gamma) \vdash \Sigma z : \mathrm{A}\,.\,\mathrm{B} : s$.
- Par inversion de $\sigma(\Gamma) \vdash \Sigma z : \mathrm{A}\,.\,\mathrm{B} : s$ (corollaire 5.2.3), il existe des sortes s, s' telles que $\sigma(\Gamma) \vdash \mathrm{A} : s$ et $\sigma(\Gamma); z : \mathrm{A} \vdash \mathrm{B} : s'$.
- Par cumulativité sur $\sigma(\Gamma) \vdash \sigma(a) : \sigma(\mathrm{T})$, nous obtenons $\sigma(\Gamma) \vdash \sigma(a) : \mathrm{A}$.
- En substituant z par $\sigma(a)$ dans $\sigma(\Gamma); z : \mathrm{A} \vdash \mathrm{B} : s'$, nous avons $\sigma(\Gamma) \vdash \mathrm{B}\,[z/\sigma(a)] : s'$. Puisque $\sigma(\mathrm{U}) \preceq \mathrm{B}$, nous avons $\sigma(\mathrm{U})\,[z/\sigma(a)] \preceq \mathrm{B}\,[z/\sigma(a)]$ (lemme 2.3.2 page 14). Par cumulativité, nous pouvons dériver $\sigma(\Gamma) \vdash \sigma(b) : \mathrm{B}\,[z/\sigma(a)]$.

□

6.3. Inversion faible de la paire existentielle. Pour la paire existentielle, nous ne pouvons pas démontrer un énoncé analogue à celui de la paire dépendante.

En effet, cet énoncé serait le suivant : si

- $\Gamma \vdash (\diamond, b) : \mathrm{R}$
- $\_ \vdash \sigma \div \Gamma$
- $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd \exists x : \mathrm{A}\,.\,\mathrm{B}$

alors il existe a tel que $\sigma(\Gamma) \vdash \sigma(b) : \mathrm{B}\,[x/\sigma(a)]$.

Mais la règle (SUB-E-2) pose alors problème car l'hypothèse d'induction liée à la prémisse $\Gamma; y : \mathrm{B}\,[x/c] \vdash (\diamond, b) : \mathrm{R}$ ne permet pas de montrer que a ne dépend pas de y.[6]

Pour contourner ce problème, nous prouvons un énoncé plus faible : l'admissibilité d'une règle d'inférence similaire à (∃-E) et où a n'apparaît pas. Cet énoncé est suffisamment fort pour la preuve de la préservation du typage.

6.3.1. Lemme (Inversion faible de la paire existentielle)**.** *Soient* $\Gamma \vdash (\diamond, b) : \mathrm{R}$ *un jugement,* σ *une substitution, et* π *une pile de termes,* A, B *des termes tels que :*

- $\_ \vdash \sigma \div \Gamma$
- $\sigma(\Gamma) \vdash \pi : \sigma(\mathrm{R}) \rhd \exists z : \mathrm{A}\,.\,\mathrm{B}$

Alors la règle suivante est admissible

$$\frac{\sigma(\Gamma) \vdash \mathrm{P} : \exists z : \mathrm{A}\,.\,\mathrm{B} \to s \qquad \sigma(\Gamma); z : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y) \qquad x \notin FV(f)}{\sigma(\Gamma) \vdash f\,[y/\sigma(b)] : \mathrm{P}\,(\diamond, \sigma(b))}$$

6. La règle (GEN) ne poserait pas de problème car nous appliquerions à la prémisse $\Gamma; x : \mathrm{A} \vdash (\diamond, b) : \mathrm{R}$ une substitution $\sigma' = \sigma \cup \{x \to \mathrm{N}\}$ et x n'apparaîtrait plus. Pour (SUB-E-2), la substitution appliquée à la prémisse serait σ, donnée en hypothèse de départ, et qui s'appliquerait également à la conclusion.

DÉMONSTRATION. Par induction sur la structure de la dérivation de $\Gamma \vdash (\diamond, b) : \mathrm{R}$.

Nous procédons pour les règles (GEN), (INST), (CUM), (SUB-I), (SUB-E-1) et (SUB-E-2) comme dans les preuves des lemmes d'inversion précédents. Seul le cas de (∃-I) diffère.

- (∃-I) Si

$$\frac{\Gamma \vdash a : \mathrm{T} \qquad \Gamma \vdash b : \mathrm{U}\,[z/a]}{\Gamma \vdash (\diamond, b) : \exists z : \mathrm{T}\,.\,\mathrm{U}}\ (\exists\text{-I})$$

avec σ et π telles que $\_ \vdash \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash \pi : \exists z : \sigma(\mathrm{T})\,.\,\sigma(\mathrm{U}) \rhd \exists z : \mathrm{A}\,.\,\mathrm{B}$.

Supposons que les jugements suivants sont dérivables :

- $\sigma(\Gamma) \vdash \mathrm{P} : \exists z : \mathrm{A}\,.\,\mathrm{B} \to s$
- $\sigma(\Gamma); z : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y)$

Montrons que nous pouvons également dériver $\sigma(\Gamma) \vdash f\,[y/\sigma(b)] : \mathrm{P}\,(\diamond, \sigma(b))$.

Pour cela, montrons que $\sigma(\Gamma) \vdash \sigma(a) : \mathrm{A}$ et $\sigma(\Gamma) \vdash \sigma(b) : \mathrm{B}\,[z/\sigma(a)]$. Nous concluons ensuite en substituant (lemme 3.4.8 page 26) d'abord x par $\sigma(a)$, puis y par $\sigma(b)$ dans le jugement $\sigma(\Gamma); z : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y)$.

- En appliquant la substitution σ aux jugements de la dérivation (lemme 3.2.7), nous avons :
  - $\sigma(\Gamma) \vdash \sigma(a) : \sigma(\mathrm{T})$
  - $\sigma(\Gamma) \vdash \sigma(b) : \sigma(\mathrm{U})\,[z/\sigma(a)]$
- D'après le lemme 4.2.1 appliqué à $\sigma(\Gamma) \vdash \pi : \exists z : \sigma(\mathrm{T})\,.\,\sigma(\mathrm{U}) \rhd \exists z : \mathrm{A}\,.\,\mathrm{B}$ et par inversion de cumulativité, nous montrons que $\sigma(\Gamma) \vdash \exists z : \mathrm{A}\,.\,\mathrm{B} : s$ et $\sigma(\mathrm{T}) \preceq \mathrm{A}$ et $\sigma(\mathrm{U}) \preceq \mathrm{B}$.
- Par inversion du type existentiel (corollaire 5.2.3), il existe des sortes s, s' telles que $\sigma(\Gamma) \vdash \mathrm{A} : s$ et $\sigma(\Gamma); z : \mathrm{A} \vdash \mathrm{B} : s'$.
- Par cumulativité, nous avons alors $\sigma(\Gamma) \vdash \sigma(a) : \mathrm{A}$. En substituant z par $\sigma(a)$ dans $\sigma(\Gamma); z : \mathrm{A} \vdash \mathrm{B} : s'$, nous avons $\sigma(\Gamma) \vdash \mathrm{B}\,[z/\sigma(a)] : s'$. Par cumulativité,nous avons alors $\sigma(\Gamma) \vdash \sigma(b) : \mathrm{B}\,[z/\sigma(a)]$.

□

6.4. Récapitulatif. De même que pour l'inversion des types, nous regroupons dans un corollaire des résultats plus pratiques et qui seront effectivement utilisés dans la preuve de la préservation du typage par βι-réduction. Le point (3) est exactement le résultat nécessaire dans la preuve de la préservation du typage dans le cas où la ι-réduction a lieu en tête de l'éliminateur de la paire existentielle.

6.4.1. Corollaire.

(1) Si $\Gamma \vdash \lambda x\,.\,\mathrm{M} : \Pi x : \mathrm{A}\,.\,\mathrm{B}$, *alors* $\Gamma; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{B}$.

(2) Si $\Gamma \vdash (a\,,\, b) : \Sigma x : \mathrm{A}\,.\,\mathrm{B}$, *alors* $\Gamma \vdash a : \mathrm{A}$ *et* $\Gamma \vdash b : \mathrm{B}\,[x/a]$.

(3) La règle suivante est admissible :

$$\frac{\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}\,.\,\mathrm{B} \to s \qquad \Gamma \vdash (\diamond, b) : \exists x : \mathrm{A}\,.\,\mathrm{B} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y) \qquad x \notin \mathit{FV}(f)}{\Gamma \vdash f\,[y/b] : \mathrm{P}\,(\diamond, b)}$$

DÉMONSTRATION. Il suffit d'appliquer respectivement les lemmes 6.1.1, 6.2.1 et 6.3.1 avec $\sigma = \emptyset$ et $\pi = \circ$. En effet, il est clair que $\_ \vdash \emptyset \div \Gamma$. De plus, nous montrons facilement, grâce au lemme du type des types, que $\Gamma \vdash \bullet : \Box x : \mathrm{A}\,.\,\mathrm{B} \rhd \Box x : \mathrm{A}\,.\,\mathrm{B}$ pour $\Box \in \{\Pi, \Sigma, \exists\}$. □

7. Préservation du typage par βι-réduction

7.1. Résultat. Nous pouvons maintenant atteindre le but de ce chapitre : démontrer que le typage est préservé par βι-réduction.

7.1.1. Proposition (Auto-réduction)**.** *Soient* Γ *un contexte et* M, M', T *des termes de* ICC_Σ *tels que* $\Gamma \vdash M : T$ *et* $M \to_{\beta\iota} M'$. *Alors* $\Gamma \vdash M' : T$.

DÉMONSTRATION. Nous procédons par induction sur la dérivation de typage du jugement $\Gamma \vdash M : T$.

- Les règles (VAR) et (SORT) ne peuvent pas être cette dernière règle car les variables et sortes sont irréductibles.
- La récurrence est immédiate pour les règles (LAM), (GEN), (INST), (CUM), (SUB-I), (SUB-E-1), (SUB-E-2) et (∃-I), (Σ-I) Pour (GEN) et (SUB-E-2), il faut utiliser le lemme 2.2.2 page 12 qui indique que si $M \to_{\beta\iota} M'$ alors $\mathsf{FV}(M') \subset \mathsf{FV}(M)$.
- (SUB-I) et (Σ-I) : si $a \to_{\beta\iota} a'$, par hypothèses d'induction, nous avons $\Gamma \vdash a' : A$. Il suffit donc de montrer que $\Gamma \vdash b : B[x/a']$. Nous avons $B[x/a'] \cong B[x/a]$ (lemme 2.2.8 page 13) et $\Gamma \vdash B[x/a'] : s$ (point 2 du corollaire 5.2.3). Nous concluons alors en appliquant (CUM).
- Pour les règles de formation, nous traitons la règle générique (□-FORM). La dernière règle de la dérivation est :

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma; x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\square}{\Gamma \vdash \square x : A.B : s_3}$$

  La réduction peut avoir lieu soit dans le codomaine ($\square x : A.B \to_{\beta\iota} \square x : A.B'$), soit dans le domaine ($\square x : A.B \to_{\beta\iota} \square x : A'.B$).
  - Dans le premier cas, par hypothèse d'induction, nous avons $\Gamma; x : A \vdash B' : s_2$. En appliquant (□-FORM), nous obtenons $\Gamma \vdash \square x : A.B' : s_3$.
  - Dans le second cas, par hypothèse d'induction, le jugement $\Gamma \vdash A' : s_1$ est dérivable. En appliquant la règle (WF-S), nous déduisons que le contexte $\Gamma; x : A'$ est bien formé. Le lemme de conversion de contexte (lemme 3.4.10 page 27) nous indique que le jugement $\Gamma; x : A' \vdash B : s_2$ est dérivable. Par une nouvelle application de (□-FORM), nous dérivons le jugement $\Gamma \vdash \square x : A'.B : s_3$.
- (APP) : La dernière étape de la dérivation est

$$\frac{\Gamma \vdash M : \Pi x : T.U \qquad \Gamma \vdash N : T}{\Gamma \vdash MN : U[x/N]}\ (\text{APP})$$

  Trois cas sont alors possibles :
  (1) la réduction s'effectue dans M : nous concluons directement par induction.
  (2) la réduction s'effectue dans N : il existe N' tel que $MN \to_{\beta\iota} MN'$. En appliquant l'hypothèse d'induction aux prémisses, nous dérivons le jugement $\Gamma \vdash MN' : U[x/N']$. Or $\Gamma \vdash U[x/N] : s$ (type des types), et $U[x/N'] \cong U[x/N]$ (lemme 2.2.8 page 13). Nous concluons en appliquant la règle (CUM).
  (3) la réduction s'effectue en position de tête : il existe un terme M' tel que $M = \lambda x.M'$ et $MN = (\lambda x.M')N \to_{\beta\iota} M'[x/N]$. Par inversion de l'abstraction (point (1) du corollaire 6.4.1) dans le jugement $\Gamma \vdash \lambda x.M' : \Pi x : T.U$, nous avons $\Gamma; x : T \vdash M' : U$. D'après le lemme de substitutivité (lemme 3.4.8 page 26), nous pouvons substituer x par N et nous obtenons $\Gamma \vdash M' : U[x/N]$.
- (∃-E) : Nous avons la dérivation suivante :

$$\frac{\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s \qquad \Gamma \vdash c : \exists x : \mathrm{A}.\mathrm{B} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y) \qquad x \notin \mathsf{FV}(f)}{\Gamma \vdash \mathsf{Elim}_\exists(y.f, c) : \mathrm{P}\,c}\ (\exists\text{-E})$$

De même que pour (App), trois cas sont possibles. La réduction peut se faire soit dans le terme f ($\mathsf{Elim}_\exists(y.f,c) \to_{\beta\iota} \mathsf{Elim}_\exists(y.f',c)$), soit dans c ($\mathsf{Elim}_\exists(y.f,c) \to_{\beta\iota} \mathsf{Elim}_\exists(y.f,c')$), soit en tête ($c = (\diamond, b)$ et $\mathsf{Elim}_\exists(y.f,(\diamond,b)) \rhd_\iota f\,[y/b]$).

Dans le premier cas, par induction et puisque $\mathsf{FV}(f') \subset \mathsf{FV}(f)$, nous dérivons directement $\Gamma \vdash \mathsf{Elim}_\exists(y.f',c) : \mathrm{P}\,c$.

Dans le second cas, par induction, nous dérivons $\Gamma \vdash c' : \exists x : \mathrm{A}.\mathrm{B}$ puis $\Gamma \vdash \mathsf{Elim}_\exists(y.f,c') : \mathrm{P}\,c'$. Puisque $\Gamma \vdash \mathrm{P}\,c' : s$, nous pouvons convertir et dériver $\Gamma \vdash \mathsf{Elim}_\exists(y.f,c') : \mathrm{P}\,c$.

Enfin, dans le troisième cas, nous concluons directement grâce au point (3) du corollaire 6.4.1.

- (Σ-E) : Nous avons la dérivation suivante :

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma x : \mathrm{A}.\mathrm{B} \to s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}.\mathrm{B} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)}{\Gamma \vdash \mathsf{Elim}_\Sigma(xy.f, c) : \mathrm{P}\,c}\ (\Sigma\text{-E})$$

De même que pour (∃-E), la réduction peut s'effectuer dans f, dans c, ou en tête. Dans les deux premiers cas, les hypothèses d'induction permettent de conclure aisément.

Dans le troisième, nous avons $c = (a, b)$ et $\mathsf{Elim}_\Sigma(xy.f,(a,b)) \rhd_\iota f\,[x/a]\,[y/b]$. Montrons que $\Gamma \vdash f\,[x/a]\,[y/b] : \mathrm{P}\,(a, b)$.

D'après le point (2) du corollaire 6.4.1, $\Gamma \vdash a : \mathrm{A}$ et $\Gamma \vdash b : \mathrm{B}\,[x/a]$. Il suffit alors de substituer successivement x par a, puis y par b dans le jugement $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)$.

□

Nous montrons maintenant un corollaire immédiat, qui sera utile pour la vérification de l'algorithme d'inférence de type extrait. Il permet de convertir le type d'un terme par un type βι-équivalent.

7.1.2. Corollaire (Réductions multiples, cumulativité)**.**

(1) Si $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *et si* $\mathrm{M} \twoheadrightarrow_{\beta\iota} \mathrm{M}'$ *alors* $\Gamma \vdash \mathrm{M}' : \mathrm{T}$.

(2) Si $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *et si* $\mathrm{T} \twoheadrightarrow_{\beta\iota} \mathrm{T}'$ *alors* $\Gamma \vdash \mathrm{M} : \mathrm{T}'$.

DÉMONSTRATION. (1) Se déduit de la proposition 7.1.1 par une récurrence immédiate sur le nombre de réductions.

(2) Nous avons $\Gamma \vdash \mathrm{M} : \mathrm{T}$ et $\mathrm{T} \preceq \mathrm{T}'$. Montrons qu'il existe une sorte s telle que $\Gamma \vdash \mathrm{T}' : s$. Nous concluons alors en appliquant la règle (Cum).

- D'après le lemme du type des types, il existe une sorte s telle que $\Gamma \vdash \mathrm{T} : s$.
- D'après le point 1, nous avons bien $\Gamma \vdash \mathrm{T}' : s$.

□

7.2. Report de la η-réduction. Le report de la η-réduction n'est pas valable en général. Ainsi nous avons

$$\mathsf{Elim}_\Sigma(xy.(x\,y), \lambda z.(x, y)\,z) \to_\eta \mathsf{Elim}_\Sigma(xy.(x\,y),(x,y)) \to_{\iota_\Sigma} x\,y$$

mais $\mathsf{Elim}_\Sigma(xy.(x\,y), \lambda z.(x, y)\,z)$ est en forme βι-normale et ne peut se βι-réduire vers $x\,y$.

Nous pouvons par contre montrer que le report de η est valable pour les termes bien typés. La preuve, détaillée à l'annexe C (proposition 2.3.1 page 237), utilise la préservation du typage par βι-réduction.

7.2.1. Proposition (Report de la règle η)**.** *Soient* $\mathrm{M}, \mathrm{N}, \mathrm{T}$ *des termes et* Γ *un contexte. Si* $\mathrm{M} \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}$ *et si* $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *alors il existe un terme* P *tel que* $\mathrm{M} \twoheadrightarrow_{\beta\iota} \mathrm{P} \twoheadrightarrow_\eta \mathrm{N}$.

L'étude syntaxique de ICC_Σ se termine avec ce chapitre. La deuxième partie, constituée de trois chapitres, est consacrée à une étude sémantique du calcul dont le but est de prouver une propriété syntaxique : sa cohérence logique.

Deuxième partie

Modèles de ICC_{Σ}

Préambule

Nous présentons dans cette partie un modèle permettant de prouver la cohérence logique de ICC_Σ.

Cette partie est constituée de trois chapitres. Le premier présente un modèle abstrait de cohérence. Ce modèle définit une structure abstraite d'interprétation qui vérifie certains axiomes. La raison d'être de ce modèle est que l'existence d'une certaine instanciation de ce modèle [7] permet de prouver la cohérence de ICC_Σ. Les deux chapitres suivants sont consacrés à la construction de cette instanciation. Le second chapitre présente ainsi des notions utiles pour l'implantation du modèle concret. Le dernier chapitre présente l'implantation du modèle abstrait, ce qui achève la preuve de la cohérence de ICC_Σ.

Ce modèle est une extension du modèle de cohérence de ICC présenté dans les chapitres 3 à 5 de la thèse de Miquel. C'est un modèle classique (le tiers-exclu y est valide) et *proof-irrelevant* (les preuves d'une proposition ne sont pas discernables par l'égalité de Leibniz). Il interprète le calcul implicite restreint, qui est une restriction de ICC sans la règle de renforcement. Cette restriction n'est pas gênante puisque ICC_Σ ne possède pas non plus cette règle.

Nous avons repris la structure de la présentation [**Miq01**] : nos trois chapitres suivent *grosso modo* la progression des chapitres 3 à 5 de la thèse de Miquel. Notamment nous utilisons nous aussi un modèle abstrait [8] qui permet de rendre la preuve de cohérence plus modulaire et plus claire car la présentation de la structure concrète d'interprétation est séparée de la démonstration des propriétés de cette structure.

7. Une instance où l'interprétation de $\mathsf{False} = \Pi A : \mathsf{Prop}.\mathsf{Prop}$ est l'ensemble vide.
8. Miquel précise que que sa démarche est similaire à celle de Fridlender [**Fri02**]

Chapitre 3

Modèle abstrait de cohérence

Nous présentons dans ce chapitre un modèle qui permet d'interpréter les termes et les jugements valides de ICC_Σ. Ce modèle est abstrait : la structure d'interprétation, et notamment l'ensemble d'interprétation, ne sont pas définis concrètement ; leur existence est simplement supposée. Ce modèle est un modèle de cohérence : nous pourrons déduire la cohérence logique de ICC_Σ à partir de l'existence d'un modèle concret vérifiant un certain critère.

Ce modèle est une extension du modèle abstrait défini par Miquel dans sa thèse. Nous avons enrichi la structure abstraite afin que le modèle abstrait puisse interpréter les termes et règles de typage supplémentaires de ICC_Σ et refléter le modèle concret qui sera présenté au chapitre 5.

Ce court chapitre est divisé en deux sections. Nous présentons tout d'abord le modèle abstrait, puis nous montrons qu'il s'agit bien d'un modèle de cohérence de ICC_Σ.

1. Présentation du modèle

1.1. Présentation formelle. Nous définissons tout d'abord une fonction qui associe à toute variable un élément de l'ensemble d'interprétation.

1.1.1. DÉFINITION (Valuation). Soient $\mathscr{M}$ un ensemble et $\mathscr{V}$ l' ensemble des variables.

- Une *valuation sur* $\mathscr{M}$ est une application $\rho : \mathscr{V} \to \mathscr{M}$.
- L'ensemble de toutes les valuations sur $\mathscr{M}$ est noté $\mathbf{Val}_\mathscr{M}$.
- Soient ρ une valuation sur $\mathscr{M}$, $x \in \mathscr{V}$ une variable et $a \in \mathscr{M}$ un élément de $\mathscr{M}$, nous notons $\rho; x \leftarrow a$ la valuation qui associe a à x et qui associe $\rho(x')$ à tout $x' \neq x$.

Nous pouvons maintenant définir un modèle abstrait du calcul implicite.

1.1.2. DÉFINITION (Modèle abstrait du calcul implicite avec types somme). Un *modèle abstrait du calcul implicite avec types somme* est une structure de la forme

$$(\mathscr{M}, \cdot, \mathsf{Pair}, [\![]\!], \leq, \mathscr{T}, \mathsf{El}, \mathsf{Gen}, \mathbf{1}, \mathbf{2}, \Pi, \forall, \Sigma, \exists, \sigma, (\mathsf{u}_i)_{i\in\omega})$$

où

- $\mathscr{M}$ est un ensemble non vide ;
- $\cdot : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ est une opération binaire sur $\mathscr{M}$;
- $[\![]\!] : \Lambda_{\mathrm{ICC}_\Sigma} \times \mathbf{Val}_\mathscr{M} \to \mathscr{M}$ est une application associant à tout terme M de ICC_Σ et à toute valuation $\rho \in \mathbf{Val}_\mathscr{M}$ un objet noté $[\![\mathrm{M}]\!]_\rho \in \mathscr{M}$;
- $\leq \subset \mathscr{M} \times \mathscr{M}$ est une relation d'ordre sur $\mathscr{M}$.
- $\mathscr{T}$ est un sous-ensemble de $\mathscr{M}$;
- $\mathsf{El} : \mathscr{T} \to \mathbb{P}(\mathscr{M})$ et $\mathsf{Gen} : \mathbb{P}(\mathscr{M}) \to \mathscr{M}$ sont des applications ;
- $\mathbf{1}, \mathbf{2}, \Pi, \forall, \Sigma, \exists, \sigma$ sont des éléments de $\mathscr{M}$;

- $\mathsf{Pair} : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$ est une opération binaire sur $\mathscr{M}$;
- $(\mathsf{u}_i)_{i\in\omega}$est une famille d'éléments de $\mathscr{T}$.

Ces paramètres vérifient les axiomes a à r et 1 à 16.

Les paramètres sont récapitulés dans la figure 8. Les axiomes des modèles abstraits de ICC_Σ sont décrits dans les figures 9 à 13.

Paramètres des modèles abstraits du calcul implicite avec sommes

- un ensemble $\mathscr{M}$
- une opération binaire $(\cdot) : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$
- une application $[\![]\!] : \Lambda_{\mathrm{ICC}_\Sigma} \times \mathbf{Val}_{\mathscr{M}} \to \mathscr{M}$
- une relation d'ordre $(\leq) \subset \mathscr{M} \times \mathscr{M}$
- un sous-ensemble $\mathscr{T} \subset \mathscr{M}$
- une application $\mathsf{El} : \mathscr{T} \to \mathbb{P}(\mathscr{M})$
- cinq constantes $\Pi, \forall, \Sigma, \exists, \sigma \in \mathscr{M}$
- une suite de constantes $(\mathsf{u}_i)_{i\in\omega} \in \mathscr{T}^\omega$
- deux constantes $\mathbf{1}, \mathbf{2} \in \mathscr{M}$
- une opération binaire $\mathsf{Pair} : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$
- une application $\mathsf{Gen} : \mathbb{P}(\mathscr{M}) \to \mathscr{M}$

FIGURE 8. Paramètres des modèles abstraits

1.1.3. REMARQUE. Comme pour l'application en λ-calcul, nous écrivons ab pour $a\cdot b$, $((a\cdot b)\cdot c)$ pour abc, et $(a\cdot(b\cdot c))$ pour $a(bc)$.

1.2. Commentaires.

Paramètres du modèle. Le modèle abstrait de ICC_Σ que nous définissons ici est bien une extension du modèle abstrait défini par Miquel. Ainsi, si $[\![]\!]|_{\Lambda\times\mathbf{Val}_{\mathscr{M}}}$ désigne la restriction de la fonction $[\![]\!]$ aux termes de ICC_Σ, $(\mathscr{M}, \cdot, [\![]\!]|_{\Lambda\times\mathbf{Val}_{\mathscr{M}}}, \leq, \mathscr{T}, \mathsf{El}, \Pi, \forall, (\mathsf{u}_i)_{i\in\omega})$, est un modèle abstrait du calcul implicite de Miquel.

Miquel a construit son modèle abstrait comme une extension d'un modèle du λ-calcul pur (λ-modèle). Ainsi ICC_Σ est interprété comme le serait un λ-calcul pur (l'opération binaire $\cdot$ interprète l'application) enrichi de constantes $\Pi, \forall$ liées aux types produit. L'ensemble $\mathscr{T}$ va interpréter les termes qui sont des types. La fonction El va associer à chaque type les termes typés par lui. La relation d'ordre $\leq$ est liée à l'interprétation de la η-réduction. Enfin les constantes $(\mathsf{u}_i)_{i\in\omega}$ vont représenter la hiérarchie des sortes présentes dans le calcul.

Afin d'étendre le modèle abstrait à ICC_Σ, nous ajoutons des constantes $\Sigma, \exists, \sigma$ pour les types somme, Pair pour la paire dépendante et $\mathbf{1}$,$\mathbf{2}$ pour l'éliminateur de la somme dépendante. La fonction Gen est nécessaire pour refléter une difficulté technique dans l'extension du modèle concret à la somme dépendante et au type existentiel.

Axiomes du modèle. Les axiomes de la fonction d'interprétation définissent inductivement l'interprétation des termes. Ceux liés à la relation d'ordre sont liés l'interprétation du calcul (règles de réduction, cumulativité). Informellement, les axiomes des codes des univers sont liés à l'interprétations de la règle (SORT) et des règles de formation. Les axiomes restants sont essentiellement utilisés pour interpréter les règles d'introduction et d'élimination.

Axiomes de $[\![]\!]$,$\leq$ et de Pair

(a) si $a \leq a'$ et $b \leq b'$ alors $ab \leq a'b'$

(b) pour tout $a, b \in \mathscr{M}$, $\mathsf{Pair}(a,b)\cdot\mathbf{1} = a$ et $\mathsf{Pair}(a,b)\cdot\mathbf{2} = b$

(c) $[\![x]\!]_\rho = \rho(x)$

(d) $[\![\mathsf{Prop}]\!]_\rho = \mathsf{u}_0$

(e) $[\![\mathsf{Type}_i]\!]_\rho = \mathsf{u}_i$, pour $i > 0$

(f) $[\![\Pi x : \mathrm{T}.\mathrm{U}]\!]_\rho = (\Pi \cdot [\![\mathrm{T}]\!]_\rho)\cdot[\![\lambda x.\mathrm{U}]\!]_\rho$

(g) $[\![\lambda x.\mathrm{M}]\!]_\rho \cdot a = [\![\mathrm{M}]\!]_{\rho;x\leftarrow a}$

(h) si $[\![\mathrm{M}]\!]_{\rho;x\leftarrow a} = [\![\mathrm{M}']\!]_{\rho';x\leftarrow a}$ pour tout $a \in \mathscr{M}$, alors $[\![\lambda x.\mathrm{M}]\!]_\rho = [\![\lambda x.\mathrm{M}']\!]_{\rho'}$

(i) $[\![\mathrm{M}\,\mathrm{N}]\!]_\rho = [\![\mathrm{M}]\!]_\rho \cdot [\![\mathrm{N}]\!]_\rho$

(j) $[\![\forall x : \mathrm{T}.\mathrm{U}]\!]_\rho = (\forall \cdot [\![\mathrm{T}]\!]_\rho)\cdot[\![\lambda x.\mathrm{U}]\!]_\rho$

(k) $[\![\Sigma x : \mathrm{A}.\mathrm{B}]\!]_\rho = (\Sigma \cdot [\![\mathrm{A}]\!]_\rho)\cdot[\![\lambda x.\mathrm{B}]\!]_\rho$

(l) $[\![(\mathrm{M},\mathrm{N})]\!]_\rho = \mathsf{Pair}([\![\mathrm{M}]\!]_\rho, [\![\mathrm{N}]\!]_\rho)$

(m) $[\![\mathsf{Elim}_\Sigma(xy.f, c)]\!]_\rho = [\![f]\!]_{\rho;x\leftarrow([\![c]\!]_\rho\cdot\mathbf{1});y\leftarrow([\![c]\!]_\rho\cdot\mathbf{2})}$

(n) $[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho = (\exists \cdot [\![\mathrm{A}]\!]_\rho)\cdot[\![\lambda x.\mathrm{B}]\!]_\rho$

(o) $[\![(\diamond, b)]\!]_\rho = [\![b]\!]_\rho$

(p) $[\![\mathsf{Elim}_\exists(y.f, c)]\!]_\rho = [\![f]\!]_{\rho;y\leftarrow[\![c]\!]_\rho}$

(q) $[\![\{x : \mathrm{A} \mid \mathrm{B}\}]\!]_\rho = (\sigma \cdot [\![\mathrm{A}]\!]_\rho)\cdot[\![\lambda x.\mathrm{B}]\!]_\rho$

(r) $[\![\lambda f x.\, f\, x]\!]_\rho \leq [\![\lambda x.\, x]\!]_\rho$

FIGURE 9. Axiomes pour la fonction d'interprétation, la relation d'ordre et la traduction de paires

Axiomes de $\mathscr{T}$, El

(1) si $t \in \mathscr{T}$ et $t \leq t'$ alors $t' \in \mathscr{T}$

(2) si $t \in \mathscr{T}$ et $t \leq t'$ alors $\mathsf{El}(t) = \mathsf{El}(t')$

(3) si $t \in \mathscr{T}$, $a \in \mathsf{El}(t)$ et $a \leq a'$ alors $a' \in \mathsf{El}(t)$

FIGURE 10. Axiomes de l'ensemble des codes de type et de la fonction d'encodage

2. Propriétés

Nous prouvons tout d'abord que le modèle abstrait permet bien d'interpréter la relation de cumulativité et les règles de typage. Puis nous montrons comment ce modèle permet bien de déduire la cohérence de ICC_Σ.

2.1. Interprétation des termes et du calcul.

2.1.1. Proposition (Propriétés de l'interprétation)**.** *L'interprétation* $[\![]\!] : \Lambda_{ICC_\Sigma} \times \boldsymbol{Val}_{\mathscr{M}}$ *satisfait les propriétés suivantes :*

(1) si $\rho(x) = \rho'(x)$ *pour tout* $x \in FV(\mathrm{M})$*, alors* $[\![\mathrm{M}]\!]_\rho = [\![\mathrm{M}]\!]_{\rho'}$

(2) $[\![\mathrm{M}[x/\mathrm{N}]]\!]_\rho = [\![\mathrm{M}]\!]_{\rho;x\leftarrow[\![\mathrm{N}]\!]_\rho}$

(3) si $\mathrm{M} \to_{\beta\mathsf{I}} \mathrm{M}'$ *alors* $[\![\mathrm{M}]\!]_\rho = [\![\mathrm{M}']\!]_\rho$

Axiomes des opérateurs de types

(4) si $\Pi tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\mathsf{El}(\Pi tu) = \{f \in \mathscr{M} \mid \forall a \in \mathsf{El}(t),\ fa \in \mathsf{El}(ua)\}$

(5) si $\forall tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\mathsf{El}(\forall tu) = \{b \in \mathscr{M} \mid \forall a \in \mathsf{El}(t),\ b \in \mathsf{El}(ua)\} = \bigcap_{a \in \mathsf{El}(t)} \mathsf{El}(ua)$

(6) si $\Sigma tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\Sigma tu = \mathsf{Gen}(\{\mathsf{Pair}(a,b) \in \mathscr{M} \mid a \in \mathsf{El}(t), b \in \mathsf{El}(ua)\})$

(7) si $\exists tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\exists tu = \mathsf{Gen}(\bigcup_{a \in \mathsf{El}(t)} \mathsf{El}(ua))$

(8) si $\sigma tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\mathsf{El}(\sigma tu) = \{a \in \mathscr{M} \mid a \in \mathsf{El}(t) \wedge \mathsf{El}(ua) \neq \emptyset\}$

FIGURE 11. Axiomes des opérateurs de types

Axiomes de Gen

(9) pour tout $\mathrm{X} \subset \mathscr{M}$, si $\mathsf{Gen}(\mathrm{X}) \in \mathscr{T}$ alors $\mathrm{X} \subset \mathsf{El}(\mathsf{Gen}(\mathrm{X}))$

(10) si $\mathrm{X} \subset \mathrm{X}'$ et si $\mathsf{Gen}(\mathrm{X}), \mathsf{Gen}(\mathrm{X}') \in \mathscr{T}$, alors $\mathsf{El}(\mathsf{Gen}(\mathrm{X})) \subset \mathsf{El}(\mathsf{Gen}(\mathrm{X}'))$.

(11) si $\Pi(\mathsf{Gen}(\mathrm{X}))u \in \mathscr{T}$ alors

$$\{f \in \mathscr{M} \mid \forall a \in \mathrm{X}, fa \in \mathsf{El}(ua)\} \subset \mathsf{El}(\Pi(\mathsf{Gen}(\mathrm{X}))u)$$

FIGURE 12. Axiomes de la fonction de génération de codes de type

Axiomes des codes des univers

(12) $\mathsf{El}(\mathsf{u}_i) \subset \mathscr{T}$ pour tout $i \in \omega$

(13) $\mathsf{u}_i \in \mathsf{El}(\mathsf{u}_{i+1})$ pour tout $i \in \omega$

(14) $\mathsf{El}(\mathsf{u}_i) \subset \mathsf{El}(\mathsf{u}_{i+1})$ pour tout $i \in \omega$

(15) si $t \in \mathsf{El}(\mathsf{u}_i)$ et $ua \in \mathsf{El}(\mathsf{u}_i)$ pour tout $a \in \mathsf{El}(t)$, alors $\Pi tu, \forall tu, \Sigma tu, \exists tu \in \mathsf{El}(\mathsf{u}_i)$, pour tout $i \in \omega$

(16) si $t \in \mathsf{El}(\mathsf{u}_i)$ et $ua \in \mathsf{El}(\mathsf{u}_0)$ pour tout $a \in \mathsf{El}(t)$, alors $\Pi tu, \forall tu \in \mathsf{El}(\mathsf{u}_0)$ pour tout $i \in \omega$.

(17) si $t \in \mathsf{El}(\mathsf{u}_i)$ et $ua \in \mathsf{El}(\mathsf{u}_j)$ pour tout $a \in \mathsf{El}(t)$, alors $\sigma tu \in \mathsf{El}(\mathsf{u}_i)$

FIGURE 13. Axiomes des codes des univers

(4) si $M \to_\eta M'$ *alors* $[\![M]\!]_\rho \leq [\![M']\!]_\rho$

(5) si $M \twoheadrightarrow_{\beta\iota\eta} M'$ *alors* $[\![M]\!]_\rho \leq [\![M']\!]_\rho$

(6) si $M \cong M'$ *alors il existe un objet* $c \in \mathscr{M}$ *tel que* $[\![M]\!]_\rho \leq c$ *et* $[\![M']\!]_\rho \leq c$.

(7) si $\rho(x) \leq \rho'(x)$ *pour tout* $x \in FV(M)$, *alors* $[\![M]\!]_\rho \leq [\![M]\!]_{\rho'}$

DÉMONSTRATION.

(1) Par induction sur la structure de M. En utilisant les axiomes de l'application $[\![]\!]$, le résultat est immédiat sauf pour l'abstraction et les constructeurs de type. [1] Pour ces cas, il suffit alors d'utiliser l'axiome (h) pour conclure.

(2) Par induction structurelle sur M. Pour l'abstraction et les constructeurs de type, nous utilisons l'axiome (h) avec $[\![M]\!]_{\rho;x\leftarrow[\![N]\!]_\rho;z\leftarrow c}$ et $[\![M[x/N]]\!]_{\rho;z\leftarrow c}$, où z est la variable abstraite et $c \in \mathscr{M}$ quelconque. Les cas des éliminateurs de la somme dépendante et du type existentiel utilisent le point (1). Les autres cas ne posent pas de difficulté.

(3) Par induction sur M. Pour les cas de base ($\beta\iota$-réduction de tête), nous appliquons le point (2), ainsi que les axiomes (g) pour une β-réduction et (b) pour une ι_Σ-réduction. [2] Les cas inductifs se traitent de manière analogue à la preuve du point (1).

(4) (Nous reprenons la preuve donnée par Miquel dans sa thèse en page 120.) Si $M \to_\eta M'$, il existe un terme N et un contexte à trou C[] tel que $M = C[\lambda x.Nx]$ et $M' = C[N]$, avec $x \notin FV(N)$. Soit z une variable fraîche n'apparaissant (libre ou liée) ni dans M, ni dans M'. Nous avons

$$\begin{array}{rcccccc} (\lambda z.C[zN])\,(\lambda f x.fx) & \to_\beta & C[(\lambda f x.fx)N] & \to_\beta & C[\lambda x.Nx] & = & M \\ (\lambda z.C[zN])\,(\lambda x.x) & \to_\beta & C[(\lambda x.x)N] & \to_\beta & C[N] & = & M'. \end{array}$$

Nous en déduisons, d'après le point (3), la réflexivité de $\leq$ et les axiomes (a), (i) et (r)

$$[\![M]\!]_\rho \;=\; [\![\lambda z.C[zN]]\!]_\rho \cdot [\![\lambda f x.fx]\!]_\rho \;\leq\; [\![\lambda z.C[zN]]\!]_\rho \cdot [\![\lambda x.x]\!]_\rho \;=\; [\![M']\!]_\rho.$$

(5) Conséquence des points (3) et (4) avec une récurrence immédiate et par transitivité de $\leq$.

(6) D'après la propriété de Church-Rosser, nous avons un terme N tel que $M \twoheadrightarrow_{\beta\iota\eta} N \twoheadleftarrow_{\beta\iota\eta} M'$. Nous concluons alors grâce au point (5) et en posant $c = [\![N]\!]_\rho$.

(7) Nous reprenons la preuve de Miquel. (Lemme 3.3.10 p. 120.) Il suffit de remarquer que $(\lambda x_1 \ldots x_n.M)\,x_1 \ldots x_n \twoheadrightarrow_\beta M$, où $x_1, \ldots, x_n$ sont les variables libres de M. Nous concluons alors en appliquant le point (3), puisque l'interprétation d'un terme clos est indépendante de la valuation (conséquence du point (1)).

□

2.1.2. NOTATION (Interprétation d'un terme clos). Si $M \in \Lambda_{ICC_\Sigma}$ est clos, la valeur de $[\![M]\!]_\rho$ ne dépend pas de ρ (point (1 de la proposition précédente). Nous la notons $[\![M]\!]$.

2.1.3. REMARQUE. Du point 2 de la proposition préçédente et de l'axiome g, nous déduisons immédiatement $[\![M[x/N]]\!]_\rho = [\![\lambda x.M]\!]_\rho \cdot [\![N]\!]_\rho$.

Interprétation de la relation de cumulativité.

2.1.4. Lemme (Interprétation de la cumulativité restreinte)**.** *Soient* M *et* M' *deux termes du calcul implicite tels que* $M \leqslant M'$. *Pour toute valuation* $\rho \in \boldsymbol{Val}_{\mathscr{M}}$, *si* $[\![M]\!]_\rho, [\![M']\!]_\rho \in \mathscr{T}$ *alors* $El([\![M]\!]_\rho) \subset El([\![M']\!]_\rho)$.

DÉMONSTRATION. Par induction sur la dérivation de $M \leqslant M'$ en considérant la dernière règle.

1. En effet dans les axiomes f, j, k , n et q, le codomaine apparaît dans le terme de droite entouré d'une abstraction et n'est donc pas un sous-terme strict du terme initial.

2. Le point (2) seul suffit pour une $\iota_\exists$-réduction.

- (CumR-Eq) : trivial.
- (CumR-Sort-1) et (CumR-Sort-2) : conséquence des axiomes (d), (e) et (14) avec une récurrence immédiate.
- (CumR-Prod) et (CumR-I-Prod) : Si

$$\frac{\mathrm{T}_2 \leqslant \mathrm{T}_1 \qquad \mathrm{U}_1 \leqslant \mathrm{U}_2}{\Pi x:\mathrm{T}_1.\mathrm{U}_1 \leqslant \Pi x:\mathrm{T}_2.\mathrm{U}_2 \text{ ou } \forall x:\mathrm{T}_1.\mathrm{U}_1 \leqslant \forall x:\mathrm{T}_2.\mathrm{U}_2}$$

  Pour $i \in \{1,2\}$, nous posons $t_i = [\![\mathrm{T}_i]\!]_\rho$ et $u_i = [\![\lambda x.\mathrm{U}_i]\!]_\rho$. Nous avons alors (axiomes (c), (g)) $\Pi t_i u_i = [\![\Pi x:\mathrm{T}_i.\mathrm{U}_i]\!]_\rho$, $\forall t_i u_i = [\![\forall x:\mathrm{T}_i.\mathrm{U}_i]\!]_\rho$. De plus, par hypothèse, $\Pi t_i u_i, \forall t_i u_i \in \mathscr{T}$. Enfin, d'après les axiomes (4a) et (5a), nous avons $t_i \in \mathscr{T}$, et, pour tout $a \in \mathsf{El}(t_i)$, $u_i a \in \mathscr{T}$.

  Montrons que $\mathsf{El}(\Pi t_1 u_1) \subset \mathsf{El}(\Pi t_2 u_2)$ et $\mathsf{El}(\forall t_1 u_1) \subset \mathsf{El}(\forall t_2 u_2)$. Par hypothèses d'induction, nous avons $\mathsf{El}(t_2) \subset \mathsf{El}(t_1)$ et $\mathsf{El}(u_1 a) = \mathsf{El}([\![\mathrm{U}_1]\!]_{\rho;x\leftarrow a}) \subset \mathsf{El}([\![\mathrm{U}_2]\!]_{\rho;x\leftarrow a}) = \mathsf{El}(u_2 a)$ pour tout $a \in \mathsf{El}(t_2)$. Les axiomes (4b) et (5b), permettent alors de conclure.
- (CumR-Σ), (CumR-U) et (CumR-Sub) :

  Si

$$\frac{\mathrm{A}_1 \leqslant \mathrm{A}_2 \qquad \mathrm{B}_1 \leqslant \mathrm{B}_2}{\Sigma x:\mathrm{A}_1.\mathrm{B}_1 \leqslant \Sigma x:\mathrm{A}_2.\mathrm{B}_2 \text{ ou } \exists x:\mathrm{A}_1.\mathrm{B}_1 \leqslant \exists x:\mathrm{A}_2.\mathrm{B}_2 \text{ ou } \{x:\mathrm{A}_1 \mid \mathrm{B}_1\} \leqslant \{x:\mathrm{A}_2 \mid \mathrm{B}_2\}}$$

  Pour $i \in \{1,2\}$, nous posons $a_i = [\![\mathrm{A}_i]\!]_\rho$ et $b_i = [\![\lambda x.\mathrm{B}_i]\!]_\rho$. Nous avons alors (axiomes (k), (n) et (q)) $\Sigma a_i b_i = [\![\Sigma x:\mathrm{A}_i.\mathrm{B}_i]\!]_\rho$, $\exists a_i b_i = [\![\exists x:\mathrm{A}_i.\mathrm{B}_i]\!]_\rho$, $\sigma a_i b_i = [\![\{x:\mathrm{A}_i \mid \mathrm{B}_i\}]\!]_\rho$. De plus, par hypothèse, $\Sigma a_i b_i, \exists a_i b_i, \sigma a_i b_i \in \mathscr{T}$. Enfin, d'après les axiomes (6a), (7a) et (8a), $a_i \in \mathscr{T}$, et, pour tout $c \in \mathsf{El}(a_i)$, $b_i c \in \mathscr{T}$.

  Montrons que $\mathsf{El}(\Sigma a_1 b_1) \subset \mathsf{El}(\Sigma a_2 b_2)$, $\mathsf{El}(\exists a_1 b_1) \subset \mathsf{El}(\exists a_2 b_2)$, et $\mathsf{El}(\sigma a_1 b_1) \subset \mathsf{El}(\sigma a_2 b_2)$. Par hypothèses d'induction, nous avons $\mathsf{El}(a_1) \subset \mathsf{El}(a_2)$ et, pour tout $c \in \mathsf{El}(a_1)$, $\mathsf{El}(b_1 c) = \mathsf{El}([\![\mathrm{B}_1]\!]_{\rho;x\leftarrow c}) \subset \mathsf{El}([\![\mathrm{B}_2]\!]_{\rho;x\leftarrow c}) = \mathsf{El}(b_2 c)$. Les axiomes (6b), (10), (7b), et (8b) permettent alors de conclure.

□

2.1.5. Proposition (Interprétation de la relation de cumulativité)**.** *Soient* M *et* M′ *deux termes du calcul implicite tels que* $\mathrm{M} \preceq \mathrm{M}'$*. Pour toute valuation* $\rho \in \boldsymbol{Val}_{\mathscr{M}}$*, si* $[\![\mathrm{M}]\!]_\rho, [\![\mathrm{M}']\!]_\rho \in \mathscr{T}$ *alors* $\mathsf{El}([\![\mathrm{M}]\!]_\rho) \subset \mathsf{El}([\![\mathrm{M}']\!]_\rho)$.

DÉMONSTRATION. D'après le point (2) du lemme 2.3.10 page 16, il existe M_0 et M'_0 tels que $\mathrm{M} \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}_0$, $\mathrm{M}' \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}'_0$ et $\mathrm{M}_0 \leqslant \mathrm{M}'_0$. D'après le point (5) de la proposition 2.1.1, nous avons $[\![\mathrm{M}]\!]_\rho \leq [\![\mathrm{M}_0]\!]_\rho$ et $[\![\mathrm{M}]\!]'_\rho \leq [\![\mathrm{M}'_0]\!]_\rho$. L'axiome (1) nous indique que $[\![\mathrm{M}_0]\!]_\rho, [\![\mathrm{M}'_0]\!]_\rho \in \mathscr{T}$ et l'axiome (2) que $\mathsf{El}([\![\mathrm{M}]\!]_\rho) = \mathsf{El}([\![\mathrm{M}_0]\!]_\rho)$ et $\mathsf{El}([\![\mathrm{M}']\!]_\rho) = \mathsf{El}([\![\mathrm{M}'_0]\!]_\rho)$. Nous concluons alors en appliquant le lemme 2.1.4 à M_0 et M'_0. □

2.2. Interprétation des règles de typage.

2.2.1. DÉFINITION (Interprétation des contextes). Soit Γ un contexte du calcul implicite. L'interprétation de ce contexte, notée $[\![\Gamma]\!]$ est définie par :

$$[\![\Gamma]\!] \quad = \quad \{\rho \in \mathbf{Val}_{\mathscr{M}} \mid \forall (x:\mathrm{T}) \in \Gamma, ([\![\mathrm{T}]\!]_\rho \in \mathscr{T}) \wedge \rho(x) \in \mathsf{El}([\![\mathrm{T}]\!]_\rho)\}.$$

2.2.2. REMARQUE. Pour le contexte vide •, $[\![\bullet]\!] = \mathbf{Val}_{\mathscr{M}}$.

2.2.3. Lemme (Interprétation des sous-contextes)**.** *Si* $\Gamma \subseteq \Gamma'$ *alors* $[\![\Gamma']\!] \subset [\![\Gamma]\!]$.

DÉMONSTRATION. Immédiat d'après les définitions du sous-contexte (définition 3.1.5 page 19) et de l'interprétation des contextes. □

2.2.4. Proposition (Validité)**.** *Si* $\Gamma \vdash M : T$ *est un jugement dérivable du calcul implicite, alors :*

$$\rho \in [\![\Gamma]\!] \quad \Rightarrow \quad [\![T]\!]_\rho \in \mathscr{T} \wedge [\![M]\!]_\rho \in \mathsf{El}([\![T]\!]_\rho).$$

DÉMONSTRATION. Par induction sur la structure de la dérivation du jugement $\Gamma \vdash M : T$. La plupart des règles se traite immédiatement. Nous ne détaillons pas les utilisations des axiomes de la figure 9 quand elles sont évidentes.

- (VAR) : immédiat par définition de $[\![\Gamma]\!]$.
- (SORT) : conséquence de l'axiome 13.
- (CUM) : immédiat d'après la proposition 2.1.5.
- Règles de formation des types : Si

$$\frac{\Gamma \vdash T : s_1 \qquad \Gamma; x : T \vdash U : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\square}{\Gamma \vdash \square x : T.U : s_3}$$

  - Nous avons clairement $[\![s_3]\!]_\rho \in \mathscr{T}$.
  - Pour montrer que $[\![\square x : T.U]\!]_\rho \in \mathsf{El}([\![s_3]\!]_\rho)$, nous distinguons les cas suivants :
    - — type sous-ensemble : si $\square x : T.U = \{x : T \mid U\}$, alors nous utilisons l'axiome (17) ;
    - — produit dépendant imprédicatif : si $\square x : T.U$ est un type produit et si $s_3 = \mathsf{Prop}$ alors nous utilisons l'axiome (16) ;
    - — autres cas (produits dépendants non-imprédicatifs, somme dépendante ou type existentiel) : alors nous utilisons les axiomes (15) et (14).
- Règles d'introduction des types produit. (LAM) et (GEN) sont similaires. Traitons (LAM). Si

$$\frac{\Gamma; x : T \vdash M : U \quad \Gamma \vdash \Pi x : T.U : s}{\Gamma \vdash \lambda x.M : \Pi x : T.U}\,(\text{LAM})$$

  montrons que $[\![\Pi x : T.U]\!]_\rho \in \mathscr{T}$ et que $[\![\lambda x.M]\!]_\rho \in \mathsf{El}([\![\Pi x : T.U]\!]_\rho)$.
  - Montrons que $[\![\Pi x : T.U]\!]_\rho \in \mathscr{T}$. Il existe $i \in \omega$ tel que $[\![s]\!]_\rho = \mathsf{u}_i$. Par hypothèse d'induction sur le jugement $\Gamma \vdash \Pi x : T.U : s$, $[\![\Pi x : T.U]\!]_\rho \in \mathsf{El}(\mathsf{u}_i)$. Par l'axiome (12), nous avons $[\![\Pi x : T.U]\!]_\rho \in \mathscr{T}$.
  - Montrons que $[\![\lambda x.M]\!]_\rho \in \mathsf{El}([\![\Pi x : T.U]\!]_\rho)$. D'après l'axiome (4b), il suffit de montrer que, pour $a \in \mathsf{El}([\![T]\!]_\rho)$ quelconque, $[\![\lambda x.M]\!]_\rho \cdot a \in \mathsf{El}([\![\lambda x.U]\!]_\rho \cdot a)$. Cela est équivalent, d'après l'axiome (g), à $[\![M]\!]_{\rho; x \leftarrow a} \in \mathsf{El}([\![U]\!]_{\rho; x \leftarrow a})$. Nous montrons sans difficulté que $\rho; x \leftarrow a \in [\![\Gamma; x : T]\!]$ et en déduisons le résultat voulu par hypothèse d'induction sur $\Gamma; x : T \vdash M : U$ avec la valuation $\rho; x \leftarrow a$.
- Règles d'élimination des types produit. Les cas (APP) et (INST) sont eux aussi analogues. Traitons (INST). Si

$$\frac{\Gamma \vdash M : \forall x : T.U \quad \Gamma \vdash N : T}{\Gamma \vdash M : U\,[x/N]}\,(\text{INST})$$

  montrons que $[\![U\,[x/N]]\!]_\rho \in \mathscr{T}$ et que $[\![M]\!]_\rho \in \mathsf{El}([\![U\,[x/N]]\!]_\rho)$.

  Par hypothèses d'induction, $[\![\forall x : T.U]\!]_\rho \in \mathscr{T}$, $[\![M]\!]_\rho \in \mathsf{El}([\![\forall x : T.U]\!]_\rho)$ et $[\![N]\!]_\rho \in \mathsf{El}([\![T]\!]_\rho)$. D'après l'axiome (5a), $[\![\lambda x.U]\!]_\rho \cdot [\![N]\!]_\rho \in \mathscr{T}$ et d'après l'axiome (5b), $[\![M]\!]_\rho \in \mathsf{El}([\![\lambda x.U]\!]_\rho \cdot [\![N]\!]_\rho)$. En appliquant la remarque 2.1.3, nous avons $[\![U\,[x/N]]\!]_\rho = [\![\lambda x.U]\!]_\rho \cdot [\![N]\!]_\rho$, d'où $[\![U\,[x/N]]\!]_\rho \in \mathscr{T}$ et $[\![M]\!]_\rho \in \mathsf{El}([\![U\,[x/N]]\!]_\rho)$.
- Règles d'introduction des types somme. Ces règles sont similaires. Si

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B\,[x/a] \quad \Gamma \vdash \exists x : A.B : s}{\Gamma \vdash (\diamond, b) : \exists x : A.B}\,(\exists\text{-I})$$

- Nous montrons $[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho \in \mathscr{T}$ par hypothèse d'induction sur $\Gamma \vdash \exists x : \mathrm{A}.\mathrm{B} : s$ (similaire à (Lam)).
- Montrons que $[\![(\diamond, b)]\!]_\rho \in \mathsf{El}([\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho)$. Par hypothèses d'induction et d'après la remarque 2.1.3, nous avons $[\![a]\!]_\rho \in \mathsf{El}([\![\mathrm{A}]\!]_\rho)$ et $[\![b]\!]_\rho \in \mathsf{El}([\![\lambda x.\mathrm{B}]\!]_\rho \cdot [\![a]\!]_\rho)$, d'où
$$[\![(\diamond, b)]\!]_\rho \in \bigcup_{c \in \mathsf{El}([\![\mathrm{A}]\!]_\rho)} \mathsf{El}([\![\lambda x.\mathrm{B}]\!]_\rho \cdot c)$$
D'après les axiomes (9) et (7b),
$$[\![(\diamond, b)]\!]_\rho \in \mathsf{El}(\mathsf{Gen}(\bigcup_{c \in \mathsf{El}([\![\mathrm{A}]\!]_\rho)} \mathsf{El}([\![\lambda x.\mathrm{B}]\!]_\rho \cdot c))) = \mathsf{El}([\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho)$$

- (Σ-E) et (∃-E) sont similaires. Traitons (∃-E) : Si
$$\frac{\begin{array}{ll}\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s & \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y) \\ \Gamma \vdash c : \exists x : \mathrm{A}.\mathrm{B} & x \notin \mathsf{FV}(f)\end{array}}{\Gamma \vdash \mathsf{Elim}_\exists(y.f, c) : \mathrm{P}\,c}\ (\exists\text{-E})$$
  - Montrons tout d'abord que $[\![\mathrm{P}\,c]\!]_\rho \in \mathscr{T}$. Il existe $i \in \omega$ tel que $[\![s]\!]_\rho = \mathsf{u}_i$. Par hypothèses d'induction sur les jugements $\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s$ et $\Gamma \vdash c : \exists x : \mathrm{A}.\mathrm{B}$ et d'après l'axiome 4b, nous avons $[\![\mathrm{P}\,c]\!]_\rho \in \mathsf{El}(\mathsf{u}_i)$. Nous concluons alors grâce à l'axiome 12.
  - Montrons maintenant que $[\![\mathsf{Elim}_\exists(y.f, c)]\!]_\rho \in \mathsf{El}([\![\mathrm{P}\,c]\!]_\rho)$. Nous avons $[\![\mathsf{Elim}_\exists(y.f, c)]\!]_\rho = [\![f]\!]_{\rho; y \leftarrow [\![c]\!]_\rho} = [\![\lambda y.f]\!]_\rho \cdot [\![c]\!]_\rho$ et $\mathsf{El}([\![\mathrm{P}\,c]\!]_\rho) = \mathsf{El}([\![\mathrm{P}]\!]_\rho \cdot [\![c]\!]_\rho)$ Or, par hypothèse d'induction, $[\![c]\!]_\rho \in \mathsf{El}([\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho)$. Pour montrer que $[\![\lambda y.f]\!]_\rho \cdot [\![c]\!]_\rho \in \mathsf{El}([\![\mathrm{P}]\!]_\rho \cdot [\![c]\!]_\rho)$, il suffit donc de montrer, d'après l'axiome (4b), que $\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\mathrm{P}]\!]_\rho \in \mathscr{T}$ et que $[\![\lambda y.f]\!]_\rho \in \mathsf{El}(\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\mathrm{P}]\!]_\rho)$.
    — Montrons que $\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\mathrm{P}]\!]_\rho \in \mathscr{T}$. Il existe $j \in \omega$ tel que $[\![s']\!]_\rho = \mathsf{u}_j$. Par hypothèses d'induction sur $\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s$ et $\Gamma \vdash \exists x : \mathrm{A}.\mathrm{B} : s'$, nous avons $[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho \in \mathsf{El}(\mathsf{u}_j)$ et $[\![\mathrm{P}]\!]_\rho \in \mathsf{El}(\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\lambda y.s]\!]_\rho)$. Soit $b \in \mathsf{El}([\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho)$. Nous avons $[\![\mathrm{P}]\!]_\rho \cdot b \in \mathsf{El}([\![s]\!]_\rho) = \mathsf{El}(\mathsf{u}_i)$ (axiome (4b)). Soit k le maximum de i et j. Nous avons alors (axiome (14)) $[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho \in \mathsf{El}(\mathsf{u}_k)$ et, pour tout $b \in \mathsf{El}([\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho)$, $[\![\mathrm{P}]\!]_\rho \cdot b \in \mathsf{El}(\mathsf{u}_k)$. Nous concluons alors grâce à l'axiome (15).
    — Montrons enfin que $[\![\lambda y.f]\!]_\rho \in \mathsf{El}(\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\mathrm{P}]\!]_\rho)$.
      — D'après l'axiome (11), et puisque $\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\mathrm{P}]\!]_\rho \in \mathscr{T}$, nous avons :
$$\{u \in \mathscr{M} \mid \forall b \in \bigcup_{a \in \mathsf{El}([\![\mathrm{A}]\!]_\rho)} \mathsf{El}([\![\lambda x.\mathrm{B}]\!]_\rho \cdot a),\ u \cdot b \in \mathsf{El}([\![\mathrm{P}]\!]_\rho \cdot b)\} \subset \mathsf{El}(\Pi[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho [\![\mathrm{P}]\!]_\rho).$$
      — Soient $a \in \mathsf{El}([\![\mathrm{A}]\!]_\rho)$ et $b \in \mathsf{El}([\![\lambda x.\mathrm{B}]\!]_\rho \cdot a)$. Il suffit donc de montrer que $[\![\lambda y.f]\!]_\rho \cdot b = [\![f]\!]_{\rho; y \leftarrow b} \in \mathsf{El}([\![\mathrm{P}]\!]_\rho \cdot b)$.
      — Soit $\tilde{\rho} = \rho; x \leftarrow a; y \leftarrow b$. Il est clair que $\tilde{\rho} \in [\![\Gamma; x : \mathrm{A}; y : \mathrm{B}]\!]$. Nous concluons alors par hypothèse d'induction sur le jugement $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,y$ avec la valuation $\tilde{\rho}$, puisque $x \notin \mathsf{FV}(f)$ et que $x, y \notin \mathsf{FV}(\mathrm{P})$.

- (Sub-E-1) : Si
$$\frac{\Gamma \vdash a : \{x : \mathrm{A} \mid \mathrm{B}\}}{\Gamma \vdash a : \mathrm{A}}\ (\text{Sub-E-1})$$
nous avons immédiatement, d'après les axiomes (8a) et (8b), $[\![\mathrm{A}]\!]_\rho \in \mathscr{T}$ et $[\![a]\!]_\rho \in [\![\mathrm{A}]\!]_\rho$.

- (Sub-E-2) : Si
$$\frac{\begin{array}{ll}\Gamma \vdash \mathrm{P} : s & \Gamma; y : \mathrm{B}\,[x/c] \vdash f : \mathrm{P} \\ \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} & y \notin \mathsf{FV}(f)\end{array}}{\Gamma \vdash f : \mathrm{P}}\ (\text{Sub-E-2})$$

- $[\![\mathrm{P}]\!]_\rho \in \mathscr{T}$ est une conséquence immédiate de l'axiome (12) et de l'hypothèse d'induction liée au jugement $\Gamma \vdash \mathrm{P} : s$.
- Montrons que $[\![f]\!]_\rho \in \mathsf{El}([\![\mathrm{P}]\!]_\rho)$. Par hypothèse d'induction, $[\![\{x : \mathrm{A} \mid \mathrm{B}\}]\!]_\rho \in \mathscr{T}$ et $[\![c]\!]_\rho \in \mathsf{El}([\![\{x : \mathrm{A} \mid \mathrm{B}\}]\!]_\rho)$. D'après les axiomes (8a) et (8b), $[\![\lambda x.\mathrm{B}]\!]_\rho \cdot [\![c]\!]_\rho = [\![\mathrm{B}[x/c]]\!]_\rho \in \mathscr{T}$ et il existe $b \in \mathsf{El}([\![\mathrm{B}[x/c]]\!]_\rho)$. Posons $\tilde{\rho} = \rho; y \leftarrow b$. Nous avons $\tilde{\rho} \in [\![\Gamma; y : \mathrm{B}[x/c]]\!]$, d'où, par hypothèse d'induction, $[\![f]\!]_{\tilde{\rho}} \in \mathsf{El}([\![\mathrm{P}]\!]_{\tilde{\rho}})$. Nous pouvons alors conclure puisque y n'est une variable libre ni de f, ni de P.

□

2.2.5. REMARQUE. Cette preuve montre que le modèle abstrait est aussi valable pour une version de ICC_Σ avec un type existentiel complètement implicite. En effet, du fait de la définition de l'interprétation pour la paire existentielle et l'éliminateur d'existentielle, le cas des règles d'un type existentiel implicite :

$$\frac{\Gamma \vdash a : \mathrm{A} \qquad \Gamma \vdash b : \mathrm{B}[x/a] \qquad \Gamma \vdash \exists x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash b : \exists x : \mathrm{A}.\mathrm{B}}$$

$$\frac{\begin{array}{cc}\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s & \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\, y \\ \Gamma \vdash c : \exists x : \mathrm{A}.\mathrm{B} & x \notin \mathsf{FV}(f)\end{array}}{\Gamma \vdash f\,[y/c] : \mathrm{P}\, c}$$

correspond exactement aux cas des règles (∃-I) et (∃-E).

2.3. Cohérence du calcul. De même que dans ICC, la cohérence logique de ICC_Σ peut être définie comme le fait que le terme $\forall \mathrm{A} : \mathsf{Prop}.\mathsf{Prop}$ n'est pas habité dans le contexte vide.

En effet, les remarques suivantes, dues à Miquel, sont également valables dans ICC_Σ.

- Si $\Gamma \vdash \forall x : \mathrm{T}.\mathrm{U} \to \Pi x : \mathrm{T}.\mathrm{U} : s$ est dérivable alors $\Gamma \vdash \lambda p x.\, p : \forall x : \mathrm{T}.\mathrm{U} \to \Pi x : \mathrm{T}.\mathrm{U}$ est dérivable. Nous avons donc en particulier $\vdash \lambda p x.\, p : \forall \mathrm{A} : \mathsf{Prop}.\mathrm{A} \to \Pi \mathrm{A} : \mathsf{Prop}.\mathrm{A}$.
- Nous avons $\vdash \lambda f.\, f\,(\forall \mathrm{A} : \mathsf{Prop}.\mathrm{A}) : \Pi \mathrm{A} : \mathsf{Prop}.\mathrm{A} \to \forall \mathrm{A} : \mathsf{Prop}.\mathrm{A}$.

2.3.1. DÉFINITION (Modèle restreint). Soit $i \in \omega$. Un modèle du calcul implicite avec types somme $(\mathscr{M}, \cdot, \mathsf{Pair}, [\![\,]\!], \leq, \mathscr{T}, \mathsf{El}, \mathsf{Gen}, \mathbf{1}, \mathbf{2}, \Pi, \forall, \Sigma, \exists, \sigma, (\mathsf{u}_i)_{i\in\omega})$ est *i-restreint* s'il existe $t \in \mathsf{El}(\mathsf{u}_i)$ tel que $\mathsf{El}(t) = \emptyset$.

2.3.2. Proposition. *Notons* $\mathsf{False}_0 = \forall \mathrm{T} : \mathsf{Prop}.\mathrm{T}$ *et, pour* $i > 0$, $\mathsf{False}_i = \forall \mathrm{T} : \mathsf{Type}_i.\mathrm{T}$. *Si* $\mathscr{M}$ *est un modèle* i*-restreint de* ICC_Σ *alors* $\mathsf{El}([\![\mathsf{False}_i]\!]) = \emptyset$.

DÉMONSTRATION.

- Nous avons $\vdash \mathsf{False}_0 : \mathsf{Prop}$ et, pour tout $i > 0$, $\Gamma \vdash \mathsf{False}_j : \mathsf{Type}_i$.
- D'après la proposition 2.2.4, et l'axiome (12), nous avons pour tout j $[\![\mathsf{False}_j]\!] \in \mathscr{T}$.
- Nous concluons alors en appliquant l'axiome (5).

□

2.3.3. Corollaire (Cohérence de ICC_Σ)**.** *S'il existe un modèle* 0*-cohérent de* ICC_Σ *alors* ICC_Σ *est cohérent.*

DÉMONSTRATION. D'après la proposition précédente, nous avons $\mathsf{El}([\![\mathsf{False}_0]\!]) = \emptyset$. Si ICC_Σ est incohérent, alors il existe un terme clos M tel que $\vdash \mathrm{M} : \mathsf{False}_0$. Nous avons donc, d'après la proposition 2.2.4, $[\![\mathrm{M}]\!] \in \mathsf{El}([\![\mathsf{False}_0]\!])$ ce qui contredit $\mathsf{El}([\![\mathsf{False}_0]\!]) = \emptyset$. □

Chapitre 4

Notions sur les espaces cohérents

Le modèle de cohérence de Miquel s'inspire du modèle construit par Jean-Yves Girard pour interpréter le Système F [**Gir86, GTL89, Gir06**]. Girard construit un domaine de Scott [**Sco76, Sco82**] vérifiant une certaine équation récursive. Pour construire ce domaine, Girard introduit les espaces cohérents, qui le mèneront ensuite vers la découverte de la logique linéaire. Il définit ainsi la catégorie des espaces cohérents, dont les morphismes sont les fonctions stables, introduites par Berry [**Ber78**]. Girard montre que c'est une catégorie cartésienne fermée réflexive [**ML98**], ce qui lui permet ainsi d'établir l'équation récursive.

Miquel reprend les ingrédients du modèle de Girard : un domaine de Scott associé à un espace cohérent, une catégorie cartésienne fermée et réflexive, et une équation récursive vérifiée par l'espace. Par rapport au Système F, ICC possède un système de types plus riche et une hiérarchie d'univers. Étendre le modèle de Girard au produit dépendant et au produit implicite ne demande pas de changements profonds. Par contre, tenir compte de la hiérarchie d'univers nécessite des ajouts substantiels. Ainsi Miquel définit les fonctions μ-stables, qui généralisent les fonctions stables, et construit une hiérarchie d'espaces cohérents rigidement inclus les uns dans les autres.

Dans ce chapitre, nous regroupons les définitions et propriétés du modèle de Miquel dont nous avons besoin pour la construction du modèle. Ainsi la première section est consacrée aux espaces cohérents. La deuxième section présente la catégorie des quasi-espaces cohérents, dont les morphismes sont les fonctions quasi-stables, et qui est une catégorie cartésienne fermée mais non-réflexive. La troisième section présente la catégorie des μ-espaces cohérents, dont les morphismes sont les fonctions μ-stables, et justifie que c'est une catégorie cartésienne fermée *et* réflexive.

Ce chapitre peut-être vu comme un résumé du chapitre 4 de la thèse de Miquel [**Miq01**] lisible de manière autonome. Les preuves des résultats repris sont en général omises, notamment les plus longues. Quelques notations sont changées. Les explications sont plus brèves.

Notons tout de même une différence, même si non-essentielle, entre notre présentation et celle de Miquel. La définition des espaces cohérents par Girard dans [**Gir06**] diffère légèrement de celle, plus ancienne, dans [**GTL89**]. Dans [**GTL89**], un espace cohérent est défini comme un ensemble d'ensembles ordonné par l'inclusion (avec une structure de domaine) et en bijection avec un graphe non-ordonné. Ainsi un espace cohérent sera vu à la fois comme un graphe ou un domaine. Dans [**Gir06**], une distinction est faite entre les deux notions : un espace cohérent est cette fois défini comme un graphe et le domaine associé est appelé domaine de cohérence. La thèse de Miquel, écrite avant [**Gir06**], suit naturellement la présentation de [**GTL89**]. Nous avons décidé de suivre la présentation la plus récente et de distinguer les espaces de cohérence de leurs domaines associés.

1. Espaces cohérents

1.1. Généralités. Dans ce paragraphe, nous donnons les définitions des espaces cohérents et de leurs domaines associés, les domaines cohérents. Nous montrons que la donnée d'un espace cohérent est équivalente à la donnée de son domaine de cohérence associé.

1.1.1. DÉFINITION (Espace cohérent). Un *espace cohérent* $\mathscr{A}$ est un couple $(|\mathscr{A}|, \frown_{\mathscr{A}})$ où

- $|\mathscr{A}|$ est un ensemble, appelé la *trame*,
- $\frown_{\mathscr{A}}$ est une relation binaire réflexive et symétrique sur $|\mathscr{A}|$, appelée *relation de cohérence*.

Les éléments de $|\mathscr{A}|$ sont appelés *atomes*. Deux atomes α et α' sont *cohérents* si $\alpha \frown_{\mathscr{A}} \alpha'$ et *incohérents* sinon.

Un espace cohérent est donc un cas particulier de graphe.

1.1.2. DÉFINITION (Clique). Soit $\mathscr{A}$ un espace cohérent. Un ensemble d'atomes a est une clique si pour tous $\alpha, \beta \in a$, $\alpha \frown_{\mathscr{A}} \beta$.

1.1.3. DÉFINITION (Famille de cliques compatibles). Soient I un ensemble et $(a_i)_{i\in I}$ est une famille de cliques de $\mathscr{A}$. $(a_i)_{i\in I}$ est *une famille de cliques compatibles dans* $\mathscr{A}$ si $\bigcup_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}}$.

1.1.4. DÉFINITION (Domaine de cohérence). Soit $\mathscr{A}$ un espace cohérent. Le *domaine de cohérence* associé à $\mathscr{A}$, noté $\mathsf{D}_{\mathscr{A}}$, est l'ensemble des cliques de $\mathscr{A}$.

1.1.5. REMARQUE. L'utilisation de l'appellation domaine est justifiée par le fait que les domaines de cohérence sont des domaines de Scott.

Nous allons maintenant caractériser les domaines de cohérence, puis montrer que connaître un espace de cohérence équivaut à connaître son domaine de cohérence associé.

1.1.6. DÉFINITION (Ordre partiel). Un *ordre partiel* est un couple $(\mathsf{D}, \leq)$ formé d'un ensemble D et d'une relation d'ordre sur cet ensemble $\leq$.

1.1.7. Proposition. *Soient $\mathscr{A}$ un espace cohérent et $\mathsf{D}_{\mathscr{A}}$ le domaine associé à $\mathscr{A}$. Alors $(\mathsf{D}_{\mathscr{A}}, \subset)$ est un ordre partiel qui vérifie les critères suivants :*

(1) (Clôture inférieure) si $a \in \mathsf{D}_{\mathscr{A}}$ et $a' \subset a$ alors $a' \in \mathsf{D}_{\mathscr{A}}$

(2) (Complétude binaire) si I est un ensemble et si $(a_i)_{i\in I}$ est une famille de cliques de $\mathscr{A}$ (i.e. d'éléments de $\mathsf{D}_{\mathscr{A}}$), alors

$$(\forall i, j \in I, a_i \cup a_j \in \mathsf{D}_{\mathscr{A}}) \quad \Rightarrow \quad \bigcup_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}}.$$

DÉMONSTRATION.

- $(\mathsf{D}_{\mathscr{A}}, \subset)$ est clairement un ordre partiel.
- (1) est évident car un sous-ensemble est bien une clique.
- Montrons (2). Soient $\alpha, \beta \in \bigcup_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}}$. Montrons que $\alpha \frown_{\mathscr{A}} \beta$.
  - Il existe $i, j \in I$ tels que $\alpha \in a_i$ et $\beta \in a_j$.
  - Par hypothèse, nous avons $a_i \cup a_j \in \mathsf{D}_{\mathscr{A}}$, d'où $\alpha \frown_{\mathscr{A}} \beta$.

□

1.1.8. Proposition. *Soit $(\mathsf{D}, \leq)$ un ordre partiel vérifiant les propriétés de complétude binaire et de clôture inférieure. L'espace de cohérence $\mathscr{A}$ dont la trame est $\{\alpha \mid \{\alpha\} \in \mathsf{D}\}$ et dont la relation réflexive et symétrique est définie par $\alpha \frown_{\mathscr{A}} \alpha' \Leftrightarrow \{\alpha, \alpha'\} \in \mathsf{D}$ est tel que $\mathsf{D}_{\mathscr{A}} = \mathsf{D}$.*

DÉMONSTRATION. Si $a \in \mathsf{D}$, montrons que $a \in \mathsf{D}_{\mathscr{A}}$. Il suffit de montrer que tous les éléments de a sont cohérents pour $\frown_{\mathscr{A}}$. Soient $\alpha, \alpha' \in a$. Par clôture inférieure, nous avons $\{\alpha, \alpha'\} \in \mathsf{D}$, soit $\alpha \frown_{\mathscr{A}} \alpha'$.

Si $a \in \mathsf{D}_{\mathscr{A}}$, montrons que $a \in \mathsf{D}$. Nous avons $a = \bigcup_{\alpha\in a} \{\alpha\}$. Si $\alpha, \alpha' \in a$, par cohérence, nous avons $\{\alpha, \alpha'\} \in \mathsf{D}$. Par complétude binaire, nous avons bien $a \in \mathsf{D}$. □

1.2. Espaces cohérents particuliers. Nous décrivons dans ce paragraphe quelques exemples d'espaces cohérents utiles pour la suite : le produit direct, qui servira de produit cartésien dans les catégorie des quasi- et des μ-espaces cohérents (cf. sections 2 et 3), la somme directe et les espaces cohérents plats, qui apparaîtront dans l'équation récursive au chapitre suivant (cf. sous-section 2.2).

Produit direct.

1.2.1. DÉFINITION (Produit direct). Soient $\mathscr{A}_1$,$\mathscr{A}_2$ des espaces cohérents. Le *produit direct* de $\mathscr{A}_1$ et $\mathscr{A}_2$, noté $\mathscr{A}_1 \,\&\, \mathscr{A}_2$, est l'espace cohérent dont la trame est

$$\begin{aligned} |\mathscr{A}_1 \,\&\, \mathscr{A}_2| &= \{1\} \times |\mathscr{A}_1| \cup \{2\} \times |\mathscr{A}_2| \\ &= \{(1,\alpha_1) \mid \alpha_1 \in |\mathscr{A}_1|\} \cup \{(2,\alpha_2) \mid \alpha_2 \in |\mathscr{A}_2|\} \end{aligned}$$

et dont la relation de cohérence est :

$$(i,\alpha) \frown_{\mathscr{A}_1 \& \mathscr{A}_2} (j,\alpha') \quad \Leftrightarrow \quad (i = j \wedge \alpha \frown_{\mathscr{A}_i} \alpha') \vee (i \neq j).$$

1.2.2. REMARQUE. Pour toute clique a de $\mathscr{A}_1 \,\&\, \mathscr{A}_2$, il existe a_1, a_2 cliques de $\mathscr{A}_1$ et $\mathscr{A}_2$ telles que $a = \{1\} \times a_1 \cup \{2\} \times a_2$. Cela justifie la notation suivante.

1.2.3. NOTATION. Nous notons $[a_1, a_2]$ la clique $\{1\} \times a_1 \cup \{2\} \times a_2$.

1.2.4. REMARQUE (Cliques compatibles dans $\mathscr{A}_1 \& \mathscr{A}_2$). Si I est un ensemble non-vide et si $(a_i)_{i\in \mathrm{I}}$ est une clique compatible de $\mathscr{A}_1 \,\&\, \mathscr{A}_2$, alors, pour $j \in \{1,2\}$ il existe une famile $(a_i^j)_{i\in \mathrm{I}}$ de cliques compatibles dans $\mathscr{A}_j$ telle que, pour tout $i \in \mathrm{I}$, $a_i = [a_i^1, a_i^2]$. Nous avons de plus

$$\bigcap_{i\in \mathrm{I}} a_i = \bigcap_{i\in \mathrm{I}} [a_i^1, a_i^2] = [\bigcap_{i\in \mathrm{I}} a_i^1, \bigcap_{i\in \mathrm{I}} a_i^2]$$

Somme directe.

1.2.5. DÉFINITION (Somme directe). Soient $\mathscr{A}_1$,$\mathscr{A}_2$ des espaces cohérents. La *somme directe* de $\mathscr{A}_1$ et $\mathscr{A}_2$, noté $\mathscr{A}_1 \oplus \mathscr{A}_2$, est l'espace cohérent dont la trame est

$$|\mathscr{A}_1 \oplus \mathscr{A}_2| \quad = \quad \{1\} \times |\mathscr{A}_1| \cup \{2\} \times |\mathscr{A}_2|$$

et dont la relation de cohérence est :

$$(i,\alpha) \frown_{\mathscr{A}_1 \oplus \mathscr{A}_2} (j,\alpha') \quad \Leftrightarrow \quad (i = j \wedge \alpha \frown_{\mathscr{A}_i} \alpha')$$

1.2.6. REMARQUE. Les cliques de $\mathscr{A}_1 \oplus \mathscr{A}_2$ sont de la forme $\{i\} \times a_i$ où $i \in \{1,2\}$ et a_i est une clique de $\mathscr{A}_i$.

Définitions simplifiées. Si les trames de $\mathscr{A}_1$ et $\mathscr{A}_2$ sont disjointes, nous pouvons définir plus simplement le produit direct $\mathscr{A}_1 \,\&\, \mathscr{A}_2$ et la somme directe $\mathscr{A}_1 \oplus \mathscr{A}_2$.

1.2.7. DÉFINITION. Si $|\mathscr{A}_1| \cap |\mathscr{A}_2| = \emptyset$ alors nous définissons $\mathscr{A}_1 \,\&\, \mathscr{A}_2$ et $\mathscr{A}_1 \oplus \mathscr{A}_2$ par :

- $|\mathscr{A}_1 \,\&\, \mathscr{A}_2| = |\mathscr{A}_1 \oplus \mathscr{A}_2| = |\mathscr{A}_1| \cup |\mathscr{A}_2|$
- $$\begin{aligned} \alpha \frown_{\mathscr{A}_1 \& \mathscr{A}_2} \alpha' \quad \Leftrightarrow \quad & (\exists i \in \{1,2\}, \alpha \frown_{\mathscr{A}_i} \alpha') \\ & \vee (\exists i, j \in \{1,2\}, i \neq j \wedge \alpha \in |\mathscr{A}_i|, \alpha' \in |\mathscr{A}_j|) \end{aligned}$$
- $$\alpha \frown_{\mathscr{A}_1 \oplus \mathscr{A}_2} \alpha' \quad \Leftrightarrow \quad (\exists i \in \{1,2\}, \alpha \frown_{\mathscr{A}_i} \alpha')$$

Nous avons alors $\mathsf{D}_{\mathscr{A}_1} \cap \mathsf{D}_{\mathscr{A}_2} = \{\emptyset\}$, $\mathsf{D}_{\mathscr{A}_1 \& \mathscr{A}_2} = \{a \cup b \mid a \in \mathsf{D}_{\mathscr{A}_1}, b \in \mathsf{D}_{\mathscr{A}_2}\}$, $\mathsf{D}_{\mathscr{A}_1 \oplus \mathscr{A}_2} = \mathsf{D}_{\mathscr{A}_1} \cup \mathsf{D}_{\mathscr{A}_2}$.

Espaces cohérent plats. Un espace cohérent plat est caractérisé par le fait que les atomes sont deux à deux incohérents. Ses seules cliques sont donc des singletons ou l'ensemble vide.

1.2.8. DÉFINITION (Espace cohérent plat). Soit X un ensemble. L'*espace cohérent plat*, noté $X_\perp$, est défini par

(1) $|X_\perp| = X$

(2) $\forall x, y \in X, x \frown_{X_\perp} y \Leftrightarrow x = y$.

1.3. Plongements rigides.

1.3.1. DÉFINITION (Plongement rigide). Soient $\mathscr{A}$,$\mathscr{A}'$ des espaces cohérents, un *plongement rigide de $\mathscr{A}$ vers $\mathscr{A}'$* est une injection $\Phi : |\mathscr{A}| \to |\mathscr{A}'|$ telle que pour tous atomes $\alpha_1, \alpha_2 \in |\mathscr{A}|$, nous avons

$$\alpha_1 \frown_{\mathscr{A}} \alpha_2 \quad \Leftrightarrow \quad \Phi(\alpha_1) \frown_{\mathscr{A}'} \Phi(\alpha_2)$$

Deux fonctions sont associées à tout plongement rigide. Le *plongement* Φ^+ défini par

$$\Phi^+ \quad : \quad \begin{array}{ccc} \mathsf{D}_{\mathscr{A}} & \to & \mathsf{D}_{\mathscr{A}'} \\ a & \mapsto & \{\Phi(\alpha) \mid \alpha \in a\} \end{array}$$

et la *rétraction* Φ^- définie par

$$\Phi^- \quad : \quad \begin{array}{ccc} \mathsf{D}_{\mathscr{A}'} & \to & \mathsf{D}_{\mathscr{A}} \\ a' & \mapsto & \{\alpha \mid \Phi(\alpha) \in a'\} \end{array}$$

1.3.2. REMARQUE (Correction des définitions du plongement et de la rétraction). Du fait de la contrainte entre les relations de cohérence pour toute clique a de $\mathscr{A}$, $\Phi^+(a)$ est bien une clique de $\mathscr{A}'$ et de même, pour toute clique a' de $\mathscr{A}$, $\Phi^-(a)$ est bien une clique de $\mathscr{A}$. Les fonctions Φ^+ et Φ^- sont donc bien définies. De plus, Φ^+ est injective, Φ^- est surjective et $\Phi^- \circ \Phi^+ = \mathsf{id}_{\mathsf{D}_{\mathscr{A}}}$. Si Φ est bijective, $\Phi^+ \circ \Phi^- = \mathsf{id}_{\mathsf{D}_{\mathscr{A}'}}$.

1.3.3. DÉFINITION (Inclusion rigide). Soient $\mathscr{A}$,$\mathscr{A}'$ des espaces cohérents, $\mathscr{A}$ est *inclus rigidement* dans $\mathscr{A}'$, ou encore $\mathscr{A}$ est un *sous-espace cohérent* de $\mathscr{A}'$ si $|\mathscr{A}| \subset |\mathscr{A}'|$ et si les relations de cohérence sont identiques sur $|\mathscr{A}|$, c'est-à-dire que, pour tous atomes $\alpha_1, \alpha_2 \in |\mathscr{A}|$, nous avons

$$\alpha_1 \frown_{\mathscr{A}} \alpha_2 \quad \Leftrightarrow \quad \alpha_1 \frown_{\mathscr{A}'} \alpha_2$$

Nous notons alors $\mathscr{A} \Subset \mathscr{A}'$.

Nous déduisons immédiatement cette proposition qui montre comment la notion d'inclusion rigide est reliée à celle de plongement rigide.

1.3.4. Proposition. *Soient $\mathscr{A}$,$\mathscr{A}'$ des espaces cohérents tels que $|\mathscr{A}| \subset |\mathscr{A}'|$. Alors $\mathscr{A} \Subset \mathscr{A}'$ si et seulement si l'injection canonique*

$$\Phi \quad : \quad \begin{array}{ccc} |\mathscr{A}| & \to & |\mathscr{A}'| \\ \alpha & \mapsto & \alpha \end{array}$$

est un plongement rigide entre $\mathscr{A}$ et $\mathscr{A}'$.

Dans ce cas le plongement et la rétraction sont respectivement les fonctions $a \mapsto a$ et $a' \mapsto a' \cap |\mathscr{A}|$ et nous avons de plus $\mathsf{D}_{\mathscr{A}} \subset \mathsf{D}_{\mathscr{A}'}$.

Colimite. La notion d'inclusion rigide nous permet d'introduire celle de colimite, utile pour la construction du modèle.

1.3.5. DÉFINITION (Ensemble dirigé). Un ordre partiel $(I, \leq)$ est un ensemble dirigé si pour tous $i, j \in I$ il existe $k \in I$ tel que $i \leq k, j \leq k$.

1.3.6. DÉFINITION (Famille croissante d'espaces cohérents). Soit $(I, \leq)$ un ensemble dirigé. Une famille d'espaces cohérents $(\mathscr{A}_i)_{i\in I}$ est une *famille croissante d'espaces cohérents* si pour tous $i, j \in I$ tels que $i \leq j$ alors $\mathscr{A}_i \Subset \mathscr{A}_j$.

1.3.7. DÉFINITION (Colimite). Soient $(I, \leq)$ est un ensemble dirigé et si $(A_i)_{i\in I}$ est une famille croissante d'espaces cohérents. La *colimite* de la famille $(A_i)_{i\in I}$ est l'espace cohérent défini par

$$\begin{array}{rcl} |\operatorname*{colim}_{i\in I}\mathscr{A}_i| & = & \bigcup_{i\in I}|\mathscr{A}_i| \\ \alpha_1 \frown_{\operatorname{colim}_{i\in I}\mathscr{A}_i} \alpha_2 & \Leftrightarrow & \exists i \in I, \alpha_1, \alpha_2 \in |\mathscr{A}_i| \wedge \alpha_1 \frown_{\mathscr{A}_i} \alpha_2 \end{array}$$

2. La catégorie des quasi-espaces cohérents

La catégorie des quasi-espaces cohérents est la catégorie dont les objets sont les espaces cohérents et dont les morphismes sont les (traces des) fonctions quasi-stables. C'est une catégorie cartésienne fermée non-réflexive.

2.1. Fonctions quasi-stables.

2.1.1. DÉFINITION (Fonctions quasi-stables). Soient $\mathscr{A}$,$\mathscr{B}$ des espaces cohérents et I un ensemble. Une fonction $F : D_{\mathscr{A}} \to D_{\mathscr{B}}$ est *quasi-stable* si, pour toute famille $(a_i)_{i\in I}$ de cliques de $D_{\mathscr{A}}$, nous avons

$$\bigcup_{i\in I} a_i \in D_{\mathscr{A}} \quad \Rightarrow \quad F(\bigcap_{i\in I} a_i) = \bigcap_{i\in I} F(a_i)$$

2.1.2. REMARQUE. Ce critère de quasi-stabilité est plus contraignant que le critère de stabilité binaire des fonctions stables. [1] Cependant Miquel a montré dans sa thèse que toute fonction stable est quasi-stable. Cela implique notamment qu'il est possible de remplacer le critère de stabilité binaire par le critère de quasi-stabilité dans la définition des fonctions stables.

La quasi-stabilité préserve certaines bonnes propriétés des fonctions stables. Ainsi, nous allons le voir, les fonctions quasi-stables sont elles aussi représentables par leur trace.

2.1.3. DÉFINITION (Trace d'une fonction quasi-stable). La *trace* d'une fonction quasi-stable $F : D_{\mathscr{A}} \to D_{\mathscr{B}}$, notée $\mathsf{Tr}(F)$, est l'ensemble défini par :

$$\mathsf{Tr}(F) \quad = \quad \{(a_0, \beta) \in D_{\mathscr{A}} \times |\mathscr{B}| \mid \beta \in F(a_0) \wedge (\forall a \subsetneq a_0, \beta \notin F(a))\}$$

La proposition suivante démontre que connaître la trace d'une fonction quasi-stable permet d'en déduire les valeurs.

2.1.4. Proposition (Application)**.** *Si* $F : D_{\mathscr{A}} \to D_{\mathscr{B}}$ *est quasi-stable alors, pour toute clique* $a \in D_{\mathscr{A}}$

$$F(a) \quad = \quad \{\beta \in |\mathscr{B}| \mid \exists a_0 \subset a, (a_0, \beta) \in \mathsf{Tr}(F)\}$$

DÉMONSTRATION. Cf. proposition 4.3.5 page 157 dans la thèse de Miquel. □

2.1.5. REMARQUE. Si $\mathsf{Tr}(F) \subset \mathsf{Tr}(G)$, alors pour tout $a \in D_{\mathscr{A}}$, $F(a) \subset G(a)$.

1. Si $a_1 \cup a_2 \in D_{\mathscr{A}}$, alors $F(a_1 \cup a_2) = F(a_1) \cup F(a_2)$.

Nous déduisons alors dans le corollaire suivant qu'une fonction quasi-stable peut être assimilée à sa trace.

2.1.6. Corollaire (Caractérisation par la trace)**.** *Si* $F,G : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}}$ *sont des fonctions quasi-stables, alors*

$$\mathsf{Tr}(F) = \mathsf{Tr}(G) \quad \Leftrightarrow \quad F = G$$

DÉMONSTRATION. Cf. corollaire 4.3.6 page 157 dans la thèse de Miquel. L'implication directe se déduit de la proposition précédente. La réciproque découle immédiatement de la définition de la trace. □

2.2. Espace fonctionnel. Nous définissons maintenant l'espace fonctionnel de la catégorie.

2.2.1. DÉFINITION (Espace des fonctions quasi-stables). Soient $\mathscr{A}$,$\mathscr{B}$ des espaces cohérents. L'*espace cohérent des fonctions quasi-stables* de $\mathscr{A}$ vers $\mathscr{B}$, noté $\mathscr{A} \xrightarrow{q} \mathscr{B}$, est l'espace cohérent dont la trame est

$$|\mathscr{A} \xrightarrow{q} \mathscr{B}| \quad = \quad \mathsf{D}_{\mathscr{A}} \times |\mathscr{B}|$$

et dont la relation de cohérence $(a_1,\beta_1) \frown_{\mathscr{A} \xrightarrow{q} \mathscr{B}} (a_2,\beta_2)$ est caractérisée par la conjonction des conditions suivantes :

(1) si $a_1 \cup a_2 \in \mathsf{D}_{\mathscr{A}}$ alors $\beta_1 \frown_{\mathscr{B}} \beta_2$;

(2) si $a_1 \cup a_2 \in \mathsf{D}_{\mathscr{A}}$ et $a_1 \neq a_2$ alors $\beta_1 \neq \beta_2$.

2.2.2. REMARQUES.

(1) La relation $\frown_{\mathscr{A} \xrightarrow{q} \mathscr{B}}$ est bien réflexive et symétrique, ce qui justifie la définition.

(2) L'espace $\mathscr{A} \xrightarrow{q} \mathscr{B}$ est l'équivalent de l'espace cohérent $\mathscr{A} \to \mathscr{B}$ pour la catégorie des espaces cohérents définie par Girard.

La proposition suivante affirme que les cliques de cet espace sont les traces des fonctions quasi-stables.

2.2.3. Proposition (Caractérisation du domaine associé à $\mathscr{A} \xrightarrow{q} \mathscr{B}$)**.** *Le domaine associé à l'espace cohérent* $\mathscr{A} \xrightarrow{q} \mathscr{B}$ *est*

$$\begin{aligned} \mathsf{D}_{\mathscr{A} \xrightarrow{q} \mathscr{B}} &= \{c \in \mathsf{D}_{\mathscr{A}} \times |\mathscr{B}| \mid \exists F : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}} \textit{ quasi-stable}, c = \mathsf{Tr}(F)\} \\ &= \{\mathsf{Tr}(F) \mid F : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}} \textit{ quasi-stable}\} \end{aligned}$$

DÉMONSTRATION. Cf. proposition 4.3.8 p :158 de la thèse de Miquel. □

2.2.4. REMARQUE. Puisque les fonctions quasi-stables sont totalement caractérisées par leur trace (cf. corollaire 2.1.6), nous pouvons définir les morphismes de la catégorie de deux manières équivalentes. Les morphismes entre les espaces cohérents $\mathscr{A}$ et $\mathscr{B}$ sont les fonctions quasi-stables entre $\mathsf{D}_{\mathscr{A}}$ et $\mathsf{D}_{\mathscr{B}}$ ou bien les cliques de l'espace fonctionnel $\mathscr{A} \xrightarrow{q} \mathscr{B}$.

2.2.5. NOTATION. Puisque les cliques de $\mathscr{A} \xrightarrow{q} \mathscr{B}$ peuvent êtres vues comme des fonctions, nous noterons dorénavant $F : \mathscr{A} \xrightarrow{q} \mathscr{B}$ pour signifier qu'une fonction $F : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}}$ est quasi-stable.

Miquel a montré que la catégorie des quasi-espaces cohérents est cartésienne fermée non-réflexive (proposition 4.3.10 page 158). La non-réflexivité empêche cette catégorie d'être le cadre pour la construction d'un espace cohérent vérifiant une équation récursive. [2]

2. Ainsi la catégorie des espaces cohérents liée aux fonctions stables du modèle de Girard pour Système F est une catégorie cartésienne fermée réflexive

Pour pallier à la non-réflexivité de la catégorie des quasi-espaces cohérents, Miquel définit, pour un cardinal μ donné, la μ-continuité (une généralisation de la continuité), puis les fonctions μ-stables (les fonctions quasi-stables et μ-continues). La catégorie induite, la catégorie des μ-espaces cohérents, est alors cartésienne fermée *et* réflexive.

3. La catégorie des μ-espaces cohérents

3.1. Fonctions μ-stables. Nous considérons dans la suite un cardinal μ, qui a vocation à être très grand (au moins égal à $\aleph_0$) afin de pouvoir coder la hiérarchie d'univers du calcul implicite.

3.1.1. NOTATION (Cardinal). Pour tout ensemble E, son *cardinal* est noté $\overline{\overline{\mathrm{E}}}$.

3.1.2. DÉFINITION (μ-approximants). Soit $\mathscr{A}$ un espace cohérent et a une clique de $\mathscr{A}$. Un μ-*approximant* de a est une clique $a_0 \subset a$ telle que $\overline{\overline{a_0}} < \mu$. L'ensemble des μ-approximants de a est noté $\mathscr{P}_\mu(a)$.

3.1.3. DÉFINITION (μ-continuité). Soient $\mathscr{A}$,$\mathscr{B}$ des espaces cohérents. Une fonction $\mathrm{F} : \mathrm{D}_{\mathscr{A}} \to \mathrm{D}_{\mathscr{B}}$ est μ-*continue* si elle est monotone[3] et si, pour tout clique $a \in \mathrm{D}_{\mathscr{A}}$, $\mathrm{F}(a) = \bigcup_{a_0 \in \mathscr{P}_\mu(a)} \mathrm{F}(a_0)$.

3.1.4. REMARQUE. Le deuxième critère implique la monotonie.

3.1.5. DÉFINITION (μ-stabilité). Soient $\mathscr{A}$,$\mathscr{B}$ des espaces cohérents. Une fonction $\mathrm{F} : \mathrm{D}_{\mathscr{A}} \to \mathrm{D}_{\mathscr{B}}$ est μ-*stable* si elle est μ-continue et quasi-stable.

3.1.6. REMARQUE. Si $\mu = \aleph_0$, nous retrouvons les notions de continuité à la Scott et de stabilité à la Berry. Ainsi une fonction $\aleph_0$-stable est une fonction stable.

3.1.7. REMARQUE. La quasi-stabilité entraîne la monotonie : si $a \subset a'$, alors $a \cup a' = a' \in \mathrm{D}_{\mathscr{A}}$ et $\mathrm{F}(a) = \mathrm{F}(a \cap a') = \mathrm{F}(a) \cap \mathrm{F}(a') \subset \mathrm{F}(a')$. Ainsi une fonction F est μ-stable si et seulement si elle est quasi-stable et si pour tout clique $a \in \mathrm{D}_{\mathscr{A}}$ $\mathrm{F}(a) = \bigcup_{a_0 \in \mathscr{P}_\mu(a)} \mathrm{F}(a_0)$.

3.1.8. Proposition (Caractérisation des fonctions μ-stables)**.** *Soient $\mathscr{A}$,$\mathscr{B}$ des espaces cohérents. Une fonction quasi-stable $\mathrm{F} : \mathscr{A} \xrightarrow{q} \mathscr{B}$ est μ-stable si et seulement si sa trace n'est constituée que d'atomes $(a,\beta) \in \mathrm{D}_{\mathscr{A}} \times |\mathscr{B}|$ tels que $\overline{\overline{a}} < \mu$.*

DÉMONSTRATION. Cf. proposition 4.3.16 p :160 de la thèse de Miquel. Conséquence de la définition de la trace. □

3.1.9. REMARQUES.

(1) Dans le cas de fonctions quasi-stables, les cliques a ne sont pas bornées. Cette non-limitation est à la base de la non-réflexivité de la catégorie des quasi-espaces cohérents.

(2) Dans le cas de fonctions stables ($\mu = \aleph_0$), les cliques a sont finies. Cela rend les fonctions stables inadaptées pour interpréter le calcul implicite du fait de la hiérarchie infinie d'univers. Dans le cadre des fonctions μ-stables, nous pourrons interpréter les univers en prenant des cardinaux suffisamment grands.

3.2. Espace fonctionnel. Nous pouvons maintenant définir l'espace des fonctions μ-stables, similaire à celui des fonctions quasi-stables, à ceci près que le cardinal des atomes est borné par le cardinal μ.

3.2.1. NOTATION. Nous notons $\mathrm{D}^{\mu}_{\mathscr{A}} = \{a \in \mathrm{D}_{\mathscr{A}} \mid \overline{\overline{a}} < \mu\}$ l'ensemble des cliques de $\mathscr{A}$ de cardinal strictement inférieur à μ.

3. Pour toutes cliques $a_0, a_1 \in \mathrm{D}_{\mathscr{A}}$, si $a_0 \subset a_1$ alors $\mathrm{F}(a_0) \subset \mathrm{F}(a_1)$.

3.2.2. DÉFINITION (Espace des fonctions μ-stables). Soient $\mathscr{A},\mathscr{B}$ des espaces cohérents. L'*espace cohérent des fonctions* μ*-stables* de $\mathscr{A}$ vers $\mathscr{B}$, noté $\mathscr{A} \xrightarrow{\mu} \mathscr{B}$, est l'espace cohérent dont la trame est

$$|\mathscr{A} \xrightarrow{\mu} \mathscr{B}| \quad = \quad \mathsf{D}^{\mu}_{\mathscr{A}} \times |\mathscr{B}|$$

et dont la relation de cohérence $\frown_{\mathscr{A}\xrightarrow{\mu}\mathscr{B}}$ est caractérisée par

$$(a_1,\beta_1) \frown_{\mathscr{A}\xrightarrow{\mu}\mathscr{B}} (a_2,\beta_2) \quad \Leftrightarrow \quad (a_1,\beta_1) \frown_{\mathscr{A}\xrightarrow{q}\mathscr{B}} (a_2,\beta_2)$$

3.2.3. REMARQUE. Nous avons $|\mathscr{A} \xrightarrow{\mu} \mathscr{B}| \subset |\mathscr{A} \xrightarrow{q} \mathscr{B}|$. et $\frown_{\mathscr{A}\xrightarrow{\mu}\mathscr{B}} = \frown_{\mathscr{A}\xrightarrow{q}\mathscr{B}}$, ce qui implique $\mathscr{A} \xrightarrow{\mu} \mathscr{B} \Subset \mathscr{A} \xrightarrow{q} \mathscr{B}$.

3.2.4. Proposition (Caractérisation du domaine associé à $\mathscr{A} \xrightarrow{\mu} \mathscr{B}$)**.** *Le domaine associé à l'espace cohérent* $\mathscr{A} \xrightarrow{\mu} \mathscr{B}$ *est*

$$\begin{aligned} \mathsf{D}_{\mathscr{A}\xrightarrow{\mu}\mathscr{B}} &= \{c \in \mathsf{D}^{\mu}_{\mathscr{A}} \times |\mathscr{B}| \mid \exists \mathrm{F} : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}} \ \mu\text{-stable}, c = \mathsf{Tr}(\mathrm{F})\} \\ &= \{\mathsf{Tr}(\mathrm{F}) \mid \mathrm{F} : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}} \ \mu\text{-stable}\} \end{aligned}$$

DÉMONSTRATION. Cf. corollaire 4.3.17 p :160 de la thèse de Miquel. Conséquence directe de la caractérisation des fonctions μ-stables (proposition 3.1.8 ci-haut). □

La composée de fonctions μ-continues (resp. μ-stables) n'est pas toujours une fonction μ-continue (resp. μ-stable) si μ est quelconque. Aussi allons nous faire des hypothèses sur la nature du cardinal.

3.2.5. DÉFINITION (Cardinal régulier). Un cardinal infini μ est *régulier* si et seulement si pour toute famille d'ensembles $(\mathrm{E}_i)_{i\in\mathrm{I}}$,

$$\bar{\bar{\mathrm{I}}} < \mu \wedge (\forall i \in \mathrm{I}, \overline{\overline{\mathrm{E}_i}} < \mu) \quad \Rightarrow \quad \overline{\overline{\bigcup_{i\in\mathrm{I}} \mathrm{E}_i}} < \mu$$

Nous supposons par la suite que μ est régulier. Dans ces conditions, la catégorie des μ-espaces cohérents[4], est une catégorie cartésienne fermée. Miquel le démontre dans la proposition 4.3.19 p.161.[5] La preuve de la fermeture repose sur le fait que l'application et la curryfication, définies ci-dessous, sont des fonctions μ-stables.

3.2.6. DÉFINITION (Application). L'*application*, notée **App** est définie par :

$$\begin{array}{rccc} & \mathsf{D}_{\mathscr{A}\&(\mathscr{A}\xrightarrow{\mu}\mathscr{B})} & \to & \mathsf{D}_{\mathscr{B}} \\ \mathbf{App}_{\mathscr{A},\mathscr{B}} : & & & \\ & (a, \mathsf{Tr}(\mathrm{F})) & \mapsto & \mathrm{F}(a) \end{array}$$

3.2.7. DÉFINITION (Isomorphisme de Curry). L'*isomorphisme de Curry*, noté **Curry** est défini par :

$$\begin{array}{rccc} & \mathsf{D}_{(\mathscr{A}\&\mathscr{B})\xrightarrow{\mu}\mathscr{C}} & \to & \mathsf{D}_{\mathscr{A}\xrightarrow{\mu}(\mathscr{B}\xrightarrow{\mu}\mathscr{C})} \\ \mathbf{Curry}_{\mathscr{A},\mathscr{B},\mathscr{C}} : & & & \\ & \mathsf{Tr}(\mathrm{F}) & \mapsto & \mathsf{Tr}(a \mapsto (b \mapsto \mathrm{F}((a,b)))) \end{array}$$

4. Ses objets sont les domaines de cohérence et ses morphismes les cliques des espaces de fonction (c'est-à-dire les traces des fonctions μ-stables).

5. La démonstration renvoie en fait à celle de la proposition 4.3.10 p.158 qui prouve la fermeture de la catégorie des quasi-espaces cohérents, dont les objets sont les domaines de cohérence et les morphismes sont les cliques des espaces de fonction quasi-stables.

Miquel a de plus montré que cette catégorie est *réflexive*,[6] c'est-à-dire qu'il existe un espace cohérent $\mathscr{A}$ non-trivial (dont la trame a au moins deux éléments) et un plongement rigide entre $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$ et $\mathscr{A}$. Cet espace est l'espace d'interprétation du modèle. Il sera présenté plus en détail dans la section 2 du chapitre suivant.

6. Cf. la même proposition 4.3.19 p.161

Chapitre 5

Modèle de cohérence de ICC_Σ

Ce chapitre est constitué de deux sections. La première décrit l'implémentation des opérateurs représentant les constructeurs de type. La seconde décrit le modèle de cohérence dans sa totalité.

1. Interprétation des types

Cette section est consacrée aux éléments du modèle de Miquel relatifs à l'interprétation des types produit du calcul implicite. Ces éléments sont l'espace cohérent des types (cf. paragraphe 1.2) et les deux opérateurs de type liés aux produits explicite et implicite du calcul (cf. paragraphe 1.3). Ces éléments reposent fortement sur les notions de type sémantique et de base de type, que nous présentons maintenant.

1.1. Types sémantiques et bases de type. Dans cette sous-section, nous considérons un espace cohérent quelconque $\mathscr{A}$ et deux ensembles de cliques de $\mathscr{A}$ X et Y.

Base de type. Une base de type est un ensemble de cliques deux à deux incompatibles.

1.1.1. DÉFINITION (Base de type). X est une *base de type de* $\mathscr{A}$ si pour toutes cliques $a_1, a_2 \in \mathrm{X}$,

$$a_1 \cup a_2 \in \mathrm{D}_{\mathscr{A}} \quad \Rightarrow \quad a_1 = a_2$$

1.1.2. REMARQUE. Si X est une base de type de $\mathscr{A}$, toute sous-partie $\mathrm{X}' \subset \mathrm{X}$ est également une base de type de $\mathscr{A}$.

Cette définition dépend là encore *a priori* de l'espace cohérent $\mathscr{A}$ considéré. La proposition suivante montre qu'en fait cette notion est invariante par inclusion rigide.

1.1.3. Proposition (Invariance des bases de type par inclusion rigide)**.** *Soit* $\mathscr{A}'$ *un espace cohérent tel que* $\mathscr{A} \Subset \mathscr{A}'$. *Alors* X *est une base de type de* $\mathscr{A}$ *si et seulement si* X *est une base de type de* $\mathscr{A}'$.

DÉMONSTRATION. Évident car tout ensemble d'atomes de $\mathscr{A}$ est une clique de $\mathscr{A}$ si et seulement si c'est une clique de $\mathscr{A}'$. □

Types sémantiques. Avant de définir les types sémantiques, nous introduisons quelques notions auxiliaires.

1.1.4. DÉFINITION (Ensemble clos supérieurement). L'ensemble X est *clos supérieurement* si :

$$\forall a \in \mathrm{X}, \forall a' \in \mathscr{A},\ a \subset a' \Rightarrow a' \in \mathrm{X}$$

1.1.5. REMARQUE. Cette notion dépend de l'espace cohérent $\mathscr{A}$ considéré.

1.1.6. DÉFINITION (Ensemble stable par intersection de cliques compatibles). Y est *stable par intersection de cliques compatibles (dans* $\mathscr{A}$*)* si pour toute famille $(a_i)_{i\in \mathrm{I}}$ de cliques de Y compatibles dans $\mathscr{A}$, nous avons $\bigcap_{i\in \mathrm{I}} a_i \in \mathrm{Y}$.

1.1.7. DÉFINITION (Type sémantique). Un ensemble de cliques de $\mathscr{A}$ est un *type sémantique* de $\mathscr{A}$ s'il est clos supérieurement et s'il est stable par intersection de cliques compatibles.

1.1.8. REMARQUE. Contrairement à la base de type, la notion de type sémantique dépend de l'espace cohérent considéré.

Relations entre bases de type et types sémantiques. Nous allons maintenant montrer la relation entre les bases de type et les types sémantiques. Pour cela nous introduisons les notions de clôture supérieure et d'ensemble des cliques minimales.

1.1.9. DÉFINITION (Clôture supérieure). La *clôture supérieure de* X, notée $\uparrow$ X, est l'ensemble défini par :

$$\uparrow \mathrm{X} \quad = \quad \{a \in \mathscr{A} \mid \exists a_0 \in \mathrm{X},\ a_0 \subset a\}$$

1.1.10. Proposition. *La clôture supérieure de* X*, est le plus petit (au sens de l'inclusion) ensemble clos supérieurement contenant* X*.*

DÉMONSTRATION. $\uparrow$ X contient X et est clairement clos supérieurement. Il est de plus facile de vérifier que tout ensemble clos supérieurement contenant X contient également $\uparrow$ X. □

Quelques résultats immédiats :

1.1.11. Lemme.

- X *est clos supérieurement si et seulement si* X $=\uparrow$ X*;*
- *si* X $\subset$ X$'$ *alors* $\uparrow$ X $\subset\uparrow$ X$'$.

1.1.12. DÉFINITION (Ensemble des cliques minimales). L'*ensemble des cliques minimales* de X, noté $\lfloor \mathrm{X} \rfloor$, est défini par :

$$\lfloor \mathrm{X} \rfloor \quad = \quad \{c \in \mathrm{X} \mid \forall c' \in \mathrm{X}, c' \subset c \Rightarrow c = c'\}.$$

1.1.13. REMARQUE. Si X est une base de type, alors X $= \lfloor \mathrm{X} \rfloor$. La réciproque est fausse, tout ensemble de cliques minimales n'est pas une base de type : il suffit de prendre un ensemble composé de deux cliques compatibles et non-incluses l'une dans l'autre.

La proposition suivante, démontrée dans la thèse de Miquel (proposition 5.1.2 page 167), illustre les liens profonds entre bases de type et types sémantiques : la clôture supérieure de toute base de type est un type sémantique et l'ensemble des cliques minimales de tout type sémantique est une base de type.

1.1.14. Proposition (Bases de type et types sémantiques)**.**

(1) Pour toute base de type X $\subset \mathrm{D}_{\mathscr{A}}$*, sa clôture supérieure* $\uparrow$ X *est un type sémantique de* $\mathscr{A}$*.*

(2) Pour tout type sémantique Y $\subset \mathrm{D}_{\mathscr{A}}$*, l'ensemble de ses cliques minimales* $\lfloor \mathrm{Y} \rfloor$ *est une base de type de* $\mathscr{A}$*.*

(3) La correspondance X $\to\uparrow$ X *constitue une bijection entre l'ensemble des bases de type de* $\mathscr{A}$ *et l'ensemble des types sémantiques de* $\mathscr{A}$*. La correspondance* Y $\to \lfloor \mathrm{Y} \rfloor$ *est sa réciproque.*

Cette dualité entre base de type et type sémantique est fondamentale dans l'interprétation des types dans le modèle. En effet, comme nous le verrons à la section suivante, (section 2) tout type du calcul implicite sera (à encodage près) interprété par une base de type, dont le type sémantique associé contiendra les interprétations des termes ayant ce type.

À la dualité entre type sémantique et base de type s'ajoute également une correspondance entre une clique quelconque d'un type sémantique et une clique de la base de type associée.

1.1.15. Lemme (Coercition)**.** *Soit* X *une base de type de* $\mathscr{A}$*. Pour toute clique* $a \in \uparrow$ X*, il existe une et une seule clique* $a_0 \in$ X *telle que* $a_0 \subset a$*. Nous notons alors* $a_0 = \lfloor a \rfloor_{\mathrm{X}}$*.*

DÉMONSTRATION. C'est le lemme 5.1.3 p :168 de la thèse de Miquel. □

1.1.16. Corollaire. *Soient $\mathscr{A}$ un espace de cohérence et* X *une base de type de $\mathscr{A}$. Soient $a, a' \in \uparrow X$ des cliques telles que $a \subset a'$. Alors $\lfloor a \rfloor_X = \lfloor a' \rfloor_X$.*

DÉMONSTRATION. $\lfloor a \rfloor_X$ est une clique de X incluse dans a donc incluse dans a'. Par unicité de la clique minimale (lemme 1.1.15), $\lfloor a \rfloor_X = \lfloor a' \rfloor_X$. □

1.1.17. REMARQUE. Si X n'est pas une base de type, il n'y a pas unicité de la clique a_0.

1.2. Espace cohérent plat des types. Nous présentons ici l'espace cohérent qui servira dans le modèle à interpréter les types du calcul implicite.

Nous considérons un cardinal μ régulier infini. Dans notre modèle, nous restreignons le cardinal des ensembles de cliques considérés. Ainsi nous utiliserons la notion de base de type μ-finie.

1.2.1. DÉFINITION (Base de type μ-finie). Soient $\mathscr{A}$ un espace cohérent et $X \subset |\mathscr{A}|$ une base de type de $\mathscr{A}$. X est dite μ-*finie* si

(1) $\overline{\overline{X}} < \mu$

(2) $\forall a \in X, \overline{\overline{a}} < \mu$.

Parmi les bases de types, nous distinguons l'ensemble vide et le singleton ensemble vide. Nous les notons $\mathbf{0} = \emptyset$ et $\mathbf{1} = \{\emptyset\}$. Ces bases de types correspondent respectivement aux types sémantiques $\emptyset$ et $|\mathscr{A}|$. Nous les désignons par le terme de *types extrémaux*.

1.2.2. DÉFINITION (Espace cohérent des types μ-finis). Soit $\mathscr{A}$ un espace cohérent. L'*espace cohérent des types* μ-*finis* est l'espace cohérent plat dont les atomes sont de la forme $\tau(X)$, où X est une base de type μ-finie de $\mathscr{A}$ quelconque :

(1) $|\mathsf{Ty}_\mu(\mathscr{A})| = \{\tau(X) \mid X \text{ base de type } \mu\text{-finie de } \mathscr{A}\}$

(2) pour X, Y bases de type μ-finies de $\mathscr{A}$, $\tau(X) \subset_{\mathsf{Ty}_\mu(\mathscr{A})} \tau(Y) \Leftrightarrow X = Y$.

Cet espace étant plat, ses cliques sont soit l'ensemble vide $\emptyset$, soit des singletons $\{\tau(X)\}$ où X est une base de type μ-finie.

L'interprétation des types dans notre modèle doit respecter deux contraintes :

- l'interprétation d'un type doit caractériser les termes typés par le type en question ;
- tout terme du calcul, y compris si c'est un type, est interprété par une clique d'un espace cohérent.

Ainsi un type T sera interprété par une clique $\{\tau(X)\}$ et les termes typés par T seront interprétés par des cliques du type sémantique $\uparrow X$.

Nous avons également besoin de définir l'espace cohérent des types sans restriction de cardinal.

1.2.3. DÉFINITION (Espace cohérent des types). Soit $\mathscr{A}$ un espace cohérent. L'*espace cohérent des types* est l'espace cohérent plat dont les atomes sont de la forme $\tau(X)$, où X est une base de type de $\mathscr{A}$ quelconque :

(1) $|\mathsf{Ty}(\mathscr{A})| = \{\tau(X) \mid X \text{ base de type de } \mathscr{A}\}$

(2) pour X, Y bases de type μ-finies de $\mathscr{A}$, $\tau(X) \subset_{\mathsf{Ty}(\mathscr{A})} \tau(Y) \Leftrightarrow X = Y$.

Nous avons clairement $\mathsf{Ty}_\mu(\mathscr{A}) \Subset \mathsf{Ty}(\mathscr{A})$.

1.3. Intersection et produit dépendant. Nous présentons ici les opérateurs de type Π et $\forall$ qui modélisent les produits du calcul des constructions implicite. Avant cela, nous définissons des bases de type utilisées pour définir ces opérateurs.

Soient donc $\mathscr{A}$ et $\mathscr{B}$ des espaces cohérents, X une base de type de $\mathscr{A}$, et $(Y_a)_{a\in X}$ une famille de bases de type de $\mathscr{B}$ indexée par X.

Type intersection. L'intersection de types sémantiques est un type sémantique car les deux critères caractérisant les types sémantiques sont stables par intersection. Aussi nous avons :

1.3.1. Proposition (Intersection de types sémantiques)**.** *L'intersection des types sémantiques*

$$\forall_{TS}(a\in X;Y_a) \quad = \quad \bigcap_{a\in X}\uparrow Y_a$$

est un type sémantique.

Nous pouvons alors définir le type intersection, relatif au produit implicite.

1.3.2. DÉFINITION (Intersection). Le *type intersection*, noté $\forall(a\in X;Y_a)$, est la base de type associée à $\forall_{TS}(a\in X;Y_a)$:

$$\begin{aligned}\forall(a\in X;Y_a) &= \lfloor\forall_{TS}(a\in X;Y_a)\rfloor\\ &= \lfloor\bigcap_{a\in X}\uparrow Y_a\rfloor\end{aligned}$$

Type produit. Le type sémantique lié au produit dépendant est un ensemble de traces de fonctions quasi-stables. Alexandre Miquel démontre la proposition suivante (proposition 5.1.10 page 170) :

1.3.3. Proposition. *L'ensemble*

$$\begin{aligned}\Pi_{TS}(a\in X;Y_a) &= \{\mathsf{Tr}(F)\in \mathsf{D}_{\mathscr{A}\xrightarrow{q}\mathscr{B}} \mid \forall a\in X, F(a)\in\uparrow Y_a\}\\ &= \{c\in \mathsf{D}_{\mathscr{A}\xrightarrow{q}\mathscr{B}} \mid \exists F:\mathscr{A}\xrightarrow{q}\mathscr{B}, (\forall a\in X, F(a)\in\uparrow Y_a)\wedge(c=\mathsf{Tr}(F))\}\end{aligned}$$

est un type sémantique de $\mathscr{A}\xrightarrow{q}\mathscr{B}$.

1.3.4. REMARQUE. Nous avons aussi

$$\Pi_{TS}(a\in X;Y_a) \quad = \quad \{\mathsf{Tr}(F)\in \mathsf{D}_{\mathscr{A}\xrightarrow{q}\mathscr{B}} \mid \forall a\in\uparrow X, F(a)\in\uparrow Y_{\lfloor a\rfloor_X}\}$$

Cette présentation sera plus pratique pour le type somme dépendante (cf. sous-section 1.5).

Nous définissons alors le type produit dépendant.

1.3.5. DÉFINITION (Produit dépendant). Le *type produit dépendant*, noté $\Pi(a\in X;Y_a)$ est la base de type associée au type sémantique $\Pi_{TS}(a\in X;Y_a)$:

$$\Pi(a\in X;Y_a) \quad = \quad \lfloor\Pi_{TS}(a\in X;Y_a)\rfloor$$

Opérateurs de type. Nous internalisons les constructions $\forall(a\in X;Y_a)$ et $\Pi(a\in X;Y_a)$ en des opérateurs de type $\forall$ et Π qui sont des fonctions μ-stables. Ces opérateurs vont naturellement correspondre aux constantes $\forall$ et Π du modèle abstrait.

1.3.6. DÉFINITION (Opérateurs de type). Soient $\mathscr{A},\mathscr{B}$ des espaces cohérents et μ un cardinal supérieur à ω. Les opérateurs de type sont les applications suivantes

$$\begin{aligned}\Pi &: \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}\times \mathsf{D}_{\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})}\to \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A}\xrightarrow{\mu}\mathscr{B})}\\ \forall &: \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}\times \mathsf{D}_{\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})}\to \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{B})}\end{aligned}$$

définies, pour tout $t\in \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}$ et pour tout $f\in \mathsf{D}_{\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})}$, par

- $$\begin{cases} \Pi(t,f) &= \{\tau(\Pi(a\in X;Y_a))\} \\ \forall(t,f) &= \{\tau(\forall(a\in X;Y_a))\} \end{cases}$$

  s'il existe une base de type de $\mathscr{A}$ X et une famille $(Y_a)_{a\in X}$ de bases de type de $\mathscr{B}$ telles que

  (1) $t = \{\tau(X)\}$

  (2) $f = \mathsf{Tr}(F)$ et $F(a) = \{\tau(Y_a)\}$ pour tout $a \in X$.

  (3) $\begin{cases} \Pi(a\in X;Y_a) \text{ est une base de type } \mu\text{-finie de } \mathscr{A} \xrightarrow{\mu} \mathscr{B} \\ \forall(a\in X;Y_a) \text{ est une base de type } \mu\text{-finie de } \mathscr{B}. \end{cases}$

- dans tous les autres cas
$$\Pi(t,f) = \forall(t,f) = \emptyset.$$

Π est appelé l'*opérateur produit* et $\forall$ l'*opérateur intersection.*

Ces opérateurs sont des fonctions partielles, puisque si X et les Y_a sont des bases de type μ-finies, $\Pi(a\in X;Y_a)$ et $\forall(a\in X;Y_a)$ ne le sont pas forcément pour un cardinal μ régulier infini quelconque.

Nous étendons ici la définition de fonctions quasi-stables ou μ-stables (pour μ cardinal quelconque) pour des fonctions à deux arguments.

1.3.7. DÉFINITION. Soient $\mathscr{A}_1$,$\mathscr{A}_2$ et $\mathscr{B}$ des espaces cohérents. Une fonction $F : D_{\mathscr{A}_1} \times D_{\mathscr{A}_2} \to D_{\mathscr{B}}$ est quasi-stable (resp. μ-stable) si et seulement si la fonction $F' : D_{\mathscr{A}_1 \& \mathscr{A}_2} \to D_{\mathscr{B}}$ définie par $F'(\{1\}\times a_1 \cup \{2\}\times a_2) = F(a_1,a_2)$ l'est.

Miquel démontre dans sa thèse que les opérateurs Π et $\forall$ sont des fonctions μ-stables. (Proposition 5.1.29 p :179.)

1.3.8. Proposition (μ-stabilité)**.** *Les opérateurs* $\Pi, \forall$ *définis précédemment sont des fonctions à deux arguments* μ*-stables.*

Par la suite, nous prendrons les versions curryfiées des opérateurs Π et $\forall$. Plus précisément nous transformons d'abord, comme dans la définition 1.3.7, temps les fonctions à deux arguments

$$\begin{aligned} \Pi &: D_{\mathsf{Ty}_\mu(\mathscr{A})} \times D_{\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})} \to D_{\mathsf{Ty}_\mu(\mathscr{A}\xrightarrow{\mu}\mathscr{B})} \\ \forall &: D_{\mathsf{Ty}_\mu(\mathscr{A})} \times D_{\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})} \to D_{\mathsf{Ty}_\mu(\mathscr{B})} \end{aligned}$$

en des fonctions à un argument

$$\begin{aligned} \Pi_1 &: D_{\mathsf{Ty}_\mu(\mathscr{A})\&(\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B}))} \to D_{\mathsf{Ty}_\mu(\mathscr{A}\xrightarrow{\mu}\mathscr{B})} \\ \forall_1 &: D_{\mathsf{Ty}_\mu(\mathscr{A})\&(\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B}))} \to D_{\mathsf{Ty}_\mu(\mathscr{B})} \end{aligned}$$

D'après la proposition 1.3.8, Π_1 et $\forall_1$ sont des fonctions μ-stables. Nous pouvons donc écrire :

$$\begin{aligned} \Pi_1 &: (\mathsf{Ty}_\mu(\mathscr{A}) \,\&\, (\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})))\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{A}\xrightarrow{\mu}\mathscr{B}) \\ \forall_1 &: (\mathsf{Ty}_\mu(\mathscr{A}) \,\&\, (\mathscr{A}\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B})))\xrightarrow{\mu}\mathsf{Ty}_\mu(\mathscr{B}) \end{aligned}$$

Nous appliquons alors les isomorphismes de Curry (définition 3.2.7) à Π_1 et $\forall_1$ pour obtenir des fonctions μ-stables (à un argument).

1.3.9. DÉFINITION (Opérateurs curryfiés). Soient $\mathscr{A},\mathscr{B}$ des espaces cohérents et μ un cardinal supérieur à ω. Les *opérateurs de type curryfiés* sont les fonctions μ-stables suivantes :

$$\Pi_c : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A} \xrightarrow{\mu} \mathscr{B}))$$
$$\forall_c : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}))$$

Dans la suite nous assimilerons respectivement Π_c et $\forall_c$ à Π et $\forall$.

1.4. Type sous-ensemble. L'interprétation du type sous-ensemble est assez simple. Intuitivement, l'interprétation de $\{x : A \mid B\}$ va être une partie de l'interprétation de A : l'ensemble des éléments a tels que l'interprétation de $B[x/a]$ est non-vide.

Puisque toute partie d'une base de type est une base de type (cf. remarque 1.1.2), la définition suivante est correcte.

1.4.1. DÉFINITION (Sous-ensemble). Soient $\mathscr{A}$ et $\mathscr{B}$ des espaces cohérents, X une base de type de $\mathscr{A}$, et $(Y_a)_{a\in X}$ une famille de bases de type de $\mathscr{B}$ indexée par X. Le *type sous-ensemble* est défini par la base de type de $\mathscr{A}$:

$$\sigma(a \in X; Y_a) \quad = \quad \{a \in X \mid Y_a \neq \emptyset\}$$

Nous définissons ensuite un opérateur de type de manière similaire à ce qui a été fait précédemment pour les produits.

1.4.2. DÉFINITION (Opérateur de type). Soient $\mathscr{A},\mathscr{B}$ des espaces cohérents et μ un cardinal régulier infini. L'*opérateur de type sous-ensemble* est l'application

$$\sigma : \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})} \times \mathsf{D}_{\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B})} \to \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}$$

définie, pour tout $t \in \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}$ et pour tout $f \in \mathsf{D}_{\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B})}$, par

- $$\sigma(t,f) \quad = \quad \{\tau(\sigma(a \in X; Y_a))\}$$
  s'il existe une base de type de $\mathscr{A}$, X et une famille $(Y_a)_{a\in X}$ de bases de type de $\mathscr{B}$ telles que
  (1) $t = \{\tau(X)\}$
  (2) $f = \mathsf{Tr}(F)$ et $F(a) = \{\tau(Y_a)\}$ pour tout $a \in X$
  (3) $\sigma(a \in X; Y_a)$ est une base de type μ-finie de $\mathscr{A}$
- dans tous les autres cas
  $$\sigma(t,f) = \emptyset.$$

1.4.3. Proposition (μ-stabilité)**.** *L'opérateur* σ *est une fonction binaire* μ*-stable.*

DÉMONSTRATION. De même que pour la preuve de la proposition 1.3.8, cette preuve est similaire à celle de la proposition 5.1.29 de la thèse de Miquel (p.179). □

1.4.4. REMARQUE (Opérateur curryfié). De même que pour les produits, nous assimilerons l'opérateur σ et sa version curryfiée :

$$\sigma : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}))$$

1.5. Type somme dépendante. Soient $\mathscr{A}$ et $\mathscr{B}$ des espaces cohérents, X une base de type de $\mathscr{A}$, et $(Y_a)_{a\in X}$ une famille de bases de type de $\mathscr{B}$ indexée par X.

Première étape. Il paraît naturel de vouloir interpréter le type $\Sigma x : \mathrm{A}.\mathrm{B}$ comme un produit cartésien dépendant. La proposition suivante montre qu'il est possible de définir un type sémantique de cette façon.

1.5.1. Proposition. *L'ensemble de cliques*

$$\begin{aligned}\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a) &= \{[a,b] \in \mathsf{D}_{\mathscr{A}\&\mathscr{B}} \mid a \in \uparrow \mathrm{X}, b \in \uparrow \mathrm{Y}_{\lfloor a \rfloor}\} \\ &= \{c \in \mathsf{D}_{\mathscr{A}\&\mathscr{B}} \mid \exists a \in \uparrow \mathrm{X}, \exists b \in \uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}, c = [a,b]\}\end{aligned}$$

est un type sémantique de $\mathscr{A}\,\&\,\mathscr{B}$.

DÉMONSTRATION. Montrons que $\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$ est clos supérieurement. Soient $c \in \Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a), c' \in \mathsf{D}_{\mathscr{A}\&\mathscr{B}}$ des cliques telles que $c \subset c'$. Montrons que $c' \in \Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$. Il existe $a, a' \in \uparrow \mathrm{X}$ et $b, b' \in \uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}$, tels que $a \subset a', b \subset b'$ et $c = [a,b]$, $c' = [a',b']$. D'après le corollaire 1.1.16 page 85, $\lfloor a \rfloor_{\mathrm{X}} = \lfloor a' \rfloor_{\mathrm{X}}$, d'où $b' \in \uparrow \mathrm{Y}_{\lfloor a' \rfloor_{\mathrm{X}}}$ et $c' \in \Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$.

Montrons que $\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$ est stable par intersection de cliques compatibles. Soit I un ensemble non-vide et $(c_i)_{i\in \mathrm{I}}$ une famille de cliques de $\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$ compatibles de $\mathscr{A}\,\&\,\mathscr{B}$. Il existe donc $(a_i)_{i\in \mathrm{I}}$, famille de cliques de $\uparrow \mathrm{X}$ compatibles dans $\mathscr{A}$ et $(b_i)_{i\in \mathrm{I}}$, familles de cliques de $\mathsf{D}_{\mathscr{B}}$ compatibles dans $\mathscr{B}$ telles que pour tout $i \in \mathrm{I}$, $c_i = [a_1, b_i]$ et $b_i \in \uparrow \mathrm{Y}_{\lfloor a_i \rfloor_{\mathrm{X}}}$.

Soient $a = \bigcap_{i\in \mathrm{I}} a_i$, $b = \bigcap_{i\in \mathrm{I}} b_i$ et $c = \bigcap_{i\in \mathrm{I}} c_i$. Nous avons $c = [a,b]$. Montrons que $c \in \Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$. Il suffit de montrer que $a \in \uparrow \mathrm{X}$ et que $b \in \uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}$. $a \in \uparrow \mathrm{X}$ car $\uparrow \mathrm{X}$ est un type sémantique et $(a_i)_{i\in \mathrm{I}}$ est une famille de cliques de $\uparrow \mathrm{X}$ compatibles dans $\mathscr{A}$. De plus, pour tout $i \in \mathrm{I}$, $a \subset a_i$, donc $\lfloor a \rfloor_{\mathrm{X}} = \lfloor a_i \rfloor_{\mathrm{X}}$ (corollaire 1.1.16 page 85) et $b \in \uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}$. $(b_i)_{i\in \mathrm{I}}$ est donc une famille de cliques de $\uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}$ compatibles dans $\mathscr{B}$. Puisque $\uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}$ est un type sémantique, nous avons bien $b \in \uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}$. □

1.5.2. NOTATION. Nous notons $\Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a)$ la base de type $\lfloor \Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a) \rfloor$ associée au type sémantique $\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$.

1.5.3. REMARQUE. Nous avons $\Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a) = \{[a,b] \in \mathsf{D}_{\mathscr{A}\&\mathscr{B}} \mid a \in \mathrm{X}, b \in \mathrm{Y}_a\}$.

Il serait donc possible de définir le type somme dépendante comme la base de type $\Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a)$. Mais cette définition va poser problème lors de la définition du modèle concret. En effet, nous allons considérer un espace de cohérence $\mathscr{A}_{\mathsf{mod}}$ qui va respecter une certaine équation récursive. Il ne sera pas possible de montrer l'inclusion rigide $\mathscr{A}_{\mathsf{mod}}\,\&\,\mathscr{A}_{\mathsf{mod}} \Subset \mathscr{A}_{\mathsf{mod}}$. Nous aurons par contre $\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}} \Subset \mathscr{A}_{\mathsf{mod}}$. Comme il est possible de définir un plongement entre le produit direct de deux espaces cohérents identiques $\mathscr{A}\,\&\,\mathscr{A}$ et l'espace des fonctions μ-stables $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$, nous allons définir le type sémantique comme l'image du type sémantique défini en première approche par le plongement rigide.

Auparavant nous devons démontrer certains résultats utiles sur l'effet des plongements rigides sur les bases de type et les types sémantiques.

Plongements rigides, bases de type et types sémantiques. Soient $\mathscr{A}, \mathscr{B}$ des espaces cohérents tels qu'il existe un plongement $\Phi : |\mathscr{A}| \to |\mathscr{B}|$. Rappelons que nous avons alors un plongement associé $\Phi^+ : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}}$ et une rétraction $\Phi^- : \mathsf{D}_{\mathscr{B}} \to \mathsf{D}_{\mathscr{A}}$ tels que pour tout $a \in \mathsf{D}_{\mathscr{A}}$ et pour tout $b \in \mathsf{D}_{\mathscr{B}}$,

$$\begin{aligned}\Phi^+(a) &= \{\Phi(\alpha) \mid \alpha \in a\} \\ \Phi^-(b) &= \{\alpha \mid \Phi(\alpha) \in b\}\end{aligned}$$

Le but de cette sous-section est d'étudier l'effet de l'application d'un plongement rigide à un type sémantique ou une base de type.

1.5.4. NOTATION. Si X est un ensemble de cliques de $\mathscr{A}$, nous notons $\Phi^{++}(\mathrm{X})$ l'image de X par le plongement Φ^+. Nous avons donc $\Phi^{++}(\mathrm{X}) = \{\Phi^+(a) \mid a \in \mathrm{X}\}$.

1.5.5. Lemme. *Soient* I *un ensemble non-vide et* $(a_i)_{i\in I}$ *une famille de cliques de* $\mathscr{A}$*, alors*

$$\Phi^+(\bigcap_{i\in I} a_i) = \bigcap_{i\in I} \Phi^+(a_i) \tag{1}$$

$$\Phi^+(\bigcup_{i\in I} a_i) = \bigcup_{i\in I} \Phi^+(a_i) \tag{2}$$

$$\bigcup_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}} \Leftrightarrow \bigcup_{i\in I} \Phi^+(a_i) \in \mathsf{D}_{\mathscr{B}} \tag{3}$$

DÉMONSTRATION. (1) et (2) sont immédiats. Montrons (3).

Si $\bigcup_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}}$, montrons que $\bigcup_{i\in I} \Phi^+(a_i) \in \mathsf{D}_{\mathscr{B}}$. D'après le point (2), $\bigcup_{i\in I} \Phi^+(a_i) = \Phi^+(\bigcup_{i\in I} a_i)$. Or $\Phi^+(\bigcup_{i\in I} a_i) \in \mathsf{D}_{\mathscr{B}}$ car $\Phi^+ : \mathsf{D}_{\mathscr{A}} \to \mathsf{D}_{\mathscr{B}}$ envoie toute clique de $\mathscr{A}$ dans une clique de $\mathscr{B}$.

Si $\bigcup_{i\in I} \Phi^+(a_i) = \Phi^+(\bigcup_{i\in I} a_i) \in \mathsf{D}_{\mathscr{B}}$. Or $\Phi^- \circ \Phi^+ = \mathsf{id}_{\mathsf{D}_{\mathscr{A}}}$, d'où $\bigcup_{i\in I} a_i = \Phi^-(\Phi^+(\bigcup_{i\in I} a_i)) \in \mathsf{D}_{\mathscr{A}}$. □

1.5.6. Lemme.

(1) $\Phi^{++}(\lfloor X \rfloor) = \lfloor \Phi^{++}(X) \rfloor$ *;*

(2) $\Phi^{++}(\uparrow X) \subset \uparrow \Phi^{++}(X)$ *;*

(3) $\uparrow \Phi^{++}(\uparrow X) = \uparrow \Phi^{++}(X)$.

DÉMONSTRATION.

(1) Si $a \in \lfloor X \rfloor$, montrons que $\Phi^+(a) \in \lfloor \Phi^{++}(X) \rfloor$. $\Phi^+(a) \in \Phi^{++}(X)$ car $\lfloor X \rfloor \subset X$. De plus, soit $a' \in X$ telle que $\Phi^+(a') \subset \Phi^+(a)$, montrons que $\Phi^+(a') = \Phi^+(a)$. Φ^- est monotone donc $\Phi^-(\Phi^+(a')) \subset \Phi^-(\Phi^+(a))$. Puisque $\Phi^- \circ \Phi^+ = \mathsf{id}_{\mathsf{D}_{\mathscr{A}}}$, nous avons $a' \subset a$. Comme $a' \in X$ et $a \in \lfloor X \rfloor$, nous avons $a' = a$, d'où $\Phi^+(a') = \Phi^+(a)$.

Soit $a \in X$ telle que $\Phi^+(a) \in \lfloor \Phi^{++}(X) \rfloor$. Montrons que $a \in \lfloor X \rfloor$. Soit $a' \in X$ telle que $a' \subset a$. Montrons que $a' = a$. Par monotonie de Φ^+, nous avons $\Phi^+(a') \subset \Phi^+(a)$. Par minimalité de $\Phi^+(a)$, nous déduisons que $\Phi^+(a') = \Phi^+(a)$, puis, en appliquant Φ^-, $a' = a$.

(2) Si $a \in \uparrow X$, montrons que $\Phi^+(a) \in \uparrow \Phi^{++}(X)$. Il existe $a_0 \in X$ telle que $a_0 \subset a$. Par monotonie de Φ^+, $\Phi^+(a_0) \subset \Phi^+(a)$. D'où $\Phi^+(a) \in \uparrow \Phi^{++}(X)$ par définition de la clôture supérieure.

(3) D'après le point 2, $\Phi^{++}(\uparrow X) \subset \uparrow \Phi^{++}(X)$, d'où $\uparrow \Phi^{++}(\uparrow X) \subset \uparrow\uparrow \Phi^{++}(X) = \uparrow \Phi^{++}(X)$.

Montrons $\uparrow \Phi^{++}(X) \subset \uparrow \Phi^{++}(\uparrow X)$. Si Y, Y' sont des ensembles de cliques tels que $Y \subset Y'$, il est clair que $\Phi^{++}(Y) \subset \Phi^{++}(Y')$ d'où $\Phi^{++}(X) \subset \Phi^{++}(\uparrow X)$, puis $\uparrow \Phi^{++}(X) \subset \uparrow \Phi^{++}(\uparrow X)$.

□

1.5.7. Proposition.

(1) Si $X \subset \mathsf{D}_{\mathscr{A}}$ *est une base de type de* $\mathscr{A}$ *alors* $\Phi^{++}(X)$ *est une base de type de* $\mathscr{B}$*.*

(2) Si $X \subset \mathsf{D}_{\mathscr{A}}$ *est stable par intersection de cliques compatibles, alors* $\Phi^{++}(X)$ *l'est également.*

(3) Si $X \subset \mathsf{D}_{\mathscr{A}}$ *est un type sémantique de* $\mathscr{A}$ *alors* $\uparrow \Phi^{++}(X)$ *est un type sémantique de* $\mathscr{B}$*.*

DÉMONSTRATION.

(1) Soient $a_1, a_2 \in X$, $b_1 = \Phi^+(a_1)$ et $b_2 = \Phi^+(a_2)$ telles que $b_1 \cup b_2 \in \mathsf{D}_{\mathscr{B}}$. Montrons que $b_1 = b_2$. Nous avons $b_1 \cup b_2 = \Phi^+(a_1 \cup a_2)$ (point (2) du lemme 1.5.5), d'où $a_1 \cup a_2 = \Phi^-(b_1 \cup b_2) \in \mathsf{D}_{\mathscr{A}}$. Or X est une base de type, donc $a_1 = a_2$, puis $b_1 = b_2$.

(2) Conséquence directe des points 1 et 3 du lemme 1.5.5.

(3) Conséquence immédiate du point 2 et du lemme 1.6.3.

□

Deuxième étape. Pour définir un plongement rigide entre $\mathscr{A}\,\&\,\mathscr{A}$ et $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$, nous devons supposer que la trame de l'espace cohérent $\mathscr{A}$ contient au moins deux éléments incohérents,[1] que nous notons $\mathbf{1}_{|\mathscr{A}|}$ et $\mathbf{2}_{|\mathscr{A}|}$. Cette perte de généralité n'est pas un problème car l'ensemble d'interprétation que nous considérerons dans le modèle concret vérifiera bien cette condition.

Nous pouvons maintenant définir un un encodage des cliques de $\mathscr{A}\,\&\,\mathscr{A}$ dans celles de $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$.

1.5.8. Proposition (Plongement du produit direct vers l'espace des fonctions μ-stables)**.** *L'application*

$$\psi_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}} \quad : \quad \begin{array}{ccc} |\mathscr{A}\,\&\,\mathscr{A}| & \to & |\mathscr{A} \xrightarrow{\mu} \mathscr{A}| \\ (i,\alpha) & \mapsto & (\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(i),\alpha) \end{array}$$

où $\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(1) = \{\mathbf{1}_{|\mathscr{A}|}\}$ *et* $\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(2) = \{\mathbf{2}_{|\mathscr{A}|}\}$ *est un plongement rigide de* $\mathscr{A}\,\&\,\mathscr{A}$ *vers* $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$.

DÉMONSTRATION. L'injectivité de $\psi_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$ est une conséquence immédiate de celle de $\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$. Soient $i, i' \in \{1,2\}$ et $\alpha, \alpha' \in |\mathscr{A}|$. Montrons que

$$(i,\alpha) \frown_{\mathscr{A}\&\mathscr{A}} (i',\alpha') \Leftrightarrow (\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(i),\alpha) \frown_{\mathscr{A} \xrightarrow{\mu} \mathscr{A}} (\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(i'),\alpha')$$

Puisque $\mathbf{1}_{|\mathscr{A}|}$ et $\mathbf{2}_{|\mathscr{A}|}$ sont incohérents, nous avons, par définition de $\frown_{\mathscr{A} \xrightarrow{\mu} \mathscr{A}}$:

$$(\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(i),\alpha) \frown_{\mathscr{A} \xrightarrow{\mu} \mathscr{A}} (\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(i'),\alpha') \Leftrightarrow i \neq i' \wedge \alpha \frown_{\mathscr{A}} \alpha'$$

Par définition de $\frown_{\mathscr{A}\&\mathscr{A}}$, nous avons bien l'équivalence recherchée. □

1.5.9. NOTATION. Le plongement associé à $\psi_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$ est noté $\psi^{+}_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$.

Nous pouvons alors utiliser la proposition 1.5.7 et définir l'interprétation du type somme dépendante.

1.5.10. Proposition (Somme dépendante)**.** *Soient*

$$\begin{aligned} \Sigma_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a) &= \uparrow \psi^{++}_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)) \\ &= \uparrow \{\psi^{+}_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(c) \mid c \in \Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)\} \end{aligned}$$

et

$$\begin{aligned} \Sigma(a \in \mathrm{X}; \mathrm{Y}_a) &= \psi^{++}_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(\Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a)) \\ &= \{\psi^{+}_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(c) \mid c \in \Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a)\} \end{aligned}$$

des ensembles de cliques de $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$.

Alors $\Sigma_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$ *est un type sémantique de* $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$ *et* $\Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$ *est sa base de type associée.*

De plus nous avons

$$\Sigma_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a) = \{\mathsf{Tr}(\mathrm{F}) \in \mathscr{A} \xrightarrow{\mu} \mathscr{A} \mid \exists a \in \uparrow \mathrm{X}, \exists b \in \uparrow \mathrm{Y}_{\lfloor a \rfloor_{\mathrm{X}}}, a \subset \mathrm{F}(\{\mathbf{1}_{|\mathscr{A}|}\}), b \subset \mathrm{F}(\{\mathbf{2}_{|\mathscr{A}|}\})\}.$$

DÉMONSTRATION. Le point 3 de la proposition 1.5.7, en remplaçant X par $\Sigma^0_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$ nous montre que $\Sigma_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$ est un type sémantique. Le point 1 de la proposition 1.5.7, en remplaçant X par $\Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a)$ nous montre que $\Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$ est une base de type. Le point 3 du lemme 1.5.6, en remplaçant X par $\Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$ nous montre que $\Sigma_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a) = \uparrow \Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$. Nous en déduisons que $\Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$ est la base de type associée à $\Sigma_{\mathrm{TS}}(a \in \mathrm{X}; \mathrm{Y}_a)$. □

1. α, α' sont incohérents si $\alpha \frown_{\mathscr{A}} \alpha'$ n'est pas vrai. Nous notons alors $\alpha \asymp \alpha'$.

1.5.11. DÉFINITION (Opérateur de type). Soit $\mathscr{A}$ un espace cohérent et μ un cardinal régulier infini. L'*opérateur de type somme dépendante* est l'application

$$\Sigma : \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})} \times \mathsf{D}_{\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A})} \to \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A} \xrightarrow{\mu} \mathscr{A})}$$

définie, pour tout $t \in \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}$ et pour tout $f \in \mathsf{D}_{\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A})}$, par

- $$\Sigma(t,f) \quad = \quad \{\tau(\Sigma(a \in \mathrm{X}; \mathrm{Y}_a))\}$$
  s'il existe une base de type de $\mathscr{A}$, X et une famille $(\mathrm{Y}_a)_{a\in\mathrm{X}}$ de bases de type de $\mathscr{A}$ telles que
  (1) $t = \{\tau(\mathrm{X})\}$
  (2) $f = \mathsf{Tr}(\mathrm{F})$ et $\mathrm{F}(a) = \{\tau(\mathrm{Y}_a)\}$ pour tout $a \in \mathrm{X}$
  (3) $\Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$ est une base de type μ-finie de $\mathscr{A} \xrightarrow{\mu} \mathscr{A}$
- dans tous les autres cas
  $$\Sigma(t,f) = \emptyset.$$

1.5.12. Proposition (μ-stabilité)**.** *L'opérateur* Σ *est une fonction binaire* μ*-stable.*

DÉMONSTRATION. De même que pour la μ-stabilité des opérateurs des types produit (proposition 1.3.8) et sous-ensemble (proposition 1.4.3), cette preuve est similaire à celle de la proposition 5.1.29 de la thèse de Miquel (p.179). □

1.5.13. REMARQUE (Opérateur curryfié). De même que précédemment, nous assimilerons l'opérateur Σ et sa version curryfiée :

$$\Sigma : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A} \xrightarrow{\mu} \mathscr{A}))$$

1.6. Types sémantiques et bases de type engendrés. Dans cette sous-section, nous considérons un espace cohérent $\mathscr{A}$ et un ensemble de cliques $\mathrm{X} \subset \mathsf{D}_{\mathscr{A}}$.[2]

Motivation. Il semble naturel de vouloir interpréter le type existentiel $\exists x : \mathrm{A}.\mathrm{B}$ par l'union des interprétations de $\mathrm{B}[x/a]$, pour les termes a de type A. Malheureusement, l'union de plusieurs types sémantiques n'est pas en général un type sémantique. Nous définissons ici un type sémantique engendré par un ensemble de cliques quelconque, ainsi que sa base de type correspondante.

Résultats préliminaires.

1.6.1. Lemme. *Soit* $\mathrm{X} \subset \mathsf{D}_{\mathscr{A}}$*, alors*

(1) $\lfloor \mathrm{X} \rfloor = \lfloor \uparrow \mathrm{X} \rfloor$.

(2) $\uparrow \lfloor \mathrm{X} \rfloor = \uparrow \mathrm{X}$.

DÉMONSTRATION.

(1) L'implication $\subset$ est évidente car $\lfloor \mathrm{X} \rfloor \subset \mathrm{X}$. $\supset$ est immédiate également.

(2) Soit $a \in \lfloor \mathrm{X} \rfloor$. Montrons que $a \in \lfloor \uparrow \mathrm{X} \rfloor$. Nous avons $\lfloor \mathrm{X} \rfloor \subset \mathrm{X} \subset \uparrow \mathrm{X}$, d'où $a \in \uparrow \mathrm{X}$. Montrons que a est une clique minimale de $\uparrow \mathrm{X}$. Soit $a' \in \uparrow \mathrm{X}$ telle que $a' \subset a$. Montrons que $a = a'$. Par définition de la clôture supérieure, il existe $a'_0 \in \mathrm{X}$ telle que $a'_0 \subset a' \subset a$. Par définition de $\lfloor \mathrm{X} \rfloor$, nous avons $a'_0 = a$, d'où $a_0 = a$.

Soit $a \in \lfloor \uparrow \mathrm{X} \rfloor$. Montrons que $a \in \lfloor \mathrm{X} \rfloor$. $a \in \uparrow \mathrm{X}$, donc il existe une clique $a_0 \in \mathrm{X}$ telle que $a_0 \subset a$. Montrons maintenant que pour toute clique $a' \in \mathrm{X}$, si $a' \subset a$ alors $a' = a$. Nous en déduirons que $a = a_0$ est une clique de X qui est de plus minimale. Puisque $\mathrm{X} \subset \uparrow \mathrm{X}$, nous avons $a' \in \uparrow \mathrm{X}$. Par minimalité de a, nous avons $a = a'$.

2. Les résultats de cette sous-section doivent beaucoup à des discussions menées avec Alexandre Miquel. En particulier, le lemme 1.6.3 et la proposition 1.6.17, cruciale, ont été prouvées par lui.

□

1.6.2. Lemme (Stabilité par intersection)**.** *Soient $\mathscr{A}$ un espace cohérent,* I *un ensemble et* $(\mathrm{X}_i)_{i\in\mathrm{I}}$ *une famille de types sémantiques de $\mathscr{A}$. Alors l'intersection des* X_i *indexée par* I, $\bigcap_{i\in\mathrm{I}}\mathrm{X}_i$, *est un type sémantique de $\mathscr{A}$.*

DÉMONSTRATION. Il suffit d'appliquer la définition. □

1.6.3. Lemme. *Soit $\mathscr{A}$ un espace cohérent et* $\mathrm{Y}\subset\mathsf{D}_{\mathscr{A}}$ *un ensemble de cliques. Si* Y *est stable par intersection de cliques compatibles. alors* $\uparrow\mathrm{Y}$ *est un type sémantique de $\mathscr{A}$.*

DÉMONSTRATION. $\uparrow\mathrm{Y}$ est clairement clos supérieurement. Montrons qu'il est stable par intersection de cliques compatibles.

Soit $(a_i)_{i\in\mathrm{I}}$ une famille de cliques de $\uparrow\mathrm{Y}$ qui est une famille de cliques compatibles dans $\mathscr{A}$. Montrons que $\bigcap_{i\in\mathrm{I}}a_i\in\uparrow\mathrm{Y}$.

Pour tout $i\in\mathrm{I}$, il existe une clique $a_i^0\in\mathrm{Y}$ telle que $a_i^0\subset a_i$. Par clôture inférieure de $\mathsf{D}_{\mathscr{A}}$, la famille de cliques de Y $(a_i^0)_{i\in\mathrm{I}}$ ainsi définie est une famille de cliques compatibles dans $\mathscr{A}$. Par hypothèse sur Y, nous avons donc $\bigcap_{i\in\mathrm{I}}a_i^0\in\mathrm{Y}$. Or $\bigcap_{i\in\mathrm{I}}a_i^0\subset\bigcap_{i\in\mathrm{I}}a_i$. Nous avons donc bien $\bigcap_{i\in\mathrm{I}}a_i\in\uparrow\mathrm{Y}$. □

Définitions.

1.6.4. DÉFINITION (Type sémantique et base de type engendrés). Le *type sémantique engendré par* X *dans $\mathscr{A}$*, noté $\mathsf{TSem}(\mathrm{X})$, est le plus petit (au sens de l'inclusion) type sémantique de $\mathscr{A}$ contenant X.

La *base de type engendrée* par X, notée $\mathsf{Base}(\mathrm{X})$, est la base de type associée au type sémantique engendré $\mathsf{TSem}(\mathrm{X})$.

1.6.5. REMARQUE. D'après le point 3 de la proposition 1.1.14 du chapitre précédent, $\mathsf{Base}(\mathrm{X})=\lfloor\mathsf{TSem}(\mathrm{X})\rfloor$.

1.6.6. REMARQUE. Contrairement aux types sémantiques, la base de type engendrée ne dépend pas de l'espace de cohérence considéré.

Nous pouvons généraliser la coercition au cas d'un ensemble de cliques quelconque.

1.6.7. Lemme (Coercition généralisée)**.** *Si $a\in\mathrm{X}$, il existe une et une seule clique $a_0\in\mathsf{Base}(\mathrm{X})$ telle que $a_0\subset a$. Nous la notons $\lfloor a\rfloor_{\mathsf{Base}(\mathrm{X})}$.*

DÉMONSTRATION. C'est une conséquence du lemme de coercition (lemme 1.1.15 du chapitre précédent) puisque $\mathrm{X}\subset\mathsf{TSem}(\mathrm{X})=\uparrow\mathsf{Base}(\mathrm{X})$. Cela justifie la réutilisation de la notation $\lfloor a\rfloor_{\mathsf{Base}(\mathrm{X})}$. □

Première caractérisation.

1.6.8. NOTATION (Ensemble des types sémantiques contenant X). Nous notons TS_{X} l'ensemble défini par :

$$\begin{aligned}\mathsf{TS}_{\mathrm{X}} &= \{\mathrm{X}'\supset\mathrm{X}\mid\mathrm{X}'\text{ type sémantique de }\mathscr{A}\}\\ &= \{\mathrm{X}'\in\mathfrak{P}(\mathscr{A})\mid\mathrm{X}\subset\mathrm{X}'\wedge\mathrm{X}'\text{ type sémantique de }\mathscr{A}\}\end{aligned}$$

1.6.9. Proposition (Caractérisation intensionnelle)**.** *Le type sémantique engendré* $\mathsf{TSem}(X)$ *existe de manière unique et nous avons*

$$\mathsf{TSem}(X) \quad = \quad \bigcap_{X' \in \mathsf{TS}_X} X'$$

DÉMONSTRATION. $\bigcap_{X' \in \mathsf{TS}_X} X'$ est un type sémantique car c'est une intersection de types sémantiques (lemme 1.6.2). C'est clairement le plus petit car il est contenu dans $\mathsf{TSem}(X)$, élément de TS_X. □

Deuxième caractérisation. La caractérisation précédente du type sémantique engendré ne décrit pas ses éléments. Cette caractérisation intensionnelle est donc peu utilisable en pratique. Aussi, nous allons maintenant donner une méthode concrète pour construire le type sémantique, puis la base de type, engendrés par une partie X de $\mathscr{A}$.

1.6.10. DÉFINITION (Intersection compatible). L'*intersection compatible de* X *dans* $\mathscr{A}$, notée $\mathsf{IC}(X)$ (ou $\mathsf{IC}_{\mathscr{A}}(X)$ si ambigüité), est l'intersection des cliques minimales de compatibles de X :

$$\begin{aligned}\mathsf{IC}(X) &= \{\bigcap_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}} \mid I \neq \emptyset \wedge (\forall i \in I, a_i \in \lfloor X \rfloor) \wedge \bigcup_{i\in I} a_i \in \mathsf{D}_{\mathscr{A}}\} \\ &= \{c \in \mathsf{D}_{\mathscr{A}} \mid \exists X_0 \subset \lfloor X \rfloor, X_0 \neq \emptyset \wedge \bigcup_{a\in X_0} a \in \mathscr{A} \wedge c = \bigcap_{a\in X_0} a\}\end{aligned}$$

1.6.11. REMARQUES.

- Nous choisissons de ne garder que l'intersection de cliques minimales pour avoir moins d'éléments. Il aurait été également possible d'utiliser X au lieu de $\lfloor X \rfloor$.
- En prenant pour I un singleton, nous montrons que $\lfloor X \rfloor \subset \mathsf{IC}(X)$.

1.6.12. Lemme.

(1) $\mathsf{IC}(X)$ *est stable par intersection de cliques compatibles.*

(2) $\uparrow \mathsf{IC}(X)$ *est un type sémantique contenant* X.

DÉMONSTRATION.

(1) Soit J un ensemble non-vide et $(c_j)_{j\in J}$ une famille d'éléments de $\mathsf{IC}(X)$ telle que $\bigcup_{j\in J} c_j \in \mathsf{D}_{\mathscr{A}}$. Montrons que $\bigcap_{j\in J} c_j \in \mathsf{IC}(X)$.

Par définition de $\mathsf{IC}(X)$, pour tout $j \in J$, nous avons un ensemble I_j et une famille d'éléments de $\lfloor X \rfloor$ $(a_i)_{i\in I_j}$ telle que $c_j = \bigcap_{i\in I_j} a_i$. Posons $I = \bigcup_{j\in J} I_j$. Nous avons donc

$$\bigcap_{j\in J} c_j \quad = \quad \bigcap_{i\in I} a_i$$

avec, pour tout i, $a_i \in \lfloor X \rfloor$ et $\bigcup_{i\in I} a_i = \bigcup_{j\in J} c_j \in \mathsf{D}_{\mathscr{A}}$ par hypothèse. Par définition de $\mathsf{IC}(X)$, nous avons bien $\bigcap_{i\in I} a_i \in \mathsf{IC}(X)$.

(2) Conséquence immédiate du lemme 1.6.3.

□

1.6.13. Proposition (Construction du type sémantique et de la base de type engendrés)**.** *Nous avons*

(1) $\mathsf{TSem}(X) = \uparrow \mathsf{IC}(X)$

(2) $\mathsf{Base}(X) = \lfloor \mathsf{IC}(X) \rfloor$

DÉMONSTRATION.

(1) L'inclusion $\mathsf{TSem}(X) \subset \uparrow \mathsf{IC}(X)$ est immédiate car $\uparrow \mathsf{IC}(X)$ est un type sémantique contenant X.

Montrons que $\uparrow \mathsf{IC}(X) \subset \mathsf{TSem}(X)$. Puisque $\mathsf{TSem}(X)$ est clos supérieurement, il suffit de montrer que $\mathsf{IC}(X) \subset \mathsf{TSem}(X)$. Soit $c \in \mathsf{IC}(X)$. Montrons que $c \in \mathsf{TSem}(X)$. Par définition de $\mathsf{IC}(X)$, il existe un ensemble I non-vide et une famille de cliques de $\lfloor X \rfloor$ compatibles dans $\mathscr{A}$ $(a_i)_{i\in I}$ tels que $c = \bigcap_{i\in I} a_i$. Or $\lfloor X \rfloor \subset X \subset \mathsf{TSem}(X)$. c est donc l'intersection d'une famille de cliques de $\mathsf{TSem}(X)$ compatibles dans $\mathscr{A}$. Puisque $\mathsf{TSem}(X)$, type sémantique, est stable par intersection de cliques compatibles, nous avons bien $c \in \mathsf{TSem}(X)$.

(2) Par définition de la base de type associée, il suffit de montrer que $\lfloor \mathsf{IC}(X) \rfloor = \lfloor \uparrow \mathsf{IC}(X) \rfloor$, ce qui est une conséquence directe du point 1 du lemme 1.6.1.

□

1.6.14. REMARQUE. Les notions de type sémantique et base de type engendrés sont bien compatibles avec celles de type sémantique et de base de type. Ainsi, si X est une base de type, nous vérifions facilement que $\mathsf{Base}(X) = \mathsf{IC}(X) = X$ et $\mathsf{TSem}(X) = \uparrow X$. De même, si Y est un type sémantique, alors $\mathsf{TSem}(Y) = Y$ et $\mathsf{Base}(Y) = \lfloor Y \rfloor$.

Lemme utile pour la fonction Gen. Analogue de la définition de base de type μ-finie (définition 1.2.1).

1.6.15. DÉFINITION (Ensemble de cliques μ-fini). X est un *ensemble de cliques μ-fini* si

(1) $\overline{\overline{X}} < \mu$

(2) $\forall a \in X, \overline{\overline{a}} < \mu$.

1.6.16. Lemme (Base engendrée et μ-finitude)**.** *Si* X *est un ensemble de cliques* μ*-fini alors*

(1) $\mathsf{IC}(X)$ *est un ensemble de cliques* μ*-fini*

(2) $\mathsf{Base}(X)$ *est une base de type* μ*-finie.*

DÉMONSTRATION. (2) est une conséquence immédiate de (1) puisque $\mathsf{Base}(X) = \lfloor \mathsf{IC}(X) \rfloor \subset \mathsf{IC}(X)$.

Soit $a \in \mathsf{IC}(X)$. Par définition de $\mathsf{IC}(X)$, il existe $a_0 \in X$, tel que $a \subset a_0$. D'où $\overline{\overline{a}} \leq \overline{\overline{a_0}} < \mu$ car X est μ-fini.

Montrons que $\overline{\overline{\mathsf{IC}(X)}} < \mu$. □

Extension du produit dépendant - lié au type existentiel. Le résultat suivant, dû à Miquel, permettra de prouver l'axiome 11. Cet axiome est utilisé dans le modèle abstrait pour l'interprétation des deux règles d'élimination des sommes (Σ-E) et (∃-E).

1.6.17. Proposition (Produit dépendant généralisé)**.** *Soient* $\mathscr{A}, \mathscr{B}$ *des espaces cohérents,* $X \subset \mathsf{D}_{\mathscr{A}}$ *un ensemble de cliques de* $\mathscr{A}$ *et* $(Y_a)_{a\in\mathsf{Base}(X)}$ *une famille de types sémantiques de* $\mathscr{B}$*. Alors*

$$\Pi_{TS}(a \in \mathsf{Base}(X); Y_a) \quad = \quad \{\mathsf{Tr}(F) \mid F : \mathscr{A} \xrightarrow{q} \mathscr{B}, \forall a \in X, F(a) \in \uparrow Y_{\lfloor a \rfloor_{\mathsf{Base}(X)}}\}$$

DÉMONSTRATION. Par définition, nous avons

$$\Pi_{TS}(a \in \mathsf{Base}(X); Y_a) \quad = \quad \{\mathsf{Tr}(F) \mid F : \mathscr{A} \xrightarrow{q} \mathscr{B}, \forall a \in \mathsf{Base}(X), F(a) \in \uparrow Y_a\}$$

Montrons l'égalité par double inclusion.

- $\subset$: Soient $F : \mathscr{A} \xrightarrow{q} \mathscr{B}$ telle que $\mathsf{Tr}(F) \in \Pi_{TS}(a \in \mathsf{Base}(X); Y_a)$ et $a_0 \in X$. Montrons que $F(a_0) \in \uparrow Y_{\lfloor a_0 \rfloor_{\mathsf{Base}(X)}}$. Par monotonie de F, nous avons $F(\lfloor a_0 \rfloor_{\mathsf{Base}(X)}) \subset F(a_0)$. Par hypothèse, $F(\lfloor a_0 \rfloor_{\mathsf{Base}(X)}) \in \uparrow Y_{\lfloor a_0 \rfloor_{\mathsf{Base}(X)}}$. Puisque $\uparrow Y_{\lfloor a_0 \rfloor_{\mathsf{Base}(X)}}$ est clos supérieurement, nous avons bien $F(a_0) \in \uparrow Y_{\lfloor a_0 \rfloor_{\mathsf{Base}(X)}}$.

- $\supset$: Soit $F : \mathscr{A} \xrightarrow{q} \mathscr{B}$ telle que $\forall a \in X, F(a) \in\uparrow Y_{\lfloor a \rfloor_{\mathsf{Base}(X)}}$ et soit $a_0 \in \mathsf{Base}(X)$. Montrons que $F(a_0) \in\uparrow Y_{a_0}$. D'après le point 2 de la proposition 1.6.13, $a_0 \in \lfloor \mathsf{IC}(X) \rfloor \subset \mathsf{IC}(X)$. Par définition de $\mathsf{IC}(X)$, il existe une famille de cliques de $\lfloor X \rfloor$ compatibles dans $\mathscr{A}$ $(a_i)_{i\in I}$ telle que $a_0 = \bigcap_{i\in I} a_i$. Pour tout $i \in I$, nous avons, $\lfloor a_i \rfloor_{\mathsf{Base}(X)} = a_0$ d'après le lemme 1.6.7. Donc, par hypothèse, $F(a_i) \in\uparrow Y_{a_0}$. La famille $(F(a_i))_{i\in I}$ est une famille de cliques de $\uparrow Y_{a_0}$ compatibles dans $\mathscr{A}$ puisque $\uparrow Y_{a_0}$ est clos supérieurement. Comme de plus $\uparrow Y_{a_0}$ est un type sémantique de $\mathscr{B}$, nous avons $\bigcap_{i\in I} F(a_i) \in\uparrow Y_{a_0}$, soit par quasi-stabilité de F, $F(a_0) = F(\bigcap_{i\in I} a_i) \in\uparrow Y_{a_0}$.

□

1.7. Interprétation du type existentiel. Soient $\mathscr{A}$ et $\mathscr{B}$ des espaces cohérents, X une base de type de $\mathscr{A}$, et $(Y_a)_{a\in X}$ une famille de bases de type de $\mathscr{B}$ indexée par X.

1.7.1. DÉFINITION (Existentielle). Nous notons $\exists_{TS}(a \in X; Y_a)$ le type sémantique engendré par $\bigcup_{a\in X} Y_a$ et $\exists(a \in X; Y_a)$ la base de type associée :

$$\begin{aligned} \exists_{TS}(a \in X; Y_a) &= \mathsf{TSem}(\bigcup_{a\in X} Y_a) \\ \exists(a \in X; Y_a) &= \lfloor \exists_{TS}(a \in X; Y_a) \rfloor. \end{aligned}$$

$\exists(a \in X; Y_a)$ est appelé le *type existentiel*.

1.7.2. REMARQUE. D'après la remarque 1.6.5, nous avons $\exists(a \in X; Y_a) = \mathsf{Base}(\bigcup_{a\in X} Y_a)$.

1.7.3. DÉFINITION (Opérateur de type). Soient $\mathscr{A}, \mathscr{B}$ des espaces cohérents et μ un cardinal régulier infini. L'*opérateur de type existentiel* est l'application

$$\exists : \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})} \times \mathsf{D}_{\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B})} \to \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{B})}$$

définie, pour tout $t \in \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A})}$ et pour tout $f \in \mathsf{D}_{\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B})}$, par

- $$\exists(t, f) = \{\tau(\exists(a \in X; Y_a))\}$$

  s'il existe une base de type de $\mathscr{A}$, X et une famille $(Y_a)_{a\in X}$ de bases de type de $\mathscr{B}$ telles que

  (1) $t = \{\tau(X)\}$

  (2) $f = \mathsf{Tr}(F)$ et $F(a) = \{\tau(Y_a)\}$ pour tout $a \in X$

  (3) $\exists(a \in X; Y_a)$ est une base de type μ-finie de $\mathscr{B}$
- dans tous les autres cas
$$\exists(t, f) = \emptyset.$$

1.7.4. Proposition (μ-stabilité)**.** *L'opérateur* $\exists$ *est une fonction binaire* μ*-stable.*

DÉMONSTRATION. De même que pour la μ-stabilité des opérateurs des types produit (proposition 1.3.8) sous-ensemble (proposition 1.4.3), et somme dépendantes (proposition 1.5.12), cette preuve est similaire à celle de la proposition 5.1.29 de la thèse de Miquel (p.179). □

1.7.5. REMARQUE (Opérateur curryfié). De même que précédemment, nous assimilerons l'opérateur $\exists$ et sa version curryfiée :

$$\exists : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}))$$

2. Modèle concret du calcul

Après avoir énoncé la théorie dans laquelle nous nous plaçons, nous présentons de manière modulaire les différents éléments du modèle concret de cohérence. Nous abordons dans un premier temps ceux repris du modèle de ICC (sous-sections 2.2 à 2.6) puis ceux propres à ICC_Σ (sous-sections 2.7 à 2.10) en montrant au fur et à mesure les axiomes du modèle abstrait. Nous terminons par la preuve de la cohérence de ICC_Σ (sous-section 2.11).

2.1. Axiomes considérés. Nous utilisons la notion de cardinal inaccessible.

2.1.1. DÉFINITION (Cardinal inaccessible). Un cardinal μ est *inaccessible* s'il vérifie les conditions suivantes

(1) $\aleph_0 < \mu$

(2) si $\alpha < \mu$ alors $2^\alpha < \mu$

(3) si $\alpha < \mu$ et $\beta_i < \mu$ pour tout $i \in \alpha$ alors $\sup_{i\in\alpha}\beta_i < \mu$

Nous nous plaçons dans la théorie des ensembles de Zermelo-Fraenkel avec axiome du choix (ZFC). Nous supposons également l'axiome de ω-inaccessibilité.

(AI^ω) *Il existe une infinité de cardinaux inaccessibles.*

2.2. Domaine concret et équation récursive. L'ensemble $\mathscr{M}$ est implanté par un domaine de cohérence $\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$ associé à un espace cohérent $\mathscr{A}_{\mathsf{mod}}$.

Nous ne détaillons pas ici la construction de l'espace cohérent[3] mais nous contentons de supposer son existence et de préciser certaines caractéristiques de cet espace nécessaires pour montrer que le modèle concret est bien une instance du modèle abstrait.

Ainsi la proposition suivante regroupe les informations utiles sur l'espace cohérent du modèle.

2.2.1. Proposition. *Il existe*

- *une suite de cardinaux inaccessibles strictement croissante* $(\mu_i)_{i>0}$ *(et nous posons* $\mu_0 = \aleph_0$*)*
- *un cardinal* μ *régulier supérieur à tous les cardinaux* μ_i
- *une suite d'espaces cohérents* $(\mathscr{A}_x)_{x<\mu}$
- *un espace cohérent* $\mathscr{A}_{\mathsf{mod}}$

tels que

(1) si $y < x$ *alors* $\mathscr{A}_x \Subset \mathscr{A}_y$,

(2) pour tout $i > 0$ *et pour tout* $x < \mu_i$, $\overline{\overline{|\mathscr{A}_x|}} < \mu_i$,

(3) pour tout $i > 0$, $\overline{\overline{|\mathscr{A}_{\mu_i}|}} < \mu_i$,

(4) $\mathscr{A}_{\mathsf{mod}} = \operatorname*{colim}_{x<\mu} \mathscr{A}_x$,

(5) $\mathscr{A}_{\mathsf{mod}} = (\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}) \oplus \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})$.

DÉMONSTRATION. Miquel a démontré ces résultats au paragraphe 5.2.2 de sa thèse (pages 182 à 185). Voir en particulier le lemme 5.2.2 pour l'équation récursive et le lemme 5.2.5 pour la cardinalité des espaces $\mathscr{A}_x$. □

3. Elle est décrite dans les sous-sections 5.2.1 et 5.2.2 (p. 181 à 185) de la thèse de Miquel.

L'ensemble d'interprétation $\mathscr{M}$ du modèle est défini par $\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$, l'ensemble des cliques de l'espace cohérent $\mathscr{A}_{\mathsf{mod}}$.

L'équation récursive vérifiée par l'espace est typique des modèles du λ-calcul pur ([**Bar84**]) car l'espace fonctionnel $\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}$ est inclus dans $\mathscr{A}_{\mathsf{mod}}$. L'espace cohérent $\mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})$ permet d'interpréter les types.

Par ailleurs, le fait que $\mathscr{A}_{\mathsf{mod}}$ est la colimite de sous-espaces cohérents rigidement inclus les uns dans les autres est lié à la hiérarchie d'univers dans le système de types du calcul implicite.

2.3. Ensemble des types, fonction de décodage et relation d'ordre.

Ensemble des types. Puisque nous pouvons utiliser la définition simplifiée de la somme directe, les cliques de l'espace cohérent $\mathscr{A}_{\mathsf{mod}}$ sont soit des (traces de) fonctions, soit de la forme $\{\tau(X)\}$.[4] Nous définissons alors l'ensemble des codes de type comme l'ensemble des cliques de $\mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})$ à l'exclusion de la clique vide.

2.3.1. DÉFINITION (Ensemble des types).

$$\begin{aligned}\mathscr{T} &= \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})} \setminus \{\emptyset\} \\ &= \{\{\tau(X)\} \mid X \text{ base de type } \mu\text{-finie de } \mathscr{A}_{\mathsf{mod}}\}\end{aligned}$$

Fonction de décodage. La fonction de décodage fait le lien entre les deux versants de l'interprétation des types : à partir de la clique interprétant un type (type vu comme un terme), elle renvoie un ensemble de cliques caractérisant les termes de ce type (type vu comme un ensemble de termes).

2.3.2. DÉFINITION (Fonction de décodage). La *fonction de décodage* est définie par

$$\begin{array}{rccc}\mathsf{El} & : & \mathscr{T} & \to & \mathbb{P}(\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}) \\ & & t = \{\tau(X)\} & \mapsto & \uparrow X\end{array}$$

Relation d'ordre. La relation d'ordre est l'inclusion ensembliste.

2.3.3. DÉFINITION (Relation d'ordre). Soient a, a' deux cliques de $\mathscr{A}_{\mathsf{mod}}$. Alors $a \leq a'$ si et seulement si $a \subset a'$.

Vérification des axiomes. Nous vérifions alors facilement les axiomes (1) à (3) du modèle abstrait.

2.3.4. Lemme (Axiomes (1) à (3))**.**

(1) si $t \in \mathscr{T}$ et $t \leq t'$ alors $t' \in \mathscr{T}$

(2) si $t \in \mathscr{T}$ et $t \leq t'$ alors $\mathsf{El}(t) = \mathsf{El}(t')$

(3) si $t \in \mathscr{T}$, $a \in \mathsf{El}(t)$ et $a \leq a'$ alors $a' \in \mathsf{El}(t)$

DÉMONSTRATION. (1) et (2) sont évidents car si $t \in \mathscr{T}, t' \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$ sont tels que $t \leq t'$, alors il existe une base de type μ-finie X de $\mathscr{A}_{\mathsf{mod}}$ telle que $t = t' = \{\tau(X)\}$. (3) est clair car tout type sémantique est clos supérieurement. □

4. La clique vide est bien sûr à la fois une clique de $\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}$ et de $\mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})$.

2.4. Opération binaire. L'équation récursive $\mathscr{A}_{\mathsf{mod}} = (\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}) \oplus \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})$, nous indique, puisque nous pouvons utiliser la définition simplifiée de la somme directe, que $\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}} \Subset \mathscr{A}_{\mathsf{mod}}$. Le plongement et la rétraction associés sont définis par

$$\begin{array}{rccc} \phi^+ & : & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}} & \to & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \\ & & a & \mapsto & a \end{array}$$

$$\begin{array}{rccc} \phi^- & : & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} & \to & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}} \\ & & a & \mapsto & \begin{cases} a & \text{si } a = \mathsf{Tr}(\mathrm{F}) \\ \emptyset & \text{si } a = \{\tau(\mathrm{X})\} \end{cases} \end{array}$$

2.4.1. REMARQUE. Nous avons $\Phi^- \circ \Phi^+ = \mathsf{id}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}}}$ et $\mathsf{Tr}(\Phi^+ \circ \Phi^-) \subset \mathsf{Tr}(\mathsf{id}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}})$.

Nous pouvons alors définir l'opération binaire $\cdot$. Nous utilisons une notation infixe.

2.4.2. DÉFINITION (Opération binaire). Nous définissons $\cdot : \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \times \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \to \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$ par

$$f \cdot a \quad = \quad \{\beta \in |\mathscr{A}_{\mathsf{mod}}| \mid \exists a_0 \subset a, (a_0, \beta) \in \phi^-(f)\}.$$

2.4.3. REMARQUE. Il est facile de voir que $\cdot$ est bien une opération binaire sur $\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$. Soient $f, a \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$, montrons que $f \cdot a \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$:

- si $f = \{\tau(\mathrm{X})\} \in \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})}$ alors $\phi^-(f) = \emptyset$, d'où, pour tout $a \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$, $f \cdot a = \emptyset$;
- sinon, $f = \mathsf{Tr}(\mathrm{F}) \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}}$, et pour tout $a \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$, d'après la proposition 2.1.4 page 77, $f \cdot a = \mathrm{F}(a) \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$.

2.4.4. Lemme (Axiome (a))**.** *Si $a \leq a'$ et $b \leq b'$ alors $ab \leq a'b'$.*

DÉMONSTRATION. Si $a \subset a'$, alors deux cas sont possibles.

- $a, a' \in \mathsf{D}_{\mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})}$ et dans ce cas $a \cdot b = a' \cdot b' = \emptyset$.
- Sinon, il existe des fonctions μ-stables F et F′ telles que $a = \mathsf{Tr}(\mathrm{F})$ et $a' = \mathsf{Tr}(\mathrm{F}')$. D'après la remarque 2.1.5 page 77, $\mathrm{F}(b) \subset \mathrm{F}'(b)$. Par monotonie de F′, $\mathrm{F}'(b) \subset \mathrm{F}'(b')$. Nous avons donc bien $a \cdot b = \mathrm{F}(b) \subset \mathrm{F}'(b') = a' \cdot b'$.

□

2.5. Opérateurs des types produit. Nous utilisons les opérateurs curryfiés de la définition 1.3.9 en prenant pour les deux espaces cohérents l'espace $\mathscr{A}_{\mathsf{mod}}$. Nous avons donc

$$\begin{array}{rcl} \Pi & : & \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}}) \xrightarrow{\mu} (\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}})) \\ \forall & : & \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}}) \xrightarrow{\mu} (\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}})). \end{array}$$

Le lemme suivant, ainsi que l'équation récursive vérifiée par $\mathscr{A}_{\mathsf{mod}}$, permet de justifier que $\Pi, \forall \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$.

2.5.1. Lemme. *Si $\mathscr{A}_1 \Subset \mathscr{A}_2$ et $\mathscr{B}_1 \Subset \mathscr{B}_2$ alors*

(1) $\mathsf{Ty}_\mu(\mathscr{A}_1) \Subset \mathsf{Ty}_\mu(\mathscr{A}_2)$

(2) $\mathscr{A}_1 \xrightarrow{\mu} \mathscr{B}_1 \Subset \mathscr{A}_2 \xrightarrow{\mu} \mathscr{B}_2$.

DÉMONSTRATION.

(1) Il suffit de montrer que toute base μ-finie de $\mathscr{A}_1$ est une base μ-finie de $\mathscr{A}_2$, ce qui est clair.

(2) Nous avons clairement $|\mathscr{A}_1 \xrightarrow{\mu} \mathscr{B}_1| \subset |\mathscr{A}_2 \xrightarrow{\mu} \mathscr{B}_2|$. Il suffit alors de vérifier que les relations de cohérence $\frown_{\mathscr{A}_1 \xrightarrow{\mu} \mathscr{B}_1}$ et $\frown_{\mathscr{A}_2 \xrightarrow{\mu} \mathscr{B}_2}$ sont équivalentes dans $|\mathscr{A}_1 \xrightarrow{\mu} \mathscr{B}_1|$.

□

La définition des opérateurs Π et ∀ nous montre que les axiomes suivants sont vérifiés.

2.5.2. Lemme (Axiomes (4 et (5) des opérateurs de types produit)**.**

(4) si $\Pi t u \in \mathscr{T}$ *alors*

(a) $t \in \mathscr{T}$ *et* $ua \in \mathscr{T}$ *pour tout* $a \in \mathsf{El}(t)$

(b) $\mathsf{El}(\Pi t u) = \{f \in \mathscr{M} \mid \forall a \in \mathsf{El}(t),\ fa \in \mathsf{El}(ua)\}$

(5) si $\forall t u \in \mathscr{T}$ *alors*

(a) $t \in \mathscr{T}$ *et* $ua \in \mathscr{T}$ *pour tout* $a \in \mathsf{El}(t)$

(b) $\mathsf{El}(\forall t u) = \{b \in \mathscr{M} \mid \forall a \in \mathsf{El}(t),\ b \in \mathsf{El}(ua)\} = \bigcap_{a \in \mathsf{El}(t)} \mathsf{El}(ua)$

2.6. Constantes d'univers.

2.6.1. DÉFINITION (Univers). Nous définissons les sous-ensembles de $\mathscr{T}$ suivants :

- $\mathsf{U}_0 = \{\{\tau(\mathbf{0})\}, \{\tau(\mathbf{1})\}\}$
- pour tout $i > 0$ $\mathsf{U}_i = \bigcup_{x < \mu_i} (\mathsf{Ty}(\mathscr{A}_x) \setminus \{\emptyset\})$

Ces ensembles sont appelés *univers*.

Puisque $\overline{\overline{\mathsf{U}_0}} = 2$ et pour tout $i > 0$, $\overline{\overline{\mathsf{U}_i}} \leq \mu_i$, U_i est une base de type μ-finie de $\mathscr{A}_{\mathsf{mod}}$. Cela justifie la définition suivante.

2.6.2. DÉFINITION (Constantes d'univers). Pour tout $i \in \omega$, nous posons $\mathsf{u}_i = \{\tau(\mathsf{U}_i)\}$.

2.6.3. Lemme (Axiomes des constantes d'univers)**.** *Nous avons pour tous* $i, j \in \omega$ *:*

(12) $\mathsf{El}(\mathsf{u}_i) \subset \mathscr{T}$

(13) $\mathsf{u}_i \in \mathsf{El}(\mathsf{u}_{i+1})$

(14) $\mathsf{El}(\mathsf{u}_i) \subset \mathsf{El}(\mathsf{u}_{i+1})$

(15) (partiel) si $t \in \mathsf{El}(\mathsf{u}_i)$ *et* $ua \in \mathsf{El}(\mathsf{u}_i)$ *pour tout* $a \in \mathsf{El}(t)$*, alors* $\Pi t u, \forall t u \in \mathsf{El}(\mathsf{u}_i)$

(16) si $t \in \mathsf{El}(\mathsf{u}_i)$ *et* $ua \in \mathsf{El}(\mathsf{u}_0)$ *pour tout* $a \in \mathsf{El}(t)$*, alors* $\Pi t u, \forall t u \in \mathsf{El}(\mathsf{u}_0)$

DÉMONSTRATION. Les trois premiers points découlent des définitions 2.6.1 et 2.6.2 ainsi que du fait que, pour tout $i \in \omega$, nous avons $\mathsf{El}(\mathsf{u}_i) = \mathsf{U}_i$. Pour les deux derniers points, nous renvoyons à la thèse de Miquel (cf. lemme 5.2.10 p.187 qui fait appel aux lemmes 5.1.22 et 5.1.23 p.176). □

2.7. Constantes de projection et encodage des paires.

2.7.1. DÉFINITION (Constantes de projection). Nous définissons les atomes $\mathbf{1}_{|\mathscr{A}_{\mathsf{mod}}|} = \tau(\mathbf{0}) \in |\mathscr{A}_{\mathsf{mod}}|$, $\mathbf{2}_{|\mathscr{A}_{\mathsf{mod}}|} = \tau(\mathbf{1}) \in |\mathscr{A}_{\mathsf{mod}}|$ ainsi que les cliques à un seul atome associées $\mathbf{1}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} = \{\mathbf{1}_{|\mathscr{A}_{\mathsf{mod}}|}\} = \{\tau(\mathbf{0})\} \in \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}}) \Subset \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$ et $\mathbf{2}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} = \{\mathbf{2}_{|\mathscr{A}_{\mathsf{mod}}|}\} = \{\tau(\mathbf{1})\} \in \mathsf{Ty}_\mu(\mathscr{A}_{\mathsf{mod}}) \Subset \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$.

2.7.2. REMARQUE. $\mathbf{1}_{|\mathscr{A}_{\mathsf{mod}}|}$ et $\mathbf{2}_{|\mathscr{A}_{\mathsf{mod}}|}$ sont clairement des atomes incohérents de $\mathscr{A}_{\mathsf{mod}}$. Nous pouvons donc bien utiliser la deuxième définition du type sémantique associé à la somme dépendante. [5]

5. cf. sous-section 1.5

Nous définissons ensuite la fonction d'encodage des paires. Cette fonction fait appel, via le plongement $\psi^+_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$ défini à la proposition 1.5.8, aux atomes $\mathbf{1}_{|\mathscr{A}_{\mathsf{mod}}|}$ et $\mathbf{2}_{|\mathscr{A}_{\mathsf{mod}}|}$.

2.7.3. DÉFINITION (Encodage des paires). Nous définissons la *fonction d'encodage des paires*, notée Pair, par

$$\begin{array}{rccc} \mathsf{Pair} & : & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \times \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} & \to & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \\ & & (a,b) & \mapsto & \Phi^+ \circ \psi^+_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}([a,b]) \end{array}$$

2.7.4. Lemme (Axiome (b))**.** *Soient* $a,b \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$. $\mathsf{Pair}(a,b)\cdot \mathbf{1}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} = a$ *et* $\mathsf{Pair}(a,b)\cdot \mathbf{2}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} = b$.

DÉMONSTRATION. Soit $c \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$. Nous avons, par définition de l'opération binaire $\cdot$:

$$\mathsf{Pair}(a,b)\cdot c = \{\gamma \in |\mathscr{A}_{\mathsf{mod}}| \mid \exists c_0 \subset c, (c_0,\gamma) \in \Phi^-(\mathsf{Pair}(a,b))\}.$$

Or, par définition de $\mathsf{Pair}(a,b)$:

$$\Phi^-(\mathsf{Pair}(a,b)) = (\Phi^- \circ \Phi^+)(\psi^+_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}([a,b])).$$

Puisque $\Phi^- \circ \Phi^+ = \mathsf{id}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}}$ et d'après les définitions de $\psi^+_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}, \psi_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$, et $\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}$:

$$\begin{array}{rcl} \Phi^-(\mathsf{Pair}(a,b)) & = & \{\psi_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}((i,\gamma)) \in |\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}| \mid (i,\gamma) \in [a,b]\} \\ & = & \{(\theta_{\mathbf{1}_{|\mathscr{A}|},\mathbf{2}_{|\mathscr{A}|}}(i),\gamma) \in |\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}| \mid (i,\gamma) \in [a,b]\} \\ & = & \{(\mathbf{1}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}},\alpha) \mid \alpha \in a\} \cup \{(\mathbf{2}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}},\beta) \mid \beta \in b\}. \end{array}$$

D'où $\mathsf{Pair}(a,b)\cdot \mathbf{1}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} = a$ et $\mathsf{Pair}(a,b)\cdot \mathbf{2}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} = b$. □

2.8. Fonction de génération de types.

2.8.1. DÉFINITION (Génération de types).

$$\begin{array}{rccc} \mathsf{Gen} & : & \mathbb{P}(\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}) & \to & \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \\ & & \mathrm{X} & \mapsto & \begin{cases} \{\tau(\mathsf{Base}(\mathrm{X}))\} \text{ si } \mathsf{Base}(\mathrm{X}) \text{ est une base de type } \mu\text{-finie} \\ \emptyset \text{ sinon} \end{cases} \end{array}$$

2.8.2. REMARQUE. Pour tout ensemble de cliques $\mathrm{X} \subset \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$, nous avons $\mathsf{Gen}(\mathrm{X}) \in \mathscr{T}$ ou $\mathsf{Gen}(\mathrm{X}) = \emptyset$. Si $\mathsf{Gen}(\mathrm{X}) \in \mathscr{T}$, nous avons $\mathsf{El}(\mathsf{Gen}(\mathrm{X})) = \mathsf{TSem}(\mathrm{X})$.

2.8.3. Lemme (Axiomes de Gen)**.**

(9) pour tout $\mathrm{X} \subset \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$*, si* $\mathsf{Gen}(\mathrm{X}) \in \mathscr{T}$*, alors* $\mathrm{X} \subset \mathsf{El}(\mathsf{Gen}(\mathrm{X}))$

(10) si $\mathrm{X} \subset \mathrm{X}'$ *et* $\mathsf{Gen}(\mathrm{X}), \mathsf{Gen}(\mathrm{X}') \in \mathscr{T}$*, alors* $\mathsf{El}(\mathsf{Gen}(\mathrm{X})) \subset \mathsf{El}(\mathsf{Gen}(\mathrm{X}'))$.

(11) si $\Pi(\mathsf{Gen}(\mathrm{X}))\, u \in \mathscr{T}$ *alors*

$$\{f \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}} \mid \forall a \in \mathrm{X}, f\,a \in \mathsf{El}(u\,a)\} \subset \mathsf{El}(\Pi\mathsf{Gen}(\mathrm{X})\,u)$$

DÉMONSTRATION.

(9) Évident d'après la remarque 2.8.2 et car $\mathrm{X} \subset \mathsf{TSem}(\mathrm{X})$.

(10) Il suffit, là encore d'après la remarque 2.8.2, de prouver que $\mathsf{TSem}(\mathrm{X}) \subset \mathsf{TSem}(\mathrm{X}')$. Puisque $\mathsf{TSem}(\mathrm{X}) = \uparrow \mathsf{IC}(\mathrm{X})$, il suffit de prouver que $\mathsf{IC}(\mathrm{X}) \subset \mathsf{TSem}(\mathrm{X}')$. Soit $c \in \mathsf{IC}(\mathrm{X})$. Montrons que $c \in \mathsf{TSem}(\mathrm{X}')$. Par définition de $\mathsf{IC}(\mathrm{X})$, c est l'intersection d'une famille de cliques de $\lfloor \mathrm{X} \rfloor$ compatibles dans $\mathscr{A}$. Puisque $\lfloor \mathrm{X} \rfloor \subset \mathrm{X} \subset \mathrm{X}' \subset \mathsf{TSem}(\mathrm{X})$, c est l'intersection d'une famille de cliques de $\mathsf{TSem}(\mathrm{X})$ compatibles dans $\mathscr{A}$. Puisque $\mathsf{TSem}(\mathrm{X})$, type sémantique, est stable par intersection de cliques compatibles, nous avons bien $c \in \mathsf{TSem}(\mathrm{X})$.

(11) Conséquence immédiate de la proposition 1.6.17.

□

2.9. Opérateurs de somme. Nous reprenons les opérateurs curryfiés définis dans la section précédente.

$$\sigma : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}))$$
$$\Sigma : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{A} \xrightarrow{\mu} \mathscr{A}))$$
$$\exists : \mathsf{Ty}_\mu(\mathscr{A}) \xrightarrow{\mu} (\mathscr{A} \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}) \xrightarrow{\mu} \mathsf{Ty}_\mu(\mathscr{B}))$$

2.9.1. Lemme (Axiomes des opérateurs de type somme)**.**

(6) si $\Sigma t u \in \mathscr{T}$ *alors*

(a) $t \in \mathscr{T}$ *et* $ua \in \mathscr{T}$ *pour tout* $a \in \mathsf{El}(t)$

(b) $\Sigma t u = \mathsf{Gen}(\{\mathsf{Pair}(a,b) \in \mathscr{M} \mid a \in \mathsf{El}(t), b \in \mathsf{El}(ua)\})$

(7) si $\exists t u \in \mathscr{T}$ *alors*

(a) $t \in \mathscr{T}$ *et* $ua \in \mathscr{T}$ *pour tout* $a \in \mathsf{El}(t)$

(b) $\exists t u = \mathsf{Gen}(\bigcup_{a \in \mathsf{El}(t)} \mathsf{El}(ua))$

(8) si $\sigma t u \in \mathscr{T}$ *alors*

(a) $t \in \mathscr{T}$ *et* $ua \in \mathscr{T}$ *pour tout* $a \in \mathsf{El}(t)$

(b) $\mathsf{El}(\sigma t u) = \{a \in \mathscr{M} \mid a \in \mathsf{El}(t) \wedge \mathsf{El}(ua) \neq \emptyset\}$

(15) (partiel) si $t \in \mathsf{El}(\mathsf{u}_i)$ *et* $ua \in \mathsf{El}(\mathsf{u}_i)$ *pour tout* $a \in \mathsf{El}(t)$*, alors* $\Sigma t u, \exists t u \in \mathsf{El}(\mathsf{u}_i)$*, pour tout* $i \in \omega$

(17) si $t \in \mathsf{El}(\mathsf{u}_i)$ *et* $ua \in \mathsf{El}(\mathsf{u}_j)$ *pour tout* $a \in \mathsf{El}(t)$*, alors* $\sigma t u \in \mathsf{El}(\mathsf{u}_i)$

DÉMONSTRATION. Nous déduisons des définitions des opérateurs que les axiomes (6) à (8) sont vérifiés. Les axiomes (15) et (17) se déduisent du fait que, pour un cardinal μ inaccessible ou dénombrable, si X et si tous les Y_a, pour $a \in \mathrm{X}$ sont μ-finis alors les bases de type $\Sigma^0(a \in \mathrm{X}; \mathrm{Y}_a)$ (qui se plonge injectivement sur $\Sigma(a \in \mathrm{X}; \mathrm{Y}_a)$), $\exists(a \in \mathrm{X}; \mathrm{Y}_a)$ et $\sigma(a \in \mathrm{X}; \mathrm{Y}_a)$ sont μ-finies, ce qui se vérifie sans difficulté à partir de leurs définitions et de celle des cardinaux inaccessibles. □

2.10. Fonction d'interprétation.

2.10.1. DÉFINITION (Fonction d'interprétation). Soit $\rho \in \mathbf{Val}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}}$ une valuation. Nous définissons $[\![]\!]_\rho : \Lambda_{\mathrm{ICC}_\Sigma} \to \mathscr{M}$ par induction à la figure 14

2.10.2. Lemme (Validité des axiomes de la fonction d'interprétation)**.** *La fonction d'interprétation* $[\![]\!]$ *satisfait les axiomes (c) à (r).*

DÉMONSTRATION. Les seuls axiomes qui correspondent pas directement à la définition 2.10.1 sont les axiomes (g), (h) et (r). Soient M, M' des termes de ICC_Σ, x, f des variables, $a \in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}$ et $\rho, \rho' \in \mathbf{Val}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}}$.

- (g) Par définition de l'opération binaire et car $\Phi^- \circ \Phi^+ = \mathsf{id}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}} \xrightarrow{\mu} \mathscr{A}_{\mathsf{mod}}}}$.
- (h) Découle de la représentabilité d'une fonction par sa trace.
- (r) Découle de $\mathsf{Tr}(\Phi^+ \circ \Phi^-) \subset \mathsf{Tr}(\mathsf{id}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}})$.

□

$$
\begin{aligned}
[\![x]\!]_\rho &= \rho(x)\\
\\
[\![\mathsf{Prop}]\!]_\rho &= \mathsf{u}_0\\
[\![\mathsf{Type}_i]\!]_\rho &= \mathsf{u}_i, \text{ pour i>0}\\
\\
[\![\Pi x : \mathrm{T}.\mathrm{U}]\!]_\rho &= (\Pi \cdot [\![\mathrm{T}]\!]_\rho) \cdot [\![\lambda x.\mathrm{U}]\!]_\rho\\
[\![\lambda x.\mathrm{M}]\!]_\rho &= \Phi^+(\mathsf{Tr}(a_{\in \mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}} \mapsto [\![\mathrm{M}]\!]_{\rho;x\leftarrow a}))\\
[\![\mathrm{M}\,\mathrm{N}]\!]_\rho &= [\![\mathrm{M}]\!]_\rho \cdot [\![\mathrm{N}]\!]_\rho\\
\\
[\![\forall x : \mathrm{T}.\mathrm{U}]\!]_\rho &= (\forall \cdot [\![\mathrm{T}]\!]_\rho) \cdot [\![\lambda x.\mathrm{U}]\!]_\rho\\
\\
[\![\Sigma x : \mathrm{A}.\mathrm{B}]\!]_\rho &= (\Sigma \cdot [\![\mathrm{A}]\!]_\rho) \cdot [\![\lambda x.\mathrm{B}]\!]_\rho\\
[\![(\mathrm{M},\mathrm{N})]\!]_\rho &= \mathsf{Pair}([\![\mathrm{M}]\!]_\rho, [\![\mathrm{N}]\!]_\rho)\\
[\![\mathsf{Elim}_\Sigma(xy.f, c)]\!]_\rho &= [\![f]\!]_{\rho;x\leftarrow([\![c]\!]_\rho\cdot\mathbf{1});y\leftarrow([\![c]\!]_\rho\cdot\mathbf{2})}\\
\\
[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho &= (\exists \cdot [\![\mathrm{A}]\!]_\rho) \cdot [\![\lambda x.\mathrm{B}]\!]_\rho\\
[\![(\diamond, b)]\!]_\rho &= [\![b]\!]_\rho\\
[\![\mathsf{Elim}_\exists(y.f, c)]\!]_\rho &= [\![f]\!]_{\rho;y\leftarrow[\![c]\!]_\rho}\\
\\
[\![\{x : \mathrm{A} \mid \mathrm{B}\}]\!]_\rho &= (\sigma \cdot [\![\mathrm{A}]\!]_\rho) \cdot [\![\lambda x.\mathrm{B}]\!]_\rho
\end{aligned}
$$

FIGURE 14. Fonction d'interprétation du modèle concret

2.11. Cohérence de ICC_Σ.

2.11.1. Proposition. *La structure* $(\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}, , \cdot \mathsf{Pair}, [\![]\!], \subset, \mathscr{T}, \mathsf{El}, \mathsf{Gen}, \mathbf{1}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}}, \mathbf{2}_{\mathsf{D}_{\mathscr{A}_{\mathsf{mod}}}}, \Pi, \forall, \Sigma, \exists, \sigma, (\mathsf{u}_{\mathsf{i}\,i})_{i\in\omega})$ *est un modèle abstrait de* ICC_Σ.

DÉMONSTRATION. Conséquence des lemmes 2.3.4 2.4.4 2.5.2 2.6.3 2.7.4 2.8.3 2.9.1 et 2.10.2. □

2.11.2. Corollaire. *Dans ZFC+AI*$^\omega$*,* ICC_Σ *est cohérent.*

DÉMONSTRATION. Similaire à Miquel (Théorème 5.2.12 p. 187).
D'après la proposition précédente et le corollaire 2.3.3 page 71, il suffit de montrer que nous avons un modèle 0-restreint. Posons $t_0 = \{\tau(\mathbf{0})\}$. Nous avons $t_0 \in \mathsf{El}(\mathsf{u}_0) = \{\{\tau(\mathbf{0})\}, \{\tau(\mathbf{1})\}\}$ et $\mathsf{El}(t_0) = \uparrow \mathbf{0} = \emptyset$. □

Perspectives

Nous avons montré dans cette partie comment étendre à ICC_Σ le modèle de cohérence établi par Miquel pour ICC_Σ. Une caractéristique intéressante de notre modèle est qu'il est en fait capable d'interpréter une version de ICC_Σ avec un type existentiel complètement implicite. Il permet donc également de prouver la cohérence de cette variante de ICC_Σ.

Notre extension du modèle de cohérence ne permet pas d'interpréter le type somme dépendante comme un produit cartésien. Comme nous l'avons vu (sous-section 1.5 du chapitre 5), cela est lié à l'équation récursive choisie pour définir l'ensemble d'interprétation. Il nous paraît intéressant de réfléchir à des équations récursives alternatives permettant une interprétation plus naturelle de la somme dépendante qui simplifierait tant le modèle concret que le modèle abstrait.

Alexandre Miquel définit dans sa thèse (chapitre 7) un deuxième modèle permettant de prouver la normalisation forte de ICC (avec la règle de renforcement). Ce modèle peut en première approche être compris comme un modèle de cohérence muni d'une fonction qui associe à chaque type un ensemble de termes saturés. Une question naturelle est alors de savoir s'il est possible d'étendre également le modèle de normalisation forte de Miquel à ICC_Σ.[6] Ce qui nous amènerait ensuite à nous interroger sur la possibilité de prouver la cohérence de ICC_Σ à partir de la normalisation forte : cela est le cas dans ICC mais cela reste pour l'instant une question ouverte pour ICC_Σ.

6. Notons que nous supposerons la normalisation forte de la $\beta\iota\eta$-réduction dans ICC_Σ dans les chapitres 8 et 9 pour pouvoir définir des algorithmes d'inférence de type pour $AICC_\Sigma$.

Troisième partie

$AICC_\Sigma$

Chapitre 6

Présentation syntaxique de AICC_Σ

Dans cette partie, nous présentons un deuxième calcul des constructions implicite, appelé *Calcul des Constructions implicite annoté avec sommes dépendantes.*. Nous utiliserons également l'acronyme AICC_Σ (*annotated implicit Calculus of Constructions with* Σ*-types*) pour désigner ce calcul.

De prime abord, ce calcul paraît être un Calcul des Constructions à la Church standard avec quelques particularités :

- deux produits dépendants et trois sommes dépendantes ;
- la présence de drapeaux (sous forme de crochets) qui marquent certains termes et certaines annotations.

Mais en fait nous allons prouver que AICC_Σ est une *variante décidable* de ICC_Σ. En effet, d'une part, AICC_Σ est muni d'un mécanisme d'extraction qui efface les parties marquées des termes et associe ainsi un terme de ICC_Σ à tout terme de AICC_Σ. D'autre part, dans AICC_Σ la comparaison de termes (règle de cumulativité) se fait entre termes extraits. Ce point est crucial : le calcul se fait dans ICC_Σ et non dans AICC_Σ [1] et cela va permettre à AICC_Σ de représenter d'être un système *exactement* aussi expressif que ICC_Σ.

Dans ce chapitre, nous présentons AICC_Σ et montrons qu'il vérifie les propriétés métathéoriques habituelles des PTS. Nous montrons également que tous les termes bien typés de AICC_Σ correspondent à des termes bien typés de ICC_Σ. Nous prouverons la réciproque au prochain chapitre, ce qui nous permettra d'affirmer que AICC_Σ est bien une représentation correcte et complète de ICC_Σ. [2]

1. Syntaxe

1.1. Termes du langages. Nous reprenons le même ensemble de variables **Var** et l'ensemble de sortes **Sort** utilisés dans ICC_Σ.

La syntaxe des termes du langage est décrite dans la définition 1.1.1. ICC_Σ et AICC_Σ étant tous les deux des calculs des constructions avec sommes dépendantes, leurs syntaxes ont des ressemblances. Mais la syntaxe de AICC_Σ est plus conséquente. D'une part, les abstractions et paires dépendantes ont des annotations de type (syntaxe à la Church), car nous voulons que le typage soit décidable. Par ailleurs, les constructeurs de type implicite vont avoir des constructeurs et des éliminateurs. En effet, dans AICC_Σ, les parties considérées comme logiques (non-utiles lors de l'exécution) doivent apparaître mais l'utilisateur a le choix de les marquer (avec des crochets) et ainsi les déclarer comme non-calculatoires.

1.1.1. DÉFINITION. [Termes annotés] L'ensemble des termes de AICC_Σ est noté Λ_{AICC}. Il est défini par la grammaire décrite dans la figure 15.

1. Il n'y a donc pas besoin de définir des règles de calcul dans AICC_Σ pour définir le système. Les règles qui seront introduites dans le dernier chapitre serviront uniquement à permettre d'exprimer le deuxième algorithme d'inférence de type.

2. Nous prouverons dans la quatrième partie que le typage dans AICC_Σ est décidable.

M, N, T, U	::=	x	(Variable)
	|	s	(Sorte)
	|	$\Pi x:\mathrm{T}.\mathrm{U}$	(Produit explicite)
	|	$\lambda x:\mathrm{T}.\mathrm{M}$	(Abstraction Explicite)
	|	$\mathrm{M}\,\mathrm{N}$	(Application Explicite)
	|	$\Pi[x:\mathrm{T}].\mathrm{U}$	(Produit implicite)
	|	$\lambda[x:\mathrm{T}].\mathrm{M}$	(Abstraction Implicite)
	|	$\mathrm{M}[\mathrm{N}]$	(Application Implicite)
	|	$\Sigma x:\mathrm{T}.\mathrm{U}$	(Somme dépendante explicite)
	|	$(\mathrm{M},\mathrm{N})_{\Sigma x:\mathrm{T}.\mathrm{U}}$	(Paire dépendante explicite)
	|	$\mathsf{Elim}_{\mathrm{E}}((x:\mathrm{A}).(y:\mathrm{B}).f,c,\mathrm{P})$	(Éliminateur de Somme dépendante explicite)
	|	$\Sigma[x:\mathrm{T}].\mathrm{U}$	(Somme dépendante implicite gauche)
	|	$([\mathrm{M}],\mathrm{N})_{\Sigma[x:\mathrm{T}].\mathrm{U}}$	(Paire dépendante implicite gauche)
	|	$\mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f,c,\mathrm{P})$	(Éliminateur de Somme dépendante implicite gauche)
	|	$\Sigma x:\mathrm{T}.[\mathrm{U}]$	(Somme dépendante implicite droite)
	|	$(\mathrm{M},[\mathrm{N}])_{\Sigma x:\mathrm{T}.[\mathrm{U}]}$	(Paire dépendante implicite droite)
	|	$\pi_1(c)$	(Première projection de la somme dépendante implicite droite)
	|	$\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}].f,c,\mathrm{P})$	(Éliminateur simplifié de la somme dépendante implicite droite)

FIGURE 15. Syntaxe des termes de AICC$_\Sigma$

Nous établissons une classification des termes analogue à celle des termes de ICC$_\Sigma$.

1.1.2. DÉFINITION (Classification des termes).

- Les *constructeurs de type* de AICC$_\Sigma$ sont les produits dépendants explicite et implicite et les sommes dépendantes explicite, implicite gauche et implicite droite.
- Les *constructeurs* de AICC$_\Sigma$ sont les abstractions explicite et implicite, et les paires dépendantes explicite, implicite gauche et implicite droite.
- Les *éliminateurs* de AICC$_\Sigma$ sont les applications explicite et implicite, les éliminateurs des sommes dépendantes explicite et implicite gauche ainsi que la première projection et l'éliminateur simplifié de la somme dépendante implicite droite.

1.1.3. NOTATION (Constructeur de type générique). De même que dans ICC$_\Sigma$, nous utiliserons une notation pour désigner un constructeur de type générique. Ainsi $\Box x:\mathrm{T}.\mathrm{U}$ désigne l'un des constructeurs de type $\Pi x:\mathrm{T}.\mathrm{U}$,$\Pi[x:\mathrm{T}].\mathrm{U}$,$\Sigma x:\mathrm{T}.\mathrm{U}$,$\Sigma[x:\mathrm{T}].\mathrm{U}$ ou $\Sigma x:\mathrm{T}.[\mathrm{U}]$.

1.1.4. DÉFINITION (Variables libres et liées). Les *variables libres* d'un terme M de AICC$_\Sigma$ sont notées $\mathsf{FV}(\mathrm{M})$. Elles sont définies inductivement à la figure 16. Les *variables liées* d'un terme M de AICC$_\Sigma$ sont notées $\mathsf{BV}(\mathrm{M})$. Elles sont définies inductivement à la figure 17.

1.1.5. CONVENTION. Les termes de la syntaxe sont définis à α-conversion près, c'est-à-dire au renommage près des variables liées.

$$
\begin{array}{rcl}
\mathsf{FV}(x) & = & \{x\} \\
\\
\mathsf{FV}(s) & = & \emptyset \\
\\
\mathsf{FV}(\square x:\mathrm{T}.\mathrm{U}) & = & \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{U}) \setminus \{x\}) \\
\\
\mathsf{FV}(\lambda x:\mathrm{T}.\mathrm{M}) & = & \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{M}) \setminus \{x\}) \\
\mathsf{FV}(\mathrm{M}\,\mathrm{N}) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) \\
\\
\mathsf{FV}(\lambda[x:\mathrm{T}].\mathrm{M}) & = & \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{M}) \setminus \{x\}) \\
\mathsf{FV}(\mathrm{M}\,[\mathrm{N}]) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) \\
\\
\mathsf{FV}((\mathrm{M},\mathrm{N})_{\Sigma x:\mathrm{T}.\mathrm{U}}) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) \cup \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{U}) \setminus \{x\}) \\
\mathsf{FV}(\mathsf{Elim}_{\mathrm{E}}((x:\mathrm{A}).(y:\mathrm{B}).f,c,\mathrm{P})) & = & \mathsf{FV}(\mathrm{A}) \cup (\mathsf{FV}(\mathrm{B}) \setminus \{x\}) \cup (\mathsf{FV}(f) \setminus \{x,y\}) \cup \mathsf{FV}(c) \cup \mathsf{FV}(\mathrm{P}) \\
\\
\mathsf{FV}(([\mathrm{M}],\mathrm{N})_{\Sigma[x:\mathrm{T}].\mathrm{U}}) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) \cup \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{U}) \setminus \{x\}) \\
\mathsf{FV}(\mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f,c,\mathrm{P})) & = & \mathsf{FV}(\mathrm{A}) \cup (\mathsf{FV}(\mathrm{B}) \setminus \{x\}) \cup (\mathsf{FV}(f) \setminus \{x,y\}) \cup \mathsf{FV}(c) \cup \mathsf{FV}(\mathrm{P}) \\
\\
\mathsf{FV}((\mathrm{M},[\mathrm{N}])_{\Sigma x:\mathrm{T}.[\mathrm{U}]}) & = & \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{N}) \cup \mathsf{FV}(\mathrm{T}) \cup (\mathsf{FV}(\mathrm{U}) \setminus \{x\}) \\
\mathsf{FV}(\pi_1(c)) & = & \mathsf{FV}(c) \\
\mathsf{FV}(\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}].f,c,\mathrm{P})) & = & \mathsf{FV}(\mathrm{B}) \cup (\mathsf{FV}(f) \setminus \{y\}) \cup \mathsf{FV}(c) \cup \mathsf{FV}(\mathrm{P})
\end{array}
$$

FIGURE 16. Variables libres des termes annotés

$$
\begin{array}{rcl}
\mathsf{BV}(x) & = & \emptyset \\
\\
\mathsf{BV}(s) & = & \emptyset \\
\\
\mathsf{BV}(\square x:\mathrm{T}.\mathrm{U}) & = & \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{U}) \cup \{x\} \\
\\
\mathsf{BV}(\lambda x:\mathrm{T}.\mathrm{M}) & = & \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{M}) \cup \{x\} \\
\mathsf{BV}(\mathrm{M}\,\mathrm{N}) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \\
\\
\mathsf{BV}(\lambda[x:\mathrm{T}].\mathrm{M}) & = & \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{M}) \cup \{x\} \\
\mathsf{BV}(\mathrm{M}\,[\mathrm{N}]) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \\
\\
\mathsf{BV}((\mathrm{M},\mathrm{N})_{\Sigma x:\mathrm{T}.\mathrm{U}}) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \cup \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{U}) \cup \{x\} \\
\mathsf{BV}(\mathsf{Elim}_{\mathrm{E}}((x:\mathrm{A}).(y:\mathrm{B}).f,c,\mathrm{P})) & = & \mathsf{BV}(\mathrm{A}) \cup \mathsf{BV}(\mathrm{B}) \cup \mathsf{BV}(f) \cup \{x,y\} \cup \mathsf{BV}(c) \cup \mathsf{BV}(\mathrm{P}) \\
\\
\mathsf{BV}(([\mathrm{M}],\mathrm{N})_{\Sigma[x:\mathrm{T}].\mathrm{U}}) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \cup \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{U}) \cup \{x\} \\
\mathsf{BV}(\mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f,c,\mathrm{P})) & = & \mathsf{BV}(\mathrm{A}) \cup \mathsf{BV}(\mathrm{B}) \cup \mathsf{BV}(f) \cup \{x,y\} \cup \mathsf{BV}(c) \cup \mathsf{BV}(\mathrm{P}) \\
\\
\mathsf{BV}((\mathrm{M},[\mathrm{N}])_{\Sigma x:\mathrm{T}.[\mathrm{U}]}) & = & \mathsf{BV}(\mathrm{M}) \cup \mathsf{BV}(\mathrm{N}) \cup \mathsf{BV}(\mathrm{T}) \cup \mathsf{BV}(\mathrm{U}) \cup \{x\} \\
\mathsf{BV}(\pi_1(c)) & = & \mathsf{BV}(c) \\
\mathsf{BV}(\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}].f,c,\mathrm{P})) & = & \mathsf{BV}(\mathrm{B}) \cup \mathsf{BV}(f) \cup \{y\} \cup \mathsf{BV}(c) \cup \mathsf{BV}(\mathrm{P})
\end{array}
$$

FIGURE 17. Variables liées des termes annotés

1.1.6. NOTATION (Types non-dépendants). Si $x \notin \mathsf{FV}(\mathrm{U})$, nous utiliserons les notations suivantes $\mathrm{T} \to \mathrm{U} = \Pi x:\mathrm{T}.\mathrm{U}$, $[\mathrm{T}] \to \mathrm{U} = \Pi[x:\mathrm{T}].\mathrm{U}$, $[\mathrm{T}] \times \mathrm{U} = \Sigma[x:\mathrm{T}].\mathrm{U}$, $\mathrm{T} \times [\mathrm{U}] = \Sigma x:\mathrm{T}.[\mathrm{U}]$, $\mathrm{T} \times \mathrm{U} = \Sigma x:\mathrm{T}.\mathrm{U}$.

Substitution de variables.

1.1.7. CONVENTION (Barendregt). Si M est un terme de AICC$_\Sigma$, nous renommons si besoin par α-conversion les variables liées de M de sorte que $\mathsf{BV}(\mathrm{M}) \cap \mathsf{FV}(\mathrm{M}) = \emptyset$.

1.1.8. DÉFINITION (Substitution). Soient M et R des termes de AICC$_\Sigma$ et z une variable. La substitution des occurrences libres de z dans M par le terme R est définie inductivement à la figure 18. Le respect de la convention de Barendregt évite la capture de variables.

$$
\begin{array}{rcl}
z\,[z/\mathrm{R}] & = & \mathrm{R} \\
x\,[z/\mathrm{R}] & = & x \quad (\text{avec } x \neq z) \\
\\
s\,[z/\mathrm{R}] & = & s \\
\\
(\Box x:\mathrm{T}.\mathrm{U})\,[z/\mathrm{R}] & = & \Box x:(\mathrm{T}\,[z/\mathrm{R}]).(\mathrm{U}\,[z/\mathrm{R}]) \\
\\
(\lambda x{:}\mathrm{T}.\mathrm{M})\,[z/\mathrm{R}] & = & \lambda x{:}(\mathrm{T}\,[z/\mathrm{R}]).\mathrm{M}\,[z/\mathrm{R}] \\
(\mathrm{M}\,\mathrm{N})\,[z/\mathrm{R}] & = & \mathrm{M}\,[z/\mathrm{R}]\,\mathrm{N}\,[z/\mathrm{R}] \\
\\
(\lambda[x:\mathrm{T}].\mathrm{M})\,[z/\mathrm{R}] & = & \lambda[x:(\mathrm{T}\,[z/\mathrm{R}])].\mathrm{M}\,[z/\mathrm{R}] \\
(\mathrm{M}\,[\mathrm{N}])\,[z/\mathrm{R}] & = & \mathrm{M}\,[z/\mathrm{R}]\,[\mathrm{N}\,[z/\mathrm{R}]] \\
\\
((\mathrm{M},\mathrm{N})_{\Sigma x:\mathrm{T}.\mathrm{U}})\,[z/\mathrm{R}] & = & (\mathrm{M}\,[z/\mathrm{R}],\mathrm{N}\,[z/\mathrm{R}])_{\Sigma x:(\mathrm{T}\,[z/\mathrm{R}]).\mathrm{U}\,[z/\mathrm{R}]} \\
(\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f,c,\mathrm{P}))\,[z/\mathrm{R}] & = & \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}\,[z/\mathrm{R}]).(y{:}\mathrm{B}\,[z/\mathrm{R}]).f\,[z/\mathrm{R}],c\,[z/\mathrm{R}],\mathrm{P}\,[z/\mathrm{R}]) \\
 & & (\text{avec } z \notin \{x,y\}) \\
\\
(([\mathrm{M}],\mathrm{N})_{\Sigma[x:\mathrm{T}].\mathrm{U}})\,[z/\mathrm{R}] & = & ([\mathrm{M}\,[z/\mathrm{R}]],\mathrm{N}\,[z/\mathrm{R}])_{\Sigma[x:(\mathrm{T}\,[z/\mathrm{R}])].\mathrm{U}\,[z/\mathrm{R}]} \\
(\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,c,\mathrm{P}))\,[z/\mathrm{R}] & = & \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}\,[z/\mathrm{R}]].(y{:}\mathrm{B}\,[z/\mathrm{R}]).f\,[z/\mathrm{R}],c\,[z/\mathrm{R}],\mathrm{P}\,[z/\mathrm{R}]) \\
 & & (\text{avec } z \notin \{x,y\}) \\
\\
((\mathrm{M},[\mathrm{N}])_{\Sigma x:\mathrm{T}.[\mathrm{U}]})\,[z/\mathrm{R}] & = & (\mathrm{M}\,[z/\mathrm{R}],[\mathrm{N}\,[z/\mathrm{R}]])_{\Sigma x:(\mathrm{T}\,[z/\mathrm{R}]).[\mathrm{U}\,[z/\mathrm{R}]]} \\
(\pi_1(c))\,[z/\mathrm{R}] & = & \pi_1((c\,[z/\mathrm{R}])) \\
(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,c,\mathrm{P}))\,[z/\mathrm{R}] & = & \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}\,[z/\mathrm{R}]].f\,[z/\mathrm{R}],c\,[z/\mathrm{R}],\mathrm{P}\,[z/\mathrm{R}]) \quad (\text{avec } z \neq y)
\end{array}
$$

FIGURE 18. Substitution dans AICC$_\Sigma$

Le lemme suivant est l'équivalent pour AICC$_\Sigma$ du lemme 1.2.13 page 10 et se montre de même par une induction sur U.

1.1.9. Lemme (Substitutions multiples)**.** *Soient x, y des variables distinctes,* M, N, U *des termes annotés tels que $y \notin FV(\mathrm{N})$, alors*

$$(\mathrm{U}\,[x/\mathrm{M}])\,[y/\mathrm{N}] = (\mathrm{U}\,[y/\mathrm{N}])\,[x/(\mathrm{M}\,[y/\mathrm{N}])]$$

1.1.10. Lemme (Substitution et variables libres)**.** *Soient x une variable et* M, N *des termes de* AICC$_\Sigma$*. Si $x \notin FV(\mathrm{M})$, alors* $\mathrm{M}\,[x/\mathrm{N}] = \mathrm{M}$.

DÉMONSTRATION. Similaire à celle du lemme 1.2.12 page 10 Par induction sur la structure de M. □

1.2. Extraction.

1.2.1. DÉFINITION (Extraction). La fonction d'extraction est définie inductivement à la figure 19.

$$
\begin{array}{rcl}
x^* & = & x \\
\\
s^* & = & s \\
\\
(\Pi x:\mathrm{T}.\mathrm{U})^* & = & \Pi x:\mathrm{T}^*.\mathrm{U}^* \\
(\lambda x{:}\mathrm{T}.\mathrm{M})^* & = & \lambda x.\mathrm{M}^* \\
(\mathrm{M}\,\mathrm{N})^* & = & \mathrm{M}^*\,\mathrm{N}^* \\
\\
(\Pi[x:\mathrm{T}].\mathrm{U})^* & = & \forall x:\mathrm{T}^*.\mathrm{U}^* \\
(\lambda[x:\mathrm{T}].\mathrm{M})^* & = & \mathrm{M}^* \\
(\mathrm{M}\,[\mathrm{N}])^* & = & \mathrm{M}^* \\
\\
(\Sigma x:\mathrm{T}.\mathrm{U})^* & = & \Sigma x:\mathrm{T}^*.\mathrm{U}^* \\
((\mathrm{M},\mathrm{N})_{\Sigma x:\mathrm{T}.\mathrm{U}})^* & = & (\mathrm{M}^*,\mathrm{N}^*) \\
((\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f,c,\mathrm{P}))^* & = & \mathsf{Elim}_{\Sigma}(xy.f^*,c^*) \\
\\
(\Sigma[x:\mathrm{T}].\mathrm{U})^* & = & \exists x:\mathrm{T}^*.\mathrm{U}^* \\
(([\mathrm{M}],\mathrm{N})_{\Sigma[x:\mathrm{T}].\mathrm{U}})^* & = & (\diamond,\mathrm{N}^*) \\
((\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,c,\mathrm{P}))^* & = & \mathsf{Elim}_{\exists}(y.f^*,c^*) \\
\\
(\Sigma x:\mathrm{T}.[\mathrm{U}])^* & = & \{x:\mathrm{T}^* \mid \mathrm{U}^*\} \\
((\mathrm{M},[\mathrm{N}])_{\Sigma x:\mathrm{T}.[\mathrm{U}]})^* & = & \mathrm{M}^* \\
\pi_1(c)^* & = & c^* \\
\mathsf{Elim}^{s}_{\mathrm{I_R}}([y:\mathrm{B}].f,c,\mathrm{P})^* & = & f^*
\end{array}
$$

FIGURE 19. Extraction

1.2.2. NOTATION. Soit $\Box x:\mathrm{T}.\mathrm{U}$ un constructeur de type de AICC_Σ et soient T',U' des termes non-annotés de ICC_Σ. $\Box^* x:\mathrm{T}'.\mathrm{U}'$ désigne le constructeur de type de ICC_Σ de structure analogue à celle de $(\Box x:\mathrm{T}.\mathrm{U})^*$. Ainsi par exemple si $\Box x:\mathrm{T}.\mathrm{U} = \Sigma x:\mathrm{T}.[\mathrm{U}]$ alors $\Box^* x:\mathrm{T}'.\mathrm{U}' = \{x:\mathrm{T}' \mid \mathrm{U}'\}$.

1.2.3. REMARQUE (Inversion de l'extraction). Il est intéressant de savoir quelle est la structure possible de M en fonction de celle de M^*. La définition de l'extraction nous indique que les cas suivants sont possibles :

- M a une structure analogue à celle de M^*,
- ou est une abstraction implicite ou paire implicite droite
- ou est une application implicite, une première projection ou un éliminateur simplifié.

Cette remarque justifie la définition suivante, qui distingue les termes selon l'effet de l'extraction sur la structure. Notons que nous ne considérons que les constructeurs ou éliminateurs car la structure des constructeurs de type est toujours préservée.

1.2.4. DÉFINITION (Termes implicites, termes explicites).

- Un *terme implicite* est un constructeur ou un éliminateur dont la structure n'est pas préservée par extraction.

- Un *terme explicite* est un constructeur ou un éliminateur dont la structure est préservée par extraction.
- Un *constructeur implicite* (resp. *explicite*) est un constructeur qui est un terme implicite (resp. explicite).
- Un *éliminateur implicite* (resp. *explicite*) est un éliminateur qui est un terme implicite (resp. explicite).

1.2.5. REMARQUE.

- Les constructeurs implicites sont donc les abstractions implicites et la paire implicite droite.
- Les constructeurs explicites sont donc les abstractions explicites, la paire dépendante explicite et la paire implicite gauche.
- Les eliminateurs implicites sont donc les applications implicites, la première projection et l'éliminateur simplifié de la somme dépendante implicite droite.
- Les eliminateurs explicites sont donc les applications explicites et les éliminateurs des sommes dépendantes explicite et implicite gauche.
- Le fait que les constructeurs et éliminateurs de la somme dépendante implicite gauche soient des termes explicites est une conséquence du fait que le type existentiel de ICC$_\Sigma$ n'est pas complètement implicite.

1.2.6. Lemme. *Soit* R *un terme de* AICC$_\Sigma$. *Supposons de plus que* R *est un constructeur de type ou une sorte.*

(1) Si $R^* \preceq (\Box x:T.U)^*$ *ou* $(\Box x:T.U)^* \preceq R^*$ *alors il existe* T_0, U_0 *tels que* $R = \Box x:T_0.U_0$.

(2) Si $R^* \preceq s$ *ou* $s \preceq R^*$ *alors* R *est une sorte.*

DÉMONSTRATION. Les deux points se prouvent de la même manière. Nous ne prouvons que le premier point.

- Si R est un constructeur de type de AICC$_\Sigma$ ou une sorte, alors R^* est un constructeur de type de ICC$_\Sigma$ ou une sorte et donc, en particulier, R^* n'est pas un éliminateur.
- Nous en déduisons, d'après le lemme 2.3.15 page 18 appliqué à $R^* \preceq (\Box x:T.U)^*$ (ou $(\Box x:T.U)^* \preceq R^*$), qu'il existe T_0', U_0' tels que $R^* = \Box^* x:T_0'.U_0'$.
- Puisque R est un constructeur de type ou une sorte, nous en déduisons qu'il existe T_0 et U_0 tels que $R = \Box x:T_0.U_0$.

□

1.2.7. Lemme (Extraction et substitution)**.** *Soient* M,N *des termes de* AICC$_\Sigma$ *et* $x \notin BV(M)$. *Alors* $(M[x/N])^* = M^*[x/N^*]$. *Si, de plus,* $x \notin FV(M^*)$, *alors* $(M[x/N])^* = M^*$.

DÉMONSTRATION. Par induction sur la structure de M et d'après les définitions de l'extraction, de la substitution dans AICC$_\Sigma$. □

1.3. Contextes et Jugements.

1.3.1. DÉFINITION (Contextes). Les contextes de AICC$_\Sigma$ sont définis exactement de la même manière que dans ICC$_\Sigma$:

$$\Gamma \quad ::= \quad \bullet \mid \Gamma; x:T$$

où T désigne un terme de AICC$_\Sigma$.

La définition de la substitution d'un contexte est similaire à celle de ICC$_\Sigma$.

1.3.2. DÉFINITION (Substitution d'un contexte). Soient Γ un contexte, x une variable et N un terme de AICC_Σ. Nous définissons inductivement la *substitution d'un contexte* :

$$\begin{array}{rcl} \bullet\,[x/\mathrm{N}] & = & \bullet \\ (\Gamma; y:\mathrm{T})\,[x/\mathrm{N}] & = & \begin{cases} \Gamma\,[x/\mathrm{N}] & \text{si } x = y \\ \Gamma\,[x/\mathrm{N}]; y:\mathrm{T}\,[x/\mathrm{N}] & \text{sinon.} \end{cases} \end{array}$$

Le lemme suivant est similaire au lemme 3.1.8 page 19. Sa démonstration est analogue.

1.3.3. Lemme (Substitution d'un contexte et variables libres)**.** *Soient* Γ *un contexte,* x *une variable et* N *un terme. Si* $x \notin DV(\Gamma) \cup FV(\Gamma)$*, alors* $\Gamma\,[x/\mathrm{N}] = \Gamma$.

Nous prolongeons l'extraction aux contextes.

1.3.4. DÉFINITION (Extraction de contextes). Soit Γ un contexte de AICC_Σ. Le *contexte extrait* de Γ, noté Γ^*, est un contexte de ICC_Σ défini inductivement par :

$$\begin{array}{rcl} \bullet^* & = & \bullet \\ (\Gamma; x:\mathrm{T})^* & = & \Gamma^*; x:\mathrm{T}^* \end{array}$$

1.3.5. REMARQUE. Nous vérifions immédiatement que $\mathsf{DV}(\Gamma) = \mathsf{DV}(\Gamma^*)$ pour tout contexte annoté Γ.

1.3.6. DÉFINITION (Jugements). Nous considérons similairement à ICC_Σ deux espèces de jugements : les jugements de bonne formation de contexte, $\Gamma \vdash$, et les jugements de typage, $\Gamma \vdash \mathrm{M} : \mathrm{T}$.

Nous étendons l'extraction aux jugements.

1.3.7. DÉFINITION (Extraction de jugements).

(1) L'extraction du jugement de bonne formation de AICC_Σ $\Gamma \vdash$ est le jugement de bonne formation de ICC_Σ $\Gamma^* \vdash$.

(2) L'extraction du jugement de typage de AICC_Σ $\Gamma \vdash \mathrm{M} : \mathrm{T}$ est le jugement de typage de ICC_Σ $\Gamma^* \vdash \mathrm{M}^* : \mathrm{T}^*$.

1.4. Règles de typage. Nous réutilisons les ensembles **Axiom**, **Rule** et **Rule′** définis à la sous-section 3.2 du chapitre 1.

Puisque AICC_Σ a pour but d'être une représentation de ICC_Σ, ses règles de typage sont strictement analogues à extraction près à celles de ICC_Σ.[3]

La principale différence est la présence de parties marquées, qui sont les arguments considérés comme non-calculatoires et qui, dans ICC_Σ, sont implicites.

1.4.1. DÉFINITION (Règles de typage). Les *règles de typage* sont les règles d'inférence décrites dans les figures 20 et 21.

2. Propriétés

2.1. Correction de l'extraction et cohérence. Nous montrons tout d'abord des propriétés liées à l'extraction. Ces propriétés seront utiles pour montrer les propriétés habituelles méta-théoriques des PTS (cf. sous-section 2.3 ci-après). Elles nous permettront également de déduire la cohérence de AICC_Σ à partir de celle de ICC_Σ.

3. Notons tout de même une différence mineure pour l'analogue de la règle (SUB-E-2), nécessaire car dans les algorithmes d'inférence de type, nous ne pouvons inférer inductivement que sur les sous-termes.

$$\frac{}{\bullet \vdash}\,(\text{WF-E}_\text{A}) \qquad \frac{\Gamma \vdash \text{T} : s \quad x \notin \mathsf{DV}(\Gamma)}{\Gamma; x : \text{T} \vdash}\,(\text{WF-S}_\text{A})$$

$$\frac{\Gamma \vdash \quad (x : \text{T}) \in \Gamma}{\Gamma \vdash x : \text{T}}\,(\text{VAR}_\text{A}) \qquad \frac{\Gamma \vdash \quad (s_1, s_2) \in \textbf{Axiom}}{\Gamma \vdash s_1 : s_2}\,(\text{SORT}_\text{A})$$

$$\frac{\Gamma \vdash \text{T} : s_1 \quad \Gamma; x : \text{T} \vdash \text{U} : s_2 \quad (s_1, s_2, s_3) \in \textbf{Rule}}{\Gamma \vdash \Pi x : \text{T}.\,\text{U} : s_3}\,(\text{E-PROD})$$

$$\frac{\Gamma; x : \text{T} \vdash \text{M} : \text{U} \quad \Gamma \vdash \Pi x : \text{T}.\,\text{U} : s}{\Gamma \vdash \lambda x{:}\text{T}.\,\text{M} : \Pi x : \text{T}.\,\text{U}}\,(\text{E-LAM})$$

$$\frac{\Gamma \vdash \text{M} : \Pi x : \text{T}.\,\text{U} \quad \Gamma \vdash \text{N} : \text{T}}{\Gamma \vdash \text{MN} : \text{U}\,[x/\text{N}]}\,(\text{E-APP})$$

$$\frac{\Gamma \vdash \text{T} : s_1 \quad \Gamma; x : \text{T} \vdash \text{U} : s_2 \quad (s_1, s_2, s_3) \in \textbf{Rule}}{\Gamma \vdash \Pi[x{:}\text{T}].\,\text{U} : s_3}\,(\text{I-PROD})$$

$$\frac{\Gamma; x : \text{T} \vdash \text{M} : \text{U} \quad \Gamma \vdash \Pi[x{:}\text{T}].\,\text{U} : s \quad x \notin \mathsf{FV}(\text{M}^*)}{\Gamma \vdash \lambda[x{:}\text{T}].\,\text{M} : \Pi[x{:}\text{T}].\,\text{U}}\,(\text{I-LAM})$$

$$\frac{\Gamma \vdash \text{M} : \Pi[x{:}\text{T}].\,\text{U} \quad \Gamma \vdash \text{N} : \text{T}}{\Gamma \vdash \text{M}\,[\text{N}] : \text{U}\,[x/\text{N}]}\,(\text{I-APP})$$

$$\frac{\Gamma \vdash \text{M} : \text{T} \quad \Gamma \vdash \text{T}' : s \quad \text{T}^* \preceq \text{T}'^*}{\Gamma \vdash \text{M} : \text{T}'}\,(\text{CUM}_\text{A})$$

FIGURE 20. Règles d'inférence de AICC$_\Sigma$ (calcul des constructions)

Extraction et variables libres. Nous montrons que pour tout terme annoté bien typé, les variables libres de l'extraction du terme sont parmi les variables libres du terme annoté. Ce résultat est faux en général pour les termes mal-typés. $\text{M} = \lambda[x{:}\mathsf{Prop}].\,x$ est un contre-exemple.

2.1.1. Lemme (Extraction et variables libres)**.** *Si* $\Gamma \vdash \text{M} : \text{T}$*, alors* $FV(\text{M}^*) \subset FV(\text{M})$.

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma \vdash \text{M} : \text{T}$, en considérant la dernière règle appliquée. (VAR$_\text{A}$) et (SORT$_\text{A}$) sont immédiates. Toutes les autres règles sauf (I-LAM), ($\Sigma_\exists$-E) et (SUB-E-2) se traitent immédiatement par induction. Le cas de ces trois autres règles est légèrement plus délicat et nécessite de faire appel aux conditions de bord de ces règles ($x \notin \mathsf{FV}(\text{M}^*)$, $x \notin \mathsf{FV}(f^*)$ et $y \notin \mathsf{FV}(f^*)$) car les termes typés ($\lambda[x{:}\text{T}].\,\text{M}$, $\mathsf{Elim}_{\text{I}_\text{L}}([x{:}\text{A}].(y{:}\text{B}).f, c, \text{P})$ et $\mathsf{Elim}^s_{\text{I}_\text{R}}([y{:}\text{B}].f, c, \text{P})$) perdent un lieur à l'extraction. □

Correction de l'extraction pour le typage. Nous montrons que l'extraction est correcte vis-à-vis du typage, c'est-à-dire que tout jugement dérivable de AICC$_\Sigma$ s'extrait vers un jugement dérivable de ICC$_\Sigma$. Nous montrons ainsi que AICC$_\Sigma$ se plonge vers ICC$_\Sigma$. Nous montrerons la réciproque au chapitre 7 et nous prouverons alors l'équivalence entre les jugements dérivables de AICC$_\Sigma$ et ceux de ICC$_\Sigma$.

2.1.2. Proposition (Correction de l'extraction)**.**

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s_3}\,(\Sigma_{\mathrm{A}})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}[x/a] \quad \Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash (a, b)_{\Sigma x:\mathrm{A}.\mathrm{B}} : \Sigma x : \mathrm{A}.\mathrm{B}}\,(\Sigma_{\mathrm{A}}\text{-I})$$

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma x : \mathrm{A}.\mathrm{B} \to s \quad \Gamma \vdash c : \Sigma x : \mathrm{A}.\mathrm{B} \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}}}{\Gamma \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}\,(\Sigma_{\mathrm{A}}\text{-E})$$

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \Sigma[x{:}\mathrm{A}].\mathrm{B} : s_3}\,(\Sigma_{\exists})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}[x/a] \quad \Gamma \vdash \Sigma[x{:}\mathrm{A}].\mathrm{B} : s}{\Gamma \vdash ([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}} : \Sigma[x{:}\mathrm{A}].\mathrm{B}}\,(\Sigma_{\exists}\text{-I})$$

$$\frac{\begin{array}{l}\Gamma \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}].\mathrm{B} \to s \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma[x\mathrm{A}].\mathrm{B}} \\ \Gamma \vdash c : \Sigma[x{:}\mathrm{A}].\mathrm{B} \qquad x \notin \mathsf{FV}(f^*)\end{array}}{\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}\,(\Sigma_{\exists}\text{-E})$$

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2}{\Gamma \vdash \Sigma x : \mathrm{A}.[\mathrm{B}] : s_1}\,(\Sigma_{\mathrm{SUB}})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}[x/a] \quad \Gamma \vdash \Sigma x : \mathrm{A}.[\mathrm{B}] : s}{\Gamma \vdash (a, [b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]} : \Sigma x : \mathrm{A}.[\mathrm{B}]}\,(\Sigma_{\mathrm{SUB}}\text{-I})$$

$$\frac{\Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}]}{\Gamma \vdash \pi_1(c) : \mathrm{A}}\,(\Sigma_{\mathrm{SUB}}\text{-E-1})$$

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}] \quad \Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P} \quad (\mathrm{B}[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \quad y \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P}) : \mathrm{P}}\,(\Sigma_{\mathrm{SUB}}\text{-E-2})$$

FIGURE 21. Règles d'inférence de AICC_Σ (Types somme)

(1) Si le jugement $\Gamma \vdash$ *est dérivable dans* AICC_Σ *alors le jugement* $\Gamma^* \vdash$ *est dérivable dans* ICC_Σ.

(2) Si le jugement $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *est dérivable dans* AICC_Σ *alors le jugement* $\Gamma^* \vdash \mathrm{M}^* : \mathrm{T}^*$ *est dérivable dans* ICC_Σ.

DÉMONSTRATION. Par induction mutuelle sur les dérivations de $\Gamma \vdash$ et $\Gamma \vdash \mathrm{M} : \mathrm{T}$ et vue la définition de l'extraction. La correspondance entre les règles de AICC_Σ et ICC_Σ est patente et la plupart des règles se traitent immédiatement. Notons toutefois :

- pour $(\mathrm{SORT_A})$ et $(\mathrm{VAR_A})$, par définition de l'extraction d'un contexte, il est clair que $\mathsf{DV}(\Gamma^*) = \mathsf{DV}(\Gamma)$ et que si $(x : \mathrm{T}) \in \Gamma$ alors $(x : \mathrm{T}^*) \in \Gamma^*$;
- pour les règles d'élimination des produits et d'introduction des sommes, du fait de la présence de substitutions, nous utilisons le lemme 1.2.7 (extraction et substitution) ;
- $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-2)$: Cette règle ne correspond pas directement à la règle $(\mathrm{SUB\text{-}E\text{-}2})$. Si

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}.\,[\mathrm{B}] \qquad \Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P} \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P}) : \mathrm{P}}$$

montrons que nous pouvons dériver

$$\frac{\Gamma^* \vdash \mathrm{P}^* : s \qquad \Gamma^* \vdash c^* : \{x : \mathrm{A}^* \mid \mathrm{B}^*\} \qquad \Gamma^*; y : \mathrm{B}^*\,[x/c^*] \vdash f^* : \mathrm{P}^* \qquad y \notin \mathsf{FV}(f^*)}{\Gamma^* \vdash f^* : \mathrm{P}^*} \text{ (Sub-E-2)}$$

La seule prémisse qui n'est pas une hypothèse d'induction est la prémisse typant f^*. Montrons donc $\Gamma; y : \mathrm{B}^*\,[x/c^*] \vdash f^* : \mathrm{P}^*$.

- Par hypothèse d'induction, nous avons $\Gamma; y : \mathrm{B}_0^* \vdash f^* : \mathrm{P}^*$.
- Puisque, par hypothèse, $(\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^*$, nous avons $\Gamma^*; y : \mathrm{B}^*\,[x/c^*] \preceq \Gamma^*; y : \mathrm{B}_0^*$.
- Montrons que $\Gamma; y : \mathrm{B}^*\,[x/c^*] \vdash$.
  - — Par hypothèse d'induction, nous avons $\Gamma^* \vdash c^* : \{x : \mathrm{A}^* \mid \mathrm{B}^*\}$. Nous en déduisons
    - — $\Gamma^* \vdash c^* : \mathrm{A}^*$ par (Sub-E-1).
    - — il existe une sorte s_0 telle que $\Gamma^* \vdash \{x : \mathrm{A}^* \mid \mathrm{B}^*\} : s_0$.
    - — Par inversion de $\Gamma^* \vdash \{x : \mathrm{A}^* \mid \mathrm{B}^*\} : s_0$, il existe une sorte s_B telle que $\Gamma^*; x : \mathrm{A}^* \vdash \mathrm{B}^* : s_\mathrm{B}$.
    - — Par substitutivité, nous avons $\Gamma^* \vdash \mathrm{B}^*\,[x/c^*] : s_\mathrm{B}$.
    - — Nous concluons en appliquant (Wf-S).
- Nous concluons alors grâce au corollaire 3.4.10 page 27 (cumulativité et changement de contexte).

□

Cohérence du système. La cohérence de AICC$_\Sigma$ découle immédiatement de la correction de l'extraction et de la cohérence de ICC$_\Sigma$.

2.1.3. Proposition. *Dans ZFC+AI$^\omega$,* AICC$_\Sigma$ *est cohérent.*

Démonstration. Par contraposition. Si le type annoté $\Pi \mathrm{A} : \mathsf{Prop}.\mathrm{A}$ (ou $\Pi[\mathrm{A}:\mathsf{Prop}].\mathrm{A}$) est habité dans le contexte vide de AICC$_\Sigma$, alors par correction de l'extraction, le type $\Pi \mathrm{A} : \mathsf{Prop}.\mathrm{A}$ (ou $\forall \mathrm{A} : \mathsf{Prop}.\mathrm{A}$) l'est dans ICC$_\Sigma$, ce qui contredit la cohérence de ICC$_\Sigma$ (proposition 2.11.2 page 103). □

2.2. Caractérisation des termes en fonction des types. Nous présentons dans cette sous-section quelques résultats qui permettent de caractériser la nature d'un terme bien typé en fonction de la nature de son type.

Ces résultats seront utilisés dans le chapitre 9, en particulier dans la sous-section 2.3 afin, notamment, de prouver la complétude de l'algorithme d'inférence de type.

Définitions.

2.2.1. Définition (Constructeurs associés).

- Le constructeur associé à un produit explicite $\Pi x : \mathrm{T}.\mathrm{U}$ est une abstraction explicite $\lambda x : \mathrm{T}_0.\mathrm{M}$.
- Le constructeur associé à un produit implicite $\Pi[x{:}\mathrm{T}].\mathrm{U}$ est une abstraction implicite $\lambda[x : \mathrm{T}_0].\mathrm{M}$.
- Le constructeur associé à une somme explicite $\Sigma x : \mathrm{A}.\mathrm{B}$ est une paire dépendante explicite $(a, b)_{\Sigma x:\mathrm{A}_0.\mathrm{B}_0}$.

- Le constructeur associé à une somme implicite gauche $\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$ est une paire dépendante implicite gauche $([a],b)_{\Sigma[x:\mathrm{A}_0]\,.\,\mathrm{B}_0}$.
- Le constructeur associé à une somme implicite droite $\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]$ est une paire dépendante implicite droite $(a,[b])_{\Sigma x:\mathrm{A}_0\,.\,[\mathrm{B}_0]}$.

2.2.2. DÉFINITION (Terme associé à un type). Soient M, T des termes dans AICC_Σ. M est un *terme associé* au type T si et seulement si un et un seul des cas suivants est vérifié :

- M et T sont des sortes ;
- M est un constructeur de type et T est une sorte ;
- T est un constructeur de type et M est un constructeur associé à T.

Résultats.

2.2.3. Lemme (Classification des termes bien typés)**.** *Si* $\Gamma\vdash \mathrm{M}:\mathrm{R}$ *alors un et un seul des cas suivants est vérifié :*

(1) il existe T *sorte ou constructeur de type* AICC_Σ *tel que*
- $\mathrm{T}^*\preceq \mathrm{R}^*$
- $\Gamma\vdash \mathrm{M}:\mathrm{T}$
- M *est un terme associé à* T *;*

(2) M *est une variable ;*

(3) M *est un éliminateur.*

DÉMONSTRATION. Nous montrons le résultat par induction sur la dérivation de $\Gamma\vdash \mathrm{M}:\mathrm{R}$:

- pour $(\mathrm{VAR_A})$, le cas (2) est trivialement vérifié ;
- pour la règle $(\mathrm{SORT_A})$, les règles de formation et d'introduction, le cas (1) est trivialement vérifié (avec $\mathrm{T}=\mathrm{R}$) ;
- pour les règles d'élimination, le cas (3) est trivialement vérifié ;
- enfin la règle $(\mathrm{CUM_A})$ se traite immédiatement par hypothèse d'induction.

□

Dans le lemme précédent, si M n'est pas une variable ni un éliminateur, nous ne connaissons pas en général le type T auquel M est associé. Les deux corollaires suivants montrent que si R est un constructeur de type ou une sorte, alors R peut-être ce terme T.

2.2.4. Corollaire. *Si* $\Gamma\vdash \mathrm{M}:\Box x:\mathrm{T}\,.\,\mathrm{U}$ *alors un et un seul des cas suivants est vérifié :*

(1) M *est un constructeur associé à* $\Box x:\mathrm{T}\,.\,\mathrm{U}$ *;*

(2) M *est une variable ;*

(3) M *est un éliminateur.*

DÉMONSTRATION. Nous appliquons le lemme précédent. Le seul cas non-trivial est celui où il existe R sorte ou constructeur de type de AICC_Σ tel que

- $\mathrm{R}^*\preceq(\Box x:\mathrm{T}\,.\,\mathrm{U})^*$
- $\Gamma\vdash \mathrm{M}:\mathrm{R}$
- M est un terme associé à R.

Montrons alors que M est un terme associé à $\Box x:\mathrm{T}\,.\,\mathrm{U}$.

- D'après le point (1) du lemme 1.2.6 page 114, il existe $\mathrm{T}_0,\mathrm{U}_0$ tels que $\mathrm{R}=\Box x:\mathrm{T}_0\,.\,\mathrm{U}_0$, donc R et $\Box x:\mathrm{T}\,.\,\mathrm{U}$ sont de même nature.

- Il suffit alors d'appliquer les définitions 2.2.1 et 2.2.2.

□

2.2.5. Corollaire. *Si* $\Gamma \vdash \mathrm{M} : s$ *alors un et un seul des cas suivants est vérifié :*

(1) M *est un constructeur de type ou une sorte ;*

(2) M *est une variable ;*

(3) M *est un éliminateur.*

DÉMONSTRATION. Similaire à celle du corollaire 2.2.4. □

2.3. Propriétés habituelles des PTS. Puisque AICC$_\Sigma$ est un langage à la Church, les propriétés usuelles des PTS sont globalement beaucoup plus faciles à prouver dans AICC$_\Sigma$ que dans ICC$_\Sigma$.

Premières propriétés. Nous commençons par montrer des propriétés métathéoriques de base qui sont également valables dans ICC$_\Sigma$.

2.3.1. Lemme (Bonne formation des contextes)**.** *Soient* Γ, Δ *des contextes et* M, T *des termes.*

(1) Toute dérivation de $\Gamma; \Delta \vdash$ *inclut une dérivation de* $\Gamma \vdash$.

(2) Toute dérivation de $\Gamma; \Delta \vdash \mathrm{M} : \mathrm{T}$ *inclut une dérivation de* $\Gamma \vdash$.

DÉMONSTRATION. Par induction mutuelle sur les dérivations de typage en considérant la dernière règle appliquée. Aucune règle ne pose de difficulté.

- (WF-E$_\mathrm{A}$) est immédiat.
- Pour (WF-S$_\mathrm{A}$), le résultat est trivial si $\Delta = \bullet$. Si $\Delta \neq \bullet$, nous concluons immédiatement par induction sur la prémisse.
- Pour toutes les autres règles, le contexte de la conclusion est identique au contexte d'au moins une prémisse ; nous concluons alors immédiatement par une induction sur cette prémisse.

□

2.3.2. Lemme (Déclaration des variables libres)**.** *Soit* $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ *un contexte*

(1) Si $\Gamma \vdash$, *alors* $FV(\mathrm{T}_i) \subset \{x_1, \ldots, x_{i-1}\}$ *pour tout* $i \in [1 \ldots n]$.

(2) Si $\Gamma \vdash \mathrm{M} : \mathrm{T}$, *alors* $FV(\mathrm{M}) \subset DV(\Gamma)$ *et* $FV(\mathrm{T}) \subset DV(\Gamma)$.

DÉMONSTRATION. Par induction mutuelle sur les dérivations de typage en considérant la dernière règle appliquée. Toutes les règles sont immédiates sauf (WF-S$_\mathrm{A}$). Pour (WF-S$_\mathrm{A}$), d'après le lemme 2.3.1, toute dérivation de $\Gamma; x : \mathrm{T} \vdash$ contient une dérivation de $\Gamma \vdash$. Nous concluons alors grâce aux hypothèses d'induction liées aux jugements $\Gamma \vdash$ et $\Gamma \vdash \mathrm{T} : s$. □

2.3.3. Corollaire. *Toute dérivation de* $\Gamma; x : \mathrm{T}; \Delta \vdash$ *contient une dérivation de* $\Gamma \vdash \mathrm{T} : s$, *où* s *est une sorte.*

DÉMONSTRATION. Toute dérivation de $\Gamma; x : \mathrm{T}; \Delta \vdash$ contient une dérivation de $\Gamma; x : \mathrm{T} \vdash$. Il suffit alors d'inverser (WF-S$_\mathrm{A}$). □

Affaiblissement.

2.3.4. Lemme (Affaiblissement)**.** *Soient* Γ, Γ' *des contextes tels que* $\Gamma \subseteq \Gamma'$ *et* $\Gamma' \vdash$. *Si* $\Gamma \vdash M : T$ *est dérivable, alors* $\Gamma' \vdash M : T$ *est dérivable.*

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma \vdash M : T$ en considérant la dernière règle appliquée.

- L'induction est immédiate pour toutes les règles pour lesquelles le contexte de chaque prémisse est identique au contexte de la conclusion ((VAR_A), (SORT_A), (CUM_A), règles d'élimination des produits, d'introduction des sommes et ($\Sigma_{\text{SUB}}-\text{E}-1$)).
- Pour les autres règles, certaines prémisses ont un contexte $\Gamma;\Delta$ qui étend le contexte Γ de la conclusion. La difficulté est alors de montrer que $\Gamma';\Delta \vdash$ est dérivable. Nous supposons ici, quitte à renommer par α-conversion les variables liées de M, que $\mathsf{DV}(\Gamma') \cap \mathsf{BV}(M) = \emptyset$.
  - Pour les règles de formation des types, il suffit d'affaiblir la prémisse typant le domaine et d'appliquer (WF-S_A).
  - Pour les règles restantes (introduction des produits et éliminations des sommes sauf ($\Sigma_{\text{SUB}}-\text{E}-1$)), il suffit d'utiliser, sur la prémisse dont le contexte est $\Gamma;\Delta$, le corollaire 2.3.3 une ou deux fois selon la longueur de Δ.

□

Substitutivité.

2.3.5. Lemme (Substitutivité)**.** *Si* $\Gamma \vdash N : U$ *alors*

(1) $\Gamma; z : U; \Delta \vdash \quad \Rightarrow \quad \Gamma; \Delta[z/N] \vdash$

(2) $\Gamma; z : U; \Delta \vdash M : T \quad \Rightarrow \quad \Gamma; \Delta[z/N] \vdash M[z/N] : T[z/N]$

DÉMONSTRATION. Par induction mutuelle sur les dérivations des jugements $\Gamma; z : U; \Delta \vdash$ et $\Gamma; z : U; \Delta \vdash M : T$ en considérant la dernière règle appliquée. Nous retrouvons les différents cas vus dans la preuve de la substitutivité dans ICC_Σ.

- (WF-E_A) ne peut être la dernière règle appliquée.
- (WF-S_A) Si $\Delta \neq \bullet$, nous concluons en appliquant (WF-S_A) à l'hypothèse d'induction. Sinon, nous déduisons $\Gamma \vdash$ de $\Gamma; z : T \vdash$ par le lemme de bonne formation des contextes (lemme 2.3.1).
- La règle (SORT_A), les règles de formation, et les règles (E-LAM), (Σ_A-E) et ($\Sigma_{\text{SUB}}-\text{E}-1$) se traitent immédiatement à partir des hypothèses d'induction.
- Pour (CUM_A), la condition de bord $T^* \preceq T_0^*$ devient $(T[z/N])^* \preceq (T_0[z/N])^*$. Nous concluons grâce aux lemmes 1.2.7 page 114 [4] et 2.3.2 page 14 [5]
- Pour la règle (VAR_A), il faut distinguer deux cas. Soit x la variable typée dans la conclusion. Si $z \neq x$, nous concluons immédiatement par induction. Sinon, par hypothèse d'induction, nous pouvons affaiblir $\Gamma \vdash N : U$ en $\Gamma; \Delta[z/N] \vdash N : U$ et conclure puisque $z \notin \mathsf{FV}(U)$.
- Pour (I-LAM) et ($\Sigma_\exists-\text{E}$), il faut également traiter la condition de bord. Ainsi si $x \notin \mathsf{FV}(R^*)$ est la condition de bord, nous devons montrer que $x \notin \mathsf{FV}((R[z/N])^*)$. Nous avons $(R[z/N])^* = R^*[z/N^*]$ d'où $\mathsf{FV}((R[z/N])^*) \subset \mathsf{FV}(R^*) \cup \mathsf{FV}(N^*) \subset \mathsf{FV}(R^*) \cup \mathsf{FV}(N) \subset \mathsf{FV}(R^*) \cup \mathsf{DV}(\Gamma)$ ce qui permet de conclure car $x \notin \mathsf{DV}(\Gamma)$.
- Pour ($\Sigma_{\text{SUB}}-\text{E}-2$) il faut traiter les deux conditions de bord $(B[x/\pi_1(c)])^* \preceq B_0^*$ et $y \notin \mathsf{FV}(f^*)$. La première se traite comme pour (CUM_A). La seconde se traite comme pour le cas de (I-LAM) et ($\Sigma_\exists-\text{E}$).

4. extraction et substitution : $(M[x/N])^* = M^*[x/N^*]$
5. cumulativité et substitution : si $M_1' \preceq M_2'$ alors $M_1'[x/N'] \preceq M_2'[x/N'])$.

- Pour les règles d'élimination des produits et d'introduction des sommes, à cause de la présence de substitutions, nous utilisons le lemme 1.1.9 page 112 (substitutions multiples).

□

Cumulativité de contextes.

2.3.6. Lemme (Cumulativité de contextes)**.** *Soient* Γ, Γ_0, Δ *des contextes annotés tels que* $\Delta^* \preceq \Gamma^*$ *et* $\Delta \vdash$.

(1) Si $\Gamma; \Gamma_0 \vdash$, *alors* $\Delta; \Gamma_0 \vdash$.

(2) Si $\Gamma; \Gamma_0 \vdash M : T$, *alors* $\Delta; \Gamma_0 \vdash M : T$.

DÉMONSTRATION. Preuve similaire à ICC$_\Sigma$ (lemme 3.4.9 page 27).
Par induction mutuelle sur les dérivations des jugements $\Gamma; \Gamma_0 \vdash$ et $\Gamma; \Gamma_0 \vdash M : T$. Toutes les règles sont immédiates sauf (VAR$_A$) si la variable appartient à Γ et non à Γ_0. Si

$$\frac{\Gamma; \Gamma_0 \vdash \qquad (x : T) \in \Gamma}{\Gamma; \Gamma_0 \vdash x : T}$$

Montrons que $\Delta; \Gamma_0 \vdash x : T$ est dérivable. Posons $\Gamma = \Gamma_1; x : T; \Gamma_2$ et $\Delta = \Delta_1; x : U; \Delta_2$. Par hypothèse d'induction, nous dérivons $\Delta; \Gamma_0 \vdash$, puis $\Delta; \Gamma_0 \vdash x : U$ en appliquant (VAR$_A$).

Nous avons $U^* \preceq T^*$, $\Delta_1^* \preceq \Gamma_1^*$ et $\Delta_2^* \preceq \Gamma_2^*$. La dérivation de $\Gamma; \Gamma_0 \vdash$ contient une dérivation de $\Gamma_1 \vdash T : s$, où s est une sorte. Par hypothèses d'induction, nous dérivons $\Delta_1 \vdash T : s$, puis, par affaiblissement, $\Delta; \Gamma_0 \vdash T : s$. Nous dérivons alors $\Delta; \Gamma_0 \vdash x : T$ par cumulativité. □

Un cas particulier du lemme précédent qui nous sera utile par la suite.

2.3.7. Corollaire. *Si* $\Gamma \vdash M : T$, $\Delta \vdash$ *et* $\Delta^* \preceq \Gamma^*$ *alors* $\Delta \vdash M : T$.

Lemmes d'inversion. Les lemmes d'inversion, nous l'avons déjà expliqué, servent à caractériser le type R d'un jugement $\Gamma \vdash M : R$ à partir de la connaissance de la nature de M. Dans ICC$_\Sigma$, de nombreuses règles empêchent le système d'être dirigé par la syntaxe. Dans AICC$_\Sigma$, comme dans tous les PTS, seule la règle de cumulativité (CUM$_A$) a une conclusion où la nature du terme typé n'est pas spécifiée. Il est donc assez aisé d'établir ces lemmes, qui seront centraux dans la suite de l'étude métathéorique de AICC$_\Sigma$.

2.3.8. Lemme (Lemmes d'inversion)**.** *Les lemmes d'inversion sont regroupés dans les figures 22 et 23.*

DÉMONSTRATION. La démonstration est identique pour tous les énoncés. Nous allons, par exemple prouver le lemme d'inversion pour l'abstraction implicite.

Considérons le jugement dérivable $\Gamma \vdash \lambda[x{:}T].M : R_0$. Montrons qu'il existe U et s tels que :

$$\begin{cases} \Gamma; x : T \vdash M : U \\ \Gamma \vdash \Pi[x{:}T].U : s \\ x \notin \mathsf{FV}(M^*) \\ (\Pi[x{:}T].U)^* \preceq R_0^* \end{cases}$$

Nous considérons l'arbre de dérivation de la preuve du jugement $\Gamma \vdash \lambda[x{:}T].M : R_0$. La forme du terme $\lambda[x{:}T].M$ nous indique que la dernière règle appliquée lors de la dérivation est (I-LAM) suivie de n applications de (CUM$_A$), où n est un entier naturel. Si n est nul, nous concluons directement. Sinon, nous remontons alors ces applications et définissons ainsi des types R_i et des sortes s_i pour i compris entre 0 et $n-1$:

(1) Variable et sortes

$$\Gamma \vdash x : \mathrm{R} \implies \exists \mathrm{T} \left\{ \begin{array}{l} (x:\mathrm{T}) \in \Gamma \\ \mathrm{T}^* \preceq \mathrm{R}^* \end{array} \right.$$

$$\Gamma \vdash s : \mathrm{R} \implies \exists s' \left\{ \begin{array}{l} (s,s') \in \textbf{Axiom} \\ s' \preceq \mathrm{R}^* \end{array} \right.$$

(2) Produit explicite

$$\Gamma \vdash \Pi x:\mathrm{T}.\mathrm{U} : \mathrm{R} \implies \exists s_1,s_2,s_3 \left\{ \begin{array}{l} \Gamma \vdash \mathrm{T} : s_1 \\ \Gamma; x:\mathrm{T} \vdash \mathrm{U} : s_2 \\ (s_1,s_2,s_3) \in \textbf{Rule} \\ s_3 \preceq \mathrm{R}^* \end{array} \right.$$

$$\Gamma \vdash \lambda x{:}\mathrm{T}.\mathrm{M} : \mathrm{R} \implies \exists s,\mathrm{U} \left\{ \begin{array}{l} \Gamma; x:\mathrm{T} \vdash \mathrm{M} : \mathrm{U} \\ \Gamma \vdash \Pi x:\mathrm{T}.\mathrm{U} : s \\ (\Pi x:\mathrm{T}.\mathrm{U})^* \preceq \mathrm{R}^* \end{array} \right.$$

$$\Gamma \vdash \mathrm{MN} : \mathrm{R} \implies \exists \mathrm{T},\mathrm{U} \left\{ \begin{array}{l} \Gamma \vdash \mathrm{M} : \Pi x:\mathrm{T}.\mathrm{U} \\ \Gamma \vdash \mathrm{N} : \mathrm{T} \\ (\mathrm{U}\,[x/\mathrm{N}])^* \preceq \mathrm{R}^* \end{array} \right.$$

(3) Produit implicite

$$\Gamma \vdash \Pi[x:\mathrm{T}].\mathrm{U} : \mathrm{R} \implies \exists s_1,s_2,s_3 \left\{ \begin{array}{l} \Gamma \vdash \mathrm{T} : s_1 \\ \Gamma; x:\mathrm{T} \vdash \mathrm{U} : s_2 \\ (s_1,s_2,s_3) \in \textbf{Rule} \\ s_3 \preceq \mathrm{R}^* \end{array} \right.$$

$$\Gamma \vdash \lambda[x:\mathrm{T}].\mathrm{M} : \mathrm{R} \implies \exists s,\mathrm{U} \left\{ \begin{array}{l} \Gamma; x:\mathrm{T} \vdash \mathrm{M} : \mathrm{U} \\ \Gamma \vdash \Pi[x:\mathrm{T}].\mathrm{U} : s \\ x \notin \mathsf{FV}(\mathrm{M}^*) \\ (\Pi[x:\mathrm{T}].\mathrm{U})^* \preceq \mathrm{R}^* \end{array} \right.$$

$$\Gamma \vdash \mathrm{M}[\mathrm{N}] : \mathrm{R} \implies \exists \mathrm{T},\mathrm{U} \left\{ \begin{array}{l} \Gamma \vdash \mathrm{M} : \Pi[x:\mathrm{T}].\mathrm{U} \\ \Gamma \vdash \mathrm{N} : \mathrm{T} \\ (\mathrm{U}\,[x/\mathrm{N}])^* \preceq \mathrm{R}^* \end{array} \right.$$

FIGURE 22. Lemmes d'inversion (variable, sortes et produits)

$$\frac{\Gamma \vdash \lambda[x:\mathrm{T}].\mathrm{M} : \mathrm{R}_{i+1} \qquad \Gamma \vdash \mathrm{R}_i : s_i \qquad \mathrm{R}^*_{i+1} \preceq \mathrm{R}^*_i}{\Gamma \vdash \lambda[x:\mathrm{T}].\mathrm{M} : \mathrm{R}_i}\ (\textsc{Cum}_\mathrm{A})$$

Une fois que toutes les applications des règles (CUM$_\mathrm{A}$) ont été remontées, nous savons que la règle appliquée pour dériver $\Gamma \vdash \lambda[x:\mathrm{T}].\mathrm{M} : \mathrm{R}_n$ est (I-LAM). Nous avons donc U et s tels que $\mathrm{R}_n = \Pi[x:\mathrm{T}].\mathrm{U}$ et :

$$\frac{\Gamma;[x:\mathrm{T}] \vdash \mathrm{M} : \mathrm{U} \qquad \Gamma \vdash \Pi[x:\mathrm{T}].\mathrm{U} : s \qquad x \notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma \vdash \lambda[x:\mathrm{T}].\mathrm{M} : \Pi[x:\mathrm{T}].\mathrm{U}}\ (\textsc{ImpLam})$$

Par transitivité de $\preceq$ et puisque, pour tout i, $\mathrm{R}^*_{i+1} \preceq \mathrm{R}^*_i$, nous avons bien $(\Pi[x:\mathrm{T}].\mathrm{U})^* \preceq \mathrm{R}^*_0$. □

2.3.9. REMARQUE. Les dérivations des jugements donnés par les lemmes sont clairement incluses dans les dérivations des jugements initiaux.

(4) Somme explicite

$$\Gamma \vdash \Sigma x:\mathrm{A}.\mathrm{B}:\mathrm{R} \implies \exists s_1,s_2,s_3 \begin{cases} \Gamma \vdash \mathrm{A}:s_1 \\ \Gamma;x:\mathrm{A} \vdash \mathrm{B}:s_2 \\ (s_1,s_2,s_3) \in \mathbf{Rule}' \\ s_3 \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}}:\mathrm{R} \implies \exists s \begin{cases} \Gamma \vdash a:\mathrm{A} \quad \Gamma \vdash b:\mathrm{B}[x/a] \\ \Gamma \vdash \Sigma x:\mathrm{A}.\mathrm{B}:s \\ (\Sigma x:\mathrm{A}.\mathrm{B})^* \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f,c,\mathrm{P}):\mathrm{R} \implies \exists s,s' \begin{cases} \Gamma \vdash \Sigma x:\mathrm{A}.\mathrm{B}:s' \\ \Gamma \vdash \mathrm{P}:\Sigma x:\mathrm{A}.\mathrm{B} \to s \\ \Gamma \vdash c:\Sigma x:\mathrm{A}.\mathrm{B} \\ \Gamma;x:\mathrm{A};y:\mathrm{B} \vdash f:\mathrm{P}\,(x,y)_{\Sigma x:\mathrm{A}.\mathrm{B}} \\ (\mathrm{P}\,c)^* \preceq \mathrm{R}^* \end{cases}$$

(5) Somme implicite gauche

$$\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\mathrm{B}:\mathrm{R} \implies \exists s_1,s_2,s_3 \begin{cases} \Gamma \vdash \mathrm{A}:s_1 \\ \Gamma;x:\mathrm{A} \vdash \mathrm{B}:s_2 \\ (s_1,s_2,s_3) \in \mathbf{Rule}' \\ s_3 \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash ([a],b)_{\Sigma[x\mathrm{A}]\,.\mathrm{B}}:\mathrm{R} \implies \exists s \begin{cases} \Gamma \vdash a:\mathrm{A} \quad \Gamma \vdash b:\mathrm{B}[x/a] \\ \Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\mathrm{B}:s \\ (\Sigma[x{:}\mathrm{A}]\,.\mathrm{B})^* \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,c,\mathrm{P}):\mathrm{R} \implies \exists s,s' \begin{cases} \Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\mathrm{B}:s' \\ \Gamma \vdash \mathrm{P}:\Sigma[x{:}\mathrm{A}]\,.\mathrm{B} \to s \\ \Gamma \vdash c:\Sigma[x{:}\mathrm{A}]\,.\mathrm{B} \\ \Gamma;x:\mathrm{A};y:\mathrm{B} \vdash f:\mathrm{P}\,([x],y)_{\Sigma[x\mathrm{A}]\,.\mathrm{B}} \\ x \notin \mathsf{FV}(f^*) \\ (\mathrm{P}\,c)^* \preceq \mathrm{R}^* \end{cases}$$

(6) Somme implicite droite

$$\Gamma \vdash \Sigma x:\mathrm{A}.[\mathrm{B}]:\mathrm{R} \implies \exists s_1,s_2 \begin{cases} \Gamma \vdash \mathrm{A}:s_1 \\ \Gamma;x:\mathrm{A} \vdash \mathrm{B}:s_2 \\ s_1 \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash (a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}:\mathrm{R} \implies \exists s \begin{cases} \Gamma \vdash a:\mathrm{A} \quad \Gamma \vdash b:\mathrm{B}[x/a] \\ \Gamma \vdash \Sigma x:\mathrm{A}.[\mathrm{B}]:s \\ (\Sigma x:\mathrm{A}.[\mathrm{B}])^* \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash \pi_1(c):\mathrm{R} \implies \exists \mathrm{A},\mathrm{B} \begin{cases} \Gamma \vdash c:\Sigma x:\mathrm{A}.[\mathrm{B}] \\ \mathrm{A}^* \preceq \mathrm{R}^* \end{cases}$$

$$\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f,c,\mathrm{P}):\mathrm{R} \implies \exists s,\mathrm{A},\mathrm{B} \begin{cases} \Gamma \vdash \mathrm{P}:s \\ \Gamma \vdash c:\Sigma x:\mathrm{A}.[\mathrm{B}] \\ \Gamma;y:\mathrm{B}_0 \vdash f:\mathrm{P} \\ y \notin \mathsf{FV}(f^*) \\ (\mathrm{B}[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \\ \mathrm{P}^* \preceq \mathrm{R}^* \end{cases}$$

FIGURE 23. Lemmes d'inversion (sommes)

Comme dans ICC_Σ, le lemme du type des types se prouve directement à partir des lemmes d'inversion des constructeurs de type. La principale différence est que dans AICC_Σ, les lemmes d'inversion s'obtiennent sans difficulté.

2.3.10. Lemme (Type des types)**.** *Si* $\Gamma \vdash M : T$ *alors il existe une sorte* s *telle que* $\Gamma \vdash T : s$.

DÉMONSTRATION. Similaire à celle effectuée pour ICC_Σ (proposition 5.2.4 page 50). Seuls les cas des règles d'élimination (E-APP), (I-APP) et $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-1)$ ne sont pas immédiats et nécessitent les lemmes d'inversion des constructeurs de type. □

Renforcement.

2.3.11. Lemme (Renforcement généralisé)**.** *Si* $z \notin FV(\Gamma) \cup FV(\Delta) \cup FV(M)$ *alors*

(1) $\Gamma; z : U; \Delta \vdash \quad \Rightarrow \quad \Gamma; \Delta \vdash$

(2) $\Gamma; z : U; \Delta \vdash M : T \quad \Rightarrow \quad \exists T_0, T_0^* \preceq T^* \wedge z \notin FV(T_0) \wedge \Gamma; \Delta \vdash M : T_0$

DÉMONSTRATION. Ces deux résultats se prouvent par induction mutuelle sur les dérivations de typage en considérant la dernière règle appliquée. La règle $(\mathrm{WF\text{-}E_A})$ ne peut être cette dernière règle. Pour la plupart des autres règles, les hypothèses d'induction sur les prémisses permettent de conclure immédiatement. Les cas suivants nécessitent des justifications supplémentaires.

- Pour $(\mathrm{VAR_A})$, il suffit de remarquer que la variable x est nécessairement distincte de y.
- Pour $(\mathrm{WF\text{-}S_A})$ si $\Delta = \bullet$, nous concluons par la bonne formation des contextes.
- Pour $(\mathrm{CUM_A})$, nous utilisons la transitivité de $\preceq$.

□

2.3.12. Lemme (Échange)**.** *Si* $x \notin FV(U)$*, alors*

(1) $\Gamma; x : T; y : U; \Delta \vdash \quad \Rightarrow \quad \Gamma; y : U; x : T; \Delta \vdash$

(2) $\Gamma; x : T; y : U; \Delta \vdash M : R \quad \Rightarrow \quad \Gamma; y : U; x : T; \Delta \vdash M : R$

DÉMONSTRATION. Ces deux résultats se prouvent par induction mutuelle sur les dérivations de typage en considérant la dernière règle appliquée. La règle $(\mathrm{WF\text{-}E_A})$ ne peut être cette dernière règle. Toutes les autres règles sauf $(\mathrm{WF\text{-}S_A})$ sont immédiates car dans ces règles, le contexte de la conclusion est un préfixe des contextes de toutes les prémisses. Pour $(\mathrm{WF\text{-}S_A})$, le seul cas non-immédiat est celui où $\Delta = \bullet$.

Si $\Gamma; x : T; y : U \vdash$ est dérivable, montrons que $\Gamma; y : U; x : T \vdash$ l'est également. Par inversion de $(\mathrm{WF\text{-}S_A})$, d'après le lemme de renforcement, et en convertissant par cumulativité, il existe s telle que $\Gamma \vdash U : s$, d'où $\Gamma; y : U \vdash$. D'après le corollaire 2.3.3, il existe s' telle que $\Gamma \vdash T : s'$. Par affaiblissement et en appliquant $(\mathrm{WF\text{-}S_A})$, nous avons bien $\Gamma; y : U x : T \vdash$ dérivable. □

2.3.13. Corollaire (Renforcement)**.** *Si* $\Gamma; z : U; \Delta \vdash M : T$ *avec* $z \notin FV(\Gamma) \cup FV(\Delta) \cup FV(M) \cup FV(T)$*, alors* $\Gamma; \Delta \vdash M : T$

DÉMONSTRATION. D'après le lemme 2.3.10 (type des types), il existe une sorte s telle que $\Gamma; z : U; \Delta \vdash T : s$. D'après le lemme 2.3.11 (renforcement généralisé) appliqué aux jugements $\Gamma; z : U; \Delta \vdash M : T$ et $\Gamma; z : U; \Delta \vdash T : s$, il existe T_0, R_0 tels que $T_0^* \preceq T^*$, $R_0^* \preceq s$ et $\Gamma; \Delta \vdash M : T_0$, $\Gamma; \Delta \vdash T : R_0$. Par cumulativité, nous dérivons $\Gamma; \Delta \vdash T : s$, puis $\Gamma; \Delta \vdash M : T$. □

Chapitre 7

Relèvement du typage

Nous voulons montrer que AICC_Σ et ICC_Σ sont des systèmes de type équivalents à extraction près. Nous avons déjà démontré que l'extraction est *correcte* : tout jugement dérivable de AICC_Σ s'extrait vers un jugement dérivable de ICC_Σ. Le but de ce chapitre est de démontrer que l'extraction est *complète* : pour tout jugement dérivable de ICC_Σ, il existe un jugement dérivable de AICC_Σ qui s'extrait vers lui. Nous appelons également cette propriété le *relèvement du typage de ICC*$_\Sigma$ car elle affirme que tout jugement dérivable de ICC_Σ se *relève* vers un jugement dérivable de AICC_Σ.

La méthode la plus naturelle pour prouver le relèvement est de raisonner par induction sur la dérivation du jugement de ICC_Σ en considérant la dernière règle appliquée. Puisque chaque règle de ICC_Σ correspond à une règle équivalente de AICC_Σ, la stratégie sera d'appliquer par induction le résultat aux prémisses en espérant obtenir les prémisses de la règle équivalente dans AICC_Σ.

La difficulté pour établir cette propriété est liée au fait que l'extraction efface un certain nombre d'informations et, notamment, ne préserve pas toujours la structure des termes. Ainsi, connaître la structure de R^* ne permet pas de déterminer la structure de R. En effet, R n'a pas nécessairement une structure analogue à celle de R^* : il peut également être un constructeur implicite ou un éliminateur implicite.

Or, la plupart des règles ont au moins une prémisse dont le terme ou le type a une structure définie. Plus précisément, deux cas sont possibles :

- la prémisse a une sorte pour type (quasiment toutes les règles)
- la prémisse a un constructeur de type pour terme ou type (toutes les règles d'introduction et d'élimination).

Le cas des sortes n'est pas bloquant car il est possible de convertir le type induit par la sorte de la prémisse (cf. lemme 2.1.1 ci-dessous). Pour les constructeurs de type, nous pouvons nous en sortir pour certaines [1] mais pas pour toutes [2]. Il est alors nécessaire de prouver un lemme intermédiaire. Le résultat suivant est suffisamment fort : tout type équivalent dans ICC_Σ à un constructeur de type peut-être mis sous cette forme : si $\Gamma \vdash \mathrm{R} : s$ et si $\mathrm{R}^* = \Box^* x : \mathrm{T}' . \mathrm{U}'$ alors il existe T et U tels que $\Gamma \vdash \Box x : \mathrm{T} . \mathrm{U} : s$ avec $\mathrm{T}^* = \mathrm{T}'$ et $\mathrm{U}^* = \mathrm{U}'$. Lors de nos tentatives pour prouver ce résultat, les difficultés rencontrées nous ont semblé similaires à celles liées à l'inversion des constructeurs de type dans ICC_Σ au chapitre 2. En adaptant à AICC_Σ les outils syntaxiques (substitutions typées et jugements de structure) utilisés précédemment, nous avons réussi à démontrer ce résultat (corollaire 2.1.3), rendant possible la preuve du relèvement.

Dans la section 1, nous présentons les outils syntaxiques inspirés du chapitre 2. La section 2 détaille la preuve du relèvement du typage ainsi que celles des lemmes auxiliaires mentionnés.

1. Substitutions et jugements de structure

Nous adaptons à nos besoins les outils présentés dans les sections 3 et 4 du chapitre 2.

1. introduction des produits, élimination des sommes explicites
2. élimination des produits et de la somme implicite droite, introduction des sommes

1.1. Substitutions non-typées. Nous présentons une notion de fonction de substitution pour AICC_Σ. Nous reprenons les définitions de la sous-section 3.1 du chapitre 2. Nous montrons également les liens entre les substitutions dans AICC_Σ et dans ICC_Σ.

Substitutions et termes.

1.1.1. DÉFINITION (Substitution). Une *substitution dans* AICC_Σ est une fonction de **Var** vers Λ_{AICC} qui est égale à l'identité sauf pour un nombre fini de variables.

De même que pour les substitutions dans ICC_Σ, nous définissons la notion de *domaine* d'une substitution et d'ensemble des *variables libres* d'une substitution, en utilisant les mêmes notations.

La substitution dont le domaine est vide est notée id_{A}.

De même que dans ICC_Σ (définition 3.1.2 page 35), nous étendons la substitution de variables à tous les termes.

1.1.2. DÉFINITION (Substitution d'un terme). Soit σ une substitution de AICC_Σ. Nous étendons σ à tous les termes de AICC_Σ en définissant inductivement la fonction $\hat{\sigma} : \Lambda_{\mathrm{AICC}} \to \Lambda_{\mathrm{AICC}}$:

$$
\begin{array}{rcl}
\hat{\sigma}(x) & = & \sigma(x) \\
\hat{\sigma}(s) & = & s \\
\\
\hat{\sigma}(\Box x:\mathrm{T}.\mathrm{U}) & = & \Box x:\hat{\sigma}(\mathrm{T}).\hat{\sigma}(\mathrm{U}) \\
\\
\hat{\sigma}(\lambda x{:}\mathrm{T}.\mathrm{M}) & = & \lambda x{:}\hat{\sigma}(\mathrm{T}).\hat{\sigma}(\mathrm{M}) \\
\hat{\sigma}(\mathrm{M}\,\mathrm{N}) & = & \hat{\sigma}(\mathrm{M})\,\hat{\sigma}(\mathrm{N}) \\
\\
\hat{\sigma}(\lambda[x{:}\mathrm{T}].\mathrm{M}) & = & \lambda[x{:}\hat{\sigma}(\mathrm{T})].\hat{\sigma}(\mathrm{M}) \\
\hat{\sigma}(\mathrm{M}\,[\mathrm{N}]) & = & \hat{\sigma}(\mathrm{M})\,[\hat{\sigma}(\mathrm{N})] \\
\\
\hat{\sigma}((a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}}) & = & (\hat{\sigma}(a),\hat{\sigma}(b))_{\Sigma x:\hat{\sigma}(\mathrm{A}).\hat{\sigma}(\mathrm{B})} \\
\hat{\sigma}(\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f,c,\mathrm{P})) & = & \mathsf{Elim}_{\mathrm{E}}((x{:}\hat{\sigma}(\mathrm{A})).(y{:}\hat{\sigma}(\mathrm{B})).\hat{\sigma}(f),\hat{\sigma}(c),\hat{\sigma}(\mathrm{P})) \\
\\
\hat{\sigma}(([a],b)_{\Sigma[x:\mathrm{A}].\mathrm{B}}) & = & ([\hat{\sigma}(a)],\hat{\sigma}(b))_{\Sigma[x:\hat{\sigma}(\mathrm{A})].\hat{\sigma}(\mathrm{B})} \\
\hat{\sigma}(\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,c,\mathrm{P})) & = & \mathsf{Elim}_{\mathrm{I_L}}([x{:}\hat{\sigma}(\mathrm{A})].(y{:}\hat{\sigma}(\mathrm{B})).\hat{\sigma}(f),\hat{\sigma}(c),\hat{\sigma}(\mathrm{P})) \\
\\
\hat{\sigma}((a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}) & = & (\hat{\sigma}(a),[\hat{\sigma}(b)])_{\Sigma x:\hat{\sigma}(\mathrm{A}).[\hat{\sigma}(\mathrm{B})]} \\
\hat{\sigma}(\pi_1(c)) & = & \pi_1(\hat{\sigma}(c)) \\
\hat{\sigma}(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,c,\mathrm{P})) & = & \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\hat{\sigma}(\mathrm{B})].\hat{\sigma}(f),\hat{\sigma}(c),\hat{\sigma}(\mathrm{P}))
\end{array}
$$

1.1.3. REMARQUE. De même que pour ICC_Σ, nous assimilons $\hat{\sigma}$ et σ et désignons $\hat{\sigma}(\mathrm{M})$ par $\sigma(\mathrm{M})$.

La convention suivante est analogue à la convention 3.1.4 page 36 utilisée dans ICC_Σ.

1.1.4. CONVENTION. Si M est un terme et σ une substitution dans AICC_Σ, nous supposons que $\mathsf{BV}(\mathrm{M}) \cap (\mathsf{Dom}(\sigma) \cup \mathsf{FV}(\sigma)) = \emptyset$.

1.1.5. Lemme. *Soient* σ_1, σ_2 *des substitutions et* M *un terme. Si, pour tout* $x \in \mathsf{FV}(\mathrm{M})$, $\sigma_1(x) = \sigma_2(x)$, *alors* $\sigma_1(\mathrm{M}) = \sigma_2(\mathrm{M})$.

DÉMONSTRATION. Immédiat par induction sur la structure de M. Similaire à celle du lemme 3.1.5 page 36. □

1.1.6. Lemme (Substitution et substitution de variables)**.** *Si* $x \notin \mathsf{Dom}(\sigma) \cup FV(\sigma)$ *alors* $\sigma(\mathrm{M}[x/\mathrm{N}]) = \sigma(\mathrm{M})[x/\sigma(\mathrm{N})]$.

DÉMONSTRATION. Preuve similaire à la preuve du lemme 3.1.7 page 36. □

Substitutions et contextes. Nous prolongeons la définition aux contextes similairement à ce qui a été fait dans ICC_Σ. (Cf. définition 3.1.10 page 37.)

1.1.7. DÉFINITION (Substitution d'un contexte).

$$\begin{array}{rcl} \sigma(\bullet) & = & \bullet \\ \sigma(\Gamma; x:\mathrm{T}) & = & \begin{cases} \sigma(\Gamma) & \text{si } x \in \mathsf{Dom}(\sigma) \\ \sigma(\Gamma); x:\sigma(\mathrm{T}) & \text{sinon.} \end{cases} \end{array}$$

1.1.8. REMARQUE. De même que pour ICC_Σ, nous avons $\mathsf{DV}(\sigma(\Gamma)) = \mathsf{DV}(\Gamma) \setminus \mathsf{Dom}(\sigma) \subset \mathsf{DV}(\Gamma)$.

1.1.9. Lemme. *Soit* Γ *un contexte bien formé et* σ_1, σ_2 *des substitutions telles que pour tout* $x \in DV(\Gamma)$, $\sigma_1(x) = \sigma_2(x)$. *Alors* $\sigma_1(\Gamma) = \sigma_2(\Gamma)$.

DÉMONSTRATION. Similaire à celle du lemme 3.1.12 page 37. Conséquence du lemme 1.1.5 □

Extraction et substitution.

1.1.10. DÉFINITION (Extraction d'une substitution). Soit σ une susbstitution dans AICC_Σ. Nous notons σ^* la substitution dans ICC_Σ telle que pour toute variable x, $\sigma^*(x) = (\sigma(x))^*$.

1.1.11. REMARQUE. Nous avons $\mathsf{Dom}(\sigma) = \mathsf{Dom}(\sigma^*)$.

1.1.12. Lemme (Extraction et substitution)**.** *Soit* σ *une substitution de* AICC_Σ.

- *Pour tout terme annoté* M, $(\sigma(\mathrm{M}))^* = \sigma^*(\mathrm{M}^*)$.
- *Pour tout contexte annoté* Γ, $(\sigma(\Gamma))^* = \sigma^*(\Gamma^*)$.

DÉMONSTRATION.

- Par induction sur M : les cas de base (variable, sortes) comme les cas inductifs sont immédiats.
- Par récurrence sur la longueur de Γ : immédiat si $\Gamma = \bullet$; si $\Gamma = \Gamma_0; x:\mathrm{T}$, nous concluons facilement puisque $\mathsf{Dom}(\sigma) = \mathsf{Dom}(\sigma^*)$ et que $(\sigma(\mathrm{T}))^* = \sigma^*\ (\mathrm{T}^*)$.

□

1.2. Substitutions typées. De même que pour ICC_Σ (cf. la sous-section 3.2 du chapitre 2), nous considérons une classe restreinte de substitutions afin d'obtenir de bonnes propriétés comme la préservation du typage.

1.2.1. DÉFINITION (Substitutions typées). Soient Γ un contexte et σ une substitution. Nous considérons le *jugement de typage de substitution*, noté $\_ \vdash_\mathrm{A} \sigma \div \Gamma$, qui indique que *la substitution* σ *est bien typée dans* AICC_Σ *sous le contexte* Γ.

Les jugements dérivables sont définis à partir des trois règles d'inférence suivantes, analogues à celles de ICC_Σ :

$$\frac{}{\_ \vdash_\mathrm{A} \mathsf{id}_\mathrm{A} \div \bullet}\ (\textsc{Subst}_\mathrm{A}\text{-Id})$$

$$\frac{\_ \vdash_{\mathrm{A}} \sigma \div \Gamma \qquad \Gamma; x : \mathrm{T} \vdash}{\_ \vdash_{\mathrm{A}} \sigma \div \Gamma; x : \mathrm{T}} \ (\textsc{Subst}_{\mathrm{A}}\text{-}\textsc{Gen})$$

$$\frac{\_ \vdash_{\mathrm{A}} \sigma \div \Gamma \qquad \sigma(\Gamma) \vdash \mathrm{N} : \sigma(\mathrm{T}) \qquad \Gamma; x : \mathrm{T} \vdash}{\_ \vdash_{\mathrm{A}} \sigma \cup \{x \mapsto \mathrm{N}\} \div \Gamma; x : \mathrm{T}} \ (\textsc{Subst}_{\mathrm{A}}\text{-}\textsc{Inst})$$

1.2.2. Lemme (Propriétés immédiates)**.** *Si* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$*, alors*

(1) $\Gamma \vdash$

(2) $\mathsf{Dom}(\sigma) \cup \mathit{FV}(\sigma) \subset \mathit{DV}(\Gamma)$.

(3) (Affaiblissement) si $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ *et si* $\Gamma; \Delta \vdash$ *alors* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma; \Delta$.

DÉMONSTRATION. Démonstration similaire à celle du lemme 3.2.2 page 38. □

Représentation ordonnée d'une substitution typée. De même que pour les substitutions typées dans ICC_Σ, nous pouvons ordonner les variables du domaine d'une substitution en suivant l'ordre du contexte qui la type.

La définition et les lemmes suivants sont strictement équivalentes à ceux de ICC_Σ.

1.2.3. DÉFINITION (Représentation d'une substitution ordonnée par son contexte). Si $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ avec $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ et $\sigma \neq \emptyset$, alors sa *représentation ordonnée par son contexte de typage* est

$$\sigma = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_k} \mapsto \mathrm{N}_k\}$$

où

- $k \leq n$ est le cardinal de $\mathsf{Dom}(\sigma)$,
- $i_1, \ldots, i_k$ sont des entiers tels que $1 \leq i_1 < i_2 < \cdots < i_k \leq n$ et $\mathsf{Dom}(\sigma) = \{x_{i_1}; \ldots; x_{i_k}\}$,
- $\mathrm{N}_j = \sigma(x_{i_j})$ pour tout j tel que $1 \leq j \leq k$.

Cette représentation est unique.

1.2.4. Lemme (Inversion de la dérivation d'une substitution bien typée)**.** *Si* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ *avec* $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ *et* $\sigma = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_k} \mapsto \mathrm{N}_k\}$ *la représentation ordonnée de* σ.

Pour j *compris entre 0 et* $k-1$*, nous notons* Γ_j *le contexte contenant toutes les déclarations précédant* $x_{i_{j+1}} : \mathrm{T}_{i_{j+1}}$*, soit*

$$\Gamma_j = x_1 : \mathrm{T}_1; \ldots; x_{i_j} : \mathrm{T}_{i_j}; x_{i_j+1} : \mathrm{T}_{i_j+1}; \ldots; x_{i_{j+1}-1} : \mathrm{T}_{i_{j+1}-1}$$

(avec $\Gamma_0 = \bullet$ *si* $i_1 = 1$*).*

Par ailleurs, nous notons $\sigma_0 = \mathsf{id}_{\mathrm{A}}$ *et pour tout* j *compris entre 1 et* k*,*

$$\sigma_j = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_j} \mapsto \mathrm{N}_j\}.$$

Nous avons alors, pour tout $1 \leq j \leq k$ *:*

$$\_ \vdash_{\mathrm{A}} \sigma_j \div \Gamma_{j-1}; x_{i_j} : \mathrm{T}_{i_j} \quad \sigma_{j-1}(\Gamma_{j-1}) \vdash \mathrm{N}_j : \sigma_{j-1}(\mathrm{T}_{i_j})$$

DÉMONSTRATION. Démonstration similaire à celle du lemme 3.2.4 page 39. □

Substitutions séquentielles. Comme pour ICC_Σ, nous allons montrer que des substitutions typées peuvent être comprises comme des séquences de substitutions à une variable.

1.2.5. Lemme. *Avec les notations du lemme précédent, nous avons, pour tout j tel que $0 \leq j \leq k-1$*

(1) pour tout terme M

$$\sigma_{j+1}(\mathrm{M}) = \sigma_j(\mathrm{M})\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$$

(2) pour tout contexte Δ

$$\sigma_{j+1}(\Delta) = \sigma_j(\Delta)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}]$$

DÉMONSTRATION. Démonstration similaire à celle du lemme 3.2.5 page 40. □

Une récurrence immédiate sur k prouve le corollaire suivant.

1.2.6. Corollaire (Substitutions séquentielles)**.** *Si* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ *avec* $\Gamma = x_1 : \mathrm{T}_1; \ldots; x_n : \mathrm{T}_n$ *et* $\sigma = \{x_{i_1} \mapsto \mathrm{N}_1; \ldots; x_{i_k} \mapsto \mathrm{N}_k\}$ *la représentation ordonnée de* σ*.*

Alors pour tout entier $j < k$, tout terme M *et tout contexte* Δ*, nous avons*

$$\sigma(\mathrm{M}) = \sigma_j(\mathrm{M})\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] \ldots [x_{i_k}/\mathrm{N}_k]$$
$$\sigma(\Delta) = \sigma_j(\Delta)\,[x_{i_{j+1}}/\mathrm{N}_{j+1}] \ldots [x_{i_k}/\mathrm{N}_k]$$

Préservation du typage.

1.2.7. Proposition (Préservation du typage par substitution)**.** *Si* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ *et* $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *alors* $\sigma(\Gamma) \vdash \sigma(\mathrm{M}) : \sigma(\mathrm{T})$*.*

DÉMONSTRATION. Similaire à la preuve de la proposition 3.2.7 page 41. □

1.2.8. Corollaire (Préservation de la bonne formation des contextes par substitution)**.** *Si* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ *alors* $\sigma(\Gamma) \vdash$*.*

DÉMONSTRATION. Similaire à la démonstration du corollaire 3.2.8 page 42. □

Extraction et substitution typée.

1.2.9. Lemme (Extraction et substitution typée)**.** *Si* $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ *alors* $\_ \vdash \sigma^* \div \Gamma^*$*.*

DÉMONSTRATION. Par induction sur la dérivation de $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$.

- (SUBST$_{\mathrm{A}}$-ID) immédiat car $\mathsf{id}^*_{\mathrm{A}} = \mathsf{id}$.
- (SUBST$_{\mathrm{A}}$-GEN) si

$$\frac{\_ \vdash_{\mathrm{A}} \sigma \div \Gamma_0 \qquad \Gamma_0; x : \mathrm{T} \vdash}{\_ \vdash_{\mathrm{A}} \sigma \div \Gamma_0; x : \mathrm{T}}\ (\text{SUBST}_{\mathrm{A}}\text{-GEN})$$

  Montrons que $\_ \vdash \sigma^* \div \Gamma_0^*; x : \mathrm{T}^*$. Par hypothèse d'induction, $\_ \vdash \sigma^* \div \Gamma_0^*$. Par correction de l'extraction, nous avons $\Gamma_0^*; x : \mathrm{T}^* \vdash$ d'où

$$\frac{\_ \vdash \sigma^* \div \Gamma_0^* \qquad \Gamma_0^*; x : \mathrm{T}^* \vdash}{\_ \vdash \sigma^* \div \Gamma_0^*; x : \mathrm{T}^*}\ (\text{SUBST-GEN})$$

- (SUBST$_{\mathrm{A}}$-INST) Si

$$\frac{\_ \vdash_{\mathrm{A}} \sigma_0 \div \Gamma_0 \qquad \sigma_0(\Gamma_0) \vdash \mathrm{N} : \sigma_0(\mathrm{T}) \qquad \Gamma_0; x : \mathrm{T} \vdash}{\_ \vdash_{\mathrm{A}} \sigma_0 \cup \{x \mapsto \mathrm{N}\} \div \Gamma_0; x : \mathrm{T}}\ (\text{SUBST}_{\mathrm{A}}\text{-INST})$$

  montrons que $\_ \vdash \sigma_0^* \cup \{x \mapsto \mathrm{N}^*\} \div \Gamma_0^*; x : \mathrm{T}^*$. Nous souhaitons appliquer

$$\frac{\_ \vdash \sigma_0^* \div \Gamma_0^* \qquad \sigma_0^*(\Gamma_0^*) \vdash N^* : \sigma_0^*(T^*) \qquad \Gamma_0^*; x : T^* \vdash}{\_ \vdash \sigma_0^* \cup \{x \mapsto N^*\} \div \Gamma_0^*; x : T^*} \text{ (SUBST-INST)}$$

- La première prémisse est vraie par hypothèse d'induction sur $\_ \vdash_A \sigma_0 \div \Gamma_0$.
- La deuxième prémisse est vraie par correction de l'extraction (point (2) du lemme 2.1.2 page 117) et d'après le lemme 1.1.12 (extraction et substitution).
- La troisième prémisse est vraie par correction de l'extraction (point (1) du lemme 2.1.2 page 117) et par définition de l'extraction de contextes (définition 1.3.4 page 115).

□

1.3. Jugements de structure. Nous définissons des jugements de structure dans AICC_Σ. Pour cela, nous adaptons les définitions et résultats de la section 4 du chapitre 2.

Définitions. Les définitions suivantes sont analogues à celles de la sous-section 4.1 du chapitre 2.

1.3.1. DÉFINITION (Pile de termes annotée). Une *pile de termes* est une liste ordonnée de termes annotés définie inductivement ainsi :

$$\pi \quad ::= \quad \circ \mid \mathsf{push}_A(N, \pi)$$

1.3.2. DÉFINITION (Jugement de structure annoté). Un *jugement de structure* est un jugement de la forme $\Gamma \vdash_A \pi : R \rhd X$ où Γ est un contexte annoté, π une pile de termes annotés, R et X des termes annotés.

1.3.3. REMARQUE. Lorsqu'il n'y a pas de risque de confusion, nous utiliserons les expressions plus courtes *pile de termes* et *jugement de structure*.

L'intuition pour les jugements de structure annotés est similaire à celle des jugements de structure de ICC_Σ. Ainsi, nous allons considérer un jugement de structure $\Gamma \vdash_A \pi : R \rhd X$ en liaison avec un jugement de typage $\Gamma \vdash N : R$ (où $N^* = \square^* x : T'^*.U'^*$). X sera d'une certaine manière le type naturel de R, c'est-à-dire ce que serait R s'il n'y avait pas eu dans la dérivation de $\Gamma \vdash N : R$ d'utilisation de (CUM$_A$) ou des règles d'introduction ou d'élimination des constructeurs de type implicite ((I-LAM), (Σ_{SUB}−I), (I-APP), (Σ_{SUB}−E−1), (Σ_{SUB}−E−2)). La pile de termes π contient des termes fournis lors de l'application de la règle (I-APP).

Nous adaptons les règles de dérivation des jugements de structure du chapitre 2 à notre situation. Nous ne reprenons qu'une seule règle de base, celle de la sorte, et ignorons les règles pour les constructeurs de type explicite. En effet, ces règles étaient liées, dans le chapitre 2 aux lemmes d'inversion des constructeurs tandis que la règle de base était liée à l'inversion des constructeurs de type. Or ici, nous ne voulons montrer qu'un seul résultat, le relèvement des constructeurs de type (typés par une sorte) qui est l'équivalent de l'inversion des constructeurs de type.

1.3.4. DÉFINITION (Règles de dérivation des jugements de structure). Les jugements de structure valides sont les jugements dérivés à partir des règles suivantes :

$$\frac{\Gamma \vdash}{\Gamma \vdash_A \circ : s \rhd s} \text{ (STRUCT}_A\text{-BASE-S)}$$

$$\frac{\Gamma \vdash_A \pi : R \rhd X \qquad R_0^* \preceq R^*}{\Gamma \vdash_A \pi : R_0 \rhd X} \text{ (STRUCT}_A\text{-IND-CONV)}$$

$$\frac{\Gamma \vdash_A \pi : R \rhd X}{\Gamma \vdash_A \pi : \Sigma x : R.\,[B] \rhd X} \text{ (STRUCT}_A\text{-IND-}\Sigma_{\text{SUB}}\text{)}$$

$$\frac{\Gamma \vdash_A \pi : R[x/N] \rhd X \qquad \Gamma \vdash N : A}{\Gamma \vdash_A \mathsf{push}_A(N,\pi) : \Pi[x{:}A]\,.\,R \rhd X}\ (\textsc{Struct}_A\text{-}\textsc{Ind}\text{-}I-\Pi)$$

1.3.5. REMARQUE. Du fait de l'unique règle de base, le terme X sera toujours une sorte. Nous avons cependant choisi de garder une présentation analogue à celle du chapitre 2, par souci de similarité et afin de faciliter une éventuelle extension des règles de ce jugement.

Résultats utiles. Nous présentons ici des résultats analogues à ceux de la sous-section 4.2 du chapitre 2.

Les deux lemmes suivants se démontrent immédiatement par induction sur la dérivation du jugement de structure. Ils sont similaires aux lemmes 4.2.1 et 4.2.3 page 44.

1.3.6. Lemme. *Si* $\Gamma \vdash_A \pi : R \rhd X$ *alors il existe une sorte* s *telle que* $\Gamma \vdash X : s$. *Si de plus* R *est une sorte, alors* $R \preceq X$ *et* $\pi = \circ$.

1.3.7. Lemme (Affaiblissement du jugement de structure)**.** *Si* $\Gamma \vdash_A \pi : R \rhd X$, $\Gamma \subseteq \Gamma'$ *et* $\Gamma' \vdash$ *alors* $\Gamma' \vdash_A \pi : R \rhd X$.

Le lemme suivant est analogue au lemme 4.2.5 page 44.

1.3.8. Lemme (Inversion du jugement de structure)**.**

(1) Si $\Gamma \vdash_A \pi : R \rhd X$ *avec* $(\Pi[x:T]\,.\,U)^* \preceq R^*$ *et* $\Gamma \vdash T : s$ *alors il existe un terme* N *et une pile* π' *tels que*

$$\pi = \mathsf{push}_A(N,\pi') \quad \Gamma \vdash N : T \quad \Gamma \vdash_A \pi' : U[x/N] \rhd X$$

(2) Si $\Gamma \vdash \pi : R \rhd X$ *avec* $(\Sigma x : A.[B])^* \preceq R^*$ *alors* $\Gamma \vdash \pi : A \rhd X$.

DÉMONSTRATION. Par induction sur la dérivation du jugement de structure. Preuve similaire à celle du lemme 4.2.5 page 44. □

2. Preuve du relèvement

2.1. Relèvement des sortes et des constructeurs de type.

Relèvement des sortes. Le lemme suivant sera utilisé pour la preuve du relèvement dans le cas des prémisses dont le type est une sorte.

2.1.1. Lemme (Relèvement des sortes)**.** *Si, dans* AICC_Σ, $\Gamma \vdash T : S$ *est dérivable et* $S^* = s$, *alors* $\Gamma \vdash T : s$ *est dérivable.*

DÉMONSTRATION. Il suffit de montrer qu'il existe une sorte s' telle que nous puissions appliquer la règle $(\textsc{Cum}_A)$:

$$\frac{\Gamma \vdash T : S \qquad \Gamma \vdash s : s' \qquad S^* \preceq s^*}{\Gamma \vdash T : s}$$

- Par absence de sorte maximale, il existe une sorte s' telle que $(s,s') \in$ **Axiom**.
- De $\Gamma \vdash T : S$, nous déduisons $\Gamma \vdash$ et nous concluons alors en appliquant $(\textsc{Sort}_A)$.

□

Relèvement des constructeurs de type. L'énoncé de ce lemme ressemble à celui à l'inversion des constructeurs de type de ICC_Σ (lemme 5.2.1 page 45). Leurs preuves sont aussi très similaires même si les détails changent, puisque nous sommes dans AICC_Σ.

2.1.2. Lemme. *Soient* $\Gamma, \mathrm{N}, \mathrm{R}$ *dans* AICC_Σ *et* T', U' *dans ICC*$_\Sigma$ *tels que* $\Gamma \vdash \mathrm{N} : \mathrm{R}$ *et* $\mathrm{N}^* = \Box^* x : \mathrm{T}' . \mathrm{U}'$ *et soient dans* AICC_Σ σ *une subsitution et* π *une pile de termes telles que*

$$\_ \vdash_{\mathrm{A}} \sigma \div \Gamma \quad \sigma(\Gamma) \vdash_{\mathrm{A}} \pi : \sigma(\mathrm{R}) \rhd s$$

Alors il existe des termes annotés T, U *tels que*

$$\mathrm{T}^* = \sigma^*(\mathrm{T}') \quad \mathrm{U}^* = \sigma^*(\mathrm{U}') \quad \sigma(\Gamma) \vdash \Box x : \mathrm{T} . \mathrm{U} : s$$

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma \vdash \mathrm{N} : \mathrm{R}$. Nous considérons la dernière règle de la dérivation.

D'après la définition de l'extraction et puisque N s'extrait vers un constructeur de type, N peut-être

- l'analogue dans AICC_Σ du constructeur de type considéré ;
- un constructeur implicite : abstraction implicite, paire dépendante implicite droite ;
- un éliminateur implicite : application implicite, première projection ou éliminateur simplifié de la somme dépendante implicite droite.

La dernière règle de la dérivation est donc parmi les règles $(\Box_{\mathrm{A}}\text{-Form})$, (I-Lam), (I-App), $(\mathrm{Cum}_{\mathrm{A}})$, $(\Sigma_{\mathrm{Sub}}-\mathrm{I})$, $(\Sigma_{\mathrm{Sub}}-\mathrm{E}-1)$, $(\Sigma_{\mathrm{Sub}}-\mathrm{E}-2)$.

- $(\Box_{\mathrm{A}}\text{-Form})$ Si $\mathrm{N} = \Box x : \mathrm{T} . \mathrm{U}$ et

$$\frac{\Gamma \vdash \mathrm{T} : s_1 \qquad \Gamma; x : \mathrm{T} \vdash \mathrm{U} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\Box}{\Gamma \vdash \Box x : \mathrm{T} . \mathrm{U} : s_3}$$

  avec de plus $\sigma(\Gamma) \vdash_{\mathrm{A}} \pi : s_3 \rhd s$, $\_ \vdash_{\mathrm{A}} \sigma \div \Gamma$ et T* = T', U* = U'.

  Montrons que les termes $\sigma(\mathrm{T})$ et $\sigma(\mathrm{U})$ conviennent.

  - Nous avons clairement $(\sigma(\mathrm{T}))^* = \sigma^*(\mathrm{T}')$ et $(\sigma(\mathrm{U}))^* = \sigma^*(\mathrm{U}')$.
  - En substituant par σ, nous dérivons $\sigma(\Gamma) \vdash \Box x : \sigma(\mathrm{T}) . \sigma(\mathrm{U}) : s_3$.
  - D'après le lemme 1.3.6, nous avons $s_3 \preceq s$. Nous pouvons alors par cumulativité dériver $\sigma(\Gamma) \vdash \Box x : \sigma(\mathrm{T}) . \sigma(\mathrm{U}) : s$.

- (I-Lam) Si

$$\frac{\Gamma; z : \mathrm{A} \vdash \mathrm{M} : \mathrm{B} \qquad \Gamma \vdash \Pi[z:\mathrm{A}] . \mathrm{B} : s_0 \qquad z \notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma \vdash \lambda[z:\mathrm{A}] . \mathrm{M} : \Pi[z:\mathrm{A}] . \mathrm{B}}$$

  avec $\mathrm{N}^* = (\lambda[z:\mathrm{A}] . \mathrm{M})^* = \mathrm{M}^*$ et s, σ et π telles que

$$\_ \vdash_{\mathrm{A}} \sigma \div \Gamma \quad \sigma(\Gamma) \vdash_{\mathrm{A}} \pi : \Pi[z:\sigma(\mathrm{A})] . \sigma(\mathrm{B}) \rhd s$$

  Montrons qu'il existe T, U tels que

$$\mathrm{T}^* = \sigma^*(\mathrm{T}') \quad \mathrm{U}^* = \sigma^*(\mathrm{U}') \quad \sigma(\Gamma) \vdash \Box x : \mathrm{T} . \mathrm{U} : s$$

  Nous procédons en trois étapes assez similaires à celles du cas (Gen) dans la preuve du lemme 5.2.1 page 45.

  (1) D'après le point (1) du lemme 1.3.8 (Inversion du jugement de structure) il existe un terme N_0 et une pile π_0 tels que

$$\pi = \mathsf{push}_{\mathrm{A}}(\mathrm{N}_0, \pi_0) \quad \sigma(\Gamma) \vdash \mathrm{N}_0 : \sigma(\mathrm{A}) \quad \sigma(\Gamma) \vdash_{\mathrm{A}} \pi_0 : \sigma(\mathrm{B})\,[z/\mathrm{N}_0] \rhd s$$

(2) Nous voulons appliquer par induction le lemme sur la première prémisse $\Gamma; z : A \vdash M : B$. Pour cela, posons $\Gamma_0 = \Gamma; z : A$ et $\sigma_0 = \sigma \cup \{z \mapsto N_0\}$. Nous avons $\Gamma_0 \vdash M : B$ et $M^* = N^* = \Box^* x : T'.U'$. Il suffit donc de montrer que $\_ \vdash_A \sigma_0 \div \Gamma_0$ et $\sigma_0(\Gamma_0) \vdash_A \pi_0 : \sigma_0(B) \rhd s$.

- Nous dérivons $\_ \vdash_A \sigma_0 \div \Gamma_0$ en appliquant (SUBST$_A$-INST) :

$$\frac{\_ \vdash_A \sigma \div \Gamma \qquad \sigma(\Gamma) \vdash N_0 : \sigma(A) \qquad \Gamma; z : A \vdash}{\_ \vdash_A \sigma \cup \{z \mapsto N_0\} \div \Gamma; z : A} \text{ (SUBST}_A\text{-INST)}$$

- Déduisons $\sigma_0(\Gamma_0) \vdash_A \pi_0 : \sigma_0(B) \rhd s$ de $\sigma(\Gamma) \vdash_A \pi_0 : \sigma(B)[z/N_0] \rhd s$:
  — $\sigma_0(B) = \sigma(B)[z/N_0]$ d'après le corollaire 1.2.6 ;
  — $\sigma_0(\Gamma_0) = \sigma_0(\Gamma)$ par définition de la substitution d'un contexte (définition 1.1.7) ; $\sigma_0(\Gamma) = \sigma(\Gamma)$ d'après le lemme 1.1.9.

(3) Nous en déduisons, par hypothèses d'induction, qu'il existe T, U tels que

$$T^* = \sigma_0^*(T') \quad U^* = \sigma_0^*(U') \quad \sigma_0(\Gamma_0) \vdash \Box x : T.U : s$$

Nous avons vu que $\sigma_0(\Gamma_0) = \sigma(\Gamma)$. Nous montrons $\sigma_0^*(T') = \sigma^*(T')$ et $\sigma_0^*(U') = \sigma^*(U')$ en appliquant le lemme 3.1.5 page 36, puisque $z \notin \mathsf{FV}(T') \cup \mathsf{FV}(U')$.

- (I-APP) Si

$$\frac{\Gamma \vdash N_0 : \Pi[z{:}A].B \qquad \Gamma \vdash N_1 : A}{\Gamma \vdash N_0[N_1] : B[z/N_1]}$$

avec $\_ \vdash_A \sigma \div \Gamma$ $\sigma(\Gamma) \vdash_A \pi : \sigma(B[z/N_1]) \rhd s$ et $(N_0[N_1])^* = N_0^* = \Box^* x : T'.U'$. Montrons qu'il existe T, U tels que

$$T^* = \sigma^*(T') \quad U^* = \sigma^*(U') \quad \sigma(\Gamma) \vdash \Box x : T.U : s$$

Pour cela, montrons que $\sigma(\Gamma) \vdash_A \mathsf{push}(\sigma(N_1), \pi) : \Pi[z{:}\sigma(A)].\sigma(B) \rhd s$. Nous conclurons alors en appliquant par induction le lemme à la prémisse $\Gamma \vdash N_0 : \Pi[z{:}A].B$.

Nous souhaitons utiliser la règle (STRUCT$_A$-IND-I$-\Pi$) :

$$\frac{\sigma(\Gamma) \vdash_A \pi : \sigma(B)[z/\sigma(N_1)] \rhd s \qquad \sigma(\Gamma) \vdash \sigma(N_1) : \sigma(A)}{\sigma(\Gamma) \vdash_A \mathsf{push}_A(\sigma(N_1), \pi) : \Pi[z{:}\sigma(A)].\sigma(B) \rhd s} \text{ (STRUCT}_A\text{-IND-I}-\Pi)$$

Montrons les deux prémisses.

- $\sigma(\Gamma) \vdash_A \pi : \sigma(B)[z/\sigma(N_1)] \rhd s$. se déduit facilement de $\sigma(\Gamma) \vdash_A \pi : \sigma(B[z/N_1]) \rhd s$: il suffit de montrer que $\sigma(B[z/N_1]) = \sigma(B)[z/\sigma(N_1)]$ (lemme 1.1.6).
- $\sigma(\Gamma) \vdash \sigma(N_1) : \sigma(A)$. se déduit de $\Gamma \vdash N_1 : A$ par la proposition 1.2.7 (substitution et jugement de typage).

- (CUM$_A$) Si

$$\frac{\Gamma \vdash N : R_0 \qquad \Gamma \vdash R : s_0 \qquad R_0^* \preceq R^*}{\Gamma \vdash N : R}$$

- Nous déduisons facilement $(\sigma(R_0))^* \preceq (\sigma(R))^*$ de $R_0^* \preceq R^*$ (lemme 1.1.12 et corollaire 3.1.9 page 36).
- Nous montrons alors que $\sigma(\Gamma) \vdash_A \pi : \sigma(R_0) \rhd s$ en appliquant la règle (STRUCT$_A$-IND-CONV) aux prémisses $\sigma(\Gamma) \vdash_A \pi : \sigma(R) \rhd s$ et $(\sigma(R_0))^* \preceq (\sigma(R))^*$.
- Nous concluons alors en appliquant par induction le lemme à la prémisse $\Gamma \vdash N : R_0$.

- ($\Sigma_{\text{SUB}}-$I) Si

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B[x/a] \qquad \Gamma \vdash \Sigma x : A.[B] : s}{\Gamma \vdash (a, [b])_{\Sigma x:A.[B]} : \Sigma x : A.[B]}$$

avec $N = (a,[b])_{\Sigma x:A.[B]}$ et $R = \Sigma x:A.[B]$ et $\_ \vdash_A \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash_A \pi : \Sigma x:\sigma(A).[\sigma(B)] \rhd s$.

La preuve est similaire, en plus simple, à celle du cas de l'application implicite. En inversant le jugement de structure $\sigma(\Gamma) \vdash_A \pi : \Sigma x:\sigma(A).[\sigma(B)] \rhd s$ (lemme 1.3.8), nous obtenons $\sigma(\Gamma) \vdash_A \pi : \sigma(A) \rhd s$. Nous pouvons alors appliquer par induction le lemme à la prémisse $\Gamma \vdash a : A$. Les hypothèses d'induction permettent de conclure.

- $(\Sigma_{SUB}-E-1)$

  Si

$$\frac{\Gamma \vdash c : \Sigma x:R.[B]}{\Gamma \vdash \pi_1(c) : R}$$

  avec $N = \pi_1(c)$, $\_ \vdash_A \sigma \div \Gamma$ et $\sigma(\Gamma) \vdash_A \pi : \sigma(R) \rhd s$.

  En appliquant la règle $(\text{STRUCT}_A\text{-IND-}\Sigma_{SUB})$, nous dérivons $\sigma(\Gamma) \vdash_A \pi : \sigma(\Sigma x:R.[B]) \rhd s$. Nous déduisons alors le résultat souhaité en appliquant par induction le lemme à la prémisse $\Gamma \vdash c : \Sigma x:R.[B]$, puisque $c^* = \pi_1(c)^* = N^*$.

- $(\Sigma_{SUB}-E-2)$

  Si

$$\frac{\Gamma \vdash R : s \qquad \Gamma \vdash c : \Sigma z:A.[B] \qquad \Gamma; y:B_0 \vdash f : R \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (B\,[z/\pi_1(c)])^* \preceq B_0^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{I_R}([y:B_0].f, c, R) : R}$$

  avec $\_ \vdash_A \sigma \div \Gamma$, $\sigma(\Gamma) \vdash_A \pi : \sigma(R) \rhd s$ et $N = \mathsf{Elim}^s_{I_R}([y:B_0].f, c, R)$. Notons que $N^* = f^* = \square^* x:T'.U'$.

  Montrons qu'il existe des termes T, U tels que

$$T^* = \sigma^*(T') \quad U^* = \sigma^*(U') \quad \sigma(\Gamma) \vdash \square x:T.U : s$$

  Nous procédons en trois étapes similaires à celles du cas de la règle (SUB-E-2) dans le lemme 5.2.1 page 45.

  (1) Appliquons par induction le lemme à la prémisse $\Gamma; y:B_0 \vdash f : R$. Pour cela, il suffit de montrer que

    (a) $\_ \vdash_A \sigma \div \Gamma; y:B_0$

    (b) $\sigma(\Gamma; y:B_0) \vdash_A \pi : \sigma(R) \rhd s$.

    (a) se déduit de $\_ \vdash_A \sigma \div \Gamma$ en appliquant $(\text{SUBST}_A\text{-GEN})$. (b) se montre par affaiblissement de $\sigma(\Gamma) \vdash_A \pi : \sigma(R) \rhd s$ (lemme 1.3.7).

    Les hypothèses d'induction nous indiquent alors qu'il existe des termes T_{HI}, U_{HI} tels que

$$T^*_{HI} = \sigma^*(T') \quad U^*_{HI} = \sigma^*(U') \quad \sigma(\Gamma); y:\sigma(B_0) \vdash \square x:T_{HI}.U_{HI} : s$$

    D'après les lemmes d'inversion, il existe des sortes s_T, s_U, s_0 telles que

$$\begin{array}{ll} \sigma(\Gamma); y:\sigma(B_0) \vdash T_{HI} : s_T & \sigma(\Gamma); y:\sigma(B_0); x:T_{HI} \vdash U_{HI} : s_U \\ (s_T, s_U, s_0) \in \mathbf{Rule}_\square & s_0 \preceq s \end{array}$$

    Posons $T = \mathsf{Elim}^s_{I_R}([y:\sigma(B_0)].T_{HI}, \sigma(c), s_T)$ et $U = \mathsf{Elim}^s_{I_R}([y:\sigma(B_0)].U_{HI}, \sigma(c), s_U)$. Nous avons $T^* = T^*_{HI} = \sigma^*(T')$ et $U^* = U^*_{HI} = \sigma^*(U')$. Il nous reste à montrer que $\sigma(\Gamma) \vdash \square x:T.U : s$. Pour cela il suffit de montrer que $\sigma(\Gamma) \vdash T : s_T$ et $\sigma(\Gamma); x:T \vdash U : s_U$ puis d'appliquer les règles $(\square_A\text{-FORM})$ et (CUM_A) pour conclure.

  (2) Dérivons $\sigma(\Gamma) \vdash T : s_T$. Par absence de sorte maximale, il existe une sorte s'_T telle que $(s_T, s'_T) \in \mathbf{Axiom}$. Montrons que nous pouvons appliquer la règle de typage $(\Sigma_{SUB}-E-2)$:

$$\frac{\left\{\begin{array}{l}\sigma(\Gamma)\vdash s_{\mathrm{T}}:s'_{\mathrm{T}}\\ \sigma(\Gamma)\vdash\sigma(c):\Sigma z:\sigma(\mathrm{A})\,.\,[\sigma(\mathrm{B})]\end{array}\right. \qquad \left\{\begin{array}{l}\sigma(\Gamma);y:\sigma(\mathrm{B}_0)\vdash \mathrm{T}_{\mathrm{HI}}:s_{\mathrm{T}}\\ (\sigma(\mathrm{B})\,[z/\pi_1(\sigma(c))])^*\preceq\sigma(\mathrm{B}_0)^*\\ y\notin \mathsf{FV}(\mathrm{T}^*_{\mathrm{HI}})\end{array}\right.}{\sigma(\Gamma)\vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\sigma(\mathrm{B}_0)].\mathrm{T}_{\mathrm{HI}},\sigma(c),s_{\mathrm{T}}):s_{\mathrm{T}}}$$

- Nous avons déjà montré $\sigma(\Gamma);y:\sigma(\mathrm{B}_0)\vdash \mathrm{T}_{\mathrm{HI}}:s_{\mathrm{T}}$.
- $\sigma(\Gamma)\vdash s_{\mathrm{T}}:s'_{\mathrm{T}}$ se dérive en appliquant $(\textsc{Sort}_{\mathrm{A}})$.
- En substituant $\Gamma\vdash c:\Sigma z:\mathrm{A}\,.\,[\mathrm{B}]$ par σ, nous obtenons $\sigma(\Gamma)\vdash\sigma(c):\Sigma z:\sigma(\mathrm{A})\,.\,[\sigma(\mathrm{B})]$.
- Montrons que $y\notin\mathsf{FV}(\mathrm{T}^*_{\mathrm{HI}})$, c'est-à-dire que $y\notin\mathsf{FV}(\sigma^*(\mathrm{T}'))$.
  - — Nous avons $\mathsf{FV}(\sigma^*(\mathrm{T}'))\subset\mathsf{FV}(\mathrm{T}')\cup\mathsf{FV}(\sigma^*)$ (lemme 3.1.6 page 36).
  - — De $y\notin\mathsf{FV}(f^*)$, nous déduisons $y\notin\mathsf{FV}(\mathrm{T}')$.
  - — De $\_\vdash_{\mathrm{A}}\sigma\div\Gamma$, nous déduisons $\_\vdash\sigma^*\div\Gamma^*$, puis $y\notin\mathsf{FV}(\sigma^*)$ (lemme 3.2.2 page 38).
- Montrons que $(\sigma(\mathrm{B})\,[z/\pi_1(\sigma(c))])^*\preceq(\sigma(\mathrm{B}_0))^*$.
  - — De $(\mathrm{B}\,[z/\pi_1(c)])^*\preceq\mathrm{B}_0^*$, nous déduisons $\sigma^*((\mathrm{B}\,[z/\pi_1(c)])^*)\preceq\sigma^*(\mathrm{B}_0^*)$ (corollaire 3.1.9 page 36).
  - — En appliquant le lemme 1.1.12, nous obtenons $(\sigma(\mathrm{B}\,[z/\pi_1(c)]))^*\preceq(\sigma(\mathrm{B}_0))^*$.
  - — Nous concluons en appliquant le lemme 1.1.6 à $(\sigma(\mathrm{B}\,[z/\pi_1(c)]))^*$, puisque $z\notin\mathsf{Dom}(\sigma)\cup\mathsf{FV}(\sigma)$.

(3) La troisième et dernière étape est de dériver $\sigma(\Gamma);x:\mathrm{T}\vdash\mathrm{U}:s_{\mathrm{U}}$.
Par absence de sorte maximale, il existe une sorte s'_{U} telle que $(s_{\mathrm{U}},s'_{\mathrm{U}})\in\mathbf{Axiom}$. De même que pour l'étape précédente, nous souhaitons utiliser la règle $(\Sigma_{\textsc{Sub}}-\mathrm{E}-2)$:

$$\frac{\left\{\begin{array}{l}\sigma(\Gamma);x:\mathrm{T}\vdash s_{\mathrm{U}}:s'_{\mathrm{U}}\\ \sigma(\Gamma);x:\mathrm{T}\vdash\sigma(c):\Sigma z:\sigma(\mathrm{A})\,.\,[\sigma(\mathrm{B})]\end{array}\right. \qquad \left\{\begin{array}{l}\sigma(\Gamma);x:\mathrm{T};y:\sigma(\mathrm{B}_0)\vdash \mathrm{U}_{\mathrm{HI}}:s_{\mathrm{U}}\\ (\sigma(\mathrm{B})\,[z/\pi_1(\sigma(c))])^*\preceq\sigma(\mathrm{B}_0)^*\\ y\notin \mathsf{FV}(\mathrm{U}^*_{\mathrm{HI}})\end{array}\right.}{\sigma(\Gamma);x:\mathrm{T}\vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\sigma(\mathrm{B}_0)].\mathrm{U}_{\mathrm{HI}},\sigma(c),s_{\mathrm{U}}):s_{\mathrm{U}}}$$

- $(\sigma(\mathrm{B})\,[z/\pi_1(\sigma(c))])^*\preceq(\sigma(\mathrm{B}_0))^*$ a été montrée à l'étape précédente.
- $y\notin\mathsf{FV}(\mathrm{U}^*_{\mathrm{HI}})$ se démontre similairement à $y\notin\mathsf{FV}(\mathrm{T}^*_{\mathrm{HI}})$ (cf. étape (2)).[3]
- De $\sigma(\Gamma)\vdash\mathrm{T}:s_{\mathrm{T}}$, nous déduisons $\sigma(\Gamma);x:\mathrm{T}\vdash$, d'où :
  - — $\sigma(\Gamma);x:\mathrm{T}\vdash s_{\mathrm{U}}:s'_{\mathrm{U}}$ en appliquant $(\textsc{Sort}_{\mathrm{A}})$;
  - — $\sigma(\Gamma);x:\mathrm{T}\vdash\pi_1(\sigma(c)):\Sigma z:\sigma(\mathrm{A})\,.\,[\sigma(\mathrm{B})]$ par affaiblissement de $\sigma(\Gamma)\vdash\pi_1(\sigma(c)):\Sigma z:\sigma(\mathrm{A})\,.\,[\sigma(\mathrm{B})]$ (démontré à la deuxième étape).
- Montrons enfin que $\sigma(\Gamma);x:\mathrm{T};y:\sigma(\mathrm{B}_0)\vdash\mathrm{U}_{\mathrm{HI}}:s_{\mathrm{U}}$ est dérivable.
  - — Par hypothèse d'induction, nous avons $\sigma(\Gamma);y:\sigma(\mathrm{B}_0);x:\mathrm{T}_{\mathrm{HI}}\vdash\mathrm{U}_{\mathrm{HI}}:s_{\mathrm{U}}$.
  - — Nous pouvons convertir le contexte et obtenir $\sigma(\Gamma);y:\sigma(\mathrm{B}_0);x:\mathrm{T}\vdash\mathrm{U}_{\mathrm{HI}}:s_{\mathrm{U}}$ puisque :
    - — $\mathrm{T}^*=\mathrm{T}^*_{\mathrm{HI}}$
    - — $\sigma(\Gamma);y:\sigma(\mathrm{B}_0);x:\mathrm{T}\vdash$ se déduit par affaiblissement de de $\sigma(\Gamma)\vdash\mathrm{T}:s_{\mathrm{T}}$.
  - — Nous pouvons conclure en appliquant le lemme d'échange (lemme 2.3.12 page 125), puisque $y\notin\mathsf{FV}(\mathrm{T})$ $(\mathrm{T}=\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\sigma(\mathrm{B}_0)].\mathrm{T}_{\mathrm{HI}},\sigma(c),s_{\mathrm{T}}))$.

3. Il faudra éventuellement α-convertir $\Sigma x:\mathrm{T}'\,.\,\mathrm{U}'$ afin d'avoir $y\neq x$.

□

Le corollaire suivant est une application directe du résultat précédent. Le deuxième point découle facilement du premier mais les deux énoncés seront utilisés dans les preuves ultérieures : le premier pour les prémisses dont le *terme* est un constructeur de type (règles d'introduction des Σ-types) ; le second pour les prémisses dont le *type* est un constructeur de type (règles d'élimination des Π-types et règles d'élimination du type sous-ensemble).

2.1.3. Corollaire (Relèvement des constructeurs de type)**.**

(1) Si $\Gamma \vdash R : s$ *et si* $R^* = \square^* x : T'.U'$ *alors il existe* T, U *tels que*

$$\Gamma \vdash \square x : T.U : s \quad T^* = T' \quad U^* = U'$$

(2) Si $\Gamma \vdash M : R$ *et si* $R^* = \square^* x : T'.U'$ *alors il existe* T, U *tels que*

$$\Gamma \vdash M : \square x : T.U \quad T^* = T' \quad U^* = U'$$

DÉMONSTRATION.

(1) Il suffit d'appliquer le lemme précédent avec $\sigma = \mathrm{id}_A$ et $\pi = \circ$.

(2) Conséquence immédiate du point précédent en appliquant le lemme du type des types puis la règle de cumulativité.

□

2.1.4. REMARQUE. Le premier point du corollaire ne peut se démontrer directement. Une preuve directe, qui ressemblerait à celle du lemme précédent, bloquerait sur le cas de (I-LAM)- et uniquement sur ce cas. C'est pour contourner ce blocage que nous avons dû adapter à AICC_Σ les substitutions de type et les jugements de structure utilisés dans le lemme précédent.

2.2. Preuve du relèvement.

2.2.1. Proposition (Relèvement du typage)**.**

(1) Si $\Gamma' \vdash$ *dans* ICC_Σ*, alors il existe un contexte annoté* Γ *tel que* $\Gamma \vdash$ *et* $\Gamma^* = \Gamma'$.

(2) Si $\Gamma' \vdash M' : T'$*, alors il existe* Γ, M, T *dans* AICC_Σ *tels que* $\Gamma \vdash M : T$ *et* $\Gamma^* = \Gamma', M^* = M', T^* = T'$.

DÉMONSTRATION. Nous prouvons simultanément les deux points en procédant par induction mutuelle sur la structure des dérivations des jugements $\Gamma' \vdash$ et $\Gamma' \vdash M' : T'$.

- (WF-E) : immédiat avec $\Gamma = \bullet$.
- (WF-S) : Si

$$\frac{\Gamma' \vdash T' : s \qquad x \notin \mathsf{DV}(\Gamma')}{\Gamma'; x : T' \vdash}$$

  par induction il existe dans AICC_Σ un contexte Γ, des termes S, T tels que

$$\Gamma \vdash T : S \quad \Gamma^* = \Gamma' \quad T^* = T' \quad S^* = s$$

  - Par relèvement des sortes (lemme 2.1.1), nous déduisons $\Gamma \vdash T : s$.
  - Nous dérivons ensuite $\Gamma; x : T \vdash$ par application de $(\mathrm{WF\text{-}S_A})$, puisque $\mathsf{DV}(\Gamma) = \mathsf{DV}(\Gamma^*)$.
- (VAR) et (SORT) : Clair par induction et d'après la définition de l'extraction de jugement.
- Règles de formation : Traitons le cas générique. Si

$$\frac{\Gamma' \vdash T' : s_1 \qquad \Gamma'; x : T' \vdash U' : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\square}{\Gamma' \vdash \square^* x : T'.U' : s_3}$$

- Par induction sur $\Gamma' \vdash T' : s_1$, il existe un contexte annoté Γ et des termes annotés T, S_1 tels que
$$\Gamma \vdash T : S_1 \quad \Gamma^* = \Gamma' \quad T^* = T' \quad S_1^* = s_1$$
De $\Gamma \vdash T : S_1$ et $S_1^* = s_1$ nous déduisons $\Gamma \vdash T : s_1$ (relèvement des sortes), puis $\Gamma \vdash T : s_1$ en appliquant (WF-S_A).
- De même, par induction sur $\Gamma'; x : T' \vdash U' : s_2$, il existe un contexte annoté Δ et des termes annotés U, S_2 tels que
$$\Delta \vdash U : S_2 \quad \Delta^* = \Gamma'; x : T' \quad U^* = U' \quad S_2^* = s_2$$
Nous en déduisons $\Delta \vdash U : s_2$ (relèvement des sortes), puis $\Gamma; x : T \vdash U : s_2$ en convertissant le contexte du jugement (corollaire 2.3.7 page 122).
- Nous concluons alors en appliquant ($\Box_A$-FORM) :
$$\frac{\Gamma \vdash T : s_1 \qquad \Gamma; x : T \vdash U : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\Box}{\Gamma \vdash \Box x : T . U : s_3}$$

- Règles d'introduction du produit : Traitons le cas de (GEN), similaire à (LAM), mais avec une condition de bord supplémentaire.

  Si
$$\frac{\Gamma'; x : T' \vdash M' : U' \qquad \Gamma' \vdash \forall x : T' . U' : s \qquad x \notin \mathsf{FV}(M')}{\Gamma' \vdash M' : \forall x : T' . U'}$$

  - Par induction sur $\Gamma'; x : T' \vdash M' : U'$, il existe un contexte annoté Γ et des termes T, M, U tels que
$$\Gamma; x : T \vdash M : U \quad \Gamma^* = \Gamma' \quad T^* = T' \quad M^* = M' \quad U^* = U'$$
  - Montrons qu'il existe une sorte s telle que nous puissions appliquer la règle (I-LAM) :
$$\frac{\Gamma; x : T \vdash M : U \qquad \Gamma \vdash \Pi[x:T] . U : s \qquad x \notin \mathsf{FV}(M^*)}{\Gamma \vdash \lambda[x:T] . M : \Pi[x:T] . U}$$
  .
    — Nous déduisons $x \notin \mathsf{FV}(M^*)$ de $M^* = M'$ et $x \notin \mathsf{FV}(M')$.
    — Montrons qu'il existe une sorte s telle que $\Gamma \vdash \Pi[x:T] . U : s$.
      — De $\Gamma; x : T \vdash M : U$, nous déduisons $\Gamma; x : T \vdash$. Par inversion de (WF-S_A), il existe une sorte s_1 telle que $\Gamma \vdash T : s_1$ est dérivable.
      — D'après le lemme du type des types appliqué à $\Gamma; x : T \vdash M : U$, il existe une sorte s_2 telle que $\Gamma; x : T \vdash U : s_2$.
      — Par complétude de **Rule**, il existe une sorte s telle que $(s_1, s_2, s) \in \mathbf{Rule}$.

- Règles d'élimination du produit : Traitons le cas de (APP), similaire à (INST).

  Si
$$\frac{\Gamma' \vdash M' : \Pi x : T' . U' \qquad \Gamma' \vdash N' : T'}{\Gamma' \vdash M' N' : U'[x/N']}$$

  - Par hypothèses d'induction, il existe des contextes annotés Γ, Δ et des termes annotés M, N, T_0, R tels que
$$\Gamma \vdash M : R \qquad \Delta \vdash N : T_0$$

avec

$$\Gamma^* = \Delta^* = \Gamma' \quad M^* = M' \quad N^* = N' \quad T_0^* = T' \quad R^* = \Pi x : T' . U'.$$

- D'après le point (2) du corollaire 2.1.3, il existe T, U tels que

$$\Gamma \vdash M : \Pi x : T . U \quad T^* = T' \quad U^* = U'$$

- Montrons que $\Gamma \vdash MN : U[x/N]$. Pour cela, montrons que $\Gamma \vdash N : T$. Nous conclurons alors en appliquant (E-APP) :

$$\frac{\Gamma \vdash M : \Pi x : T . U \qquad \Gamma \vdash N : T}{\Gamma \vdash MN : U[x/N]}$$

  — Puisque $\Gamma \vdash$ et que $\Gamma^* = \Delta^* = \Gamma'$, nous déduisons $\Gamma \vdash N : T_0$ de $\Delta \vdash N : T_0$ en convertissant les contextes (corollaire 2.3.7 page 122).

  — De $\Gamma \vdash M : \Pi x : T . U$, nous déduisons qu'il existe une sorte s_0 telle que $\Gamma \vdash \Pi x : T . U : s_0$ (lemme du type des types). Par lemme d'inversion, il existe une sorte s telle que $\Gamma \vdash T : s$.

  — Nous pouvons alors appliquer (CUM$_A$), puisque $T_0^* = T^* = T'$:

$$\frac{\Gamma \vdash N : T_0 \qquad \Gamma \vdash T : s \qquad T_0^* \preceq T^*}{\Gamma \vdash N : T}$$

- D'après le lemme 1.2.7 page 114, nous avons $(U[x/N])^* = U'[x/N']$, ce qui nous permet de conclure.

- (CUM) Si

$$\frac{\Gamma' \vdash M' : T' \qquad \Gamma' \vdash T'_0 : s \qquad T' \preceq T'_0}{\Gamma' \vdash M' : T'_0}$$

- Par hypothèses d'induction, il existe $\Gamma, \Delta, M, T, T_0, S$ dans AICC$_\Sigma$ tels que

$$\Gamma \vdash M : T \qquad \Delta \vdash T_0 : S$$

avec

$$\Gamma^* = \Delta^* = \Gamma' \quad M^* = M' \quad T^* = T' \quad T_0^* = T'_0 \quad S^* = s$$

- Par relèvement de la sorte et par conversion des contextes, nous déduisons $\Gamma \vdash T_0 : s$ de $\Delta \vdash T_0 : S$, $S^* = s$, $\Gamma^* = \Delta^*$ et $\Gamma \vdash$.
- Nous concluons alors en appliquant (CUM$_A$) :

$$\frac{\Gamma \vdash M : T \qquad \Gamma \vdash T_0 : s \qquad T^* \preceq T_0^*}{\Gamma \vdash M : T_0}$$

- Règles d'introduction de la somme. Nous traitons uniquement le cas de la règle (Σ-I). (∃-I) et (SUB-I) sont similaires.

  Si

$$\frac{\Gamma' \vdash a' : A' \qquad \Gamma' \vdash b' : B'[x/a'] \qquad \Gamma' \vdash \Sigma x : A' . B' : s}{\Gamma' \vdash (a', b') : \Sigma x : A' . B'}$$

- Par hypothèses d'induction, il existe dans AICC$_\Sigma$ trois contextes Γ, Δ, Δ_0 et des termes a, A_0, b, U, R, S tels que

$$\Gamma \vdash a : A_0 \qquad \Delta \vdash b : U \qquad \Delta_0 \vdash R : S$$

avec

$$\Gamma^* = \Delta^* = \Delta_0^* = \Gamma' \quad a^* = a' \quad b^* = b' \quad \mathrm{R}^* = \Sigma x : \mathrm{A}'.\mathrm{B}'$$

et

$$\mathrm{A}_0^* = \mathrm{A}' \quad \mathrm{U}^* = \mathrm{B}'[x/a'] \quad \mathrm{S}^* = s.$$

- En convertissant les contextes et en remplaçant S par s (relèvement des sortes), nous avons

$$\Gamma \vdash a : \mathrm{A}_0 \qquad \Gamma \vdash b : \mathrm{U} \qquad \Gamma \vdash \mathrm{R} : s$$

- D'après le point (1) du corollaire 2.1.3 appliqué à $\Gamma \vdash \mathrm{R} : s$, il existe des termes annotés A, B tels que

$$\Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s \qquad \mathrm{A}^* = \mathrm{A'} \qquad \mathrm{B}^* = \mathrm{B'}$$

- Montrons alors que nous pouvons appliquer $(\Sigma_{\mathrm{A}}\text{-I})$:

$$\frac{\Gamma \vdash a : \mathrm{A} \qquad \Gamma \vdash b : \mathrm{B}[x/a] \qquad \Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash (a, b)_{\Sigma x:\mathrm{A}.\mathrm{B}} : \Sigma x : \mathrm{A}.\mathrm{B}}$$

  — Par inversion de $\Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s$, il existe des sortes s_1, s_2 telles que :

$$\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2$$

  — Nous déduisons alors $\Gamma \vdash a : \mathrm{A}$ de $\Gamma \vdash a : \mathrm{A}_0$ en appliquant $(\mathrm{Cum}_{\mathrm{A}})$.

  — Nous pouvons alors substituer x par a dans $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2$ et en déduire $\Gamma \vdash \mathrm{B}[x/a] : s_2$.

  — Nous concluons alors en appliquant $(\mathrm{Cum}_{\mathrm{A}})$ à $\Gamma \vdash b : \mathrm{U}$ puisque $(\mathrm{B}[x/a])^* = \mathrm{B}'[x/a']$.

- $(\Sigma\text{-E})$ et $(\exists\text{-E})$ sont similaires. Traitons $(\exists\text{-E})$.

  Si

$$\frac{\Gamma' \vdash \mathrm{P}' : \exists x : \mathrm{A}'.\mathrm{B}' \to s \qquad \Gamma' \vdash c' : \exists x : \mathrm{A}'.\mathrm{B}' \qquad \Gamma'; x : \mathrm{A}'; y : \mathrm{B}' \vdash f' : \mathrm{P}'(\diamond, y) \qquad x \notin \mathsf{FV}(f')}{\Gamma' \vdash \mathsf{Elim}_{\exists}(y.f', c') : \mathrm{P}'\,c'}$$

  - Par hypothèses d'induction, et en convertissant comme précédemment les contextes, nous avons $\Gamma, \mathrm{A}, \mathrm{B}, \mathrm{P}, c, f, \mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3$ annotés tels que

$$\Gamma \vdash \mathrm{P} : \mathrm{R}_1 \qquad \Gamma \vdash c : \mathrm{R}_2 \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{R}_3$$

    avec

$$\Gamma^* = \Gamma' \quad \mathrm{P}^* = \mathrm{P}' \quad c^* = c' \quad \mathrm{A}^* = \mathrm{A}' \quad \mathrm{B}^* = \mathrm{B}' \quad f^* = f'$$

    et

$$\begin{cases} \mathrm{R}_1^* & = & \exists x : \mathrm{A}'.\mathrm{B}' \to s & = & (\Sigma[x{:}\mathrm{A}].\mathrm{B} \to s)^* \\ \mathrm{R}_2^* & = & \exists x : \mathrm{A}'.\mathrm{B}' & = & (\Sigma[x{:}\mathrm{A}].\mathrm{B})^* \\ \mathrm{R}_3^* & = & \mathrm{P}'(x, y) & = & (\mathrm{P}([x], y)_{\Sigma[x\mathrm{A}].\mathrm{B}})^* \end{cases}$$

  - Montrons que nous pouvons appliquer $(\Sigma_{\exists}-\mathrm{E})$:

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}].\mathrm{B} \to s \qquad \Gamma \vdash c : \Sigma[x{:}\mathrm{A}].\mathrm{B} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}([x], y)_{\Sigma[x\mathrm{A}].\mathrm{B}} \qquad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}$$

    La condition de bord est trivialement vraie. Les trois prémisses vont se déduire des hypothèses d'induction en appliquant la règle de cumulativité.

— Par inversion de $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash$ et par complétude de **Rule** et **Rule**′, il existe des sortes s_1, s_2 telles que

$$\Gamma \vdash \Sigma[x{:}\mathrm{A}].\mathrm{B} : s_1 \qquad \Gamma \vdash \Sigma[x{:}\mathrm{A}].\mathrm{B} \to s : s_2$$

— En appliquant $(\textsc{Cum}_\mathrm{A})$ dans $\Gamma \vdash c : \mathrm{R}_2$ et $\Gamma \vdash \mathrm{P} : \mathrm{R}_1$, nous dérivons :

$$\Gamma \vdash c : \Sigma[x{:}\mathrm{A}].\mathrm{B} \qquad \Gamma \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}].\mathrm{B} \to s$$

— Par affaiblissement de $\Gamma \vdash \mathrm{P} : \Sigma[x:\mathrm{A}].\mathrm{B} \to s$ et en appliquant (E-App), nous dérivons $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}].\mathrm{B}} : s$. En appliquant $(\textsc{Cum}_\mathrm{A})$ à $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash f : \mathrm{R}_3$, nous dérivons la dernière prémisse $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}].\mathrm{B}}$.

- $(\textsc{Sub}\text{-E-1})$: Si

$$\frac{\Gamma' \vdash a' : \{x:\mathrm{A}\,|'\}\mathrm{B}'}{\Gamma' \vdash a' : \mathrm{A}'}\ (\textsc{Sub}\text{-E-1})$$

  - Par hypothèses d'induction, il existe dans AICC_Σ Γ, a, R tels que $\Gamma \vdash a : \mathrm{R}$, $\Gamma^* = \Gamma'$, $a^* = a'$ et $\mathrm{R}^* = \{x : \mathrm{A}' \mid \mathrm{B}'\}$.
  - D'après le point (2) du corollaire 2.1.3 appliqué à $\Gamma \vdash a : \mathrm{R}$, il existe des termes annotés A, B tels que $\Gamma \vdash a : \Sigma x : \mathrm{A}.[\mathrm{B}]$ et $\mathrm{A}^* = \mathrm{A}'$, $\mathrm{B}^* = \mathrm{B}'$.
  - Nous dérivons alors $\Gamma \vdash a : \mathrm{A}$ en appliquant $(\Sigma_{\textsc{Sub}}\text{-E-1})$ à $\Gamma \vdash a : \Sigma x : \mathrm{A}.[\mathrm{B}]$.

- $(\textsc{Sub}\text{-E-2})$:

  Si

$$\frac{\Gamma' \vdash \mathrm{P}' : s \qquad \Gamma' \vdash c' : \{x:\mathrm{A}' \mid \mathrm{B}'\} \qquad \Gamma'; y:\mathrm{B}'[x/c'] \vdash f' : \mathrm{P}' \qquad y \notin \mathsf{FV}(f')}{\Gamma' \vdash f' : \mathrm{P}'}\ (\textsc{Sub}\text{-E-2})$$

  - Par hypothèses d'induction, et en convertissant les contextes, il existe dans AICC_Σ $\Gamma, \mathrm{P}, c, f, \mathrm{S}, \mathrm{R}_1, \mathrm{R}_2, \mathrm{R}_3$ tels que

$$\Gamma \vdash \mathrm{P} : \mathrm{S} \qquad \Gamma \vdash c : \mathrm{R}_1 \qquad \Gamma; y : \mathrm{R}_2 \vdash f : \mathrm{R}_3$$

    avec

$$\Gamma^* = \Gamma' \quad \mathrm{P}^* = \mathrm{P}' \quad c^* = c' \quad f^* = f' \quad \mathrm{R}_2^* = \mathrm{B}'[x/c']$$

    et

$$\mathrm{S}^* = s \quad \mathrm{R}_1^* = \{x : \mathrm{A}' \mid \mathrm{B}'\} \quad \mathrm{R}_3^* = \mathrm{P}'.$$

  - Par relèvement de la sorte, nous déduisons $\Gamma \vdash \mathrm{P} : s$ de $\Gamma \vdash \mathrm{P} : \mathrm{S}$ et $\mathrm{S}^* = s$.
  - D'après le point (2) du corollaire 2.1.3 appliqué à $\Gamma \vdash c : \mathrm{R}_1$, il existe des termes annotés A, B tels que $\Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}]$, $\mathrm{A}^* = \mathrm{A}'$ et $\mathrm{B}^* = \mathrm{B}'$.
  - Montrons que nous pouvons appliquer la règle $(\Sigma_{\textsc{Sub}}\text{-E-2})$ et dériver :

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}] \qquad \Gamma; y : \mathrm{B}[x/\pi_1(c)] \vdash f : \mathrm{P} \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (\mathrm{B}[x/\pi_1(c)])^* \preceq (\mathrm{B}[x/\pi_1(c)])^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}[x/\pi_1(c)]].f, c, \mathrm{P}) : \mathrm{P}}$$

    Les deux conditions de bord se montrent immédiatement et nous savons déjà que les deux prémisses typant P et c sont dérivables. Il nous reste donc à montrer que $\Gamma; y : \mathrm{B}[x/\pi_1(c)] \vdash f : \mathrm{P}$ est dérivable.

    — Nous obtenons $\Gamma; y : \mathrm{R}_2 \vdash f : \mathrm{P}$ en appliquant $(\textsc{Cum}_\mathrm{A})$ à $\Gamma; y : \mathrm{R}_2 \vdash f : \mathrm{R}_3$.

    — Montrons que $\Gamma; y : \mathrm{B}[x/\pi_1(c)] \vdash$.

— Nous déduisons $\Gamma; x : A; y : B \vdash$ à partir de $\Gamma \vdash c : \Sigma x : A.\,[B]$ en appliquant le lemme du type des types, le lemme d'inversion de la somme dépendante implicite droite, puis la règle de typage (WF-S_A).

— Nous dérivons $\Gamma \vdash \pi_1(c) : A$ en appliquant $(\Sigma_{\text{SUB}}-\text{E}-1)$ à $\Gamma \vdash c : \Sigma x : A.\,[B]$.

— Nous avons alors $\Gamma; y : B[x/\pi_1(c)] \vdash$ en substituant x par $\pi_1(c)$ dans $\Gamma; x : A; y : B \vdash$.

— Nous avons $R_2^* = (B[x/\pi_1(c)])^*$, d'où $(\Gamma; y : R_2)^* = (\Gamma; y : B[x/\pi_1(c)])^*$.

— Nous pouvons donc convertir le contexte du jugement $\Gamma; y : R_2 \vdash f : P$ (corollaire 2.3.7 page 122) et finalement obtenir $\Gamma; y : B[x/\pi_1(c)] \vdash f : P$.

□

2.2.2. REMARQUE. Notons que, comme annoncé en introduction, la preuve du relèvement n'a fait appel au corollaire 2.1.3 uniquement pour les règles d'élimination des produits, d'introduction des sommes et d'élimination du type sous-ensemble.

Quatrième partie

Décidabilité du typage

Préambule

AICC_Σ a pour but d'être un système implémentable et utilisable en pratique. Pour cela, il est souhaitable que le typage soit décidable, c'est-à-dire qu'il existe un algorithme de *vérification* de type correct et complet qui s'assure que les programmes écrits par l'utilisateur sont bien typés. Cet algorithme prend en entrée un contexte Γ et deux termes M et T et répond oui si $\Gamma \vdash M : T$ et non sinon.

Habituellement, la question de la vérification de type se ramène à celle de l'*inférence* de type. L'inférence de type est le fait de construire un type à partir d'un terme et d'un contexte. Plus précisément, un algorithme d'*inférence de type* prend en entrée un contexte Γ, supposé bien formé, et un terme M et retourne, s'il existe, un terme T tel que $\Gamma \vdash M : T$, ou bien faux si ce n'est pas le cas. En général, le terme T retourné est un *type principal*, c'est-à-dire plus petit (au sens de la cumulativité) que tout autre type de M dans Γ. L'inférence de type permet la vérification de type : pour vérifier que $\Gamma \vdash M : T$, il suffit d'inférer un type R pour M dans Γ et un type S pour T (pour s'assurer que T est bien typé) et vérifier que R est plus petit que T. Réciproquement, la vérification de type fait appel à l'inférence de type : vérifier par exemple qu'une application MN a pour type R dans Γ impliquera de pouvoir inférer un type pour M. C'est pourquoi le problème de la décidabilité du typage d'un système de types se confond en général avec celui de l'inférence de type. Nous montrons dans les deux chapitres qui composent cette partie que les choses sont différentes dans AICC_Σ.

Tout d'abord, remarquons que l'inférence de type ne peut se faire de la manière habituelle car la réduction se fait pour les termes extraits et non pour les termes annotés. En effet, dans le calcul des constructions, ou plus généralement dans un PTS où la syntaxe ne distingue pas les types des termes ($\lambda\Pi$ mais pas le λ-calcul simplement typé par exemple), l'algorithme d'inférence de type va faire appel à un sous-algorithme de réduction. Ainsi, inférer, par exemple, le type d'une application MN demandera notamment d'inférer un type pour M puis de réduire ce type inféré, par exemple en prenant sa forme normale (ou juste sa forme normale de tête), afin d'obtenir un produit dépendant $\Pi x : T.U$ qui sera utilisé pour exprimer le type $U[x/N]$ inféré pour MN. Il est impossible d'imiter cette démarche pour AICC_Σ. Tout d'abord, il n'y a pas de mécanisme de réduction dans AICC_Σ. Par ailleurs, utiliser ce qui existe déjà, en l'occurrence, la réduction dans ICC_Σ, ne nous permet pas de conclure, car trop d'information est perdue dans l'extraction. Reprenons l'exemple de l'application : si R est le type inféré pour M et si nous savons que $\mathsf{nf}(R^*) = \Pi x : T'.U'$, il paraît difficile de pouvoir obtenir à partir de $\Pi x : T'.U'$ un produit $\Pi x : T.U$ dans AICC_Σ tel que $\Gamma \vdash M : \Pi x : T.U$ et $(\Pi x : T.U)^* = \Pi x : T'.U'$.[4]

Face à ce problème nous proposons deux solutions. La première, développée dans le chapitre 8, est de définir un algorithme de vérification de typage à partir d'un algorithme d'inférence qui retourne un type de ICC_Σ et non de AICC_Σ. La seconde, développée dans le chapitre 9, est d'enrichir AICC_Σ avec un mécanisme de réduction de termes qui va permettre d'exprimer un algorithme d'inférence de type.

4. Ce problème est proche de celui du relèvement. Néanmoins, même si la preuve du relèvement que nous avons fournie est vraisemblablement constructive, elle utilisait la connaissance de la *dérivation* d'un jugement alors que pour un algorithme d'inférence de type, nous ne connaissons que le terme et son contexte.

L'algorithme d'inférence du chapitre 8, appélé algorithme d'inférence de type *extrait*, permet, grâce à la propriété du relèvement du typage, de définir un algorithme de vérification de type. Ainsi pour AICC_Σ, le problème de la vérification de type n'est pas couplé à celui de l'inférence de type mais à celui de l'inférence de type *extrait*.

Cette première solution est suffisante pour résoudre la décidabilité du typage, puisque nous obtenons un algorithme de vérification de type. Toutefois, il nous semble intéressant d'obtenir un algorithme d'inférence de type annoté. Tout d'abord, parce qu'il est satisfaisant en soi d'avoir un tel algorithme. Ensuite, parce qu'il apparaît nécessaire, dans le cadre d'une éventuelle implémentation future, de pouvoir inférer les types annotés afin de pouvoir élaborer des sous-termes et alléger ainsi la tâche du programmeur. Enfin, ajouter un mécanisme de réduction dans AICC_Σ qui relèverait la réduction de ICC_Σ permet de montrer l'équivalence entre les deux systèmes au niveau du calcul.

Inversement, l'existence d'un algorithme d'inférence de type annoté ne rend pas inutile l'existence d'un algorithme d'inférence de type extrait. En effet, il est intéressant de montrer que la décidabilité du typage ne nécessite pas d'inférer un type annoté. Par ailleurs l'algorithme d'inférence de type extrait est plus simple, plus robuste et ne nécessite pas d'enrichir le système. Enfin, le mécanisme de réduction développé dans le chapitre 9 pour l'inférence de type extrait fait appel à l'algorithme d'inférence de type extrait : le premier algorithme est donc utilisé par le second.

Chapitre 8

Inférence de type extrait

Dans les systèmes de types usuels, un algorithme d'inférence de type infère le type principal d'un terme bien typé. Avant de présenter au chapitre suivant un tel algorithme pour AICC_Σ, nous présentons ici un algorithme d'inférence de type inférant l'*extraction* d'un type principal. avec des caractéristiques un peu différentes. Cet algorithme prend en entrée un terme de AICC_Σ un contexte de ICC_Σ, et échoue ou retourne un terme de ICC_Σ. La spécification de cet algorithme est qu'il retourne pour les termes de AICC_Σ bien typés l'*extraction* de leur type principal, d'où son nom d'algorithme d'inférence de type *extrait*.

Cette spécification est inhabituelle car le type retourné n'est pas un terme du même langage que le terme donné en entrée. Inférer un type extrait ne suffit pas pour décider du typage dans AICC_Σ car le type extrait ne contient pas toutes les informations nécessaires. Il est néanmoins *utile* pour deux raisons. D'une part, il permet, nous le verrons à la section 5, de définir des algorithmes de vérification de bonne formation de contexte et de vérification de type. D'autre part il sera utilisé par l'algorithme d'inférence de type défini au prochain chapitre.

L'algorithme est présenté dans les deux premières sections : la section 1 est consacrée aux sous-algorithmes utilisés dans l'algorithme principal ; la section 2 à l'algorithme principal lui-même. Les propriétés principales d'un algorithme d'inférence de type sont sa terminaison, sa correction et sa complétude. La terminaison de l'algorithme découlera immédiatement de sa définition. Les deux autres propriétés seront démontrées respectivement dans les sections 3 et 4.

1. Sous-algorithmes

Nous présentons dans cette section des algorithmes auxiliaires utilisés dans l'algorithme principal.

1.1. Second membre d'une paire de Axiom ou troisième membre d'un triplet de Rule ou Rule′.

1.1.1. DÉFINITION. Nous définissons trois algorithmes `Axiom`, `Rule` et `RulePrime` par :

(1)
$$\begin{aligned} \texttt{Axiom}(\mathsf{Prop}) &= \mathsf{Type}_0 \\ \texttt{Axiom}(\mathsf{Type}_i) &= \mathsf{Type}_{i+1} \end{aligned}$$

(2)
$$\begin{aligned} \texttt{Rule}(s, \mathsf{Prop}) &= \mathsf{Prop} \\ \texttt{Rule}(\mathsf{Prop}, \mathsf{Type}_i) &= \mathsf{Type}_i \\ \texttt{Rule}(\mathsf{Type}_i, \mathsf{Type}_j) &= \mathsf{Type}_{max(i,j)} \end{aligned}$$

(3)
$$\begin{aligned} \texttt{RulePrime}(s, \mathsf{Prop}) &= s \\ \texttt{RulePrime}(\mathsf{Prop}, \mathsf{Type}_i) &= \mathsf{Type}_i \\ \texttt{RulePrime}(\mathsf{Type}_i, \mathsf{Type}_j) &= \mathsf{Type}_{max(i,j)} \end{aligned}$$

La proposition suivante se prouve sans difficulté.

1.1.2. Proposition. *Les algorithmes* `Axiom`, `Rule` *et* `RulePrime` *terminent, sont corrects et complets.*

1.2. Mise en forme normale. Pour pouvoir définir des algorithmes de mise en forme βι- et βιη-normale, nous admettons la normalisation forte de la βιη-réduction pour les termes bien typés de ICC_Σ. [1]

1.2.1. Conjecture (Normalisation forte)**.** *Il n'existe pas de séquence infinie de* βιη*-réductions commençant par un terme bien typé de* ICC_Σ.

Nous en déduisons immédiatement la normalisation forte de la βι-réduction pour les termes bien typés de ICC_Σ. Par ailleurs, nous prouvons, par confluence de βι et βιη, l'unicité des formes normales (proposition 2.1.5 page 11). Nous pouvons donc définir des algorithmes de mise en forme normale pour la βι-réduction et la βιη-réduction.

1.2.2. DÉFINITION (Algorithmes de mise en forme normale).

- Nous notons `NF_BI` (resp. `NF_BIE`) un algorithme qui implémente une stratégie de réduction pour la βι-réduction (resp. βιη-réduction) et qui retourne la forme βι-normale (resp. βιη-normale) d'un terme si elle existe.
- Si M' est un terme de ICC_Σ tel que l'algorithme `NF_BI` (resp. `NF_BIE`) appliqué à M' retourne une valeur, cette valeur est notée $\mathtt{NF\_BI}(M')$ (resp. $\mathtt{NF\_BIE}(M')$).

Ces algorithmes

- *terminent* pour tout terme bien typé dans ICC_Σ, du fait de la normalisation forte de ICC_Σ pour la βι-réduction et la βιη-réduction ;
- sont *corrects* : pour tout terme M' de ICC_Σ,
  - si $\mathtt{NF\_BI}(M')$ existe alors $M' \twoheadrightarrow_{\beta\iota} \mathtt{NF\_BI}(M')$ et $\mathtt{NF\_BI}(M')$ est en forme βι-normale,
  - si $\mathtt{NF\_BIE}(M')$ existe alors $M' \twoheadrightarrow_{\beta\iota\eta} \mathtt{NF\_BIE}(M')$ et $\mathtt{NF\_BIE}(M')$ est en forme βιη-normale ;
- sont *complets* : pour tout terme M' bien typé dans ICC_Σ,
  - $\mathtt{NF\_BI}(M')$ existe et $\mathtt{NF\_BI}(M') = \mathsf{nf}_{\beta\iota}(M')$,
  - $\mathtt{NF\_BIE}(M')$ existe et $\mathtt{NF\_BIE}(M') = \mathsf{nf}(M')$.

1.3. Décidabilité de la cumulativité.

1.3.1. DÉFINITION (Cumulativité). Nous définissons l'algorithme `cumu` par la règle d'inférence suivante :

$$\frac{\mathtt{NF\_BIE}(M') \leqslant \mathtt{NF\_BIE}(N')}{\mathtt{cumu}(M', N')}$$

1.3.2. Proposition. *L'algorithme* `cumu`

(1) termine : pour tous termes M', N' *typables,* $\mathtt{cumu}(M', N')$ *est défini ;*

(2) est correct : pour tous termes M', N' *de* ICC_Σ*, si* $\mathtt{cumu}(M', N')$ *alors* $M' \preceq N'$ *;*

(3) est complet : pour tous termes M', N' *de* ICC_Σ *typables, si* $M' \preceq N'$ *alors* $\mathtt{cumu}(M', N')$.

DÉMONSTRATION. (1) `NF_BIE` termine pour tout terme typable ; la cumulativité restreinte $\leqslant$ est décidable.

(2) Correction : soient M', N' tels que $\mathtt{cumu}(M', N')$; montrons que $M' \preceq N'$:

1. Nous avons vu dans la deuxième partie qu'il nous semblait envisageable de montrer cette conjecture en étendant à ICC_Σ le modèle de normalisation forte défini par Miquel pour ICC_Σ.

- posons $M'_0 = \texttt{NF\_BIE}(M')$ et $N'_0 = \texttt{NF\_BIE}(N')$;
- nous avons $M'_0 \leqslant N'_0$ donc $M'_0 \preceq N'_0$;
- par correction de NF_BIE, nous avons $M' \twoheadrightarrow_{\beta\iota\eta} M'_0$ et $N' \twoheadrightarrow_{\beta\iota\eta} N'_0$, d'où $M' \preceq M'_0$ et $N'_0 \preceq N'$;
- par transitivité, nous déduisons $M' \preceq N'$.

(3) Complétude : soient M', N' typables tels que $M' \preceq N'$; montrons que $\texttt{cumu}(M', N')$.

- M' et N' sont typables dans ICC_Σ donc fortement normalisables.
- D'après le corollaire 2.3.12 page 17, nous avons $\mathsf{nf}(M') \leqslant \mathsf{nf}(N')$.
- Par correction de NF_BIE, nous avons $\mathsf{nf}(M') = \texttt{NF\_BIE}(M')$ et $\mathsf{nf}(M') = \texttt{NF\_BIE}(M')$, d'où le résultat souhaité.

□

2. Algorithme principal

2.1. Définitions.

2.1.1. DÉFINITION (Jugement d'inférence de type extrait). Soient Γ' un contexte de ICC_Σ, M un terme de AICC_Σ et T' un terme de ICC_Σ. Le *jugement d'inférence de type extrait*, noté $\Gamma' \vdash M \Uparrow^I T'$, signifie que le terme T' est le type extrait inféré de M dans le contexte Γ'.

2.1.2. DÉFINITION (Algorithme d'inférence de type extrait). Les règles d'inférence définissant l'algorithme d'inférence de type extrait sont détaillées dans les figures 24 et 25.

Nous remarquons que ces règles sont *dirigées par la syntaxe*, c'est-à-dire qu'aucune structure n'est représentée dans la conclusion de plus de deux règles. Cela permet de prouver par une induction structurelle immédiate le déterminisme de l'algorithme :

2.1.3. Lemme (Unicité du type inféré)**.** *Si* $\Gamma' \vdash M \Uparrow^I T'_1$ *et* $\Gamma' \vdash M \Uparrow^I T'_2$ *alors* $T'_1 = T'_2$.

2.1.4. Proposition (Terminaison)**.** *L'algorithme d'inférence de type extrait termine.*

DÉMONSTRATION. Conséquence immédiate

- de la terminaison des sous-algorithmes utilisés : Axiom, Rule, RulePrime, NF_BI et cumu ;
- de la décidabilité de l'appartenance à un contexte, l'appartenance à l'ensemble **Sort**, l'égalité syntaxique, du calcul des variables libres d'un terme de ICC_Σ, de l'appartenance à un ensemble de variables,
- du fait que les appels récursifs se font sur des sous-termes stricts.

□

2.2. Notation générique pour les constructeurs de type. Nous introduisons, comme dans les chapitres précédents, des notations communes aux différents constructeurs de type.

2.2.1. NOTATION (Algorithme générique pour les ensembles de règles). Nous notons $\texttt{Rule}_\square$, l'algorithme tel que

- si $\square \in \{\Pi, \forall\}$, $\texttt{Rule}_\square = \texttt{Rule}$;
- si $\square \in \{\Sigma, \exists\}$, $\texttt{Rule}_\square = \texttt{RulePrime}$;
- si $\square \in \{\mathsf{Sub}\}$, pour toutes sortes s_1, s_2, $\texttt{Rule}_\square(s_1, s_2) = s_1$.

2.2.2. NOTATION (Règle d'inférence générique pour les constructeurs de type). Nous utilisons la règle générique suivante :

(1) Variable

$$\frac{(x:\mathrm{T}')\in\Gamma'}{\Gamma'\vdash x\Uparrow^{\mathrm{I}}\mathrm{T}'}\ (\textsc{Inf-Var})$$

(2) Sorte

$$\frac{}{\Gamma'\vdash s_1\Uparrow^{\mathrm{I}}\mathtt{Axiom}(s_1)}\ (\textsc{Inf-Sort})$$

(3) Produit explicite

$$\frac{\Gamma'\vdash \mathrm{T}\Uparrow^{\mathrm{I}}\mathrm{S}_1' \qquad \mathtt{NF\_BI}(\mathrm{S}_1')\in\mathbf{Sort} \qquad \Gamma';x:\mathrm{T}^*\vdash \mathrm{U}\Uparrow^{\mathrm{I}}\mathrm{S}_2' \qquad \mathtt{NF\_BI}(\mathrm{S}_2')\in\mathbf{Sort}}{\Gamma'\vdash \Pi x:\mathrm{T}.\,\mathrm{U}\Uparrow^{\mathrm{I}}\mathtt{Rule}(\mathtt{NF\_BI}(\mathrm{S}_1'),\mathtt{NF\_BI}(\mathrm{S}_2'))}\ (\textsc{Inf-E-Prd})$$

$$\frac{\Gamma'\vdash \mathrm{T}\Uparrow^{\mathrm{I}}\mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}')\in\mathbf{Sort} \qquad \Gamma';x:\mathrm{T}^*\vdash \mathrm{M}\Uparrow^{\mathrm{I}}\mathrm{U}'}{\Gamma'\vdash \lambda x{:}\mathrm{T}.\,\mathrm{M}\Uparrow^{\mathrm{I}}\Pi x:\mathrm{T}^*.\,\mathrm{U}'}\ (\textsc{Inf-E-Lam})$$

$$\frac{\Gamma'\vdash \mathrm{M}\Uparrow^{\mathrm{I}}\mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}')=\Pi x:\mathrm{T}'.\,\mathrm{U}' \qquad \Gamma'\vdash \mathrm{N}\Uparrow^{\mathrm{I}}\mathrm{T}_0' \qquad \mathtt{cumu}(\mathrm{T}_0',\mathrm{T}')}{\Gamma'\vdash \mathrm{M}\,\mathrm{N}\Uparrow^{\mathrm{I}}\mathrm{U}'[x/\mathrm{N}^*]}\ (\textsc{Inf-E-App})$$

(4) Produit implicite

$$\frac{\Gamma'\vdash \mathrm{T}\Uparrow^{\mathrm{I}}\mathrm{S}_1' \qquad \mathtt{NF\_BI}(\mathrm{S}_1')\in\mathbf{Sort} \qquad \Gamma';x:\mathrm{T}^*\vdash \mathrm{U}\Uparrow^{\mathrm{I}}\mathrm{S}_2' \qquad \mathtt{NF\_BI}(\mathrm{S}_2')\in\mathbf{Sort}}{\Gamma'\vdash \Pi[x{:}\mathrm{T}].\,\mathrm{U}\Uparrow^{\mathrm{I}}\mathtt{Rule}(\mathtt{NF\_BI}(\mathrm{S}_1'),\mathtt{NF\_BI}(\mathrm{S}_2'))}\ (\textsc{Inf-I-Prd})$$

$$\frac{\Gamma'\vdash \mathrm{T}\Uparrow^{\mathrm{I}}\mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}')\in\mathbf{Sort} \qquad \Gamma';x:\mathrm{T}^*\vdash \mathrm{M}\Uparrow^{\mathrm{I}}\mathrm{U}' \qquad x\notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma'\vdash \lambda[x{:}\mathrm{T}].\,\mathrm{M}\Uparrow^{\mathrm{I}}\forall x:\mathrm{T}^*.\,\mathrm{U}'}\ (\textsc{Inf-I-Lam})$$

$$\frac{\Gamma'\vdash \mathrm{M}\Uparrow^{\mathrm{I}}\mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}')=\forall x:\mathrm{T}'.\,\mathrm{U}' \qquad \Gamma'\vdash \mathrm{N}\Uparrow^{\mathrm{I}}\mathrm{T}_0' \qquad \mathtt{cumu}(\mathrm{T}_0',\mathrm{T}')}{\Gamma'\vdash \mathrm{M}\,[\mathrm{N}]\Uparrow^{\mathrm{I}}\mathrm{U}'[x/\mathrm{N}^*]}\ (\textsc{Inf-I-App})$$

FIGURE 24. Règles d'inférence du jugement d'inférence de type extrait (Variable, sorte et produits)

$$\frac{\Gamma'\vdash \mathrm{T}\Uparrow^{\mathrm{I}}\mathrm{S}_1' \qquad \mathtt{NF\_BI}(\mathrm{S}_1')\in\mathbf{Sort} \qquad \Gamma';x:\mathrm{T}^*\vdash \mathrm{U}\Uparrow^{\mathrm{I}}\mathrm{S}_2' \qquad \mathtt{NF\_BI}(\mathrm{S}_2')\in\mathbf{Sort}}{\Gamma'\vdash \square x:\mathrm{T}.\,\mathrm{U}\Uparrow^{\mathrm{I}}\mathtt{Rule}_{\square}(\mathtt{NF\_BI}(\mathrm{S}_1'),\mathtt{NF\_BI}(\mathrm{S}_2'))}\ (\textsc{Inf-CstrTyp})$$

pour désigner une règle quelconque de l'algorithme d'inférence de type extrait relative aux constructeurs de type.

En étendant la proposition 1.1.2 avec le cas $\square\in\{\mathsf{Sub}\}$, nous avons, pour $\square\in\{\Pi,\forall,\Sigma,\exists,\mathsf{Sub}\}$,

- $\mathtt{Rule}_{\square}$ termine ;
- $\mathtt{Rule}_{\square}$ est correct : pour toutes sortes s_1,s_2, $(s_1,s_2,\mathtt{Rule}_{\square}(s_1,s_2))\in\mathbf{Rule}_{\square}$;
- $\mathtt{Rule}_{\square}$ est complet : pour toutes sortes s_1,s_2,s_3, si $(s_1,s_2,s_3)\in\mathbf{Rule}_{\square}$ alors $s_3=\mathtt{Rule}_{\square}(s_1,s_2)$.

3. Correction

Le but de cette section est d'établir que le type inféré pour un terme par l'algorithme est l'extraction d'un type de ce terme.

(5) Somme explicite

$$\frac{\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Sigma x : \mathrm{A}.\, \mathrm{B} \Uparrow^{\mathrm{I}} \mathtt{RulePrime}(\mathtt{NF\_BI}(\mathrm{S}'_1), \mathtt{NF\_BI}(\mathrm{S}'_2))}\ (\textsc{Inf-}\Sigma)$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma' \vdash (a, b)_{\Sigma x : \mathrm{A}.\, \mathrm{B}} \Uparrow^{\mathrm{I}} \Sigma x : \mathrm{A}^*.\, \mathrm{B}^*}\ (\textsc{Inf-}\Sigma\textsc{-Pair})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{P} \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}'_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{R}'_1) = \Sigma x : \mathrm{A}'_1.\, \mathrm{B}'_1 \to s_1 \\ \mathtt{NF\_BI}(\mathrm{R}'_2) = \Sigma x : \mathrm{A}'_2.\, \mathrm{B}'_2 \\ \Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{cumu}(\Sigma x : \mathrm{A}^*.\, \mathrm{B}^*, \Sigma x : \mathrm{A}'_1.\, \mathrm{B}'_1) \\ \mathtt{cumu}(\Sigma x : \mathrm{A}'_2.\, \mathrm{B}'_2, \Sigma x : \mathrm{A}^*.\, \mathrm{B}^*) \\ \Gamma'; x : \mathrm{A}^*; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_3 \\ \mathtt{cumu}(\mathrm{R}'_3, \mathrm{P}^*\,(x, y))\end{array}\right.}{\Gamma' \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) \Uparrow^{\mathrm{I}} \mathrm{P}^*\, c^*}\ (\textsc{Inf-}\Sigma\textsc{-Elim})$$

(6) Somme implicite droite

$$\frac{\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Sigma[x{:}\mathrm{A}]\,.\, \mathrm{B} \Uparrow^{\mathrm{I}} \mathtt{RulePrime}(\mathtt{NF\_BI}(\mathrm{S}'_1), \mathtt{NF\_BI}(\mathrm{S}'_2))}\ (\textsc{Inf-}\exists)$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma' \vdash ([a], b)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} \Uparrow^{\mathrm{I}} \exists x : \mathrm{A}^*.\, \mathrm{B}^*}\ (\textsc{Inf-}\exists\textsc{-Pair})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{P} \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}'_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{R}'_1) = \exists x : \mathrm{A}'_1.\, \mathrm{B}'_1 \to s_1 \\ \mathtt{NF\_BI}(\mathrm{R}'_2) = \exists x : \mathrm{A}'_2.\, \mathrm{B}'_2 \\ \Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{cumu}(\exists x : \mathrm{A}^*.\, \mathrm{B}^*, \exists x : \mathrm{A}'_1.\, \mathrm{B}'_1) \\ \mathtt{cumu}(\exists x : \mathrm{A}'_2.\, \mathrm{B}'_2, \exists x : \mathrm{A}^*.\, \mathrm{B}^*) \\ \Gamma'; x : \mathrm{A}^*; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_3 \\ x \notin \mathsf{FV}(f^*) \\ \mathtt{cumu}(\mathrm{R}'_3, \mathrm{P}^*\,(\diamond, y))\end{array}\right.}{\Gamma' \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) \Uparrow^{\mathrm{I}} \mathrm{P}^*\, c^*}\ (\textsc{Inf-}\exists\textsc{-Elim})$$

(7) Somme implicite gauche

$$\frac{\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Sigma x : \mathrm{A}.\, [\mathrm{B}] \Uparrow^{\mathrm{I}} \mathtt{NF\_BI}(\mathrm{S}'_1)}\ (\textsc{Inf-Sub})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma' \vdash (a, [b])_{\Sigma x : \mathrm{A}.\,[\mathrm{B}]} \Uparrow^{\mathrm{I}} \{x : \mathrm{A}^* \mid \mathrm{B}^*\}}\ (\textsc{Inf-Sub-Pair})$$

$$\frac{\Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}') = \{x : \mathrm{A}' \mid \mathrm{B}'\}}{\Gamma' \vdash \pi_1(c) \Uparrow^{\mathrm{I}} \mathrm{A}'}\ (\textsc{Inf-Sub-Elim-1})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{P} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \\ \Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \Gamma' \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \\ \mathtt{NF\_BI}(\mathrm{R}'_1) = \{x : \mathrm{A}'_1 \mid \mathrm{B}'_1\} \\ \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma'; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, \mathrm{P}^*) \\ \mathtt{cumu}(\mathrm{B}'_1\,[x/c^*], \mathrm{B}^*) \\ y \notin \mathsf{FV}(f^*)\end{array}\right.}{\Gamma' \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P}) \Uparrow^{\mathrm{I}} \mathrm{P}^*}\ (\textsc{Inf-Sub-Elim-2})$$

FIGURE 25. Règles d'inférence du jugement d'inférence de type extrait (Sommes dépendantes)

3.1. Lemme auxiliaire. Nous montrons un lemme auxiliaire qui est la conséquence du corollaire 2.1.3 page 138.

3.1.1. Lemme (Relèvement des constructeurs de type à βι-réduction près)**.**

(1) Si $\Gamma \vdash R : s$ *et si* $R^* \twoheadrightarrow_{\beta\iota} \Box^* x : T' . U'$ *alors il existe* T, U *tels que*

- $\Gamma \vdash \Box x : T . U : s$
- $T^* = T'$
- $U^* = U'$.

(2) Si $\Gamma \vdash M : R$ *et si* $R^* \twoheadrightarrow_{\beta\iota} \Box^* x : T' . U'$ *alors il existe* T, U *tels que*

- $\Gamma \vdash M : \Box x : T . U$
- $T^* = T'$
- $U^* = U'$.

DÉMONSTRATION.

(1) D'après le point (1) du corollaire 2.1.3 page 138, il suffit de montrer qu'il existe N tel que $\Gamma \vdash N : s$ et $N^* = \Box^* x : T' . U'$.
- Par correction de l'extraction sur $\Gamma \vdash R : s$, nous déduisons $\Gamma^* \vdash R^* : s$.
- Par préservation du typage par βι-réduction, nous obtenons $\Gamma^* \vdash \Box^* x : T' . U' : s$.
- Par relèvement de $\Gamma^* \vdash \Box^* x : T' . U' : s$ il existe Δ, N, S tels que
  - $\Delta \vdash N : S$
  - $\Delta^* = \Gamma^*, N^* = \Box^* x : T' . U', S^* = s$.
- Par conversion de contexte, puisque $\Gamma \vdash$, nous avons $\Gamma \vdash N : S$.
- Par cumulativité, nous obtenons $\Gamma \vdash N : s$.

(2) Preuve similaire à celle du point (2) du corollaire 2.1.3 page 138.
- Il existe une sorte s telle que $\Gamma \vdash R : s$.
- D'après le point (1), il existe T, U tels que
  - $\Gamma \vdash \Box x : T . U : s$
  - $T^* = T'$
  - $U^* = U'$.
- Par cumulativité, nous dérivons $\Gamma \vdash M : \Box x : T . U$.

□

3.2. Proposition principale.

3.2.1. Proposition (Correction de l'algorithme d'inférence de type extrait)**.** *Si* $\Gamma \vdash$ *et si* $\Gamma^* \vdash M \Uparrow^I T'$ *alors il existe* T *tel que* $\Gamma \vdash M : T$ *et* $T^* = T'$.

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma^* \vdash M \Uparrow^I T'$. Nous considérons la dernière règle appliquée.

- (INF-VAR) Si $\Gamma^* \vdash x \Uparrow^I T'$, alors $(x : T') \in \Gamma^*$. Par définition de l'extraction d'un contexte, il existe T tel que $(x : T) \in \Gamma$ et $T^* = T'$. En appliquant (VAR_A), puisque $\Gamma \vdash$, nous dérivons bien $\Gamma \vdash x : T$.
- (INF-SORT) Si $\Gamma^* \vdash s_1 \Uparrow^I \texttt{Axiom}(s_1)$, montrons que $\Gamma \vdash s_1 : \texttt{Axiom}(s_1)$. Puisque $\Gamma \vdash$ par hypothèse et que $(s_1, \texttt{Axiom}(s_1)) \in$ **Axiom** par correction de `Axiom`, nous pouvons conclure en appliquant la règle (SORT_A).

- Les règles pour les constructeurs de type sont similaires. Traitons la règle générique.
  Si

$$\frac{\Gamma^* \vdash T \Uparrow^I S_1' \qquad \texttt{NF\_BI}(S_1') \in \textbf{Sort} \qquad \Gamma^*; x : T^* \vdash U \Uparrow^I S_2' \qquad \texttt{NF\_BI}(S_2') \in \textbf{Sort}}{\Gamma^* \vdash \square x : T . U \Uparrow^I \texttt{Rule}_\square(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2'))} \text{ (INF-CSTRTYP)}$$

  Montrons que $\Gamma \vdash \square x : T . U : \texttt{Rule}_\square(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2'))$.
  - Montrons que $\Gamma \vdash T : \texttt{NF\_BI}(S_1')$.
    — Par induction sur le jugement $\Gamma^* \vdash T \Uparrow^I S_1'$, il existe S_1 tel que $\Gamma \vdash T : S_1$ et $S_1^* = S_1'$.
    — Puisque $S_1^* \cong \texttt{NF\_BI}(S_1')$ et que $\texttt{NF\_BI}(S_1')$ est une sorte, donc typée par une sorte dans Γ bien formé, nous dérivons par cumulativité $\Gamma \vdash T : \texttt{NF\_BI}(S_1')$.
  - Montrons que $\Gamma; x : T \vdash U : \texttt{NF\_BI}(S_2')$.
    — Puisque $\Gamma \vdash T : \texttt{NF\_BI}(S_1')$ et que $\texttt{NF\_BI}(S_1') \in \textbf{Sort}$, nous dérivons $\Gamma; x : T \vdash$ par (WF-S_A).
    — Nous pouvons donc par induction appliquer le résultat à la prémisse $\Gamma^*; x : T^* \vdash U \Uparrow^I S_2'$: il existe donc S_2 tel que $\Gamma; x : T \vdash U : S_2$ et $S_2^* = S_2'$.
    — Nous montrons comme précédemment que, par cumulativité, $\Gamma; x : T \vdash U : \texttt{NF\_BI}(S_2')$.
  - Par correction de $\texttt{Rule}_\square$, nous avons $(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2'), \texttt{Rule}_\square(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2'))) \in \textbf{Rule}_\square$.
  - Nous concluons alors en appliquant la règle de formation générique ($\square_A$-FORM).
- (INF-E-LAM) et (INF-I-LAM) se traitent similairement. Étudions par exemple le cas de (INF-I-LAM). Si

$$\frac{\Gamma^* \vdash T \Uparrow^I S' \qquad \texttt{NF\_BI}(S') \in \textbf{Sort} \qquad \Gamma^*; x : T^* \vdash M \Uparrow^I U' \qquad x \notin \mathsf{FV}(M^*)}{\Gamma^* \vdash \lambda[x{:}T] . M \Uparrow^I \forall x : T^* . U'} \text{ (INF-I-LAM)}$$

  montrons qu'il existe U tel que $U^* = U'$ et $\Gamma \vdash \lambda[x{:}T] . M : \Pi[x{:}T] . U$. Pour cela nous voulons appliquer la règle (I-LAM) :

$$\frac{\Gamma; x : T \vdash M : U \qquad \Gamma \vdash \Pi[x{:}T] . U : s \qquad x \notin \mathsf{FV}(M^*)}{\Gamma \vdash \lambda[x{:}T] . M : \Pi[x{:}T] . U}$$

  La condition de bord $x \notin \mathsf{FV}(M^*)$ est vraie par hypothèse. Montrons les deux prémisses.
  (1) Montrons qu'il existe U tel que $U^* = U'$ et $\Gamma; x : T \vdash M : U$.
    - Montrons que $\Gamma; x : T \vdash$.
      — Par hypothèse d'induction, il existe S tel que $\Gamma \vdash T : S$ et $S^* = S'$.
      — Par cumulativité, nous dérivons $\Gamma \vdash T : \texttt{NF\_BI}(S')$, avec $\texttt{NF\_BI}(S') \in \textbf{Sort}$.
      — Nous concluons par (WF-S_A).
    - Puisque $\Gamma; x : T \vdash$, nous pouvons appliquer par induction le résultat à la prémisse $\Gamma^*; x : T^* \vdash M \Uparrow^I U'$, et en déduisons qu'il existe U tel que $U^* = U'$ et $\Gamma; x : T \vdash M : U$.
  (2) Montrons qu'il existe une sorte s telle que $\Gamma \vdash \Pi[x{:}T] . U : s$.
    - D'après le lemme du type des types appliqué à $\Gamma; x : T \vdash M : U$, il existe une sorte s_0 telle que $\Gamma; x : T \vdash U : s_0$.
    - Puisque **Rule** est fonctionnel, il existe une sorte s telle que $(\texttt{NF\_BI}(S'), s_0, s) \in \textbf{Rule}$.

- Nous dérivons alors, par application de (I-Prod), $\Gamma \vdash \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U} : s$.

- (Inf-E-App) et (Inf-I-App) sont similaires. Traitons (Inf-I-App). Si

$$\frac{\Gamma^* \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}') = \forall x : \mathrm{T}'\,.\,\mathrm{U}' \qquad \Gamma^* \vdash \mathrm{N} \Uparrow^{\mathrm{I}} \mathrm{T}'_0 \qquad \mathtt{cumu}(\mathrm{T}'_0, \mathrm{T}')}{\Gamma^* \vdash \mathrm{M}\,[\mathrm{N}] \Uparrow^{\mathrm{I}} \mathrm{U}'\,[x/\mathrm{N}^*]}\ (\textsc{Inf-I-App})$$

  Montrons qu'il existe U_0 tel que $\Gamma \vdash \mathrm{M}\,[\mathrm{N}] : \mathrm{U}_0$ et $\mathrm{U}_0^* = \mathrm{U}'\,[x/\mathrm{N}^*]$.

  - Par hypothèses d'induction,
    - il existe R tel que $\Gamma \vdash \mathrm{M} : \mathrm{R}$ et $\mathrm{R}^* = \mathrm{R}'$;
    - il existe T_0 tel que $\Gamma \vdash \mathrm{N} : \mathrm{T}_0$ et $\mathrm{T}_0^* = \mathrm{T}'_0$.
  - Montrons qu'il existe T, U tels que $\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}$, $\mathrm{T}^* = \mathrm{T}'$ et $\mathrm{U}^* = \mathrm{U}'$: cela découle du point (2) du lemme 3.1.1, puisque $\Gamma \vdash \mathrm{M} : \mathrm{R}$ et que, par correction de NF_BI, $\mathrm{R}^* = \mathrm{R}' \twoheadrightarrow_{\beta\iota} \mathtt{NF\_BI}(\mathrm{R}') = \forall x : \mathrm{T}'\,.\,\mathrm{U}'$.
  - Montrons que $\Gamma \vdash \mathrm{N} : \mathrm{T}$.
    - Par correction de cumu, nous déduisons $\mathrm{T}'_0 \preceq \mathrm{T}'$ de $\mathtt{cumu}(\mathrm{T}'_0, \mathrm{T}')$, d'où $\mathrm{T}_0^* \preceq \mathrm{T}^*$.
    - En appliquant le lemme du type des types à $\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}$, puis par inversion, il existe une sorte s telle que $\Gamma \vdash \mathrm{T} : s$.
    - Par cumulativité sur $\Gamma \vdash \mathrm{N} : \mathrm{T}_0$, nous dérivons bien $\Gamma \vdash \mathrm{N} : \mathrm{T}$.
  - En appliquant (I-App), nous dérivons bien $\Gamma \vdash \mathrm{M}\,[\mathrm{N}] : \mathrm{U}\,[x/\mathrm{N}]$ avec $(\mathrm{U}\,[x/\mathrm{N}])^* = \mathrm{U}'\,[x/\mathrm{N}^*]$.

- Règles d'introduction des sommes : elles (Inf-Σ-Pair), (Inf-Sub-Pair) et (Inf-Sub-Pair) sont similaires. Traitons par exemple (Inf-∃-Pair).

  Si

$$\frac{\left\{\begin{array}{l}\Gamma^* \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma^*; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma^* \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma^* \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma^* \vdash ([a], b)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} \Uparrow^{\mathrm{I}} \{x : \mathrm{A}^* \mid \mathrm{B}^*\}}\ (\textsc{Inf-∃-Pair})$$

  Montrons que $\Gamma \vdash ([a], b)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$.

  - Montrons qu'il existe une sorte s telle que $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s$.
    - Par hypothèse d'induction, il existe S_{A} tel que $\Gamma \vdash \mathrm{A} : \mathrm{S}_{\mathrm{A}}$ et $\mathrm{S}_{\mathrm{A}}^* = \mathrm{S}'_{\mathrm{A}} \twoheadrightarrow_{\beta\iota} \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}) \in \mathbf{Sort}$.
    - Posons $s_{\mathrm{A}} = \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}})$. Puisque s_{A} est une sorte, donc bien typée dans Γ bien formé, nous dérivons, par cumulativité, $\Gamma \vdash \mathrm{A} : s_{\mathrm{A}}$.
    - Par application de (Wf-S_{A}), nous dérivons $\Gamma; x : \mathrm{A} \vdash$. Nous pouvons donc appliquer par induction le résultat à la prémisse $\Gamma^*; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}}$: nous en déduisons qu'il existe S_{B} tel que $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : \mathrm{S}_{\mathrm{B}}$ et $\mathrm{S}_{\mathrm{B}}^* = \mathrm{S}'_{\mathrm{B}} \twoheadrightarrow_{\beta\iota} \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}$.
    - Posons $s_{\mathrm{B}} = \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}})$. De même que précédemment, nous dérivons par cumulativité $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_{\mathrm{B}}$.
    - Par fonctionnalité de $\mathbf{Rule}'$, il existe une sorte s telle que $(s_{\mathrm{A}}, s_{\mathrm{B}}, s) \in \mathbf{Rule}'$.
    - Nous concluons alors en appliquant ($\Sigma_\exists$).
  - Montrons que $\Gamma \vdash a : \mathrm{A}$ et $\Gamma \vdash b : \mathrm{B}\,[x/a]$.
    - Par inversion de $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s$, il existe des sortes s_1, s_2 telles que
      - $\Gamma \vdash \mathrm{A} : s_1$
      - $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2$.

— Montrons que $\Gamma \vdash a : A$.

— Par hypothèse d'induction, il existe R_1 tel que $\Gamma \vdash a : R_1$ et $R_1^* = R_1'$.

— Par correction de `cumu` et puisque $\texttt{cumu}(R_1', A^*)$, nous déduisons $R_1^* \preceq A^*$.

— Puisque $\Gamma \vdash A : s_1$, nous dérivons $\Gamma \vdash a : A$ par cumulativité sur $\Gamma \vdash a : R_1$.

— Montrons que $\Gamma \vdash b : B[x/a]$.

— Par hypothèse d'induction, il existe R_2 tel que $\Gamma \vdash b : R_2$ et $R_2^* = R_2'$.

— Par correction de `cumu` et puisque $\texttt{cumu}(R_2', (B[x/a])^*)$, nous déduisons $R_2^* \preceq (B[x/a])^*$.

— De $\Gamma \vdash a : A$ et $\Gamma; x : A \vdash B : s_2$, nous déduisons par substitutivité que $\Gamma \vdash B[x/a] : s_2$.

— Nous concluons par cumulativité sur $\Gamma \vdash b : R_2$.

- Nous concluons en appliquant $(\Sigma_\exists - \text{I})$:

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B[x/a] \qquad \Gamma \vdash \Sigma[x{:}A].B : s}{\Gamma \vdash ([a], b)_{\Sigma[x:A].B} : \Sigma[x{:}A].B}$$

- (Inf-Σ-Elim) et (Inf-∃-Elim) : leurs démonstrations sont similaires, traitons le cas de (Inf-∃-Elim). Si

$$\frac{\left\{\begin{array}{l}\Gamma^* \vdash P \Uparrow^I R_1' \\ \Gamma^* \vdash c \Uparrow^I R_2'\end{array}\right. \quad \left\{\begin{array}{l}\texttt{NF\_BI}(R_1') = \exists x : A_1'.B_1' \to s_1 \\ \texttt{NF\_BI}(R_2') = \exists x : A_2'.B_2' \\ \Gamma^* \vdash A \Uparrow^I S_A' \\ \Gamma^*; x : A^* \vdash B \Uparrow^I S_B' \\ \texttt{NF\_BI}(S_A'), \texttt{NF\_BI}(S_B') \in \textbf{Sort}\end{array}\right. \quad \left\{\begin{array}{l}\texttt{cumu}(\exists x : A^*.B^*, \exists x : A_1'.B_1') \\ \texttt{cumu}(\exists x : A_2'.B_2', \exists x : A^*.B^*) \\ \Gamma^*; x : A^*; y : B^* \vdash f \Uparrow^I R_3' \\ x \notin \mathsf{FV}(f^*) \\ \texttt{cumu}(R_3', P^*(\diamond, y))\end{array}\right.}{\Gamma^* \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, c, P) \Uparrow^I P^*\, c^*} \text{(Inf-U-Elim)}$$

Montrons que $\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, c, P) : P\, c$. Pour cela nous voulons appliquer la règle $(\Sigma_\exists - \text{E})$:

$$\frac{\Gamma \vdash P : \Sigma[x{:}A].B \to s_1 \qquad \Gamma \vdash c : \Sigma[x{:}A].B \qquad \Gamma; x : A; y : B \vdash f : P([x], y)_{\Sigma[x:A].B} \qquad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, c, P) : P\, c}$$

La condition de bord $x \notin \mathsf{FV}(f^*)$ est vraie par hypothèse. Avant de montrer que les trois prémisses sont dérivables, nous montrons, en procédant comme dans la preuve du cas précédent (règles d'introduction des sommes) qu'il existe une sorte s telle que $\Gamma \vdash \Sigma[x{:}A].B : s$.[2] Montrons maintenant les trois prémisses.

(1) Montrons que $\Gamma \vdash c : \Sigma[x{:}A].B$.

- Par hypothèse d'induction, il existe R_2 tel que $\Gamma \vdash c : R_2$ et $R_2^* = R_2'$.
- Par hypothèses et par correction de `cumu`, nous obtenons $\texttt{NF\_BI}(R_2') \preceq (\Sigma[x{:}A].B)^*$. Par correction de `NF_BI`, nous déduisons $R_2^* = R_2' \twoheadrightarrow_{\beta\iota} \texttt{NF\_BI}(R_2')$ d'où $R_2^* \preceq (\Sigma[x{:}A].B)^*$.
- Puisque $\Gamma \vdash \Sigma[x{:}A].B : s$, nous dérivons $\Gamma \vdash c : \Sigma[x{:}A].B$ par cumulativité sur $\Gamma \vdash c : R_2$.

(2) Montrons que $\Gamma \vdash P : \Sigma[x{:}A].B \to s_1$.

2. Cela est possible car les deux règles contiennent les prémisses $\Gamma^* \vdash A \Uparrow^I S_A'$ et $\Gamma^*; x : A \vdash B \Uparrow^I S_B'$ et la condition de bord $\texttt{NF\_BI}(S_A'), \texttt{NF\_BI}(S_B') \in \textbf{Sort}$.

- Par hypothèse d'induction, il existe R_1 tel que $\Gamma \vdash P : R_1$ et $R_1^* = R_1'$.
- Puisque $\Gamma \vdash \Sigma[x{:}A].B : s$, nous montrons facilement qu'il existe une sorte s' telle que $\Gamma \vdash \Sigma[x{:}A].B \to s_1 : s'$.
- De même que plus haut pour R_2, nous obtenons, par hypothèses et par correction de `cumu` et `NF_BI`, que $R_1^* \preceq (\Sigma[x{:}A].B \to s_1)^*$.
- Par cumulativité sur $\Gamma \vdash P : R_1$. nous dérivons bien $\Gamma \vdash P : \Sigma[x{:}A].B \to s_1$.

(3) Montrons que $\Gamma; x : A; y : B \vdash f : P\,([x], y)_{\Sigma[x{:}A].B}$.

- Montrons que $\Gamma; x : A; y : B \vdash$: par inversion de $\Gamma \vdash \Sigma[x : A].B : s$, il existe une sorte s_0 telle que $\Gamma; x : A \vdash B : s_0$; nous concluons en appliquant (Wf-S$_A$).
- Nous pouvons donc appliquer par induction le résultat à la prémisse $\Gamma^*; x : A^*; y : B^* \vdash f \Uparrow^I R_3'$ et en déduire qu'il existe R_3 tel que $\Gamma; x : A; y : B \vdash f : R_3$ avec $R_3^* = R_3'$.
- Remplaçons par cumulativité le type de f :
  - — par correction de `cumu`, nous avons $R_3^* \preceq P^*\,(\diamond, y)$;
  - — montrons que $\Gamma; x : A; y : B \vdash P\,([x], y)_{\Sigma[x{:}A].B} : s_1$:
    - — nous obtenons $\Gamma; x : A; y : B \vdash P : \Sigma[x : A].B \to s_1$ par affaiblissement de $\Gamma \vdash P : \Sigma[x{:}A].B \to s_1$;
    - — en appliquant ($\Sigma_\exists - I$), nous dérivons $\Gamma; x : A; y : B \vdash ([x], y)_{\Sigma[x{:}A].B} : \Sigma[x : A].B$;
    - — nous concluons alors en appliquant (E-App).

- (Inf-Sub-Elim-1) Si

$$\frac{\Gamma^* \vdash c \Uparrow^I R' \qquad \texttt{NF\_BI}(R') = \{x : A' \mid B'\}}{\Gamma^* \vdash \pi_1(c) \Uparrow^I A'}\ \text{(Inf-Sub-Elim-1)}$$

Montrons qu'il existe A tel que $\Gamma \vdash \pi_1(c) : A$ et $A^* = A'$.

- Montrons qu'il existe A, B tels que $\Gamma \vdash c : \Sigma x : A.\,[B]$ et $A^* = A', B^* = B'$.
  - — Par hypothèse d'induction, il existe R tel que $\Gamma \vdash c : R$ et $R^* = R'$.
  - — Puisque $R^* \twoheadrightarrow_{\beta\iota} \{x : A' \mid B'\}$, nous concluons en appliquant le point (2) du lemme 3.1.1 à $\Gamma \vdash c : R$.
- En appliquant ($\Sigma_{SUB} - E - 1$) à $\Gamma \vdash c : \Sigma x : A.\,[B]$, nous dérivons bien $\Gamma \vdash \pi_1(c) : A$.

- (Inf-Sub-Elim-2) Si

$$\frac{\left\{\begin{array}{l}\Gamma^* \vdash P \Uparrow^I S_1' \\ \Gamma^* \vdash c \Uparrow^I R_1' \\ \Gamma^* \vdash B \Uparrow^I S_2'\end{array}\right. \qquad \left\{\begin{array}{l}\texttt{NF\_BI}(S_1') \in \mathbf{Sort} \\ \texttt{NF\_BI}(R_1') = \{x : A_1' \mid B_1'\} \\ \texttt{NF\_BI}(S_2') \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma^*; y : B^* \vdash f \Uparrow^I R_2' \\ \texttt{cumu}(R_2', P^*) \\ \texttt{cumu}(B_1'[x/c^*], B^*) \\ y \notin \mathsf{FV}(f^*)\end{array}\right.}{\Gamma^* \vdash \mathsf{Elim}^s_{I_R}([y{:}B].f, c, P) \Uparrow^I P^*}\ \text{(Inf-Sub-Elim-2)}$$

Montrons que $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].f, c, P) : P$. Pour cela nous voulons appliquer la règle ($\Sigma_{SUB} - E - 2$) :

$$\frac{\Gamma \vdash P : s_1 \qquad \Gamma \vdash c : \Sigma x : A_1.\,[B_1] \qquad \Gamma; y : B \vdash f : P \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (B_1\,[x/\pi_1(c)])^* \preceq B^*\end{array}}{\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].f, c, P) : P}$$

$y \notin \mathsf{FV}(f^*)$ est vraie par hypothèse. Montrons les trois prémisses et la seconde condition de bord.

(1) Posons $s_1 = \mathtt{NF\_BI}(\mathrm{S}'_1)$. Montrons que $\Gamma \vdash \mathrm{P} : s_1$.
- Par hypothèse d'induction, il existe S_1 tel que $\Gamma \vdash \mathrm{P} : \mathrm{S}_1$ et $\mathrm{S}_1^* = \mathrm{S}'_1$.
- Par cumulativité, puisque $\mathtt{NF\_BI}(\mathrm{S}'_1) = s_1 \in \mathbf{Sort}$, nous obtenons bien $\Gamma \vdash \mathrm{P} : s_1$.

(2) Montrons qu'il existe $\mathrm{A}_1, \mathrm{B}_1$ tels que $\Gamma \vdash c : \Sigma x : \mathrm{A}_1 . [\mathrm{B}_1]$ et $(\mathrm{B}_1 [x/\pi_1(c)])^* \preceq \mathrm{B}^*$.
- Par hypothèse d'induction, il existe R_1 tel que $\Gamma \vdash c : \mathrm{R}_1$ et $\mathrm{R}_1^* = \mathrm{R}'_1$.
- Puisque $\mathrm{R}_1^* \twoheadrightarrow_{\beta\iota} \{x : \mathrm{A}'_1 \mid \mathrm{B}'_1\}$ et $\Gamma \vdash c : \mathrm{R}_1$, le point (2) du lemme 3.1.1 nous indique qu'il existe $\mathrm{A}_1, \mathrm{B}_1$ tels que $\Gamma \vdash c : \Sigma x : \mathrm{A}_1 . [\mathrm{B}_1]$ et $\mathrm{A}_1^* = \mathrm{A}'_1, \mathrm{B}_1^* = \mathrm{B}'_1$.
- Nous déduisons $\mathrm{B}'_1 [x/c^*] = (\mathrm{B}_1 [x/\pi_1(c)])^* \preceq \mathrm{B}^*$ par correction de `cumu` et puisque $\mathtt{cumu}(\mathrm{B}'_1 [x/c^*], \mathrm{B}^*)$.

(3) Montrons que $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$.
- Posons $s_2 = \mathtt{NF\_BI}(\mathrm{S}'_2)$. Nous montrons que $\Gamma \vdash \mathrm{B} : s_2$ de la même manière que nous avons montré $\Gamma \vdash \mathrm{P} : s_1$. Nous en déduisons par $_{(\text{WF-S}_\text{A})}$ que $\Gamma; y : \mathrm{B} \vdash$.
- Puisque $\Gamma; y : \mathrm{B} \vdash$, par induction sur la prémisse $\Gamma^*; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_2$, il existe R_2 tel que $\Gamma; y : \mathrm{B} \vdash f : \mathrm{R}_2$ et $\mathrm{R}_2^* = \mathrm{R}'_2$.
- Puisque $\Gamma \vdash \mathrm{P} : s_1$ et $\mathtt{cumu}(\mathrm{R}'_2, \mathrm{P}^*)$, nous dérivons par cumulativité $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$.

□

3.2.2. REMARQUE. Nous considérons pour la correction uniquement des contextes bien formés. Il nous faut donc pouvoir définir un algorithme qui vérifie la bonne formation de contexte. Dans les systèmes de types usuels, un tel algorithme se déduit facilement de l'algorithme d'inférence de type. Pour AICC_Σ Nous verrons à la sous-section 5.2, qu'il n'est pas nécessaire d'inférer un type annoté pour cela et que l'algorithme d'inférence de type extrait suffit.

3.3. Corollaires. Nous présentons ici quelques conséquences immédiates de la propriété de correction que nous venons de prouver.

3.3.1. Corollaire (Correction généralisée)**.** *Si* $\Gamma' \vdash$ *et si* $\Gamma' \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{T}'$ *alors il existe un contexte* Γ *et un terme* T *tels que* $\Gamma \vdash \mathrm{M} : \mathrm{T}$, $\Gamma^* = \Gamma'$ *et* $\mathrm{T}^* = \mathrm{T}'$.

DÉMONSTRATION. Par relèvement de $\Gamma' \vdash$ (point (1) de la proposition 2.2.1 page 138), il existe un contexte de AICC_Σ Γ tel que $\Gamma \vdash$ et $\Gamma^* = \Gamma'$. Nous concluons alors en appliquant la proposition 3.2.1 à $\Gamma \vdash$ et $\Gamma^* \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{T}'$. □

3.3.2. REMARQUE. Prouver directement cette généralisation aurait demandé plus de travail que de prouver d'abord la proposition 3.2.1 puis d'en déduire la généralisation : il aurait fallu traiter le contexte à chaque règle au lieu d'une seule fois ici.

3.3.3. Corollaire (Propriétés des types inférés)**.** *Si* $\Gamma' \vdash$ *et si* $\Gamma' \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{T}'$ *alors il existe une sorte* s *telle que*

(1) $\Gamma' \vdash \mathrm{T}' : s$

(2) $\mathtt{NF\_BI}(\mathrm{T}')$ *est défini et* $\Gamma' \vdash \mathtt{NF\_BI}(\mathrm{T}') : s$.

DÉMONSTRATION.
- D'après le corollaire 3.3.1, il existe Γ et T tels que $\Gamma \vdash \mathrm{M} : \mathrm{T}$, $\Gamma^* = \Gamma'$ et $\mathrm{T}^* = \mathrm{T}'$.
- Par correction de l'extraction, nous déduisons $\Gamma' \vdash \mathrm{M}^* : \mathrm{T}'$.
- D'après le lemme du type des types, il existe une sorte s telle que $\Gamma' \vdash \mathrm{T}' : s$.

- T' étant typable, sa forme $\beta\iota$-normale existe et est, par complétude de `NF_BI`, égale à $\texttt{NF\_BI}(T')$.
- Par préservation du typage, nous obtenons $\Gamma' \vdash \texttt{NF\_BI}(T') : s$.

□

4. Complétude

4.1. Lemmes auxiliaires.

4.1.1. Lemme (Forme normale et cumulativité)**.** *Soient* Γ *un contexte de* AICC_Σ, M *un terme et s une sorte tels que* $\Gamma \vdash M : s_M$.

(1) si $M \preceq s$ *(resp.* $s \preceq M$*) alors* $\mathsf{nf}_{\beta\iota}(M) \in \textbf{\textit{Sort}}$ *et* $\mathsf{nf}_{\beta\iota}(M) \preceq s$ *(resp.* $s \preceq \mathsf{nf}_{\beta\iota}(M)$*);*

(2) si $M \preceq \Box x : T.U$ *(resp.* $\Box x : T.U \preceq M$*) alors il existe* A, B *tels que* $\mathsf{nf}_{\beta\iota}(M) = \Box x : A.B$ *et* $\Box x : A.B \preceq \Box x : T.U$ *(resp.* $\Box x : T.U \preceq \Box x : A.B$*).*

DÉMONSTRATION. Nous prouvons uniquement le point (2), car le point (1) se prouve à la fois similairement et plus simplement. Par ailleurs, pour des raisons de symétrie, nous considérons uniquement le cas $M \preceq \Box x : T.U$.

- Puisque M est bien typé, il est $\beta\iota$-normalisable par normalisation forte de la $\beta\iota$-réduction dans ICC_Σ. Il est alors clair que $\mathsf{nf}_{\beta\iota}(M) \preceq \Box x : T.U$.
- D'après le point (2) du lemme 2.3.13 page 17 appliqué à $\mathsf{nf}_{\beta\iota}(M) \preceq \Box x : T.U$, il existe T_M, U_M tels que $\mathsf{nf}_{\beta\iota}(M) \twoheadrightarrow_{\beta\iota\eta} \Box x : T_M.U_M$ et $\Box x : T_M.U_M \preceq \Box x : T.U$.
- Puisque le typage est préservé par la $\beta\iota$-réduction, nous avons $\Gamma \vdash \mathsf{nf}_{\beta\iota}(M) : s_M$. Nous pouvons donc appliquer le report de la η-réduction dans $\mathsf{nf}_{\beta\iota}(M) \twoheadrightarrow_{\beta\iota\eta} \Box x : T_M.U_M$ (proposition 7.2.1 page 56) et en déduire, puisque $\mathsf{nf}_{\beta\iota}(M)$ est en forme $\beta\iota$-normale, que $\mathsf{nf}_{\beta\iota}(M) \twoheadrightarrow_\eta \Box x : T_M.U_M$.
- D'après le lemme 2.2.4 page 12, $\mathsf{nf}_{\beta\iota}(M)$ est soit de même nature que $\Box x : T_M.U_M$, soit une abstraction. Le corollaire 3.4.18 page 31 appliqué à $\Gamma \vdash \mathsf{nf}_{\beta\iota}(M) : s_M$ nous indique que $\mathsf{nf}_{\beta\iota}(M)$ n'est pas une abstraction, ce qui permet de conclure.

□

4.1.2. REMARQUE. Nous imposons que M soit typé par une sorte car sinon $\mathsf{nf}_{\beta\iota}(M)$ pourrait être une abstraction, puisque nous ne η-normalisons pas.

4.1.3. Lemme. *Soient* Γ *un contexte annoté,* M, T *des termes annotés et* T' *un terme non-annoté tels que* $\Gamma \vdash M : T$ *et* $\Gamma^* \vdash M \Uparrow^I T'$.

(1) Si $T' \preceq s$*, alors* $\texttt{NF\_BI}(T') \in \textbf{\textit{Sort}}$ *et* $\texttt{NF\_BI}(T') \preceq s$.

(2) Si $T' \preceq \Box x : A'.B'$*, alors il existe des termes non-annotés* A'_0, B'_0 *tels que*

$$\texttt{NF\_BI}(T') = \Box x : A'_0.B'_0 \preceq \Box x : A'.B'$$

DÉMONSTRATION. De $\Gamma \vdash M : T$, nous déduisons $\Gamma^* \vdash$. D'après le corollaire 3.3.3 appliqué à $\Gamma^* \vdash$ et $\Gamma^* \vdash M \Uparrow^I T'$, il existe une sorte s' telle que $\Gamma^* \vdash T' : s'$. Nous concluons en appliquant le lemme 4.1.1 à T', puis la complétude de `NF_BI`, puisque T' est bien typé. □

4.2. Proposition principale.

4.2.1. Proposition (Complétude de l'inférence de type)**.** *Soient* Γ *un contexte annoté,* M, T *des termes annotés tels que* $\Gamma \vdash M : T$*. Alors il existe un terme non-annoté* T' *tel que* $\Gamma^* \vdash M \Uparrow^I T'$ *et* $T' \preceq T^*$*.*

DÉMONSTRATION. Nous procédons par induction sur la structure de la dérivation du jugement de typage.

- (VAR_A) Immédiat par application de (INF-VAR).
- (SORT_A) : si $\Gamma \vdash s_1 : s_2$, nous avons $s_2 = \texttt{Axiom}(s_1)$ et concluons immédiatement en appliquant (INF-SORT).
- Les règles de formation (E-PROD), (I-PROD), (Σ_A), ($\Sigma_\exists$), (Σ_{SUB}) se prouvent de manière similaire. Traitons le cas de la règle générique.

  Si

$$\frac{\Gamma \vdash A : s_1 \qquad \Gamma; x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}_\square}{\Gamma \vdash \square x : A . B : s_3}$$

  montrons qu'il existe S_1', S_2' tels que

$$\frac{\Gamma^* \vdash A \Uparrow^I S_1' \qquad \texttt{NF\_BI}(S_1') \in \mathbf{Sort} \qquad \Gamma^*; x : A^* \vdash B \Uparrow^I S_2' \qquad \texttt{NF\_BI}(S_2') \in \mathbf{Sort}}{\Gamma^* \vdash \square x : A . B \Uparrow^I \texttt{Rule}_\square(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2'))} \text{ (INF-CSTRTYP)}$$

  avec $\texttt{Rule}_\square(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2')) \preceq s_3$.

  - Par hypothèses d'induction, il existe S_1', S_2' tels que
    - $\Gamma^* \vdash A \Uparrow^I S_1'$ et $S_1' \preceq s_1$,
    - $\Gamma^*; x : A^* \vdash B \Uparrow^I S_1'$ et $S_2' \preceq s_2$.
  - D'après le point (1) du lemme 4.1.3, nous avons $\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2') \in \mathbf{Sort}$.
  - Nous avons $\texttt{Rule}_\square(\texttt{NF\_BI}(S_1'), \texttt{NF\_BI}(S_2')) \preceq s_3$ par compatibilité de $\mathbf{Rule}_\square$ avec la cumulativité.

- (E-LAM) et (I-LAM) se traitent de manière similaire. Prenons par exemple le cas de (I-LAM). Nous avons

$$\frac{\Gamma; x : T \vdash M : U \qquad \Gamma \vdash \Pi[x:T] . U : s \qquad x \notin \mathsf{FV}(M^*)}{\Gamma \vdash \lambda[x:T] . M : \Pi[x:T] . U}$$

  Montrons qu'il existe S' et U' tels que

$$\frac{\Gamma^* \vdash T \Uparrow^I S' \qquad \texttt{NF\_BI}(S') \in \mathbf{Sort} \qquad \Gamma^*; x : T^* \vdash M \Uparrow^I U' \qquad x \notin \mathsf{FV}(M^*)}{\Gamma^* \vdash \lambda[x:T] . M \Uparrow^I \forall x : T^* . U'} \text{ (INF-I-LAM)}$$

  avec $\forall x : T^* . U' \preceq (\Pi[x:T] . U)^*$.

  - La condition de bord est vraie par hypothèse.
  - Par hypothèse d'induction, il existe U' tel que $\Gamma^*; x : T^* \vdash M \Uparrow^I U'$ et $U' \preceq U^*$, d'où $\forall x : T^* . U' \preceq (\Pi[x:T] . U)^*$.
  - Montrons qu'il existe S' tel que $\Gamma^* \vdash T \Uparrow^I S'$ et $\texttt{NF\_BI}(S') \in \mathbf{Sort}$.
    - Par inversion de $\Gamma \vdash \Pi[x:T] . U : s$, il existe une sorte s' telle qu'une dérivation de $\Gamma \vdash T : s'$ soit incluse dans celle $\Gamma \vdash \Pi[x:T] . U : s$.

— Nous pouvons donc appliquer par induction le résultat à $\Gamma \vdash T : s'$: il existe ainsi S' tel que $\Gamma^* \vdash T \Uparrow^I S'$ et $S' \preceq s'$.

— D'après le point (1) du lemme 4.1.3 appliqué à $\Gamma \vdash T : s'$, $\Gamma^* \vdash T \Uparrow^I S'$ et $S' \preceq s'$, nous avons $\texttt{NF\_BI}(S') \in \textsf{Sort}$.

- (E-App) et (I-App) sont similaires. Traitons le cas de l'application implicite. Si

$$\frac{\Gamma \vdash M : \Pi[x{:}T]\,.\,U \qquad \Gamma \vdash N : T}{\Gamma \vdash M\,[N] : U\,[x/N]}$$

montrons qu'il existe R', T', U', T'_0 tels que

$$\frac{\Gamma^* \vdash M \Uparrow^I R' \qquad \texttt{NF\_BI}(R') = \forall x : T'\,.\,U' \qquad \Gamma^* \vdash N \Uparrow^I T'_0 \qquad \texttt{cumu}(T'_0, T')}{\Gamma^* \vdash M\,[N] \Uparrow^I U'\,[x/N^*]}\ (\textsc{Inf-I-App})$$

avec $U'\,[x/N^*] \preceq (U\,[x/N])^*$.

- Par hypothèses d'induction, il existe T'_0, R' tels que $\Gamma^* \vdash M \Uparrow^I R'$ et $\Gamma^* \vdash N \Uparrow^I T'_0$, avec $R' \preceq \forall x : T^*\,.\,U^*$ et $T'_0 \preceq T^*$.
- D'après le point (2) du lemme 4.1.3 appliqué à $\Gamma \vdash M : \Pi[x{:}T]\,.\,U$, $\Gamma^* \vdash M \Uparrow^I R'$ et $R' \preceq \forall x : T^*\,.\,U^*$, il existe T', U' tels que $\texttt{NF\_BI}(R') = \forall x : T'\,.\,U'$.
- De $\texttt{NF\_BI}(R') = \forall x : T'\,.\,U'$ et $R' \preceq \forall x : T^*\,.\,U^*$, nous déduisons $\forall x : T'\,.\,U' \preceq \forall x : T^*\,.\,U^*$, puis par inversion $T^* \preceq T'$ et $U' \preceq U^*$.
- Montrons que $\texttt{cumu}(T'_0, T')$.
  — T'_0 est typable d'après le point (1) du corollaire 3.3.3 appliqué à $\Gamma^* \vdash$ et $\Gamma^* \vdash N \Uparrow^I T'_0$.
  — Montrons que T' est typable.
    — D'après le point (2) du corollaire 3.3.3 appliqué à $\Gamma^* \vdash$ et $\Gamma^* \vdash M \Uparrow^I R'$, il existe une sorte s telle que $\Gamma^* \vdash \forall x : T'\,.\,U' : s$.
    — Par inversion, il existe une sorte s' telle que $\Gamma^* \vdash T' : s'$.
  — Puisque $T'_0 \preceq T^*$ et $T^* \preceq T'$, nous déduisons par transitivité que $T'_0 \preceq T'$.
  — Par complétude de cumu, puisque T'_0 et T' sont typables, nous déduisons $\texttt{cumu}(T'_0, T')$ de $T'_0 \preceq T'$.
- Montrons enfin que $U'\,[x/N^*] \preceq (U\,[x/N])^*$.
  — De $U' \preceq U^*$, nous déduisons par le lemme 2.3.2 page 14 que $U'\,[x/N^*] \preceq U^*\,[x/N^*]$.
  — D'après le lemme 1.2.7 page 114, nous avons $U^*\,[x/N^*] = (U\,[x/N])^*$.
- ($\textsc{Cum}_A$) Immédiat par hypothèse d'induction et transitivité de la cumulativité.
- Les règles d'introduction des types somme (Σ_A-I), ($\Sigma_\exists - I$) et ($\Sigma_{SUB} - I$) sont similaires entre elles. Traitons par exemple ($\Sigma_{SUB} - I$). Si

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B\,[x/a] \qquad \Gamma \vdash \Sigma x : A\,.\,[B] : s}{\Gamma \vdash (a, [b])_{\Sigma x:A\,.\,[B]} : \Sigma x : A\,.\,[B]}$$

Montrons qu'il existe S'_A, S'_B, R'_1, R'_2 tels que

$$\frac{\left\{\begin{array}{l}\Gamma^* \vdash A \Uparrow^I S'_A \\ \Gamma^*; x : A^* \vdash B \Uparrow^I S'_B \\ \texttt{NF\_BI}(S'_A), \texttt{NF\_BI}(S'_B) \in \textsf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma^* \vdash a \Uparrow^I R'_1 \\ \texttt{cumu}(R'_1, A^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma^* \vdash b \Uparrow^I R'_2 \\ \texttt{cumu}(R'_2, (B\,[x/a])^*)\end{array}\right.}{\Gamma^* \vdash (a, [b])_{\Sigma x:A\,.\,[B]} \Uparrow^I \{x : A^* \mid B^*\}}\ (\textsc{Inf-Sub-Pair})$$

Nous procédons en trois étapes.

(1) Montrons qu'il existe S'_A, S'_B tels que $\Gamma^* \vdash A \Uparrow^I S'_A$, $\Gamma^*; x : A^* \vdash B \Uparrow^I S'_B$, $\texttt{NF\_BI}(S'_A), \texttt{NF\_BI}(S'_B) \in \textsf{Sort}$.

- Par inversion du jugement $\Gamma \vdash \Sigma x : A.\,[B] : s$, il existe des sortes s_A, s_B telles que des dérivations des jugements $\Gamma \vdash A : s_A$ et $\Gamma; x : A \vdash B : s_B$ sont incluses dans celle de $\Gamma \vdash \Sigma x : A.\,[B] : s$.
- Nous pouvons donc raisonner par induction sur les jugements $\Gamma \vdash A : s_A$ et $\Gamma; x : A \vdash B : s_B$ et en déduire qu'il existe S'_A, S'_B tels que
  - $\Gamma^* \vdash A \Uparrow^I S'_A$,
  - $\Gamma^*; x : A^* \vdash B \Uparrow^I S'_B$,
  - $S'_A \preceq s_A$ et $S'_B \preceq s_B$.
- D'après le point (1) du lemme 4.1.3, nous déduisons $\texttt{NF\_BI}(S'_A), \texttt{NF\_BI}(S'_B) \in \textsf{Sort}$.

(2) Montrons qu'il existe R'_1 tel que $\Gamma^* \vdash a \Uparrow^I R'_1$ et $\texttt{cumu}(R'_1, A^*)$.

- Par hypothèse d'induction sur la prémisse $\Gamma \vdash a : A$, il existe R'_1 tel que $\Gamma^* \vdash a \Uparrow^I R'_1$ et $R'_1 \preceq A^*$.
- Montrons que $\texttt{cumu}(R'_1, A^*)$.
  - Par correction de l'extraction, A^* est typable dans ICC_Σ car A l'est dans AICC_Σ.
  - R'_1 est typable dans ICC_Σ puisque c'est un type inféré par l'algorithme à partir d'un contexte bien formé (corollaire 3.3.3).
  - Puisque $R'_1 \preceq A^*$ et que R'_1 et A^* sont typables, nous déduisons par complétude de cumu que $\texttt{cumu}(R'_1, A^*)$.

(3) De même, par induction, il existe R'_2 tel que $\Gamma^* \vdash b \Uparrow^I R'_2$ et $R'_2 \preceq (B\,[x/a])^*$. Nous montrons comme précédemment que R'_2 et $(B\,[x/a])^*$ sont bien typés et déduisons, par complétude de cumu, que $\texttt{cumu}(R'_2, (B\,[x/a])^*)$.

- $(\Sigma_A\text{-E})$ et $(\Sigma_\exists - \mathrm{E})$ sont similaires. Traitons $(\Sigma_\exists - \mathrm{E})$. Si

$$\frac{\Gamma \vdash P : \Sigma[x{:}A].\,B \to s \qquad \Gamma \vdash c : \Sigma[x{:}A].\,B \qquad \Gamma; x : A; y : B \vdash f : P\,([x], y)_{\Sigma[x{:}A].\,B} \qquad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, c, P) : P\,c}$$

Montrons qu'il existe des termes de ICC_Σ $R'_1, R'_2, R'_3, S'_A, S'_B, A'_1, B'_1, A'_2, B'_2$ et une sorte s_1 tels que

$$\frac{\left\{\begin{array}{l}\Gamma^* \vdash P \Uparrow^I R'_1\\ \Gamma^* \vdash c \Uparrow^I R'_2\end{array}\right. \quad \left\{\begin{array}{l}\texttt{NF\_BI}(R'_1) = \exists x : A'_1.\,B'_1 \to s_1\\ \texttt{NF\_BI}(R'_2) = \exists x : A'_2.\,B'_2\\ \Gamma^* \vdash A \Uparrow^I S'_A\\ \Gamma^*; x : A^* \vdash B \Uparrow^I S'_B\\ \texttt{NF\_BI}(S'_A), \texttt{NF\_BI}(S'_B) \in \textsf{Sort}\end{array}\right. \quad \left\{\begin{array}{l}\texttt{cumu}(\exists x : A^*.\,B^*, \exists x : A'_1.\,B'_1)\\ \texttt{cumu}(\exists x : A'_2.\,B'_2, \exists x : A^*.\,B^*)\\ \Gamma^*; x : A^*; y : B^* \vdash f \Uparrow^I R'_3\\ x \notin \mathsf{FV}(f^*)\\ \texttt{cumu}(R'_3, P^*\,(\diamond, y))\end{array}\right.}{\Gamma^* \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, c, P) \Uparrow^I P^*\,c^*}\ (\textsc{Inf-U-Elim})$$

La condition de bord $x \notin \mathsf{FV}(f^*)$ est vraie par hypothèse. Montrons les cinq prémisses et les autres conditions de bord. Nous allons procéder en quatre étapes.

(1) Montrons qu'il existe R'_3 tel que $\Gamma^*; x : A^*; y : B^* \vdash f \Uparrow^I R'_3$ et $\texttt{cumu}(R'_3, P^*\,(\diamond, y))$.

- Par induction sur la prémisse $\Gamma; x : A; y : B \vdash f : P\,([x], y)_{\Sigma[x{:}A].\,B}$, il existe R'_3 tel que $\Gamma^*; x : A^*; y : B^* \vdash f \Uparrow^I R'_3$ et $R'_3 \preceq P^*\,(\diamond, y)$.

- Montrons que $\texttt{cumu}(R'_3, P^*(\diamond, y))$.
  - R'_3 est typable car c'est le terme inféré par l'algorithme à partir d'un contexte bien formé.
  - $P^*(\diamond, y)$ est typable comme terme extrait d'un terme typable dans AICC_Σ.
  - Puisque $R'_3 \preceq P^*(\diamond, y)$ et que R'_3 et $P^*(\diamond, y)$ sont typables, nous déduisons par complétude de cumu que $\texttt{cumu}(R'_3, P^*(\diamond, y))$.

(2) Montrons qu'il existe S'_A, S'_B tels que $\Gamma^* \vdash A \Uparrow^I S'_A$, $\Gamma^*; x : A^* \vdash B \Uparrow^I S'_B$, $\texttt{NF\_BI}(S'_A), \texttt{NF\_BI}(S'_B) \in \textbf{Sort}$.

- Montrons qu'il existe des sortes s_A, s_B telles qu'une dérivation de $\Gamma \vdash A : s_A$ et une dérivation de $\Gamma; x : A \vdash B : s_B$ sont incluses dans celle de $\Gamma; x : A; y : B \vdash f : P([x], y)_{\Sigma[x:A].B}$.
  - D'après le lemme de bonne formation de contextes, la dérivation de $\Gamma; x : A; y : B \vdash f : P([x], y)_{\Sigma[x:A].B}$, contient une dérivation de $\Gamma; x : A \vdash$ et une dérivation de $\Gamma; x : A; y : B \vdash$.
  - Nous concluons en inversant (WF-S_A) dans $\Gamma; x : A \vdash$ et $\Gamma; x : A; y : B \vdash$.
- Par hypothèses d'induction sur $\Gamma \vdash A : s_A$ et $\Gamma; x : A \vdash B : s_B$, il existe S'_A, S'_B tels que
  - $\Gamma^* \vdash A \Uparrow^I S'_A$,
  - $\Gamma^*; x : A^* \vdash B \Uparrow^I S'_B$,
  - $S'_A \preceq s_A$ et $S'_B \preceq s_B$.
- D'après le point (1) du lemme 4.1.3, nous déduisons $\texttt{NF\_BI}(S'_A), \texttt{NF\_BI}(S'_B) \in \textbf{Sort}$.

(3) Montrons qu'il existe R'_2, A'_2, B'_2 tels que $\Gamma^* \vdash c \Uparrow^I R'_2$, $\texttt{NF\_BI}(R'_2) = \exists x : A'_2 . B'_2$ et $\texttt{cumu}(\exists x : A'_2 . B'_2, \exists x : A^* . B^*)$.

- Par hypothèse d'induction, il existe R'_2 tel que $\Gamma^* \vdash c \Uparrow^I R'_2$ et $R'_2 \preceq \exists x : A^* . B^*$.
- D'après point (2) du lemme 4.1.3 il existe A'_2, B'_2 tels que $\texttt{NF\_BI}(R'_2) = \exists x : A'_2 . B'_2 \preceq \exists x : A^* . B^*$.
- Nous déduisons $\texttt{cumu}(\exists x : A'_2 . B'_2, \exists x : A^* . B^*)$ par complétude de cumu puisque $\exists x : A^* . B^*$ et $\exists x : A'_2 . B'_2$ sont typables :
  - $\exists x : A^* . B^*$ est typable en tant que terme extrait d'un terme typable dans AICC_Σ ;
  - $\exists x : A'_2 . B'_2 = \texttt{NF\_BI}(R'_2)$ est typable car c'est la forme normale d'un type inféré par l'algorithme appliqué à un contexte bien formé (corollaire 3.3.3).

(4) Montrons qu'il existe R'_1, A'_1, B'_1 et une sorte s_1 tels que $\Gamma^* \vdash P \Uparrow^I R'_1$, $\texttt{NF\_BI}(R'_1) = \exists x : A'_1 . B'_1 \to s_1$ et $\texttt{cumu}(\exists x : A^* . B^*, \exists x : A'_1 . B'_1)$.

- Par hypothèse d'induction, il existe R'_1 tel que $\Gamma^* \vdash P \Uparrow^I R'_1$ et $R'_1 \preceq \exists x : A^* . B^* \to s$.
- Montrons qu'il existe A'_1, B'_1 et une sorte s_1 tels que $\texttt{NF\_BI}(R'_1) = \exists x : A'_1 . B'_1 \to s_1$.
  - D'après le point (2) du lemme 4.1.3, il existe T'_1, U'_1 tels que $\texttt{NF\_BI}(R'_1) = \Pi x : T'_1 . U'_1 \preceq \exists x : A^* . B^* \to s$.
  - Par inversion, nous avons $U'_1 \preceq s$ et $\exists x : A^* . B^* \preceq T'_1$.
  - D'après le point (2) du corollaire 3.3.3, il existe une sorte s_0 telle que $\Gamma^* \vdash \Pi x : T'_1 . U'_1 : s_0$.
  - Par inversion, il existe des sortes s_T, s_U telles que $\Gamma^* \vdash T'_1 : s_T$ et $\Gamma^*; x : T'_1 \vdash U'_1 : s_U$.

— D'après le lemme 4.1.1 appliqué respectivement à U'_1 et T'_1, et par complétude de `NF_BI`, puisque U'_1 et T'_1 sont bien typés, il existe une sorte s_1 et des termes A'_1, B'_1 tels que $\texttt{NF\_BI}(U'_1) = s_1$ et $\texttt{NF\_BI}(T'_1) = \exists x : A'_1 . B'_1$.

- Montrons enfin que $\texttt{cumu}(\exists x : A^* . B^*, \exists x : A'_1 . B'_1)$.
  — Nous avons vu que $\exists x : A^* . B^* \preceq T'_1 \twoheadrightarrow_{\beta\iota} \exists x : A'_1 . B'_1$.
  — $\exists x : A^* . B^*$ est typable comme terme extrait d'un terme typable dans AICC_Σ.
  — $\exists x : A'_1 . B'_1$ est typable car c'est la forme $\beta\iota$-normale de T'_1 bien typé.
  — Nous concluons par complétude de `cumu`.

- $(\Sigma_{\text{SUB}} - \text{E} - 1)$ Si

$$\frac{\Gamma \vdash c : \Sigma x : A . [B]}{\Gamma \vdash \pi_1(c) : A}$$

montrons qu'il existe R', A', B' tels que

$$\frac{\Gamma^* \vdash c \Uparrow^{\text{I}} R' \qquad \texttt{NF\_BI}(R') = \{x : A' \mid B'\}}{\Gamma^* \vdash \pi_1(c) \Uparrow^{\text{I}} A'} \ (\text{INF-SUB-ELIM-1})$$

avec $A' \preceq A^*$.

- Par hypothèse d'induction, il existe R' tel que $\Gamma^* \vdash c \Uparrow^{\text{I}} R'$ et $R' \preceq \{x : A^* \mid B^*\}$.
- D'après le point (2) du lemme 4.1.3, il existe A', B' tels que $\texttt{NF\_BI}(R') = \{x : A' \mid B'\} \preceq \{x : A^* \mid B^*\}$.
- Enfin, par inversion de $\{x : A' \mid B'\} \preceq \{x : A^* \mid B^*\}$, nous avons bien $A' \preceq A^*$.

- $(\Sigma_{\text{SUB}} - \text{E} - 2)$ Si

$$\frac{\Gamma \vdash P : s \qquad \Gamma \vdash c : \Sigma x : A . [B] \qquad \Gamma; y : B_0 \vdash f : P \qquad \begin{array}{l} y \notin \text{FV}(f^*) \\ (B\,[x/\pi_1(c)])^* \preceq B_0^* \end{array}}{\Gamma \vdash \text{Elim}^s_{\text{I}_\text{R}}([y{:}B_0].f, c, P) : P}$$

Montrons qu'il existe des termes de ICC_Σ $S'_1, S'_2, R'_1, R'_2, A'_1, B'_1$ tels que :

$$\frac{\left\{\begin{array}{l} \Gamma^* \vdash P \Uparrow^{\text{I}} S'_1 \\ \Gamma^* \vdash c \Uparrow^{\text{I}} R'_1 \\ \Gamma^* \vdash B_0 \Uparrow^{\text{I}} S'_2 \end{array}\right. \quad \left\{\begin{array}{l} \texttt{NF\_BI}(S'_1) \in \textbf{Sort} \\ \texttt{NF\_BI}(R'_1) = \{x : A'_1 \mid B'_1\} \\ \texttt{NF\_BI}(S'_2) \in \textbf{Sort} \end{array}\right. \quad \left\{\begin{array}{l} \Gamma^*; y : B_0^* \vdash f \Uparrow^{\text{I}} R'_2 \\ \texttt{cumu}(R'_2, P^*) \\ \texttt{cumu}(B'_1\,[x/c^*], B_0^*) \\ y \notin \text{FV}(f^*) \end{array}\right.}{\Gamma^* \vdash \text{Elim}^s_{\text{I}_\text{R}}([y{:}B_0].f, c, P) \Uparrow^{\text{I}} P^*} \ (\text{INF-SUB-ELIM-2})$$

La condition de bord $y \notin \text{FV}(f^*)$ est vraie par hypothèse. Montrons les quatre prémisses et les autres conditions de bord. Nous allons procéder en quatre étapes.

(1) Montrons qu'il existe S'_1 tel que $\Gamma^* \vdash P \Uparrow^{\text{I}} S'_1$ et $\texttt{NF\_BI}(S'_1) \in \textbf{Sort}$.
  - Par hypothèse d'induction liée à la prémisse $\Gamma \vdash P : s$, il existe un terme de ICC_Σ S'_1 tel que $\Gamma^* \vdash P \Uparrow^{\text{I}} S'_1$ et $S'_1 \preceq s$.
  - D'après le le point (1) du lemme 4.1.3, $\texttt{NF\_BI}(S'_1)$ est une sorte.

(2) Montrons qu'il existe R'_1, A'_1, B'_1 tels que $\Gamma^* \vdash c \Uparrow^{\text{I}} R'_1$, $\texttt{NF\_BI}(R'_1) = \{x : A'_1 \mid B'_1\}$ et $\texttt{cumu}(B'_1\,[x/c^*], B_0^*)$.
  - De même que précédemment, par hypothèse d'induction, il existe un terme non-annoté R'_1 tel que $\Gamma^* \vdash c \Uparrow^{\text{I}} R'_1$ et $R'_1 \preceq \{x : A^* \mid B^*\}$.

- D'après le point (2) du lemme 4.1.3, il existe A'_1, B'_1 tels que $\texttt{NF\_BI}(R'_1) = \{x : A'_1 \mid B'_1\} \preceq \{x : A^* \mid B^*\}$.
- Montrons que $\texttt{cumu}(B'_1[x/c^*], B^*_0)$.
  — Nous obtenons $B'_1[x/c^*] \preceq B^*_0$ par inversion de $\{x : A'_1 \mid B'_1\} \preceq \{x : A^* \mid B^*\}$ et d'après l'hypothèse $(B[x/\pi_1(c)])^* \preceq B^*_0$.
  — Par correction de l'extraction, nous déduisons de $\Gamma; y : B_0 \vdash f : P$ que B^*_0 est bien typé dans ICC_Σ.
  — Montrons qu'il existe une sorte s'_1 telle que $\Gamma^*; x : A^* \vdash B'_1 : s'_1$.
    — Puisque $\Gamma^* \vdash$, d'après le point (2) du corollaire 3.3.3 appliqué à $\Gamma^* \vdash c \Uparrow^I R'_1$, il existe une sorte s_1 telle que $\Gamma^* \vdash \{x : A'_1 \mid B'_1\} : s_1$.
    — Par inversion de $\Gamma^* \vdash \{x : A'_1 \mid B'_1\} : s_1$, il existe une sorte s'_1 telle que $\Gamma^*; x : A'_1 \vdash B'_1 : s'_1$.
    — Nous montrons facilement à partir de l'hypothèse $\Gamma \vdash c : \Sigma x : A.[B]$ que $\Gamma; x : A \vdash$ puis $\Gamma^*; x : A^* \vdash$.
    — Puisque $A^* \preceq A'_1$, en changeant le contexte, nous déduisons que $\Gamma^*; x : A^* \vdash B'_1 : s'_1$.
  — Montrons que $B'_1[x/c^*]$ est bien typé dans ICC_Σ.
    — De $\Gamma \vdash c : \Sigma x : A.[B]$, nous déduisons $\Gamma^* \vdash c^* : A^*$.
    — Par substitutivité sur $\Gamma^*; x : A^* \vdash B'_1 : s'_1$, nous obtenons $\Gamma^* \vdash B'_1[x/c^*] : s'_1$.
  — Nous concluons alors par complétude de cumu, puisque B^*_0 et $B'_1[x/c^*]$ sont typables.

(3) Montrons qu'il existe S'_2 tel que $\Gamma^* \vdash B_0 \Uparrow^I S'_2$ et $\texttt{NF\_BI}(S'_2) \in \textbf{Sort}$.
- La dérivation de $\Gamma; y : B_0 \vdash$ est incluse dans celle de $\Gamma; y : B_0 \vdash f : P$. Par inversion de $\Gamma; y : B_0 \vdash$, il existe une sorte s_2 telle qu'une dérivation de $\Gamma \vdash B_0 : s_2$ existe et est incluse dans celle de $\Gamma; y : B_0 \vdash f : P$.
- Nous avons donc comme hypothèse d'induction qu'il existe un terme S'_2 tel que $\Gamma^* \vdash B_0 \Uparrow^I S'_2$ et $S'_2 \preceq s_2$.
- D'après le le point (1) du lemme 4.1.3, $\texttt{NF\_BI}(S'_2)$ est bien une sorte.

(4) Montrons qu'il existe R'_2 tel que $\Gamma^*; y : B^*_0 \vdash f \Uparrow^I R'_2$ et $\texttt{cumu}(R'_2, P^*)$.
- Par hypothèse d'induction, il existe R'_2 tel que $\Gamma^*; y : B^*_0 \vdash f \Uparrow^I R'_2$ et $R'_2 \preceq P^*$.
- Montrons que $\texttt{cumu}(R'_2, P^*)$.
  — P^* est bien typé car P est bien typé dans AICC_Σ.
  — D'après le point (1) du corollaire 3.3.3 appliqué à $\Gamma^*; y : B^*_0 \vdash$ et $\Gamma^*; y : B^*_0 \vdash f \Uparrow^I R'_2$, R'_2 est bien typé.
  — Puisque R'_2 et P^* sont bien typés, nous déduisons $\texttt{cumu}(R'_2, P^*)$ de $R'_2 \preceq P^*$ par complétude de cumu.

□

4.2.2. REMARQUE. La complétude nous montre que le type inféré extrait, s'il existe, est le plus petit possible dans ICC_Σ. Il est donc l'extraction d'un type principal annoté.

5. Algorithmes dérivés

Nous définissons dans cette section deux algorithmes : le premier vérifie la bonne formation d'un contexte ; le deuxième est un algorithme de vérification de type. Dans les systèmes de type habituels, ces deux algorithmes nécessitent de pouvoir inférer un type. Nous montrons dans cette section que, dans AICC_Σ, pouvoir inférer un type *extrait* suffit.

5.1. Lemme auxiliaire. Le résultat suivant est utile pour les preuves de correction et de complétude des deux algorithmes de cette section.

5.1.1. Lemme. *Soient* Γ *un contexte annoté et* T *un terme annoté alors les deux propositions suivantes sont équivalentes*

(1) il existe une sorte s *telle que* $\Gamma \vdash T : s$

(2) $\Gamma \vdash$ *et il existe un terme* S' *de* ICC_Σ *tel que* $\Gamma^* \vdash T \Uparrow^I S'$ *et* $\mathtt{NF\_BI}(S') \in \mathbf{Sort}$.

DÉMONSTRATION.

- (1) ⇒ (2) : Si $\Gamma \vdash T : s$, par le lemme de bonne formation de contexte, nous déduisons $\Gamma \vdash$. Montrons qu'il existe S' dans ICC_Σ tel que $\Gamma^* \vdash T \Uparrow^I S'$ et $\mathtt{NF\_BI}(S') \in \mathbf{Sort}$.
  - Par complétude de l'algorithme d'inférence de type extrait, il existe S' dans ICC_Σ tel que $\Gamma^* \vdash T \Uparrow^I S'$ et $S' \preceq s$.
  - Montrons que $\mathtt{NF\_BI}(S')$ existe et que $\mathtt{NF\_BI}(S') \in \mathbf{Sort}$.
    — D'après le point (1) du corollaire 3.3.3 appliqué à $\Gamma^* \vdash$ et $\Gamma^* \vdash T \Uparrow^I S'$, S' est bien typé dans ICC_Σ.
    — Puisque S' est bien typé et que $S' \preceq s$, le lemme 4.1.1 nous indique que $\mathsf{nf}_{\beta\iota}(S')$.
    — Puisque S' est bien typé, la complétude de NF_BI nous indique que $\mathtt{NF\_BI}(S') = \mathsf{nf}_{\beta\iota}(S')$.
- (2) ⇒ (1) : Si $\Gamma \vdash$, $\Gamma^* \vdash T \Uparrow^I S'$ et $s = \mathtt{NF\_BI}(S') \in \mathbf{Sort}$, montrons que $\Gamma \vdash T : s$.
  - Par correction de l'algorithme d'inférence de type, puisque $\Gamma \vdash$, il existe un terme annoté S tel que $\Gamma \vdash T : S$ et $S^* = S'$.
  - Par correction de NF_BI, nous avons $S^* \twoheadrightarrow_{\beta\iota} s$.
  - Nous concluons alors par cumulativité, puisque toute sorte est typable dans un contexte bien typé.

□

5.1.2. REMARQUE. Ce lemme signifie que l'algorithme d'inférence de type extrait est suffisant pour décider si un terme de $AICC_\Sigma$ est typable par une sorte ou pas. Ceci explique pourquoi les deux algorithmes suivants ne nécessitent pas de pouvoir inférer un type annoté.

5.2. Bonne formation d'un contexte. Nous avons observé dans la remarque 3.2.2 que puisque l'algorithme d'inférence n'est correct que pour les contextes bien formés, il est nécessaire de définir un algorithme vérifiant la bonne formation d'un contexte.

5.2.1. DÉFINITION (Vérification de la bonne formation d'un contexte). Pour tout contexte annoté Γ, nous notons $\mathtt{wf}(\Gamma)$ l'algorithme de *vérification de la bonne formation* de Γ.

Cet algorithme est défini inductivement par les deux règles d'inférence suivantes :

$$\frac{}{\mathtt{wf}(\bullet)}$$

$$\frac{\mathtt{wf}(\Gamma) \qquad \Gamma^* \vdash T \Uparrow^I S' \qquad \mathtt{NF\_BI}(S') \in \mathbf{Sort}}{\mathtt{wf}(\Gamma; x : T)}$$

5.2.2. Lemme (Spécification)**.** *L'algorithme de vérification de la bonne formation d'un contexte est :*

(1) termine ;

(2) correct : pour tout contexte annoté Γ, *si* $\mathtt{wf}(\Gamma)$ *alors* $\Gamma \vdash$;

(3) complet : pour tout contexte annoté Γ, *si* $\Gamma \vdash$ *alors* $\mathtt{wf}(\Gamma)$.

DÉMONSTRATION. Soit Γ un contexte annoté.

(1) est évident.

(2) Si $\mathtt{wf}(\Gamma)$, montrons par induction sur la structure de Γ que $\Gamma \vdash$.
- Nous avons bien $\bullet \vdash$;
- Si $\Gamma = \Gamma_0; x : \mathrm{T}$ alors

$$\frac{\mathtt{wf}(\Gamma_0) \qquad \Gamma_0^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}') \in \mathsf{Sort}}{\mathtt{wf}(\Gamma_0; x : \mathrm{T})}$$

  Montrons que $\Gamma_0; x : \mathrm{T} \vdash$.
  - Par induction sur $\mathtt{wf}(\Gamma_0)$, nous déduisons $\Gamma_0 \vdash$.
  - Posons $s = \mathtt{NF\_BI}(\mathrm{S}')$. Le lemme 5.1.1 nous indique que $\Gamma_0 \vdash \mathrm{T} : s$.
  - Nous concluons en appliquant $(\mathrm{WF\text{-}S_A})$.

(3) Si $\Gamma \vdash$, montrons par induction sur la structure de Γ que $\mathtt{wf}(\Gamma)$.
- Nous avons bien $\mathtt{wf}(\bullet)$.
- Si $\Gamma = \Gamma_0; x : \mathrm{T}$, montrons qu'il existe un terme S' de ICC_Σ tel que

$$\frac{\mathtt{wf}(\Gamma_0) \qquad \Gamma_0^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}') \in \mathsf{Sort}}{\mathtt{wf}(\Gamma_0; x : \mathrm{T})}$$

  - L'hypothèse d'induction sur Γ_0 nous indique que $\mathtt{wf}(\Gamma_0)$.
  - Montrons qu'il existe S' tel que $\Gamma_0^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'$ et $\mathtt{NF\_BI}(\mathrm{S}') \in \mathsf{Sort}$.
    — Par inversion de $\Gamma_0; x : \mathrm{T} \vdash$, il existe une sorte s telle que $\Gamma_0 \vdash \mathrm{T} : s$.
    — Nous concluons en appliquant le lemme 5.1.1.

□

5.3. Vérification de type. Nous présentons et étudions dans cette sous-section un algorithme de vérification de type dans AICC_Σ.

L'algorithme est défini inductivement par un jugement reliant un contexte et deux termes de AICC_Σ.

5.3.1. DÉFINITION (Jugement de vérification de type). Soient Γ un contexte de AICC_Σ, M, T des termes de AICC_Σ. Le *jugement de vérification de type*, noté $\Gamma \vdash \mathrm{M} \Downarrow \mathrm{T}$, signifie que nous vérifions que T est un type de M dans Γ.

5.3.2. DÉFINITION (Algorithme de vérification de type). L'algorithme de vérification de type prend en entrée un contexte annoté Γ et deux termes annotés M, T et est défini par l'unique règle d'inférence suivante :

$$\frac{\Gamma^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}') \in \mathsf{Sort} \qquad \mathtt{cumu}(\mathrm{T}', \mathrm{T}^*) \qquad \Gamma^* \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{T}'}{\Gamma \vdash \mathrm{M} \Downarrow \mathrm{T}}$$

L'algorithme de vérification de typage fait donc essentiellement deux choses :

(1) vérifier que le type donné en entrée est typable par une sorte ;

(2) inférer un type extrait pour le terme et vérifier que ce type est plus petit (pour $\preceq$) que l'extraction du type donné en entrée.

La première opération ne nécessite pas d'inférer un type annoté d'après le lemme 5.1.1 ; la seconde non plus car les termes comparés sont des termes extraits.

5.3.3. Proposition. *L'algorithme de vérification de type*

(1) termine si $\Gamma \vdash$ *;*

(2) est correct : si $\Gamma \vdash$ *et* $\Gamma \vdash M \Downarrow T$ *alors* $\Gamma \vdash M : T$ *;*

(3) est complet : si $\Gamma \vdash M : T$ *alors* $\Gamma \vdash M \Downarrow T$.

DÉMONSTRATION.

(1) Terminaison :

- l'algorithme d'inférence de type extrait termine pour tout contexte non-annoté et tout terme annoté ;
- $\texttt{NF\_BI}(S')$ existe puisque S', type inféré, est typable car $\Gamma^* \vdash$;
- l'appartenance à **Sort** et l'extraction sont décidables,
- l'algorithme `cumu` termine pour des termes typables dans un contexte :
  - T' est le type inféré par l'algorithme d'inférence de type extrait donc est bien typé puisque $\Gamma^* \vdash$,
  - par correction de l'algorithme d'inférence de type extrait, nous avons $\Gamma^* \vdash T^* : S'$.

(2) Correction : si $\Gamma \vdash$ et si

$$\frac{\Gamma^* \vdash T \Uparrow^I S' \qquad \texttt{NF\_BI}(S') \in \textbf{Sort} \qquad \texttt{cumu}(T', T^*) \qquad \Gamma^* \vdash M \Uparrow^I T'}{\Gamma \vdash M \Downarrow T}$$

montrons que $\Gamma \vdash M : T$.

- Par correction de l'algorithme d'inférence de type extrait sur $\Gamma^* \vdash M \Uparrow^I T'$, il existe un terme de AICC_Σ T_0 tel que $\Gamma \vdash M : T_0$ et $T_0^* = T'$.
- Posons $s = \texttt{NF\_BI}(S')$. Le lemme 5.1.1 nous indique que $\Gamma \vdash T : s$.
- Par correction de `cumu`, nous avons $T_0^* = T' \preceq T^*$.
- Nous pouvons donc appliquer (CUM_A) et dériver $\Gamma \vdash M : T$.

(3) Complétude : si $\Gamma \vdash M : T$, montrons qu'il existe S', T' tels que

$$\frac{\Gamma^* \vdash T \Uparrow^I S' \qquad \texttt{NF\_BI}(S') \in \textbf{Sort} \qquad \texttt{cumu}(T', T^*) \qquad \Gamma^* \vdash M \Uparrow^I T'}{\Gamma \vdash M \Downarrow T}$$

- D'après le lemme du type des types appliqué à $\Gamma \vdash M : T$, il existe une sorte s telle que $\Gamma \vdash T : s$.
- D'après le lemme 5.1.1, il existe un terme S' de ICC_Σ tel que $\Gamma^* \vdash T \Uparrow^I S'$ et $\texttt{NF\_BI}(S') \in$ **Sort**.
- Par complétude de l'algorithme d'inférence de type extrait sur $\Gamma \vdash M : T$, il existe un terme T' tel que $\Gamma^* \vdash M \Uparrow^I T'$ et $T' \preceq T^*$.
- T' et T^* sont typables d'où $\texttt{cumu}(T', T)$.

□

Chapitre 9

Inférence de type

Nous présentons et étudions dans ce chapitre un algorithme d'inférence de type (annoté) correct et complet. Cet algorithme prend en entrée un contexte Γ et un terme M annotés et retourne un terme annoté R ou bien échoue. Si l'algorithme n'échoue pas et si Γ est bien formé, alors le type retourné R est un type de M dans Γ (correction). Réciproquement, si M est bien typé dans Γ alors l'algorithme n'échoue pas et le type retourné est un type principal (complétude).

Comme nous l'avons précisé dans l'introduction de la quatrième partie, il manque un mécanisme de réduction dans AICC_Σ pour pouvoir établir un tel algorithme d'inférence de type. Le cas difficile est, une fois de plus, celui des éliminateurs. En effet, inférer le type d'une sorte ou d'une variable n'est pas difficile. Pour les constructeurs de type et les constructeurs, inférer un type semble possible car s'ils sont bien formés, alors l'*extraction* de certains types inférés doit $\beta\iota$-normaliser vers des sortes. Mais pour les éliminateurs, l'inférence de type est plus délicate. Inférer les types des sous-termes et $\beta\iota$-réduire l'extraction de ces types inférés ne suffit plus : nous devons obtenir des constructeurs de type de AICC_Σ et il n'est pas évident de construire un constructeur de type annoté à partir d'un constructeur de type de ICC_Σ.

Pour résoudre ce problème, nous proposons dans ce chapitre un algorithme de réduction dans AICC_Σ. Celui-ci permet d'obtenir un constructeur de type de AICC_Σ à partir d'un terme annoté qui (1) est typé par une sorte et (2) s'extrait vers un terme comparable à un constructeur de type de ICC_Σ. Cet algorithme est présenté et étudié dans les quatre premières sections. La première introduit les règles de réduction dans AICC_Σ utilisées par l'algorithme. La seconde montre que ces règles sont complètes, c'est-à-dire qu'elles permettent bien, dans les conditions indiquées, d'obtenir un constructeur de type. La troisième montre que les règles sont correctes, c'est-à-dire qu'elles sont compatibles avec la réduction dans ICC_Σ et que le typage est préservé. La section 4 définit l'algorithme de réduction et montre qu'il vérifie bien la spécification souhaitée. Enfin la cinquième et dernière section présente l'algorithme d'inférence de type et montre qu'il est correct et complet.

1. Règles de réduction

Nous présentons dans cette section des règles de réduction pour AICC_Σ. Ces règles sont à la base de l'algorithme de réduction que nous utiliserons pour l'algorithme d'inférence de type.

Comme nous l'avons énoncé en introduction de ce chapitre, nous avons besoin d'un mécanisme qui, à partir d'un terme annoté R typé par une sorte et tel que $\mathrm{R}^* \preceq \Box' x:\mathrm{T}'_0.\mathrm{U}'_0$, retourne un constructeur de type de AICC_Σ $\Box x:\mathrm{T}.\mathrm{U}$. Pour pouvoir utiliser ce mécanisme dans un algorithme d'inférence correct et complet, le constructeur de type retourné doit être bien typé (donc typable par une sorte) et tel que $\mathrm{R}^* \twoheadrightarrow_{\beta\iota} (\Box x:\mathrm{T}.\mathrm{U})^*$. [1]

Pour simplifier le raisonnement sur ce mécanisme, nous allons imposer que le typage soit préservé, c'est-à-dire que si $\Gamma \vdash \mathrm{R} : s$ alors $\Gamma \vdash \Box x:\mathrm{T}.\mathrm{U} : s$. Notons que cette contrainte supplémentaire n'est pas strictement nécessaire.

1. Ces contraintes sont nécessaires pour que si $\Gamma \vdash \mathrm{M} : \mathrm{R}$ avec R type principal, nous puissions avoir également, par cumulativité, $\Gamma \vdash \mathrm{M} : \Box x:\mathrm{T}.\mathrm{U}$ avec $\Box x:\mathrm{T}.\mathrm{U}$ type principal.

Expliquons informellement ce que le mécanisme est censé faire.

Remarquons tout d'abord que R^* se $\beta\iota$-réduit vers un constructeur de type de ICC_Σ. En effet, puisque $R^* \preceq \Box' x : T'_0 . U'_0$, nous savons que R^* se $\beta\iota\eta$-réduit vers un constructeur de type de ICC_Σ. Par report de la η-réduction nous avons donc $R^* \twoheadrightarrow_{\beta\iota} R'_1 \twoheadrightarrow_\eta \Box' x : T' . U'$. R'_1 est donc soit une abstraction (s'il y a une η-réduction à la racine) ou un constructeur de type (si toutes les réductions ont lieu dans des sous-termes stricts). R est typé par une sorte, donc R^* également par correction de l'extraction, puis R'_1 aussi car la $\beta\iota$-réduction préserve le typage. Puisqu'une abstraction ne peut être typée par une sorte, R'_1 est un constructeur de type. Notons-le $\Box' x : T'_1 . U'_1$. Nous avons alors bien $R^* \twoheadrightarrow_{\beta\iota} \Box' x : T'_1 . U'_1$.

Puisque $R^* \twoheadrightarrow_{\beta\iota} \Box' x : T'_1 . U'_1$, une idée naturelle va être de chercher à simuler dans $AICC_\Sigma$ les réductions se déroulant dans ICC_Σ. Cela n'est néanmoins pas suffisant. Si nous parvenons à transformer R en R_1 tel que $R_1^* = \Box' x : T'_1 . U'_1$, R_1 n'est pas nécessairement un constructeur de type. Par inversion de l'extraction R_1 peut-être un constructeur de type ou un constructeur implicite ou un éliminateur implicite. Le fait que R_1 est typé par une sorte permet d'éliminer le cas du constructeur implicite mais R_1 peut toujours être un éliminateur implicite. Il faut donc également un mécanisme a priori indépendant des réductions de ICC_Σ qui fera monter en tête la structure de constructeur de type cachée sous des éliminateurs implicites.

1.1. Réductions standard analogues à ICC_Σ. Une première étape pour simuler dans $AICC_\Sigma$ les réductions de ICC_Σ est d'ajouter dans des règles analogues aux règles de réduction de ICC_Σ.

Nous définissons ainsi l'équivalent de la β-réduction :

$$(\lambda x{:}\mathrm{T} . \mathrm{M})\,\mathrm{N} \quad \rhd^{\mathrm{A}}_{\beta^e} \quad \mathrm{M}[x/\mathrm{N}]$$

puis deux règles analogues aux règles de ι-réduction :

$$\begin{aligned} \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{T}).(y{:}\mathrm{U}).f, (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}}, \mathrm{P}) \quad &\rhd^{\mathrm{A}}_{\iota_\Sigma} \quad f[x/a][y/b] \\ \mathsf{Elim}_{\mathrm{I}_\exists}([x{:}\mathrm{T}].(y{:}\mathrm{U}).f, ([a],b)_{\Sigma[x\mathrm{A}].\mathrm{B}}, \mathrm{P}) \quad &\rhd^{\mathrm{A}}_{\iota_\exists} \quad f[x/a][y/b] \end{aligned}$$

1.2. Réductions standard implicites. Ces règles ne sont néanmoins pas suffisantes pour simuler toutes les réductions dans ICC_Σ. Ainsi, par exemple, le terme $\pi_1((\lambda x{:}\mathrm{T} . \mathrm{M}, [b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]})\,\mathrm{N}$, bien typé a priori, ne contient pas de redex pour les règles précédentes mais s'extrait vers le β-redex $(\lambda x . \mathrm{M}^*)\,\mathrm{N}^*$.

Le problème vient de ce que les éléments déclenchant la réduction dans ICC_Σ, qui sont des constructeurs[2], sont extraits à partir de termes annotés qui sont a priori des constructeurs explicites, des constructeurs implicites ou des éliminateurs implicites. Ainsi $(\lambda x . \mathrm{M}^*)\,\mathrm{N}^*$ peut être l'extraction de $(\lambda x{:}\mathrm{T} . \mathrm{M})\,\mathrm{N}$ ou de $(\lambda[z{:}\mathrm{A}] . (\lambda x{:}\mathrm{T} . \mathrm{M}))\,\mathrm{N}$ ou encore de $\pi_1((\lambda x{:}\mathrm{T} . \mathrm{M}, [b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]})\,\mathrm{N}$.

Si le constructeur est l'extraction d'un constructeur explicite, alors l'une des trois règles ci-dessus s'applique (ainsi $(\lambda x{:}\mathrm{T} . \mathrm{M})\,\mathrm{N} \rhd^{\mathrm{A}}_{\beta^e} \mathrm{M}[x/\mathrm{N}]$). Si le constructeur est l'extraction d'un constructeur implicite, alors nous avons un terme mal typé ($(\lambda[z{:}\mathrm{A}] . \lambda x{:}\mathrm{T} . \mathrm{M})\,\mathrm{N}$ par exemple), ce qui nous permet de mettre de côté ce cas. Pour résoudre le cas où le constructeur est l'extraction d'un éliminateur implicite, nous devons rajouter des nouvelles règles de réduction.

2. L'abstraction pour la β-réduction, la paire explicite et la paire existentielle pour les règles de ι-réduction.

Les deux règles suivantes traitent les cas de l'application implicite et de la première projection. Celui de l'éliminateur simplifié est différent et est traité dans la prochaine sous-section.

$$\begin{aligned}(\lambda[x{:}\mathrm{T}]\,.\,\mathrm{M})\,[\mathrm{N}] &\quad \triangleright^{\mathrm{A}}_{\beta^i} \quad \mathrm{M}\,[x/\mathrm{N}]\\ \pi_1((a,[b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]}) &\quad \triangleright^{\mathrm{A}}_{\iota^1_\sigma} \quad a\end{aligned}$$

Ces règles sont similaires à celles rajoutées en premier : ce sont des éliminateurs ayant comme sous-terme un constructeur associé : l'abstraction implicite pour l'application implicite, la paire implicite droite pour la première projection. Elles permettent bien l'application des règles $\triangleright^{\mathrm{A}}_{\beta^e}$,$\triangleright^{\mathrm{A}}_{\iota_\Sigma}$, et $\triangleright^{\mathrm{A}}_{\iota_\exists}$ quand un constructeur explicite est encoquillé dans une application implicite ou une première projection. Ainsi nous avons (en notant $\rightarrow^{\mathrm{A}}_{\iota^1_\sigma}$ l'extension au contexte de $\triangleright^{\mathrm{A}}_{\iota^1_\sigma}$) :

$$\pi_1((\lambda x{:}\mathrm{T}\,.\,\mathrm{M},[b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]})\,\mathrm{N} \rightarrow^{\mathrm{A}}_{\iota^1_\sigma} (\lambda x{:}\mathrm{T}\,.\,\mathrm{M})\,\mathrm{N} \triangleright^{\mathrm{A}}_{\beta^e} \mathrm{M}\,[x/\mathrm{N}]$$

qui correspond dans ICC_Σ à $(\lambda x\,.\,\mathrm{M}^*)\,\mathrm{N}^* \triangleright_\beta \mathrm{M}^*\,[x/\mathrm{N}^*]$.

1.3. Règles pour l'éliminateur simplifié. Nous pourrions être tentés d'ajouter la règle suivante, analogue pour l'éliminateur simplifié à ce que sont $\triangleright^{\mathrm{A}}_{\beta^i}$ pour l'application implicite et $\triangleright^{\mathrm{A}}_{\iota^1_\sigma}$ pour la première projection :

$$\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{T}].f,(a,[b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]},\mathrm{P}) \quad \triangleright^{\mathrm{A}}_{\iota^2_\sigma} \quad f\,[y/b]$$

Cette règle paraît naturelle et permet de résoudre certains blocages. Ainsi nous aurions :

$$\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].(\lambda z{:}\mathrm{T}\,.\,\mathrm{M}),(a,[b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]},\mathrm{P})\,\mathrm{N} \triangleright^{\mathrm{A}}_{\iota^2_\sigma} (\lambda z{:}(\mathrm{T}\,[y/b])\,.\,(\mathrm{M}\,[y/b]))\,\mathrm{N} \triangleright^{\mathrm{A}}_{\beta^e} \mathrm{M}\,[y/b]\,[z/\mathrm{N}]$$

qui simulerait dans ICC_Σ, en supposant $y \notin \mathsf{FV}(\mathrm{M}^*)$:

$$(\lambda z\,.\,\mathrm{M}^*)\,\mathrm{N}^* \triangleright_\beta \mathrm{M}^*\,[z/\mathrm{N}^*]$$

Malheureusement elle est trop restrictive et ne permet de simuler certaines réductions. Ainsi le terme $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\lambda z{:}\mathrm{T}\,.\,\mathrm{M},x,\mathrm{P})\,\mathrm{N}$, a priori bien typé, ne contient pas de redex pour cette règle (ni pour aucune des règles précédentes) mais s'extrait vers le β-redex $(\lambda z\,.\,\mathrm{M}^*)\,\mathrm{N}^*$.

Ce blocage est lié à une caractéristique de l'éliminateur simplifié qui le distingue des deux autres éliminateurs implicites : l'objet éliminé c d'un éliminateur simplifié $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,c,\mathrm{P})$ est effacé par l'extraction. Ainsi, avoir des informations sur $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,c,\mathrm{P})^* = f^*$ ne nous donne aucun renseignement supplémentaire sur c. Nous sommes alors bloqués quand c n'est pas une paire implicite droite (et ne se réduit pas vers une paire implicite droite).

Pour ces raisons, il ne paraît pas nécessaire à ce stade d'ajouter la règle $\triangleright^{\mathrm{A}}_{\iota^2_\sigma}$. Nous allons à la place ajouter des règles dont la contrainte se porte sur la branche f plutôt que sur l'objet éliminé c.

Éliminateur simplifié et constructeurs. Nous considérons tout d'abord trois règles dont le terme de gauche est un éliminateur simplifié $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,c,\mathrm{P})$ où c est quelconque mais où f est un constructeur de type et dont le terme de droite est un constructeur de type contenant un (pour les abstractions) ou deux (pour la paire explicite) éliminateurs simplifiés.

Abstractions

$$\begin{aligned}\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\lambda z{:}\mathrm{T}\,.\,\mathrm{M},c,\Pi z:\mathrm{T}_0\,.\,\mathrm{U}_0) &\quad \triangleright^{\mathrm{A}}_{\lambda-\mathsf{Simpl}} \quad \lambda z{:}\mathrm{T}_0\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M},c,\mathrm{U}_0)\\ \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\lambda[z{:}\mathrm{T}]\,.\,\mathrm{M},c,\Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0) &\quad \triangleright^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}} \quad \lambda[z{:}\mathrm{T}_0]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M},c,\mathrm{U}_0)\end{aligned}$$

Paire dépendante explicite

$$\frac{a_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].a, c, \mathrm{A}_0) \qquad b_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].b, c, \mathrm{B}_0\,[z/a_0])}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].(a,b)_{\Sigma z:\mathrm{A}.\mathrm{B}}, c, \Sigma z:\mathrm{A}_0\,.\,\mathrm{B}_0) \rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}} (a_0, b_0)_{\Sigma z:\mathrm{A}_0\,.\,\mathrm{B}_0}}$$

Notons tout d'abord que, dans les termes de gauche, les annotations de type P sont restreintes à des constructeurs de type. Cette restriction est nécessaire pour exprimer les annotations de type des différents éliminateurs simplifiés dans les termes de droite.

Par ailleurs, nous avons seulement trois règles et non cinq car l'équivalent de la règle $\rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$ pour les paires implicites ne préserverait pas le typage. [3] Ainsi certains blocages restent. Par exemple $\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].([a],b)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}}, d, \mathrm{U}), \mathrm{P})$ s'extrait vers un ι-redex mais ne peut se réduire dans AICC_Σ.

Éliminateur simplifié et autres éliminateurs de paire implicite. Pour résoudre ce problème, nous introduisons deux nouvelles règles dont le terme de droite, au lieu d'être un éliminateur simplifié avec un constructeur comme branche, va être un autre éliminateur avec comme objet éliminé un éliminateur simplifié. [4]

$$\frac{c = \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U})}{\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) \rhd^{\mathrm{A}}_{\exists-\mathsf{Simpl}} \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, g, \mathrm{P}), d, \mathrm{P}\,c)}$$

$$\pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x:\mathrm{T}\,.\,[\mathrm{U}])) \quad \rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}} \quad \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T})$$

Notons de même que précédemment la restriction sur l'annotation de type de l'éliminateur simplifié pour la règle $\rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$ afin de pouvoir exprimer l'annotation de type de l'éliminateur du terme de droite.

Ces deux règles jouent pour les règles $\rhd^{\mathrm{A}}_{\iota^1_\sigma}$ et $\rhd^{\mathrm{A}}_{\iota_\exists}$ le même rôle que les règles $\rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}}$, $\rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}}$ et $\rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$ jouent pour les règles $\rhd^{\mathrm{A}}_{\beta^e}$, $\rhd^{\mathrm{A}}_{\beta^i}$ et $\rhd^{\mathrm{A}}_{\iota_\Sigma}$: débloquer les cas où elles ne peuvent s'appliquer parce qu'un constructeur est encoquillé dans un éliminateur simplifié.

1.4. Éliminateur simplifié et constructeurs de type. Les règles présentées ci-dessus servent à simuler les réductions de ICC_Σ dans AICC_Σ. Comme nous l'avons vu, ce n'est pas suffisant car tout terme annoté qui s'extrait vers un constructeur de type de ICC_Σ n'est pas nécessairement un constructeur de type de AICC_Σ.

Ainsi si $\mathrm{R}^* = \Box' x:\mathrm{T}'\,.\,\mathrm{U}'$, R peut être un constructeur de type ou un éliminateur implicite ou encore un constructeur implicite. Nous pouvons éliminer le cas du constructeur implicite si nous supposons, ce qui sera le cas dans le cadre de l'algorithme d'inférence de type, que R est typé par une sorte. Il faut donc un mécanisme pour transformer un éliminateur implicite R qui s'extrait vers un constructeur de type de ICC_Σ $\Box' x:\mathrm{T}'\,.\,\mathrm{U}'$ en un constructeur de type de AICC_Σ.

Nous allons procéder pour les constructeurs de type encoquillés dans des constructeurs implicites comme nous l'avons fait pour les constructeurs.

Ainsi les règles $\rhd^{\mathrm{A}}_{\beta^i}$ et $\rhd^{\mathrm{A}}_{\iota^1_\sigma}$ définies ci-dessus peuvent s'appliquer quand R est une application implicite, une première projection.

3. Plus de détails dans la preuve du lemme 3.2.2 (cas de la règle $\rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$).

4. qui pourra notamment avoir comme sous-terme une paire implicite

Pour l'éliminateur simplifié, nous devons ajouter une règle similaire à $\triangleright^{\mathrm{A}}_{\lambda-\mathsf{Simpl}}$: le terme de droite est un éliminateur simplifié dont la branche est un constructeur de type.

Nous ajoutons donc la règle suivante, qui transforme un éliminateur simplifié $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\Box x:\mathrm{T}\,.\,\mathrm{U}, c, \mathrm{P})$ en constructeur de type en insérant l'éliminateur simplifié dans les domaine et codomaine :

$$\frac{\begin{cases}\Gamma^*; y:\mathrm{B}^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{T}} \\ \Gamma^*; y:\mathrm{B}^*; z:\mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{U}}\end{cases} \qquad \begin{cases}\mathrm{T}_0 = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{T}, c, \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{T}})) \\ \mathrm{U}_0 = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{U}, c, \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{U}}))\end{cases}}{\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\Box z:\mathrm{T}\,.\,\mathrm{U}, c, \mathrm{P}) \triangleright^{\mathrm{A}}_{\Gamma,\ \Box-\mathsf{Simpl}} \Box z:\mathrm{T}_0\,.\,\mathrm{U}_0}$$

Remarquons que la règle fait appel à l'inférence de type extrait. [5] Cela semble inévitable si nous voulons exprimer l'annotation de type dans les éliminateurs simplifiés du terme de droite. Imposer, de manière analogue aux règles $\triangleright^{\mathrm{A}}_{\mathsf{Constr}-\mathsf{Simpl}}$ et $\triangleright^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$, où P est un constructeur de type, que P soit une sorte fonctionnerait pour $\Sigma x:\mathrm{A}\,.\,\mathrm{B}$ et $\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$ mais pas pour les produits ou $\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]$. [6]

Puisqu'inférer un type nécessite d'avoir un contexte, cette règle de réduction est donc une relation ternaire avec un contexte comme troisième argument.

1.5. Règles supplémentaires. Nous ajoutons trois règles qui ne sont pas nécessaires pour aboutir à un algorithme de réduction, mais qui permettent d'avoir (1) un résultat de classification des termes plus simple et (2) un résultat de canonicité.

La règle suivante réduit un éliminateur simplifié dont la branche est une sorte vers cette sorte elle-même. Elle permet de simplifier l'énoncé du lemme 2.3.1 ci-dessous.

$$\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].s, c, \mathrm{P}) \quad \triangleright^{\mathrm{A}}_{s-\mathsf{Simpl}} \quad s$$

Les deux règles suivantes permettent d'établir un résultat de canonicité (cf. corollaire 2.3.4 ci-dessous).

Nous ajoutons la règle $\triangleright^{\mathrm{A}}_{\iota^2_\sigma}$ déjà mentionnée ci-dessus :

$$\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{T}].f, (a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}, \mathrm{P}) \quad \triangleright^{\mathrm{A}}_{\iota^2_\sigma} \quad f\,[y/b]$$

Nous ajoutons une dernière règle qui joue pour $\triangleright^{\mathrm{A}}_{\iota^2_\sigma}$ le même rôle que $\triangleright^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$ pour $\triangleright^{\mathrm{A}}_{\iota^1_\sigma}$ ou $\triangleright^{\mathrm{A}}_{\exists-\mathsf{Simpl}}$ pour $\triangleright^{\mathrm{A}}_{\iota_\exists}$: elle permet de débloquer les cas où $\triangleright^{\mathrm{A}}_{\iota^2_\sigma}$ ne peut s'appliquer car l'objet éliminé est une paire implicite encoquillée dans un éliminateur simplifié :

$$\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) \quad \triangleright^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}} \quad \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, g, \mathrm{P}), d, \mathrm{P})$$

1.6. Classification des règles de réduction. Nous regroupons les règles en cinq catégories définies dans figure 26 ci-dessous. Cette classification est proche du découpage fait dans la présentation des règles dans les sous-sections précédentes. Notons que la règle $\triangleright^{\mathrm{A}}_{\iota^2_\sigma}$ est considérée comme une règle de ι-réduction. La règle $\triangleright^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}}$ est associée aux règles $\triangleright^{\mathrm{A}}_{\exists-\mathsf{Simpl}}$ et $\triangleright^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$.

Nous définissons également la réunion des règles de réduction à la racine.

5. Cela est donc une justification supplémentaire à l'établissement d'un algorithme d'inférence de type extrait.

6. L'imprédicativité des produits fait que la sorte typant le domaine peut-être plus grande (au sens de la cumulativité) que celle typant le codomaine et le produit. De même pour $\Sigma x:\mathrm{A}.[\mathrm{B}]$, la sorte typant le codomaine peut être plus grande que celle typant le domaine et la somme.

- βι-réduction $\rhd^{\mathrm{A}}_{\beta\iota}$

$$\begin{aligned}
(\lambda x{:}\mathrm{T}.\,\mathrm{M})\,\mathrm{N} &\rhd^{\mathrm{A}}_{\beta^e} \mathrm{M}\,[x/\mathrm{N}]\\
(\lambda[x{:}\mathrm{T}].\,\mathrm{M})\,[\mathrm{N}] &\rhd^{\mathrm{A}}_{\beta^i} \mathrm{M}\,[x/\mathrm{N}]\\
\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{T}).(y{:}\mathrm{U}).f,(a,b)_{\Sigma x:\mathrm{A}.\,\mathrm{B}},\mathrm{P}) &\rhd^{\mathrm{A}}_{\iota_\Sigma} f\,[x/a]\,[y/b]\\
\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{T}].(y{:}\mathrm{U}).f,([a],b)_{\Sigma[x:\mathrm{A}].\,\mathrm{B}},\mathrm{P}) &\rhd^{\mathrm{A}}_{\iota_\exists} f\,[x/a]\,[y/b]\\
\pi_1((a,[b])_{\Sigma x:\mathrm{A}.\,[\mathrm{B}]}) &\rhd^{\mathrm{A}}_{\iota^1_\sigma} a\\
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{T}].f,(a,[b])_{\Sigma x:\mathrm{A}.\,[\mathrm{B}]},\mathrm{P}) &\rhd^{\mathrm{A}}_{\iota^2_\sigma} f\,[y/b]
\end{aligned}$$

- Éliminateur simplifié et constructeurs $\rhd^{\mathrm{A}}_{\mathsf{Constr-Simpl}}$

$$\begin{aligned}
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\lambda z{:}\mathrm{T}.\,\mathrm{M},c,\Pi z:\mathrm{T_0}.\,\mathrm{U_0}) &\rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}} \lambda z{:}\mathrm{T_0}.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\mathrm{M},c,\mathrm{U_0})\\
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\lambda[z{:}\mathrm{T}].\,\mathrm{M},c,\Pi[z{:}\mathrm{T_0}].\,\mathrm{U_0}) &\rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}} \lambda[z{:}\mathrm{T_0}].\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\mathrm{M},c,\mathrm{U_0})
\end{aligned}$$

$$\frac{a_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U_0}].a,c,\mathrm{A_0}) \qquad b_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U_0}].b,c,\mathrm{B_0}\,[z/a_0])}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U_0}].(a,b)_{\Sigma z:\mathrm{A}.\,\mathrm{B}},c,\Sigma z:\mathrm{A_0}.\,\mathrm{B_0}) \rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}} (a_0,b_0)_{\Sigma z:\mathrm{A_0}.\,\mathrm{B_0}}}$$

- Éliminateur simplifié et autres éliminateurs $\rhd^{\mathrm{A}}_{\mathsf{Elim-Simpl}}$

$$\frac{c = \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g,d,\mathrm{U})}{\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,c,\mathrm{P}) \rhd^{\mathrm{A}}_{\exists-\mathsf{Simpl}} \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,g,\mathrm{P}),d,\mathrm{P}\,c)}$$

$$\begin{aligned}
\pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,c,\Sigma x:\mathrm{T}.\,[\mathrm{U}])) &\rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}} \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f),c,\mathrm{T})\\
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,\mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g,d,\mathrm{U}),\mathrm{P}) &\rhd^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}} \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f,g,\mathrm{P}),d,\mathrm{P})
\end{aligned}$$

- Éliminateur simplifié et constructeurs de type $\rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}}$

$$\frac{\begin{cases}\Gamma^*;y:\mathrm{B}^*\vdash \mathrm{T}\Uparrow^{\mathrm{I}}\mathrm{S}'_{\mathrm{T}}\\ \Gamma^*;y:\mathrm{B}^*;z:\mathrm{T}^*\vdash \mathrm{U}\Uparrow^{\mathrm{I}}\mathrm{S}'_{\mathrm{U}}\end{cases} \qquad \begin{cases}\mathrm{T_0}=\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{T},c,\mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{T}}))\\ \mathrm{U_0}=\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{U},c,\mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{U}}))\end{cases}}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\square z:\mathrm{T}.\,\mathrm{U},c,\mathrm{P}) \rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}} \square z:\mathrm{T_0}.\,\mathrm{U_0}}$$

- Éliminateur simplifié et sortes $\rhd^{\mathrm{A}}_{s-\mathsf{Simpl}}$

$$\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].s,c,\mathrm{P}) \quad \rhd^{\mathrm{A}}_{s-\mathsf{Simpl}} \quad s$$

FIGURE 26. Récapitulatif des règles de réduction dans AICC_Σ

1.6.1. DÉFINITION. Pour tout contexte Γ de AICC_Σ, nous notons $\rhd^{\mathrm{A}}_{\Gamma}$ la réunion des relations $\rhd^{\mathrm{A}}_{\beta\iota}$, $\rhd^{\mathrm{A}}_{\mathsf{Elim-Simpl}}$,$\rhd^{\mathrm{A}}_{\mathsf{Constr-Simpl}}$, $\rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}}$ et $\rhd^{\mathrm{A}}_{s-\mathsf{Simpl}}$.

1.6.2. REMARQUE. Comme nous l'avons expliqué, nous cherchons à résoudre deux problèmes

(1) simulation des réductions de AICC_Σ : transformer R tel que $\mathrm{R}^* \twoheadrightarrow_{\beta\iota} \square' x:\mathrm{T}'.\,\mathrm{U}'$ en $\mathrm{R_0}$ tel que $\mathrm{R_0^*} = \square' x:\mathrm{T}'_0.\,\mathrm{U}'_0$;

(2) obtention d'un constructeur de type : transformer $\mathrm{R_0}$ en $\square x:\mathrm{T_0}.\,\mathrm{U_0}$ avec $\mathrm{R_0^*} = (\square x:\mathrm{T_0}.\,\mathrm{U_0})^* = \square' x:\mathrm{T}'_0.\,\mathrm{U}'_0$.

Nous pourrions croire que ces deux problèmes sont indépendants. Ainsi nous pourrions être tentés de vouloir séparer la règle $\rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}}$ des autres en l'appliquant dans un deuxième temps. Cela n'est en fait pas possible car les règles de réduction sont interdépendantes. Intuitivement, cela est

dû aux restrictions sur les annotations de type P dans les règles $\triangleright^{\mathrm{A}}_{\lambda-\mathsf{Simpl}}$, $\triangleright^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}}$, $\triangleright^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$ et $\triangleright^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$.

1.7. Réduction dans un sous-terme. Le mécanisme de réduction a pour but d'obtenir des constructeurs de type, nous n'allons donc pas réduire dans tous les sous-termes.

1.7.1. DÉFINITION. Soit Γ un contexte annoté. Nous notons $\to^h_\Gamma$ la relation de réduction définie par les règles présentées à la figure 27. Nous notons $\twoheadrightarrow^h_\Gamma$ la clôture réflexive et transitive de $\to^h_\Gamma$.

$$\frac{\mathrm{M}_0 \triangleright^{\mathrm{A}}_\Gamma \mathrm{M}_1}{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}\ (\textsc{HRed-Base})$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathrm{M}_0\,\mathrm{N} \to^h_\Gamma \mathrm{M}_1\,\mathrm{N}}\ (\textsc{HRed-E-App})$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathrm{M}_0\,[\mathrm{N}] \to^h_\Gamma \mathrm{M}_1\,[\mathrm{N}]}\ (\textsc{HRed-I-App})$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, \mathrm{M}_0, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, \mathrm{M}_1, \mathrm{P})}\ (\textsc{HRed-}\Sigma\textsc{-Elim})$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}_{\mathrm{I}_\mathrm{L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathrm{M}_0, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}_{\mathrm{I}_\mathrm{L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathrm{M}_1, \mathrm{P})}\ (\textsc{HRed-}\exists\textsc{-Elim})$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\pi_1(\mathrm{M}_0) \to^h_\Gamma \pi_1(\mathrm{M}_1)}\ (\textsc{HRed-}\pi_1)$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y{:}\mathrm{B}].f, \mathrm{M}_0, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y{:}\mathrm{B}].f, \mathrm{M}_1, \mathrm{P})}\ (\textsc{HRed-Simpl-Object})$$

$$\frac{\mathrm{M}_0 \to^h_{\Gamma;y:\mathrm{B}} \mathrm{M}_1}{\mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y{:}\mathrm{B}].\mathrm{M}_0, c, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y{:}\mathrm{B}].\mathrm{M}_1, c, \mathrm{P})}\ (\textsc{HRed-Simpl-Branch})$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y{:}\mathrm{B}].f, c, \mathrm{M}_0) \to^h_\Gamma \mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y{:}\mathrm{B}].f, c, \mathrm{M}_1)}\ (\textsc{HRed-Simpl-Type})$$

FIGURE 27. Réduction dans un contexte

Dans toutes les règles de réduction présentées ci-dessus, le terme de gauche est un éliminateur dont certains sous-termes ont une forme bien définie.[7] Nous allons seulement réduire dans ces sous-termes afin éventuellement de déclencher une réduction à la racine. Ainsi pour les applications MN ou M[N], nous allons réduire dans M car $\triangleright^{\mathrm{A}}_{\beta^e}$ ou $\triangleright^{\mathrm{A}}_{\beta^i}$ peuvent s'appliquer si M est une abstraction explicite ou implicite. Pour les éliminateurs de somme dépendante explicite $\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P})$, nous allons réduire c car $\triangleright^{\mathrm{A}}_{\iota_\Sigma}$ peut s'appliquer si c est une paire dépendante explicite. Pour les éliminateurs de somme dépendante implicite gauche $\mathsf{Elim}_{\mathrm{I}_\mathrm{L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P})$ et la première projection $\pi_1(c)$, nous réduisons c également car $\triangleright^{\mathrm{A}}_{\iota_\exists}$ ou $\triangleright^{\mathrm{A}}_{\iota^1_\sigma}$ peuvent s'appliquer si c est une paire dépendante implicite (gauche ou droite) et $\triangleright^{\mathrm{A}}_{\exists-\mathsf{Simpl}}$ ou $\triangleright^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$ peuvent s'appliquer si c

7. Comme nous l'avons vu, c'est aussi le cas dans ICC_Σ pour les règles β et ι mais pas pour la règle η.

est un éliminateur simplifié. Enfin, pour l'éliminateur simplifié $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P})$, nous réduisons dans c, f et P car selon la forme de c, les règles $\rhd^{\mathrm{A}}_{\iota^2_\sigma}$ ou $\rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$ peuvent s'appliquer ; selon les formes de f et P, les règles $\rhd^{\mathrm{A}}_{\mathsf{Constr}-\mathsf{Simpl}}$, $\rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$ $\rhd^{\mathrm{A}}_{\Gamma,\,\square-\mathsf{Simpl}}$ et $\rhd^{\mathrm{A}}_{s-\mathsf{Simpl}}$ peuvent s'appliquer.

Nous réduisons donc *faiblement* (pas sous les lieurs sauf pour f dans $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P})$) et en position de tête (si la *tête* d'un terme est définie comme toute position qui peut déclencher une réduction).

1.7.2. REMARQUE (Confluence). Pour un contexte donné, $\rightarrow^h_\Gamma$ n'est pas confluente : la règle $\rhd^{\mathrm{A}}_{\iota^2_\sigma}$ introduit des paires critiques qui ne se réduisent pas vers un élément commun.

Ainsi nous avons l'exemple suivant d'un terme pouvant être réduit par $\rhd^{\mathrm{A}}_{\iota^2_\sigma}$ ou $\rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}}$:

$$\begin{array}{lcl}
\mathrm{M} & = & \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathsf{Type}_0].\lambda z{:}\mathsf{Prop}.\mathsf{Prop}, ([\mathsf{Prop}], \mathsf{Prop})_{\Sigma[x:\mathsf{Type}_0].\,\mathsf{Type}_0}, \mathsf{Type}_4 \to \mathsf{Type}_4) \\
\mathrm{M} & \rhd^{\mathrm{A}}_{\iota^2_\sigma} & \lambda z{:}\mathsf{Prop}.\mathsf{Prop} \\
\mathrm{M} & \rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}} & \lambda z{:}\mathsf{Type}_4.\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathsf{Type}_0].\mathsf{Prop}, ([\mathsf{Prop}], \mathsf{Prop})_{\Sigma[x:\mathsf{Type}_0].\,\mathsf{Type}_0}, \mathsf{Type}_4)
\end{array}$$

Les deux termes réduits sont distincts et irréductibles pour $\rightarrow^h_\Gamma$.

Néanmoins, si nous considérons une confluence à extraction près [8], il est raisonnable de penser que $\rightarrow^h_\Gamma$ est confluente : les paires critiquent concernent seulement des règles telles que si $\mathrm{M} \rhd^{\mathrm{A}} \mathrm{N}$ alors $\mathrm{M}^* = \mathrm{N}^*$.

2. Complétude des règles de réduction

Dans la section précédente, nous avons introduit des règles de réduction en justifiant informellement leur *nécessité* par des exemples de réductions souhaitables rendues possibles par l'ajout de la nouvelle règle. Dans cette section, nous allons montrer que nous avons ajouté *suffisamment* de règles et que toutes les réductions nécessaires peuvent avoir lieu.

Pour cela, nous établissons un résultat de classification des termes bien typés qui affine les résultats de la sous-section 2.2 du chapitre 6 (cf. page 118). Celui-ci nous indique que tout terme bien typé est soit sous forme canonique (constructeur de type ou sorte typé par un type comparable à une sorte, constructeur typé par un type comparable à un constructeur de type), soit une variable, soit un éliminateur.

Dans cette section, nous nous penchons sur le cas des éliminateurs. Nous définissons ainsi deux familles d'éliminateurs, les termes neutres et les paires implicites projetées, puis nous montrons qu'un éliminateur bien typé dans Γ est un terme neutre ou une paire implicite projetée ou réductible pour $\rightarrow^h_\Gamma$.

Nous en déduisons que si $\Gamma \vdash \mathrm{R} : s$ et $\mathrm{R}^* \twoheadrightarrow_{\beta\iota} \square' x : \mathrm{T}'.\mathrm{U}'$ alors R est soit réductible pour $\rightarrow^h_\Gamma$ soit un constructeur de type de AICC_Σ (de nature analogue à celle de $\square' x : \mathrm{T}'.\mathrm{U}'$), ce qui montre que les règles ajoutées sont suffisantes pour l'obtention d'un constructeur de type.

2.1. Termes neutres. Nous définissions des termes neutres dans ICC_Σ et AICC_Σ. Intuitivement, les termes neutres sont des éliminateurs contenant des variables bloquant toute réduction qui pourrait aboutir à transformer le terme en redex. Ainsi les termes neutres vont rester des termes neutres de même nature en se réduisant. En particulier, un terme neutre ne se réduit pas vers un constructeur de type.

8. si $\mathrm{M} \twoheadrightarrow^h_\Gamma \mathrm{M}_1$ et $\mathrm{M} \twoheadrightarrow^h_\Gamma \mathrm{M}_2$ alors il existe $\mathrm{N}_1, \mathrm{N}_2$ tels que $\mathrm{N}_1^* = \mathrm{N}_2^*$, $\mathrm{M}_1 \twoheadrightarrow^h_\Gamma \mathrm{N}_1$ et $\mathrm{M}_2 \twoheadrightarrow^h_\Gamma \mathrm{N}_2$

2.1.1. *Termes neutres dans ICC$_\Sigma$.*

2.1.1. DÉFINITION (Termes neutres). Les *termes neutres* de ICC$_\Sigma$ sont définis par la grammaire suivante :

$$\begin{array}{rcl} \mathsf{Neu} & ::= & x \\ & & \mid \mathsf{Neu}\,\mathrm{N}' \\ & & \mid \mathsf{Elim}_\Sigma(xy.f', \mathsf{Neu}) \\ & & \mid \mathsf{Elim}_\exists(y.f', \mathsf{Neu}) \end{array}$$

où x, y sont des variables, N', f' des termes quelconques de ICC$_\Sigma$. L'ensemble des termes neutres de ICC$_\Sigma$ est noté $\Lambda^{\mathsf{Neu}}_{\mathrm{ICC}_\Sigma}$.

2.1.2. Lemme (Termes neutres et calcul)**.** *Soit* $\mathrm{M}'_{\mathsf{Neu}}$ *un terme neutre de ICC*$_\Sigma$.

(1) $\mathrm{M}'_{\mathsf{Neu}}$ *n'est pas réductible à la racine.*

(2) si $\mathrm{M}'_{\mathsf{Neu}} \to_{\beta\iota\eta} \mathrm{M}'$ *alors* M' *est neutre et de même nature que* $\mathrm{M}'_{\mathsf{Neu}}$.

(3) si $\mathrm{M}'_{\mathsf{Neu}} \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}'$ *alors* M' *est neutre et de même nature que* $\mathrm{M}'_{\mathsf{Neu}}$.

(4) si $\mathrm{M}'_{\mathsf{Neu}} \leqslant \mathrm{M}'$ *ou* $\mathrm{M}' \leqslant \mathrm{M}'_{\mathsf{Neu}}$ *alors* $\mathrm{M}' = \mathrm{M}'_{\mathsf{Neu}}$.

(5) si $\mathrm{M}'_{\mathsf{Neu}} \preceq \mathrm{M}'$ *ou* $\mathrm{M}' \preceq \mathrm{M}'_{\mathsf{Neu}}$ *alors* M' *est un éliminateur ou une abstraction ou* $\mathrm{M}' = \mathrm{M}'_{\mathsf{Neu}} \in$ ***Var***.

DÉMONSTRATION.

(1) Montrons que $\mathrm{M}'_{\mathsf{Neu}}$ n'est pas un βιη-redex :
- Montrons que $\mathrm{M}'_{\mathsf{Neu}}$ n'est pas un β-redex :
  - si $\mathrm{M}'_{\mathsf{Neu}}$ est un β-redex alors il existe $x, \mathrm{M}', \mathrm{N}'$ tels que $\mathrm{M}'_{\mathsf{Neu}} = (\lambda x.\mathrm{M}')\,\mathrm{N}'$
  - par définition des termes neutres, $\lambda x.\mathrm{M}' \in \Lambda^{\mathsf{Neu}}_{\mathrm{ICC}_\Sigma}$ ce qui est impossible car les termes neutres sont des éliminateurs.
- De même, $\mathrm{M}'_{\mathsf{Neu}}$ n'est pas un ι-redex : sinon $\mathrm{M}'_{\mathsf{Neu}}$ serait de la forme $\mathsf{Elim}_\Sigma(xy.f', (a', b'))$ ou $\mathsf{Elim}_\exists(y.f', (\diamond, b'))$ avec les constructeurs (a', b') ou $(\diamond, b')$ des termes neutres, ce qui est impossible.
- Enfin $\mathrm{M}'_{\mathsf{Neu}}$ n'est pas un η-redex car un terme neutre ne peut être une abstraction.

(2) Par induction sur la structure de $\mathrm{M}'_{\mathsf{Neu}}$ en utilisant le point (1) qui nous indique que la réduction ne peut avoir lieu en tête.

(3) Découle du point (2) par une récurrence sur le nombre de pas de réductions.

(4) Par inversion de la cumulativité restreinte (lemme 2.3.5 page 15), $\mathrm{M}' = \mathrm{M}'_{\mathsf{Neu}}$ ou M' et $\mathrm{M}'_{\mathsf{Neu}}$ sont des constructeurs de type de même nature. $\mathrm{M}'_{\mathsf{Neu}}$ ne pouvant être un constructeur de type, nous déduisons $\mathrm{M}' = \mathrm{M}'_{\mathsf{Neu}}$.

(5) Montrons que M' est un éliminateur ou une abstraction ou alors il existe une variable x telle que $\mathrm{M}' = \mathrm{M}'_{\mathsf{Neu}} = x$.
- D'après le point (2) du lemme 2.3.10 page 16, il existe des termes $\mathrm{M}'_0, \mathrm{N}'_0$ tels que $\mathrm{M}' \twoheadrightarrow_{\beta\iota\eta} \mathrm{M}'_0$, $\mathrm{M}'_{\mathsf{Neu}} \twoheadrightarrow_{\beta\iota\eta} \mathrm{N}'_0$ et $\mathrm{M}'_0 \leqslant \mathrm{N}'_0$ ou $\mathrm{N}'_0 \leqslant \mathrm{M}'_0$.
- D'après le point (3) de ce lemme, N'_0 est un terme neutre de même nature que $\mathrm{M}'_{\mathsf{Neu}}$.
- D'après le point (4) de ce lemme, nous en déduisons que $\mathrm{M}'_0 = \mathrm{N}'_0$.
- D'après le lemme 2.2.4 page 12 appliqué à M' et M'_0, M' est un éliminateur ou une abstraction ou bien M' et M'_0 ont même structure.
- Si M' n'est pas un éliminateur ou une abstraction, montrons que $\mathrm{M}' = \mathrm{M}'_{\mathsf{Neu}} \in$ **Var**.

- M' et M'_0 sont de même nature. M'_0 est neutre, donc un éliminateur ou une variable. M' n'est pas un éliminateur, donc M' et M'_0 sont des variables.
- M' est donc irréductible d'où $M' = M'_0$.
- M'_{Neu} et N'_0 sont de même nature et $M'_0 = N'_0$ donc M'_{Neu} est une variable.
- M'_{Neu} est donc irréductible d'où $M'_{\mathsf{Neu}} = N'_0$.
- Nous en déduisons $M' = M'_{\mathsf{Neu}} \in \mathbf{Var}$.

□

2.1.3. REMARQUE. Nous pouvons montrer que tout terme neutre est irréductible pour la βι-réduction de tête faible, définie par les quatre règles suivantes :

$$\frac{M'_0 \rhd_{\beta\iota} M'_1}{M'_0 \to^h_{\beta\iota} M'_1} \qquad \frac{M'_0 \to^h_{\beta\iota} M'_1}{\mathsf{Elim}_\Sigma(xy.f', M'_0) \to^h_{\beta\iota} \mathsf{Elim}_\Sigma(xy.f', M'_1)}$$

$$\frac{M'_0 \to^h_{\beta\iota} M'_1}{M'_0\,N \to^h_{\beta\iota} M'_1\,N} \qquad \frac{M'_0 \to^h_{\beta\iota} M'_1}{\mathsf{Elim}_\exists(y.f', M'_0) \to^h_{\beta\iota} \mathsf{Elim}_\exists(y.f', M'_1)}$$

La βι-réduction de tête faible, est une restriction de la relation $\to_{\beta\iota}$ où nous ne réduisons pas sous les lieurs (réduction *faible*) mais seulement dans les sous-termes en position de tête (réduction de *tête*).

2.1.2. *Termes neutres de* AICC_Σ. Les termes neutres de AICC_Σ sont l'équivalent dans le monde annoté des termes neutres de ICC_Σ. Ainsi les termes neutres de AICC_Σ vont rester des termes neutres en se réduisant et vont s'extraire vers des termes neutres de ICC_Σ.

2.1.4. DÉFINITION (Termes neutres). Les *termes neutres* de AICC_Σ sont définis par la grammaire suivante :

$$\begin{aligned}
\mathsf{Neu}_A \quad ::= \quad & x \\
& \mid \mathsf{Neu}_A\,N \mid \mathsf{Neu}_A\,[N] \\
& \mid \mathsf{Elim}_E((x{:}A).(y{:}B).f, \mathsf{Neu}_A, P) \\
& \mid \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, \mathsf{Neu}_A, P) \\
& \mid \pi_1(\mathsf{Neu}_A) \mid \mathsf{Elim}^s_{I_R}([y{:}B].\mathsf{Neu}_A, \mathsf{Neu}_A, P)
\end{aligned}$$

où x, y sont des variables, N, f, P des termes quelconques de AICC_Σ. L'ensemble des termes neutres de AICC_Σ est noté $\Lambda^{\mathsf{Neu}}_{\mathrm{AICC}}$.

2.1.5. Lemme (Termes neutres et extraction)**.** *Si* M_{Neu_A} *est un terme neutre de* AICC_Σ*, alors* $M^*_{\mathsf{Neu}_A}$ *est un terme neutre de* ICC_Σ.

DÉMONSTRATION. Immédiat par induction sur la structure de M_{Neu_A}. □

2.1.6. Lemme (Termes neutres et réduction)**.** *Si* M_{Neu_A} *est un terme neutre de* AICC_Σ *et si* $M_{\mathsf{Neu}_A} \to^h_\Gamma N$ *alors* N *est également un terme neutre de* AICC_Σ.

DÉMONSTRATION. Par induction sur la structure de M_{Neu_A}.

- Tous les cas sauf celui de l'éliminateur simplifié sont immédiats.
- Si $M_{\mathsf{Neu}_A} = \mathsf{Elim}^s_{I_R}([y{:}B].f_N, c, P) \to^h_\Gamma N$ avec f_N terme neutre de AICC_Σ, alors, par inversion de $\to^h_\Gamma$ trois cas peuvent se produire.
  (1) Montrons que la réduction n'a pas lieu à la racine : f_N terme neutre, ne peut être ni un constructeur, ni un constructeur de type, ni une sorte, donc $\mathsf{Elim}^s_{I_R}([y{:}B].f_N, c, P)$ ne peut être un redex pour aucune des règles de réduction présentées ci-dessus.

(2) La réduction a lieu à la racine (règle (HRED-BASE)) : f_{N}, terme neutre, ne peut être ni un constructeur, ni un constructeur de type, ni une sorte, donc $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f_{\mathrm{N}}, c, \mathrm{P})$ ne peut être un redex que pour la règle $\rhd^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}}$: nous pouvons donc poser $c = \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U})$ et avons

$$\mathrm{M}_{\mathsf{Neu}_{\mathrm{A}}} = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f_{\mathrm{N}}, \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) \rhd^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}} \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f_{\mathrm{N}}, g, \mathrm{P}), d, \mathrm{P}) = \mathrm{N}$$

Montrons que N est un terme neutre.

- f_{N} est un terme neutre donc $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f_{\mathrm{N}}, g, \mathrm{P})$ est un terme neutre ;
- N est un terme neutre car $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f_{\mathrm{N}}, g, \mathrm{P})$ est un terme neutre.

(3) La réduction a lieu dans f_{N} (règle (HRED-Simpl-BRANCH)) : par induction f_{N} se réduit vers un terme neutre, donc N est un terme neutre également.

(4) La réduction a lieu dans P (règle (HRED-Simpl-TYPE)) : la branche de N est f_{N}, terme neutre par hypothèse, donc N est un terme neutre.

□

2.1.7. REMARQUE (Termes neutres et termes clos). Nous montrons facilement par induction structurelle qu'un terme neutre de ICC_{Σ} ou AICC_{Σ} n'est pas clos.

2.2. Paires implicites projetées. Les paires implicites projetées sont des paires implicites encoquillées (au niveau de la branche) dans un ou plusieurs éliminateurs simplifiés.

Nous définissons cette famille d'éliminateurs car elle caractérise, comme nous le verrons dans la preuve du lemme 2.3.1 les éliminateurs de AICC_{Σ} bien typés qui ne sont ni des termes neutres, ni réductibles pour $\rightarrow^{h}_{\Gamma}$ (avec Γ un contexte de typage pour les éliminateurs considérés).

2.2.1. DÉFINITION (Paires implicites projetées).

- Les *paires implicites gauches projetées* sont définies par la grammaire suivante :

$$\mathsf{IpairL}_{\mathsf{E}} \quad ::= \quad \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}}, c, \mathrm{P}) \mid \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathsf{IpairL}_{\mathsf{E}}, c, \mathrm{P})$$

- Les *paires implicites droites projetées* sont définies par la grammaire suivante :

$$\mathsf{IpairR}_{\mathsf{E}} \quad ::= \quad \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].(a, [b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}, c, \mathrm{P}) \mid \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathsf{IpairR}_{\mathsf{E}}, c, \mathrm{P})$$

Contrairement aux termes neutres, les paires implicites projetées peuvent s'extraire vers des constructeurs de type. $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].(\mathsf{Prop} \rightarrow \mathsf{Prop}, [y])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}, x, \mathrm{P})$ en est un exemple.

En fait nous pouvons être plus précis et montrer que termes neutres et paires implicites projetées sont deux familles disjointes.

2.2.2. Lemme. *Une paire implicite projetée n'est pas un terme neutre.*

DÉMONSTRATION. Traitons le cas de la paire implicite droite projetée. Le cas de la paire implicite gauche projetée se résout exactement de la même manière. Si $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P})$ est une paire implicite droite, montrons par induction sur sa structure que ce n'est pas un terme neutre :

- si $f = ([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}}$ alors f n'est pas un terme neutre donc $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P})$ non plus ;
- sinon, f est une paire implicite droite : par hypothèse d'induction, ce n'est pas un terme neutre, donc $\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P})$ non plus.

□

Un petit résultat utile pour la preuve du lemme 2.3.1 ci-dessous.

2.2.3. Lemme. *Si* $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *avec* T *un constructeur de type ou une sorte alors*

- *si* M *est une paire implicite droite projetée,* T *est une somme implicite droite ;*
- *si* M *est une paire implicite gauche projetée,* T *est une somme implicite gauche.*

DÉMONSTRATION. Montrons le résultat pour la paire implicite gauche projetée. Celui de la paire implicite droite projetée se traite de manière analogue. Nous procédons par induction sur la structure de M.

- Si $\mathrm{M} = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}}, c, \mathrm{P})$:
  - par inversion de $\Gamma \vdash \mathrm{M} : \mathrm{T}$, nous avons $\Gamma; y : \mathrm{B} \vdash ([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}} : \mathrm{P}$ et $\mathrm{P}^* \preceq \mathrm{T}^*$;
  - par inversion de $\Gamma; y : \mathrm{B} \vdash ([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}} : \mathrm{P}$, $(\Sigma[x{:}\mathrm{A}].\mathrm{B})^* \preceq \mathrm{P}^*$;
  - par transitivité, $(\Sigma[x{:}\mathrm{A}].\mathrm{B})^* \preceq \mathrm{T}^*$;
  - puisque T est un constructeur de type ou une sorte, nous déduisons que T est bien une somme dépendante implicite gauche.
- Si $\mathrm{M} = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P})$, avec f une paire implicite gauche projetée :
  - par inversion de $\Gamma \vdash \mathrm{M} : \mathrm{T}$, nous avons $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$ et $\mathrm{P}^* \preceq \mathrm{T}^*$;
  - par cumulativité, nous déduisons $\Gamma; y : \mathrm{B} \vdash f : \mathrm{T}$;
  - nous concluons alors par hypothèse d'induction sur f.

□

2.3. Classification des éliminateurs bien typés. Nous présentons dans cette sous-section le résultat qui justifie le choix des règles de réduction ajoutées dans AICC_Σ ainsi que les définitions des termes neutres et des paires implicites projetées.

Nous allons ainsi montrer que tout éliminateur annoté typé dans un contexte Γ est soit un terme neutre, soit une paire implicite projetée, soit réductible pour $\rightarrow^{h}_{\Gamma}$. Notons que les trois catégories d'éliminateurs ne sont pas mutuellement exclusives : un terme neutre ou une paire implicite projetée peuvent se réduire.

Cette classification nous permettra d'utiliser la réduction $\rightarrow^{h}_{\Gamma}$ dans AICC_Σ pour obtenir des constructeurs de type : nous montrons qu'un éliminateur typé par une sorte dont l'extraction se réduit vers un constructeur de type n'est ni un terme neutre (dont l'extraction ne peut se réduire vers un constructeur de type) ni une paire implicite projetée (qui ne peut être typée par une sorte).

2.3.1. Lemme (Classification des éliminateurs bien typés)**.** *Si* $\Gamma \vdash \mathrm{M} : \mathrm{R}$ *et si* M *est un éliminateur alors* M *est un terme neutre ou réductible pour* $\rightarrow^{h}_{\Gamma}$ *ou une paire implicite projetée.*

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma \vdash \mathrm{M} : \mathrm{R}$. M étant un éliminateur nous considérons uniquement les règles d'élimination et la règle (Cum$_\mathrm{A}$). Les cas les plus compliqués sont la première projection et l'éliminateur simplifié.

- (Cum$_\mathrm{A}$) : immédiat par hypothèse d'induction.
- (E-App) et (I-App) sont similaires. Traitons (I-App) : si

$$\frac{\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}].\mathrm{U} \qquad \Gamma \vdash \mathrm{N} : \mathrm{T}}{\Gamma \vdash \mathrm{M}\,[\mathrm{N}] : \mathrm{U}\,[x/\mathrm{N}]}$$

  montrons que $\mathrm{M}\,[\mathrm{N}]$ est un terme neutre ou réductible pour $\rightarrow^{h}_{\Gamma}$.
  - D'après le corollaire 2.2.4 page 119 appliqué à $\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}].\mathrm{U}$, trois cas sont possibles :
    (1) M est un constructeur associé à $\Pi[x{:}\mathrm{T}].\mathrm{U}$, donc une abstraction implicite,
    (2) M est une variable,

(3) M est un éliminateur.

- Si M est une abstraction implicite $\lambda[x{:}\mathrm{T}]\,.\,\mathrm{M}_0$ alors $\mathrm{M}\,[\mathrm{N}] = (\lambda[x{:}\mathrm{T}]\,.\,\mathrm{M}_0)\,[\mathrm{N}] \rhd^{\mathrm{A}}_{\beta^i} \mathrm{M}_0\,[x/\mathrm{N}]$.
- Si M est une variable alors M est un terme neutre donc $\mathrm{M}\,[\mathrm{N}]$ également.
- Si M est un éliminateur, par hypothèse d'induction sur $\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}$, un des trois cas suivants est vrai :
  — M est un terme neutre, d'où $\mathrm{M}\,[\mathrm{N}]$ neutre également ;
  — M est réductible pour $\rightarrow^h_\Gamma$, alors $\mathrm{M}\,[\mathrm{N}]$ l'est également (règle (HRED-I-APP)) ;
  — M est une paire implicite projetée : impossible car alors $\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}$ contredit le lemme 2.2.3.

- $(\Sigma_{\mathrm{A}}\text{-E})$: Si

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma x : \mathrm{A}\,.\,\mathrm{B} \to s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}\,.\,\mathrm{B} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x,y)_{\Sigma x:\mathrm{A}\,.\,\mathrm{B}}}{\Gamma \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}$$

montrons que $\mathsf{Elim}_{\mathrm{E}}((x:\mathrm{A}).(y:\mathrm{B}).f, c, \mathrm{P})$ est un terme neutre ou réductible pour $\rightarrow^h_\Gamma$. La preuve est similaire à celle du cas précédent. D'après le corollaire 2.2.4 page 119 appliqué à $\Gamma \vdash c : \Sigma x : \mathrm{A}\,.\,\mathrm{B}$, trois cas sont possibles :

(1) c est un constructeur associé à $\Sigma x : \mathrm{A}\,.\,\mathrm{B}$, d'où $c = (a,b)_{\Sigma x:\mathrm{A}_0\,.\,\mathrm{B}_0}$ et $\mathsf{Elim}_{\mathrm{E}}((x:\mathrm{A}).(y:\mathrm{B}).f, c, \mathrm{P}) = \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, (a,b)_{\Sigma x:\mathrm{A}_0\,.\,\mathrm{B}_0}, \mathrm{P}) \rhd^{\mathrm{A}}_{\iota_\Sigma} f\,[x/a]\,[y/b]$;

(2) c est une variable alors $\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P})$ est un terme neutre ;

(3) c est un éliminateur : nous concluons, exactement comme pour l'application implicite, à partir de l'hypothèse d'induction.

- $(\Sigma_{\exists}-\mathrm{E})$: Si

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s \qquad \Gamma \vdash c : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x],y)_{\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}} \qquad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}$$

montrons que $\mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f, c, \mathrm{P})$ est un terme neutre ou réductible pour $\rightarrow^h_\Gamma$.

La preuve est très similaire à celles des deux cas précédents. Nous appliquons le corollaire 2.2.4 page 119 à $\Gamma \vdash c : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$, puis nous raisonnons également par induction pour traiter le cas des éliminateurs.

Une différence importante, toutefois, est que c peut être une paire implicite projetée gauche (puisque $\Gamma \vdash c : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$ ne contredit pas le lemme 2.2.3). Dans ce cas, c est un éliminateur simplifié et $\mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f, c, \mathrm{P})$ est un redex pour la règle $\rhd^{\mathrm{A}}_{\exists-\mathsf{Simpl}}$.

- $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-1)$ Si

$$\frac{\Gamma \vdash c : \Sigma x : \mathrm{A}\,.\,[\mathrm{B}]}{\Gamma \vdash \pi_1(c) : \mathrm{A}}$$

montrons que $\pi_1(c)$ est neutre ou réductible pour $\rightarrow^h_\Gamma$.

Nous procédons comme précédemment en appliquant le corollaire 2.2.4 page 119 à $\Gamma \vdash c : \Sigma x : \mathrm{A}\,.\,[\mathrm{B}]$. Pour le cas où c est un éliminateur, nous utilisons l'hypothèse d'induction et le seul cas non-immédiat est celui où c est une paire implicite projetée.

Posons $c = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, d, \mathrm{P})$. Contrairement au cas précédent, $\pi_1(c) = \pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, d, \mathrm{P}))$ n'est pas nécessairement un redex pour $\rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$ car P n'est pas a priori une somme implicite droite.

Montrons que $\pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, d, \mathrm{P}))$ est réductible pour $\rightarrow^h_\Gamma$.

- Montrons que P est une somme implicite droite ou un terme réductible pour $\rightarrow_\Gamma^h$.
  - — Par inversion du jugement $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f,d,\mathrm{P}) : \Sigma x : \mathrm{A}.[\mathrm{B}]$, nous déduisons :
    - — $\mathrm{P}^* \preceq (\Sigma x : \mathrm{A}.[\mathrm{B}])^*$
    - — il existe une sorte s telle qu'une dérivation de $\Gamma \vdash \mathrm{P} : s$ est incluse dans celle de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f,d,\mathrm{P}) : \Sigma x : \mathrm{A}.[\mathrm{B}]$.
  - — D'après le corollaire 2.2.5 page 120 appliqué à $\Gamma \vdash \mathrm{P} : s$, trois cas sont possibles :
    (1) P est un constructeur de type ou une sorte,
    (2) P est une variable,
    (3) P est un éliminateur.
  - — Si P est un constructeur de type ou une sorte, alors, d'après le point (1) du lemme 1.2.6 page 114 et puisque $\mathrm{P}^* \preceq (\Sigma x : \mathrm{A}.[\mathrm{B}])^*$, il existe $\mathrm{A}_1, \mathrm{B}_1$ tels que $\mathrm{P} = \Sigma x : \mathrm{A}_1.[\mathrm{B}_1]$.
  - — P n'est pas une variable car sinon $\mathrm{P}^* \preceq (\Sigma x : \mathrm{A}.[\mathrm{B}])^*$ contredit le lemme 2.3.15 page 18.
  - — Si P est un éliminateur alors, puisque $\Gamma \vdash \mathrm{P} : s$ est inclus dans la dérivation, nous pouvons appliquer par induction le lemme à $\Gamma \vdash \mathrm{P} : s$:
    (1) P n'est pas neutre car sinon P^* est neutre dans ICC_Σ (lemme 2.1.5) et $\mathrm{P}^* \preceq (\Sigma x : \mathrm{A}.[\mathrm{B}])^*$ contredit le point (5) du lemme 2.1.2 page 179.
    (2) P ne peut être une paire implicite projetée car alors $\Gamma \vdash \mathrm{P} : s$ contredirait le lemme 2.2.3.
    (3) P est donc réductible pour $\rightarrow_\Gamma^h$.
- Si $\mathrm{P} = \Sigma x : \mathrm{A}_1.[\mathrm{B}_1]$, alors $\pi_1(c) = \pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f,d,\Sigma[x{:}\mathrm{A}_1].\mathrm{B}_1)) \rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}} \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\pi_1(f),c,\mathrm{A}_1)$, d'où $\pi_1(c) \rightarrow_\Gamma^h \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\pi_1(f),c,\mathrm{A}_1)$ (règle (HRed-Base)).
- Si P est réductible pour $\rightarrow_\Gamma^h$ alors c est réductible (règle (HRed-Simpl-Type)) puis $\pi_1(c)$ également (règle (HRed-π_1)).

- ($\Sigma_{\mathrm{Sub}}-\mathrm{E}-2$) Si

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}] \qquad \Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P} \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (\mathrm{B}[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f,c,\mathrm{P}) : \mathrm{P}}$$

montrons que $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f,c,\mathrm{P})$ est un terme neutre ou réductible ou une paire implicite projetée.

- Montrons pour commencer trois résultats préliminaires : un pour c et deux pour P.
  (1) Montrons que c est
  - — un terme neutre
  - — ou une paire implicite droite
  - — ou une paire implicite droite projetée
  - — ou un terme réductible pour $\rightarrow_\Gamma^h$.

  D'après le corollaire 2.2.4 page 119 appliqué à $\Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}]$, trois cas sont possibles :
  (a) c est un constructeur associé à $\Sigma x : \mathrm{A}.[\mathrm{B}]$, donc d'après la définition 2.2.1 page 118, c est une paire implicite droite ;

(b) c est une variable : alors c est un terme neute neutre.

(c) c est un éliminateur : par induction sur $\Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}]$, trois cas sont possibles :

(i) c est un terme neutre ;

(ii) c est réductible pour $\rightarrow_\Gamma^h$;

(iii) c est une paire implicite projetée (droite ou gauche) :

— $\Sigma x : \mathrm{A}.[\mathrm{B}]$ n'est pas une somme implicite gauche, donc d'après le lemme 2.2.3 appliqué à $\Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}]$, c n'est pas une paire implicite projetée gauche ;

— nous en déduisons que c est bien une paire implicite projetée droite.

(2) Montrons que si $(\square x : \mathrm{T}.\mathrm{U})^* \preceq \mathrm{P}^*$ et si P n'est pas un éliminateur alors il existe $\mathrm{T}_0, \mathrm{U}_0$ tels que $\mathrm{P} = \square x : \mathrm{T}_0.\mathrm{U}_0$.

— Montrons que P est un constructeur de type ou une sorte.

— D'après le corollaire 2.2.5 page 120 appliqué à $\Gamma \vdash \mathrm{P} : s$, P est une variable, un éliminateur, un constructeur de type ou une sorte.

— P n'est pas un éliminateur par hypothèse.

— P n'est pas une variable car si $\mathrm{P} = x_0$ alors nous avons $x_0 \preceq \square^* x : \mathrm{T}^*.\mathrm{U}^*$ ce qui contredit le lemme 2.3.15 page 18.

— Nous concluons en appliquant le point (1) du lemme 1.2.6 page 114 à P.

(3) Montrons que si $(\square x : \mathrm{T}.\mathrm{U})^* \preceq \mathrm{P}^*$ et si P est un éliminateur, alors P est réductible pour $\rightarrow_\Gamma^h$.

— Par induction sur $\Gamma \vdash \mathrm{P} : s$, P est un terme neutre ou bien réductible pour $\rightarrow_\Gamma^h$ ou une paire implicite projetée.

— P n'est pas neutre car sinon P^* est neutre dans ICC_Σ (lemme 2.1.5) et $(\square x : \mathrm{A}.\mathrm{B})^* \preceq \mathrm{P}^*$ contredit le point (5) du lemme 2.1.2 page 179.

— P n'est pas une paire implicite projetée car sinon $\Gamma \vdash \mathrm{P} : s$ contredirait le lemme 2.2.3.

- Revenons à la preuve principale. D'après le lemme 2.2.3 page 119 appliqué à $\Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P}$ trois cas sont possibles :

(1) f est une variable,

(2) f est un éliminateur,

(3) il existe R sorte ou constructeur de type de AICC_Σ tel que f est un terme associé à R et $\mathrm{R}^* \preceq \mathrm{P}^*$.

Traitons chacun de ces cas.

(1) si f est une variable : d'après le résultat préliminaire sur c nous avons quatre possibilités :

— c est un terme neutre : f est neutre donc $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P})$ est neutre

— une paire implicite droite : nous pouvons appliquer $\triangleright^{\mathrm{A}}_{\iota^2_\sigma}$ à $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P})$;

— une paire implicite (droite) projetée : nous pouvons appliquer $\triangleright^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}}$ à $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P})$.

— un terme réductible pour $\rightarrow_\Gamma^h$: alors en appliquant (HRED-Simpl-OBJECT), $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P})$ est également réductible.

(2) Si f est un éliminateur alors par induction sur $\Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P}$, nous avons trois cas :

— f est un terme neutre : nous concluons en procédant comme ci-dessus dans le cas où f est une variable ;

— f est réductible pour $\rightarrow^h_{\Gamma; y:\mathrm{B}_0}$ alors, en appliquant (HRed-Simpl-Branch), $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ est réductible pour $\rightarrow^h_\Gamma$;[9]

— f est une paire implicite projetée : alors $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ est également une paire implicite projetée.

(3) Si f est un terme associé à R, montrons que $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}].f, c, \mathrm{P})$ est réductible pour $\rightarrow^h_\Gamma$ ou une paire implicite projetée. Par définition de terme associé, nous avons les trois cas suivants :

(a) f est une sorte (et R également) : $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}].f, c, \mathrm{P})$ est un redex pour $\rhd^{\mathrm{A}}_{s-\mathsf{Simpl}}$;

(b) f est un constructeur de type $\square z : \mathrm{T}\,.\,\mathrm{U}$ (et R est une sorte) : montrons que nous pouvons appliquer $\rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}}$ à $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$:

— par inversion de $\Gamma; y : \mathrm{B}_0 \vdash \square z : \mathrm{T}\,.\,\mathrm{U} : \mathrm{P}$, il existe $s_\mathrm{T}, s_\mathrm{U}$ tels que $\Gamma; y : \mathrm{B}_0 \vdash \mathrm{T} : s_\mathrm{T}$ et $\Gamma; y : \mathrm{B}_0; z : \mathrm{T} \vdash \mathrm{U} : s_\mathrm{U}$;

— par complétude de l'algorithme d'inférence de type extrait (proposition 4.2.1 page 161), il existe des termes de ICC_Σ $\mathrm{S}'_\mathrm{T}, \mathrm{S}'_\mathrm{U}$ tels que $\Gamma^*; y : \mathrm{B}^*_0 \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_\mathrm{T}$ et $\Gamma^*; y : \mathrm{B}^*_0; z : \mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_\mathrm{U}$;

— nous pouvons donc bien appliquer $\rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}}$:

$$\frac{\left\{\begin{array}{l}\Gamma^*; y : \mathrm{B}^*_0 \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_\mathrm{T} \\ \Gamma^*; y : \mathrm{B}^*_0; z : \mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_\mathrm{U}\end{array}\right. \qquad \left\{\begin{array}{l}\mathrm{T}_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].\mathrm{T}, c, \mathtt{NF\_BI}(\mathrm{S}'_\mathrm{T})) \\ \mathrm{U}_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].\mathrm{U}, c, \mathtt{NF\_BI}(\mathrm{S}'_\mathrm{U}))\end{array}\right.}{\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].\square z : \mathrm{T}\,.\,\mathrm{U}, c, \mathrm{P}) \rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}} \square z : \mathrm{T}_0\,.\,\mathrm{U}_0}$$

(c) f est un constructeur associé à R constructeur de type de AICC_Σ. Posons $\mathrm{R} = \square x : \mathrm{T}\,.\,\mathrm{U}$. Nous avons $(\square x : \mathrm{T}\,.\,\mathrm{U})^* \preceq \mathrm{P}^*$ pour un certain $\square$.

Montrons que $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ est réductible pour $\rightarrow^h_\Gamma$ ou une paire implicite projetée.

— Si P est un éliminateur, d'après le troisième résultat préliminaire, P est réductible pour $\rightarrow^h_\Gamma$. En appliquant règle (HRed-Simpl-Type), $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ est donc réductible pour $\rightarrow^h_\Gamma$.

— Si P n'est pas un éliminateur alors, d'après le deuxième résultat préliminaire ci-dessus, il existe $\mathrm{T}_0, \mathrm{U}_0$ tels que $\mathrm{P} = \square x : \mathrm{T}_0\,.\,\mathrm{U}_0$. f, constructeur associé à R est donc également un constructeur associé à P. Cinq cas sont possibles :

— si $\square z : \mathrm{T}_1\,.\,\mathrm{U}_1 = \Pi z : \mathrm{T}_1\,.\,\mathrm{U}_1$ alors $f = \lambda z{:}\mathrm{T}\,.\,\mathrm{M}$ et nous pouvons appliquer $\rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}}$ à $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$;

— si $\square z : \mathrm{T}_1\,.\,\mathrm{U}_1 = \Pi[z:\mathrm{T}_1]\,.\,\mathrm{U}_1$ alors $f = \lambda[z:\mathrm{T}]\,.\,\mathrm{M}$ et nous pouvons appliquer $\rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}}$ à $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$;

— si $\square z : \mathrm{T}_1\,.\,\mathrm{U}_1 = \Sigma z : \mathrm{T}_1\,.\,\mathrm{U}_1$ alors $f = (a, b)_{\Sigma z:\mathrm{A}\,.\,\mathrm{B}}$ et nous pouvons appliquer $\rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$ à $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$;

9. Notons que le contexte de typage dans la réduction a changé.

— si $\square z : \mathrm{T}_1 . \mathrm{U}_1 = \Sigma[z : \mathrm{T}_1] . \mathrm{U}_1$ alors $f = ([a], b)_{\Sigma[z:\mathrm{A}].\mathrm{B}}$; d'après la définition 2.2.1, $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ est une paire implicite gauche projetée ;

— si $\square z : \mathrm{T}_1 . \mathrm{U}_1 = \Sigma z : \mathrm{T}_1 . [\mathrm{U}_1]$ alors $f = ([a], b)_{\Sigma[z:\mathrm{A}].\mathrm{B}}$; d'après la définition 2.2.1, $\mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ est une paire implicite droite projetée.

□

2.3.2. REMARQUE. Cette preuve assez longue justifie bien l'ajout des différentes règles et les définitions des termes neutres et des paires implicites projetées : toutes les règles de réduction à la racine et dans un sous-terme sont utilisées ainsi que tous les différentes formes possibles de termes neutres ou paires implicites projetées. Le choix de ces règles et définitions s'est évidemment fait en parallèle de l'élaboration de la preuve du lemme de classification des éliminateurs.

2.3.3. REMARQUE. Si des règles analogues à $\rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$ avaient pu être définies pour les paires implicites, la définition de paire implicite projetée aurait été inutile et la preuve plus simple. Nous aurions pu montrer que tout éliminateur bien typé est soit un terme neutre soit réductible. Le cas $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-1)$ en particulier aurait été significativement raccourci.

2.3.1. *Canonicité.* Nous pouvons maintenant montrer une propriété usuelle des systèmes de type : les termes clos irréductibles typés par une sorte ou un constructeur de type ont une forme canonique.

2.3.4. Corollaire (Classification des termes clos)**.**

(1) Si $\vdash \mathrm{M} : \square x : \mathrm{T} . \mathrm{U}$*, alors* M *est un constructeur associé à* $\square x : \mathrm{T} . \mathrm{U}$ *ou réductible pour* $\rightarrow^h_\bullet$.

(2) Si $\vdash \mathrm{M} : s$*, alors* M *est une sorte ou un constructeur de type ou réductible pour* $\rightarrow^h_\bullet$.

DÉMONSTRATION.

(1) Nous procédons par induction sur la structure de M.

- En appliquant les résultats de classification établis précédemment à $\vdash \mathrm{M} : \square x : \mathrm{T} . \mathrm{U}$ (corollaire 2.2.4 page 119 et lemme 2.3.1), M est
  (a) un constructeur associé à $\square x : \mathrm{T} . \mathrm{U}$;
  (b) ou un terme neutre (y compris une variable) ;
  (c) ou réductible pour $\rightarrow^h_\bullet$;
  (d) ou une paire implicite projetée.
- (a) et (c) permettent de conclure.
- (b) est impossible car les termes neutres ne sont pas clos (remarque 2.1.7).
- si (d), posons $\mathrm{M} = \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P})$ et montrons que M est réductible pour $\rightarrow^h_\bullet$.
  - Par inversion de $\vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}].f, c, \mathrm{P}) : \square x : \mathrm{T} . \mathrm{U}$, il existe A, B tels que $\vdash c : \Sigma x : \mathrm{A} . [\mathrm{B}]$.
  - Par induction sur c, c est (i) une paire implicite droite ou (ii) réductible pour $\rightarrow^h_\bullet$. Si (i) alors M est un ι^2_σ-redex, donc réductible pour $\rightarrow^h_\bullet$. Si (ii) M est réductible pour $\rightarrow^h_\bullet$.

(2) La preuve est similaire sauf que le cas de la paire implicite projetée est traité en appliquant le point 1. Il n'est pas nécessaire de raisonner par induction structurelle.

□

3. Correction des règles de réduction

Nous montrons dans cette section que si $\Gamma \vdash \mathrm{M} : \mathrm{R}$ et si $\mathrm{R} \rightarrow^h_\Gamma \mathrm{R}_0$ alors $\Gamma \vdash \mathrm{M} : \mathrm{R}_0$. Pour cela nous commençons par prouver que la réduction dans AICC_Σ est compatible avec la réduction dans ICC_Σ, puis nous prouverons que $\rightarrow^h_\Gamma$ préserve le typage.

3.1. Correction pour la réduction. Nous montrons dans cette sous-section que si R se réduit vers R_0 dans $AICC_\Sigma$ alors $R^* = R_0^*$ ou $R^* \to_{\beta\iota} R_0^*$. Ce résultat n'est valide que pour les termes dont les variables implicites liées sont bien positionnées. Nous montrons la correction d'abord pour les réductions à la racine, puis pour la réduction dans un contexte.

3.1.1. *Réduction à la racine.*

3.1.1. Proposition (Correction pour la réduction)**.** *Si* $\Gamma \vdash M_0 : R$ *et si* $M_0 \rhd^A_\Gamma M_1$ *alors* $M_0^* = M_1^*$ *ou* $M_0^* \rhd_{\beta\iota} M_1^*$.

DÉMONSTRATION. Nous distinguons les différentes règles appliquées dans $M_0 \rhd^A_\Gamma M_1$. Deux cas se présentent :

(1) Nous n'utilisons pas le fait que les variables liées de M_0 sont bien positionnées : nous avons deux sous-cas :
   (a) pour les règles $\rhd^A_{\beta^e}$ et $\rhd^A_{\iota_\Sigma}$, nous montrons respectivement que $M_0^* \rhd_\beta M_1^*$ et que $M_0^* \rhd_\iota M_1^*$.
   (b) pour les règles $\rhd^A_{\iota^1_\sigma}$, $\mathsf{Constr{-}Simpl}$, $\mathsf{Elim{-}Simpl}$, $\rhd^A_{\Gamma,\ \square\text{-}\mathsf{Simpl}}$ et $\rhd^A_{s\text{-}\mathsf{Simpl}'}$ nous montrons directement que $M_0^* = M_1^*$.

(2) Nous utilisons l'hypothèse sur la position des variables liées implicites de M_0. Là encore, deux sous-cas :
   (a) $\rhd^A_{\iota_\exists}$: Nous avons $M_0^* = \mathsf{Elim}_\exists(y.f^*, (\diamond, b^*)) \rhd_{\iota_\exists} f^*[y/b^*]$ et $M_1^* = f^*[x/a^*][y/b^*]$. Montrons que $M_1^* = f^*[y/b^*]$.
      - Par hypothèse sur les variables implicites liées de M_0, nous avons $x \notin \mathsf{FV}(f^*)$.
      - Le lemme 1.2.12 page 10 nous indique alors que $f^*[x/a^*] = f^*$, d'où $M_1^* = f^*[y/b^*]$.
   (b) Pour les règles $\rhd^A_{\beta^i}$ et $\rhd^A_{\iota^2_\sigma}$, nous montrons que $M_0^* = M_1^*$.
      - Pour $\rhd^A_{\beta^i}$: Nous avons $M_0^* = M^*$ et $M_1^* = M^*[x/N^*]$. De même que pour le cas précédent, nous avons par hypothèse $x \notin \mathsf{FV}(M^*)$ et concluons grâce au même lemme 1.2.12 page 10.
      - $\rhd^A_{\iota^2_\sigma}$: Nous avons $M_0^* = f^*$ et $M_1^* = f^*[y/b^*]$. Similairement à $\rhd^A_{\iota_\exists}$ et $\rhd^A_{\beta^i}$, nous avons par hypothèse que $y \notin \mathsf{FV}(f^*)$ d'où $f^*[y/b^*] = f^*$.

□

3.1.2. *Réduction dans un contexte.*

3.1.2. Proposition (Correction pour la réduction)**.** *Si* $\Gamma \vdash N_0 : R$ *et* $N_0 \to^h_\Gamma N_1$ *alors* $N_0^* = N_1^*$ *ou* $N_0^* \to_{\beta\iota} N_1^*$.

DÉMONSTRATION. Par induction sur la dérivation de $N_0 \to^h_\Gamma N_1$.

- Le cas de base (HRED-BASE) découle de la proposition 3.1.1.
- Le cas (HRED-Simpl-TYPE) est trivial car $N_0^* = N_1^* = f^*$.
- Les autres cas se traitent immédiatement par induction : le résultat peut s'appliquer aux sous-termes car ils sont bien typés. Notons que le changement de contexte pour la règle (HRED-Simpl-BRANCH) ne pose pas de problème car si $N_i = \mathsf{Elim}^s_{I_R}([y{:}B].M_i, c, P)$ pour $i \in \{1,2\}$ et si $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_0, c, P) : R$ alors par inversion $\Gamma; y : B \vdash M_0 : P$ et nous pouvons bien supposer par induction que $M_0 \to^h_{\Gamma;y:B} M_1$.

□

3.2. Correction pour le typage.

3.2.1. *Réduction à la racine.*

Réductions standard.

3.2.1. Lemme (Correction pour $\triangleright^{\mathrm{A}}_{\beta\iota}$)**.** *Si* $\mathrm{M}_0 \triangleright^{\mathrm{A}}_{\beta\iota} \mathrm{M}_1$ *et si* $\Gamma \vdash \mathrm{M}_0 : \mathrm{R}$ *alors* $\Gamma \vdash \mathrm{M}_1 : \mathrm{R}$.

DÉMONSTRATION. Nous distinguons les différentes règles de $\triangleright^{\mathrm{A}}_{\beta\iota}$.

- $\triangleright^{\mathrm{A}}_{\beta^e}$ et $\triangleright^{\mathrm{A}}_{\beta^i}$ sont similaires. Traitons $\triangleright^{\mathrm{A}}_{\beta^e}$. Si $\Gamma \vdash (\lambda x{:}\mathrm{T}.\mathrm{M})\,\mathrm{N} : \mathrm{R}$. Montrons que $\Gamma \vdash \mathrm{M}[x/\mathrm{N}] : \mathrm{R}$.
  - Par inversion de $\Gamma \vdash (\lambda x{:}\mathrm{T}.\mathrm{M})\,\mathrm{N} : \mathrm{R}$, il existe A, B tels que
    - $\Gamma \vdash \lambda x{:}\mathrm{T}.\mathrm{M} : \Pi x : \mathrm{A}.\mathrm{B}$
    - $\Gamma \vdash \mathrm{N} : \mathrm{A}$
    - $(\mathrm{B}[x/\mathrm{N}])^* \preceq \mathrm{R}^*$
  - Par inversion de $\Gamma \vdash \lambda x{:}\mathrm{T}.\mathrm{M} : \Pi x : \mathrm{A}.\mathrm{B}$, il existe une sorte s et un terme U tels que
    - $\Gamma; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{U}$
    - $\Gamma \vdash \Pi x : \mathrm{T}.\mathrm{U} : s$
    - $(\Pi x : \mathrm{T}.\mathrm{U})^* \preceq (\Pi x : \mathrm{A}.\mathrm{B})^*$
  - Par inversion de $(\Pi x : \mathrm{T}.\mathrm{U})^* \preceq (\Pi x : \mathrm{A}.\mathrm{B})^*$ nous avons $\mathrm{A}^* \preceq \mathrm{T}^*$ et $\mathrm{U}^* \preceq \mathrm{B}^*$.
  - Nous montrons que $(\Gamma; x : \mathrm{A})^* \preceq (\Gamma; x : \mathrm{T})^*$ et que $\Gamma; x : \mathrm{A} \vdash$. Nous en déduisons grâce au corollaire 2.3.7 page 122 que $\Gamma; x : \mathrm{A} \vdash \mathrm{M} : \mathrm{U}$.
  - Par substitutivité, nous avons $\Gamma \vdash \mathrm{M}[x/\mathrm{N}] : \mathrm{U}[x/\mathrm{N}]$.
  - Par cumulativité, nous montrons que $\Gamma \vdash \mathrm{M}[x/\mathrm{N}] : \mathrm{B}[x/\mathrm{N}]$ puis que $\Gamma \vdash \mathrm{M}[x/\mathrm{N}] : \mathrm{R}$.
- $\triangleright^{\mathrm{A}}_{\iota\Sigma}$: Nous avons $\Gamma \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{T}).(y{:}\mathrm{U}).f, (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}}, \mathrm{P}) : \mathrm{R}$. Montrons que $\Gamma \vdash f[x/a][y/b] : \mathrm{R}$.
  - Par inversion de $\Gamma \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{T}).(y{:}\mathrm{U}).f, (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}}, \mathrm{P}) : \mathrm{R}$,
    - $\Gamma \vdash (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}} : \Sigma x : \mathrm{T}.\mathrm{U}$
    - $\Gamma; x : \mathrm{T}; y : \mathrm{U} \vdash f : \mathrm{P}\,(x,y)_{\Sigma x:\mathrm{T}.\mathrm{U}}$
    - $(\mathrm{P}\,(a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}})^* \preceq \mathrm{R}^*$
  - Par inversion de $\Gamma \vdash (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}} : \Sigma x : \mathrm{T}.\mathrm{U}$:
    - $\Gamma \vdash a : \mathrm{A}$
    - $\Gamma \vdash b : \mathrm{B}[x/a]$
    - $(\Sigma x : \mathrm{A}.\mathrm{B})^* \preceq (\Sigma x : \mathrm{T}.\mathrm{U})^*$,
  - Par inversion de $(\Sigma x : \mathrm{A}.\mathrm{B})^* \preceq (\Sigma x : \mathrm{T}.\mathrm{U})^*$, nous avons
    - $\mathrm{A}^* \preceq \mathrm{T}^*$
    - $\mathrm{B}^* \preceq \mathrm{U}^*$, puis $(\mathrm{B}[x/a])^* \preceq (\mathrm{U}[x/a])^*$.
  - Par cumulativité, nous montrons que
    - $\Gamma \vdash a : \mathrm{T}$
    - $\Gamma \vdash b : \mathrm{U}[x/a]$.
  - Par substitutivité sur $\Gamma; x : \mathrm{T}; y : \mathrm{U} \vdash f : \mathrm{P}\,(x,y)_{\Sigma x:\mathrm{T}.\mathrm{U}}$ nous avons :
    - en substituant x par $a : \Gamma; y : \mathrm{U}[x/a] \vdash f[x/a] : \mathrm{P}\,(a,y)_{\Sigma x:\mathrm{T}.\mathrm{U}}$
    - puis, en substituant y par $b : \Gamma \vdash f[x/a][y/b] : \mathrm{P}\,(a,b)_{\Sigma x:\mathrm{T}.\mathrm{U}}$.

- Par cumulativité sur $\Gamma \vdash f\,[x/a]\,[y/b] : \mathrm{P}\,(a,b)_{\Sigma x:\mathrm{T}.\mathrm{U}}$, nous montrons finalement $\Gamma \vdash f\,[x/a]\,[y/b] : \mathrm{R}$.

- $\rhd^{\mathrm{A}}_{\iota_\exists}$: Similaire à $\rhd^{\mathrm{A}}_{\iota_\Sigma}$.
- $\rhd^{\mathrm{A}}_{\iota^1_\sigma}$: Similaire, mais en plus simple, à $\rhd^{\mathrm{A}}_{\iota_\Sigma}$: double utilisation des lemmes d'inversion, changement de type par cumulativité, mais pas besoin du lemme de substitutivité.
- $\rhd^{\mathrm{A}}_{\iota^2_\sigma}$: Nous avons $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].(a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}, f, \mathrm{P}) : \mathrm{R}$. Montrons que $\Gamma \vdash f\,[y/b] : \mathrm{R}$.
  - Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].(a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]}, f, \mathrm{P}) : \mathrm{R}$, il existe $s, \mathrm{T}, \mathrm{U}$ tels que
    - $\Gamma \vdash (a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]} : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]$
    - $\Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P}$
    - $\Gamma \vdash \mathrm{P} : s$
    - $(\mathrm{U}\,[x/\pi_1((a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]})])^* = \mathrm{U}^*\,[x/a^*] \preceq \mathrm{B}_0^*$
    - $\mathrm{P}^* \preceq \mathrm{R}^*$.
  - Par inversion de $\Gamma \vdash (a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]} : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]$, nous avons
    - $\Gamma \vdash b : \mathrm{B}\,[x/a]$,
    - $(\Sigma x : \mathrm{A}\,.\,[\mathrm{B}])^* \preceq (\Sigma x : \mathrm{T}\,.\,[\mathrm{U}])^*$, d'où $\mathrm{A}^* \preceq \mathrm{T}^*$ et $\mathrm{B}^* \preceq \mathrm{U}^*$ par inversion de la cumulativité.
  - Montrons que $\Gamma \vdash b : \mathrm{B}_0$ par cumulativité sur $\Gamma \vdash b : \mathrm{B}\,[x/a]$:
    - Montrons qu'il existe une sorte s_0 telle que $\Gamma \vdash \mathrm{B}_0 : s_0$.
      - De $\Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P}$ nous déduisons $\Gamma; y : \mathrm{B}_0 \vdash$.
      - Nous concluons par inversion de $(\mathrm{WF\text{-}S_A})$.
    - Montrons que $\mathrm{B}^*\,[x/a^*] \preceq \mathrm{B}_0^*$.
      - De $\mathrm{B}^* \preceq \mathrm{U}^*$ nous déduisons $\mathrm{B}^*\,[x/a^*] \preceq \mathrm{U}^*\,[x/a^*]$.
      - Nous concluons par transitivité puisque $\mathrm{U}^*\,[x/a^*] \preceq \mathrm{B}_0^*$.
  - Par substitutivité sur $\Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P}$ et $\Gamma \vdash b : \mathrm{B}_0$, puisque $y \notin \mathsf{FV}(\mathrm{P})$, $\Gamma \vdash f\,[y/b] : \mathrm{P}$.
  - Par cumulativité sur $\Gamma \vdash f\,[y/b] : \mathrm{P}$, nous avons bien $\Gamma \vdash f\,[y/b] : \mathrm{R}$.

□

Constructeurs et éliminateur simplifié.

3.2.2. Lemme (Correction pour $\rhd^{\mathrm{A}}_{\mathsf{Constr-Simpl}}$)**.** *Si* $\mathrm{M}_0 \rhd^{\mathrm{A}}_{\mathsf{Constr-Simpl}} \mathrm{M}_1$ *et si* $\Gamma \vdash \mathrm{M}_0 : \mathrm{R}$ *alors* $\Gamma \vdash \mathrm{M}_1 : \mathrm{R}$.

DÉMONSTRATION. Nous distinguons les différentes règles appliquées dans $\mathrm{M}_0 \rhd^{\mathrm{A}}_{\mathsf{Constr-Simpl}} \mathrm{M}_1$.

- $\rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}}$ et $\rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}}$ sont similaires. Traitons $\rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}}$. Si

  $\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\lambda[z{:}\mathrm{T}]\,.\,\mathrm{M}, c, \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0) \rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}} \lambda[z{:}\mathrm{T}_0]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0)$

  et si $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\lambda[z{:}\mathrm{T}]\,.\,\mathrm{M}, c, \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0) : \mathrm{R}$, montrons que $\Gamma \vdash \lambda[z{:}\mathrm{T}_0]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \mathrm{R}$.
  - Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\lambda[z{:}\mathrm{T}]\,.\,\mathrm{M}, c, \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0) : \mathrm{R}$, il existe $\mathrm{A}, \mathrm{B}, s_0$ tels que tels que
    - $\Gamma \vdash \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0 : s_0$
    - $\Gamma \vdash c : \Sigma x : \mathrm{A}\,.\,[\mathrm{B}]$
    - $\Gamma; y : \mathrm{B}_0 \vdash \lambda[z{:}\mathrm{T}]\,.\,\mathrm{M} : \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0$
    - $y \notin \mathsf{FV}((\lambda[z{:}\mathrm{T}]\,.\,\mathrm{M})^*) = \mathsf{FV}(\mathrm{M}^*)$

— $(\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^*$

— $(\Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0)^* \preceq \mathrm{R}^*$.

- Par inversion de $\Gamma; y:\mathrm{B}_0 \vdash \lambda[z{:}\mathrm{T}]\,.\,\mathrm{M} : \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0$ il existe U, s tels que
  — $\Gamma; y:\mathrm{B}_0; z:\mathrm{T} \vdash \mathrm{M} : \mathrm{U}$
  — $\Gamma; y:\mathrm{B}_0 \vdash \Pi[z{:}\mathrm{T}]\,.\,\mathrm{U} : s$
  — $z \notin \mathsf{FV}(\mathrm{M}^*)$
  — $(\Pi[z:\mathrm{T}]\,.\,\mathrm{U})^* \preceq (\Pi[z:\mathrm{T}_0]\,.\,\mathrm{U}_0)^*$, d'où $\mathrm{T}_0^* \preceq \mathrm{T}^*$ et $\mathrm{U}^* \preceq \mathrm{U}_0^*$ par inversion de la cumulativité.
- Par inversion de $\Gamma \vdash \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0 : s_0$, il existe s_1, s_2 telles que
  — $\Gamma \vdash \mathrm{T}_0 : s_1$
  — $\Gamma; z:\mathrm{T}_0 \vdash \mathrm{U}_0 : s_2$.
- Montrons que $\Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{M} : \mathrm{U}_0$ à partir de $\Gamma; y:\mathrm{B}_0; z:\mathrm{T} \vdash \mathrm{M} : \mathrm{U}$ en trois étapes : changement de T en T_0, échange des variables y et z, changement de U en U_0.
  (1) Montrons $\Gamma; y:\mathrm{B}_0; z:\mathrm{T}_0 \vdash \mathrm{M} : \mathrm{U}$ par changement de contexte sur $\Gamma; y:\mathrm{B}_0; z:\mathrm{T} \vdash \mathrm{M} : \mathrm{U}$ (lemme 2.3.6 page 122) :
     — nous avons $\mathrm{T}_0^* \preceq \mathrm{T}^*$ d'où $(\Gamma; y:\mathrm{B}_0; z:\mathrm{T})^* \preceq (\Gamma; y:\mathrm{B}_0; z:\mathrm{T}_0)^*$;
     — montrons que $\Gamma; y:\mathrm{B}_0; z:\mathrm{T}_0 \vdash$:
       — nous déduisons $\Gamma; y:\mathrm{B}_0 \vdash$ de $\Gamma; y:\mathrm{B}_0 \vdash \Pi[z{:}\mathrm{T}]\,.\,\mathrm{U} : s$;
       — par affaiblissement de $\Gamma \vdash \mathrm{T}_0 : s_1$, nous obtenons $\Gamma; y:\mathrm{B}_0 \vdash \mathrm{T}_0 : s_1$;
       — nous concluons en appliquant $(\mathrm{WF\text{-}S_A})$.
  (2) Montrons que $\Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{M} : \mathrm{U}$:
     — puisque $\Gamma; y:\mathrm{B}_0 \vdash$, nous avons $z \notin \mathsf{FV}(\mathrm{B}_0)$;
     — nous pouvons donc, d'après le lemme 2.3.12 page 125, échanger les variables y et z dans $\Gamma; y:\mathrm{B}_0; z:\mathrm{T} \vdash \mathrm{M} : \mathrm{U}$.
  (3) Montrons finalement que $\Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{M} : \mathrm{U}_0$ par cumulativité sur $\Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{M} : \mathrm{U}$:
     — nous avons $\mathrm{U}^* \preceq \mathrm{U}_0^*$
     — par affaiblissement de $\Gamma; z:\mathrm{T}_0 \vdash \mathrm{U}_0 : s_2$, nous obtenons $\Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{U}_0 : s_2$.
- Montrons que $\Gamma; z:\mathrm{T}_0 \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \mathrm{U}_0$. Nous voulons appliquer $(\Sigma_{\mathrm{SUB}}\mathrm{-E}\mathrm{-2})$:

$$\frac{\Gamma; z:\mathrm{T}_0 \vdash \mathrm{U}_0 : s_2 \qquad \Gamma; z:\mathrm{T}_0 \vdash c : \Sigma x:\mathrm{A}\,.\,[\mathrm{B}] \qquad \Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{M} : \mathrm{U}_0 \qquad \begin{array}{l} y \notin \mathsf{FV}(\mathrm{M}^*) \\ (\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \end{array}}{\Gamma; z:\mathrm{T}_0 \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \mathrm{U}_0}$$

  — Les conditions de bord $y \notin \mathsf{FV}(\mathrm{M}^*)$ et $(\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^*$ ont déjà été démontrées.
  — Les prémisses $\Gamma; z:\mathrm{T}_0 \vdash \mathrm{U}_0 : s_2$ et $\Gamma; z:\mathrm{T}_0; y:\mathrm{B}_0 \vdash \mathrm{M} : \mathrm{U}_0$ ont déjà été montrées dérivables.
  — La dernière prémisse $\Gamma; z:\mathrm{T}_0 \vdash c : \Sigma x:\mathrm{A}\,.\,[\mathrm{B}]$ est dérivable par affaiblissement de $\Gamma \vdash c : \Sigma x:\mathrm{A}\,.\,[\mathrm{B}]$.

- Montrons enfin que $\Gamma \vdash \lambda[z{:}\mathrm{T}_0]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \mathrm{R}$.
  - — Puisque $z \notin \mathsf{FV}(\mathrm{M}^*) = \mathsf{FV}((\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0))^*)$, nous pouvons appliquer (I-LAM) à $\Gamma; z : \mathrm{T}_0 \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \mathrm{U}_0$ et dériver
    $$\Gamma \vdash \lambda[z{:}\mathrm{T}_0]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \Pi[z{:}\mathrm{T}_0]\,.\,\mathrm{U}_0.$$
  - — Par cumulativité, nous déduisons que $\Gamma \vdash \lambda[z{:}\mathrm{T}_0]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}_0].\mathrm{M}, c, \mathrm{U}_0) : \mathrm{R}$.

- $\rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}}$: Si $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].(a,b)_{\Sigma z:\mathrm{A}.\mathrm{B}}, c, \Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0) : \mathrm{R}$ montrons que $\Gamma \vdash (a_0, b_0)_{\Sigma z:\mathrm{A}_0\,.\,\mathrm{B}_0} : \mathrm{R}$ avec $a_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].a, c, \mathrm{A}_0)$ et $b_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].b, c, \mathrm{B}_0\,[x/a_0])$.
  - Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].(a,b)_{\Sigma z:\mathrm{A}.\mathrm{B}}, c, \Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0) : \mathrm{R}$, il existe $s_0, \mathrm{T}, \mathrm{U}$ tels que
    - — $\Gamma \vdash \Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0 : s_0$
    - — $\Gamma \vdash c : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]$
    - — $\Gamma; y : \mathrm{U}_0 \vdash (a,b)_{\Sigma z:\mathrm{A}.\mathrm{B}} : \Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0$
    - — $y \notin \mathsf{FV}(((a,b)_{\Sigma z:\mathrm{A}.\mathrm{B}})^*)$ d'où $y \notin \mathsf{FV}(a^*)$ et $y \notin \mathsf{FV}(b^*)$. [10]
    - — $(\mathrm{U}\,[x/\pi_1(c)])^* \preceq \mathrm{U}_0^*$
    - — $(\Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0)^* \preceq \mathrm{R}^*$.
  - Par inversion de $\Gamma; y : \mathrm{U}_0 \vdash (a,b)_{\Sigma z:\mathrm{A}.\mathrm{B}} : \Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0$, il existe s tel que
    - — $\Gamma; y : \mathrm{U}_0 \vdash a : \mathrm{A}$
    - — $\Gamma; y : \mathrm{U}_0 \vdash b : \mathrm{B}\,[z/a]$
    - — $\Gamma; y : \mathrm{U}_0 \vdash \Sigma z : \mathrm{A}.\mathrm{B} : s$
    - — $(\Sigma z : \mathrm{A}.\mathrm{B})^* \preceq (\Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0)^*$, d'où $\mathrm{A}^* \preceq \mathrm{A}_0^*$ et $\mathrm{B}^* \preceq \mathrm{B}_0^*$, par inversion de la cumulativité.
  - Par inversion de $\Gamma \vdash \Sigma z : \mathrm{A}_0\,.\,\mathrm{B}_0 : s_0$, il existe s_1, s_2 telles que
    - — $\Gamma \vdash \mathrm{A}_0 : s_1$
    - — $\Gamma; z : \mathrm{A}_0 \vdash \mathrm{B}_0 : s_2$.
  - Nous pouvons dériver $\Gamma \vdash a_0 : \mathrm{A}_0$ en appliquant ($\Sigma_{\mathrm{SUB}}-\mathrm{E}-2$)
    $$\frac{\Gamma \vdash \mathrm{A}_0 : s_1 \qquad \Gamma \vdash c : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}] \qquad \Gamma; y : \mathrm{U}_0 \vdash a : \mathrm{A}_0 \qquad \begin{array}{l} y \notin \mathsf{FV}(a^*) \\ (\mathrm{U}\,[x/\pi_1(c)])^* \preceq \mathrm{U}_0^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].a, c, \mathrm{A}_0) : \mathrm{A}_0}$$
    puisque nous dérivons $\Gamma \vdash a : \mathrm{A}_0$ par cumulativité sur $\Gamma \vdash a : \mathrm{A}$. [11]
  - Montrons que $\Gamma \vdash b_0 : \mathrm{B}_0\,[z/a_0]$: nous voulons appliquer ($\Sigma_{\mathrm{SUB}}-\mathrm{E}-2$) [12]
    $$\frac{\Gamma \vdash \mathrm{B}_0\,[z/a_0] : s_2 \qquad \Gamma \vdash c : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}] \qquad \Gamma; y : \mathrm{U}_0 \vdash b : \mathrm{B}_0\,[z/a_0] \qquad \begin{array}{l} y \notin \mathsf{FV}(b^*) \\ (\mathrm{U}\,[x/\pi_1(c)])^* \preceq \mathrm{U}_0^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U}_0].b, c, \mathrm{B}_0\,[z/a_0]) : \mathrm{B}_0\,[z/a_0]}$$
    Il suffit de montrer que les prémisses $\Gamma \vdash \mathrm{B}_0\,[z/a_0] : s_2$ et $\Gamma; y : \mathrm{U}_0 \vdash b : \mathrm{B}_0\,[z/a_0]$ sont dérivables :

10. Dans le cas de la paire implicite gauche $([a], b)_{\Sigma[z\mathrm{A}]\,.\,\mathrm{B}}$, nous avons seulement $y \notin \mathsf{FV}(b^*)$; dans celui de la paire implicite droite $(a, [b])_{\Sigma z:\mathrm{A}.\,[\mathrm{B}]}$, $y \notin \mathsf{FV}(a^*)$.

11. Cette étape serait impossible pour $([a], b)_{\Sigma[z\mathrm{A}]\,.\,\mathrm{B}}$ car nous n'avons pas $y \notin \mathsf{FV}(a^*)$.

12. De même, cela serait impossible pour $(a, [b])_{\Sigma z:\mathrm{A}.\,[\mathrm{B}]}$ car nous n'avons pas $y \notin \mathsf{FV}(b^*)$.

— par substitutivité sur $\Gamma; z : \mathrm{A}_0 \vdash \mathrm{B}_0 : s_2$ et $\Gamma \vdash a_0 : \mathrm{A}_0$, nous obtenons $\Gamma \vdash \mathrm{B}_0[z/a_0] : s_2$;

— nous montrons facilement que $(\mathrm{B}[z/a])^* \preceq (\mathrm{B}_0[z/a_0])^*$ et en déduisons $\Gamma; y : \mathrm{U}_0 \vdash b : \mathrm{B}_0[z/a_0]$ par cumulativité sur $\Gamma; y : \mathrm{U}_0 \vdash b : \mathrm{B}[z/a]$.

- Montrons finalement que $\Gamma \vdash (a_0, b_0)_{\Sigma z:\mathrm{A}_0.\mathrm{B}_0} : \mathrm{R}$

— nous dérivons $\Gamma \vdash (a_0, b_0)_{\Sigma z:\mathrm{A}_0.\mathrm{B}_0} : \Sigma z : \mathrm{A}_0.\mathrm{B}_0$ en appliquant $(\Sigma_{\mathrm{A}}\text{-I})$

$$\frac{\Gamma \vdash a_0 : \mathrm{A}_0 \qquad \Gamma \vdash b_0 : \mathrm{B}_0[z/a_0] \qquad \Gamma \vdash \Sigma z : \mathrm{A}_0.\mathrm{B}_0 : s_0}{\Gamma \vdash (a_0, b_0)_{\Sigma z:\mathrm{A}_0.\mathrm{B}_0} : \Sigma z : \mathrm{A}_0.\mathrm{B}_0}$$

— nous concluons par cumulativité.

□

3.2.2. *Éliminateurs de paire implicite et éliminateur simplifié.*

3.2.3. Lemme (Correction pour $\triangleright^{\mathrm{A}}_{\mathsf{Elim-Simpl}}$)**.** *Si* $\mathrm{M}_0 \triangleright^{\mathrm{A}}_{\mathsf{Elim-Simpl}} \mathrm{M}_1$ *et si* $\Gamma \vdash \mathrm{M}_0 : \mathrm{R}$ *alors* $\Gamma \vdash \mathrm{M}_1 : \mathrm{R}$.

DÉMONSTRATION. Nous distinguons les différentes règles appliquées dans $\mathrm{M}_0 \triangleright^{\mathrm{A}}_{\mathsf{Elim-Simpl}} \mathrm{M}_1$.

- $\triangleright^{\mathrm{A}}_{\exists-\mathsf{Simpl}}$: Supposons que $\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f, \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$ et montrons que $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].\mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f, g, \mathrm{P}), d, \mathrm{P}\,(\mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U}))) : \mathrm{R}$. Nous procédons comme précédemment en deux étapes :

(1) Commençons par déconstruire, par inversion des règles $(\Sigma_\exists - \mathrm{E})$ et $(\Sigma_{\mathrm{SUB}} - \mathrm{E} - 2)$, le jugement de départ.

- Par inversion de $\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f, \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$, il existe des sortes s, s' telles que
  - $\Gamma \vdash \Sigma[x:\mathrm{A}].\mathrm{B} : s'$
  - $\Gamma \vdash \mathrm{P} : \Sigma[x:\mathrm{A}].\mathrm{B} \to s$
  - $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U}) : \Sigma[x:\mathrm{A}].\mathrm{B}$
  - $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}].\mathrm{B}}$
  - $x \notin \mathsf{FV}(f^*)$
  - $(\mathrm{P}\,\mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U}))^* \preceq \mathrm{R}^*$.
- Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U}) : \Sigma[x:\mathrm{A}].\mathrm{B}$, il existe $s_1, \mathrm{A}_1, \mathrm{B}_1$ tels que
  - $\Gamma \vdash \mathrm{U} : s_1$
  - $\Gamma \vdash d : \Sigma u : \mathrm{A}_1.[\mathrm{B}_1]$
  - $\Gamma; z : \mathrm{T} \vdash g : \mathrm{U}$
  - $z \notin \mathsf{FV}(g^*)$
  - $(\mathrm{B}_1[u/\pi_1(d)])^* \preceq \mathrm{T}^*$
  - $\mathrm{U}^* \preceq (\Sigma[x:\mathrm{A}].\mathrm{B})^*$.

(2) Construisons maintenant le jugement final en appliquant $(\Sigma_\exists - \mathrm{E})$, $(\Sigma_{\mathrm{SUB}} - \mathrm{E} - 2)$ puis $(\mathrm{CUM_A})$. Posons $f_0 = \mathsf{Elim}_{\mathrm{I_L}}([x:\mathrm{A}].(y:\mathrm{B}).f, g, \mathrm{P})$ et $g_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].g, d, \mathrm{U})$. Nous voulons donc montrer que $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z:\mathrm{T}].f_0, d, \mathrm{P}\,g_0) : \mathrm{R}$.

- Montrons que $\Gamma; z : \mathrm{T} \vdash f_0 : \mathrm{P}\,g$. Pour cela, montrons que nous pouvons appliquer la règle $(\Sigma_\exists - \mathrm{E})$:

$$\frac{\Gamma;z:\mathrm{T}\vdash \mathrm{P}:\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}\to s \qquad \Gamma;z:\mathrm{T}\vdash g:\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \qquad \Gamma;z:\mathrm{T};x:\mathrm{A};y:\mathrm{B}\vdash f:\mathrm{P}\,([x],y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} \qquad \Gamma;z:\mathrm{T}\vdash\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}:s' \qquad x\notin \mathsf{FV}(f^*)}{\Gamma;z:\mathrm{T}\vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f,g,\mathrm{P}):\mathrm{P}\,g}$$

— La condition de bord $x\notin \mathsf{FV}(f^*)$ a déjà été démontrée.

— Montrons que $\Gamma;z:\mathrm{T}\vdash\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}:s'$:

— De $\Gamma;z:\mathrm{T}\vdash g:\mathrm{U}$ nous déduisons $\Gamma;z:\mathrm{T}\vdash$.

— Nous concluons par affaiblissement de $\Gamma\vdash\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}:s'$.

— De même, nous montrons que $\Gamma;z:\mathrm{T}\vdash \mathrm{P}:\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}\to s$, par affaiblissement sur $\Gamma\vdash \mathrm{P}:\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}\to s$.

— Par cumulativité sur $\Gamma;z:\mathrm{T}\vdash g:\mathrm{U}$, nous dérivons $\Gamma;z:\mathrm{T}\vdash g:\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}$ puisque

— $\Gamma;z:\mathrm{T}\vdash\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}:s'$;

— $\mathrm{U}^*\preceq(\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B})^*$.

— Montrons enfin par affaiblissement sur $\Gamma;x:\mathrm{A};y:\mathrm{B}\vdash f:\mathrm{P}\,([x],y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}}$ que $\Gamma;z:\mathrm{T};x:\mathrm{A};y:\mathrm{B}\vdash f:\mathrm{P}\,([x],y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}}$; il suffit pour cela de montrer que $\Gamma;z:\mathrm{T};x:\mathrm{A};y:\mathrm{B}\vdash$:

— Nous avons $\Gamma;z:\mathrm{T}\vdash$.

— Par inversion de $\Gamma;x:\mathrm{A};y:\mathrm{B}\vdash$, nous avons des sortes $s_\mathrm{A},s_\mathrm{B}$ telles que $\Gamma\vdash\mathrm{A}:s_\mathrm{A}$ et $\Gamma;x:\mathrm{A}\vdash\mathrm{B}:s_\mathrm{B}$.

— Par affaiblissement de $\Gamma\vdash\mathrm{A}:s_\mathrm{A}$, nous avons $\Gamma;z:\mathrm{T}\vdash\mathrm{A}:s_\mathrm{A}$ puis, en appliquant $(\textsc{Wf-S}_\mathrm{A})$, $\Gamma;z:\mathrm{T};x:\mathrm{A}\vdash$.

— Par affaiblissement de $\Gamma;x:\mathrm{A}\vdash\mathrm{B}:s_\mathrm{B}$, nous avons $\Gamma;z:\mathrm{T};x:\mathrm{A}\vdash\mathrm{B}:s_\mathrm{B}$, puis $\Gamma;z:\mathrm{T};x:\mathrm{A};y:\mathrm{B}\vdash$.

- Montrons que $\Gamma;z:\mathrm{T}\vdash f_0:\mathrm{P}\,g_0$ par cumulativité sur $\Gamma;z:\mathrm{T}\vdash f_0:\mathrm{P}\,g$:

— Par définition de g_0, nous avons $g_0^*=g^*$ d'où $(\mathrm{P}\,g)^*\preceq(\mathrm{P}\,g_0)^*$.

— Montrons que $\Gamma;z:\mathrm{T}\vdash\mathrm{P}\,g_0:s$:

— Par application de $(\textsc{E-App})$ sur $\Gamma\vdash\mathrm{P}:\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}\to s$ et $\Gamma\vdash\mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g,d,\mathrm{U}):\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$, nous dérivons $\Gamma\vdash\mathrm{P}\,\mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g,d,\mathrm{U}):s$ soit $\Gamma\vdash\mathrm{P}\,g_0:s$.

— Nous concluons par affaiblissement puisque $\Gamma;z:\mathrm{T}\vdash$.

- Montrons que $\Gamma\vdash\mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0,d,\mathrm{P}\,g_0):\mathrm{P}\,g_0$. Pour cela, montrons que nous pouvons appliquer la règle $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-2)$:

$$\frac{\Gamma\vdash\mathrm{P}\,g_0:s \qquad \Gamma\vdash d:\Sigma u:\mathrm{A}_1\,.\,[\mathrm{B}_1] \qquad \Gamma;z:\mathrm{T}\vdash f_0:\mathrm{P}\,g_0 \qquad z\notin\mathsf{FV}(f_0^*) \qquad (\mathrm{B}_1\,[u/\pi_1(d)])^*\preceq\mathrm{T}^*}{\Gamma\vdash\mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0,d,\mathrm{P}\,g_0):\mathrm{P}\,g_0}$$

— Toutes les prémisses et la condition de bord $(\mathrm{B}_1\,[u/\pi_1(d)])^*\preceq\mathrm{T}^*$ ont été démontrées plus haut.

— Montrons que $z\notin\mathsf{FV}(f_0^*)$.

— Par correction de l'extraction sur $\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$ nous obtenons $\Gamma^* \vdash \mathsf{Elim}_{\exists}(y.f^*, g^*) : \mathrm{R}^*$, soit, par définition de f_0, $\Gamma^* \vdash f_0^* : \mathrm{R}^*$.

— D'après le lemme de déclaration des variables libres (lemme 3.4.3 page 25) appliqué à $\Gamma^* \vdash f_0^* : \mathrm{R}^*$, nous avons $\mathsf{FV}(f_0^*) \subset \mathsf{DV}(\Gamma^*)$.

— Puisque $\mathsf{DV}(\Gamma^*) = \mathsf{DV}(\Gamma)$ et que $z \notin \mathsf{DV}(\Gamma)$, nous déduisons bien $z \notin \mathsf{FV}(f_0^*)$.

- Montrons finalement, par cumulativité sur $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}\, g_0) : \mathrm{P}\, g_0$, que $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}\, g_0) : \mathrm{R}$:

  — Le lemme du type des types appliqué à $\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$ (le jugement initial) nous indique qu'il existe une sorte s telle que $\Gamma \vdash \mathrm{R} : s$.

  — Nous avons vu que $(\mathrm{P}\, g_0)^* = (\mathrm{P}\, \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}))^* \preceq \mathrm{R}^*$.

- $\rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}}$: Supposons que $\Gamma \vdash \pi_1(\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x : \mathrm{T}\,.\,[\mathrm{U}])) : \mathrm{R}$.
Montrons que $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T}) : \mathrm{R}$.

Comme précédemment, nous avons une étape d'inversion de jugement puis une étape d'applications de règles.

(1) Inversons les règles $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-1)$ puis $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-2)$ à partir du jugement initial.

- Par inversion de $\Gamma \vdash \pi_1(\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x : \mathrm{T}\,.\,[\mathrm{U}])) : \mathrm{R}$, nous avons $\mathrm{A}_1, \mathrm{B}_1$ tels que
  — $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]) : \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1]$.
  — $\mathrm{A}_1^* \preceq \mathrm{R}^*$.
- Par inversion de $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]) : \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1]$, il existe $s, \mathrm{A}_2, \mathrm{B}_2$ tels que
  — $\Gamma \vdash \Sigma x : \mathrm{T}\,.\,[\mathrm{U}] : s$
  — $\Gamma \vdash c : \Sigma x : \mathrm{A}_2\,.\,[\mathrm{B}_2]$
  — $\Gamma; y : \mathrm{B} \vdash f : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]$
  — $y \notin \mathsf{FV}(f^*)$
  — $(\Sigma x : \mathrm{T}\,.\,[\mathrm{U}])^* \preceq (\Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1])^*$, d'où $\mathrm{T}^* \preceq \mathrm{A}_1^*$ par inversion de la cumulativité,
  — $(\mathrm{B}_2\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*$.

(2) Appliquons maintenant les règles $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-1)$, $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-2)$ et $(\mathrm{CUM_A})$ pour construire le jugement final :

- En appliquant $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-1)$ à $\Gamma; y : \mathrm{B} \vdash f : \Sigma x : \mathrm{T}\,.\,[\mathrm{U}]$, nous dérivons $\Gamma; y : \mathrm{B} \vdash \pi_1(f) : \mathrm{T}$.
- Montrons que nous pouvons dériver $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T}) : \mathrm{T}$. Pour cela, montrons qu'il existe une sorte s_{T} telle que nous pouvons appliquer $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-2)$:

$$\frac{\Gamma \vdash \mathrm{T} : s_{\mathrm{T}} \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}_2\,.\,[\mathrm{B}_2] \qquad \Gamma; y : \mathrm{B} \vdash \pi_1(f) : \mathrm{T} \qquad \begin{array}{l} y \notin \mathsf{FV}(\pi_1(f)^*) \\ (\mathrm{B}_2\,[x/\pi_1(c)])^* \preceq \mathrm{B}^* \end{array}}{\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T}) : \mathrm{T}}$$

— Les prémisses $\Gamma \vdash c : \Sigma x : \mathrm{A}_2\,.\,[\mathrm{B}_2]$ et $\Gamma; y : \mathrm{B} \vdash \pi_1(f) : \mathrm{T}$, ainsi que la condition de bord $(\mathrm{B}_2\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*$ ont été obtenues plus haut.

— Nous déduisons $y \notin \mathsf{FV}(\pi_1(f)^*)$ de $y \notin \mathsf{FV}(f^*)$ puisque $\pi_1(f)^* = f^*$.

— Par inversion de $\Gamma \vdash \Sigma x : \mathrm{T}\,.\,[\mathrm{U}] : s$, il existe une sorte s_T telle que $\Gamma \vdash \mathrm{T} : s_\mathrm{T}$.

- Montrons par cumulativité sur $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T}) : \mathrm{T}$ que $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T}) : \mathrm{R}$:
  — d'après le lemme du type des types appliqué à $\Gamma \vdash \pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x : \mathrm{T}\,.\,[\mathrm{U}])) : \mathrm{R}$, il existe une sorte s telle que $\Gamma \vdash \mathrm{R} : s$;
  — par transitivité sur $\mathrm{T}^* \preceq \mathrm{A}_1^*$ et sur $\mathrm{A}_1^* \preceq \mathrm{R}^*$ nous déduisons $\mathrm{T}^* \preceq \mathrm{R}^*$.

- $\rhd^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}}$: Supposons que $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$ et montrons que nous pouvons dériver $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, g, \mathrm{P}), d, \mathrm{P}) : \mathrm{R}$. Nous procédons comme précédemment en deux phases.

(1) Déconstruisons $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$ par double inversion.

- Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) : \mathrm{R}$, nous avons $s, \mathrm{A}_1, \mathrm{B}_1$ tels que
  — $\Gamma \vdash \mathrm{P} : s$
  — $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}) : \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1]$
  — $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$
  — $y \notin \mathsf{FV}(f^*)$
  — $(\mathrm{B}_1\,[x/\pi_1(g)])^* = (\mathrm{B}_1\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*$, où $c = \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U})$.
  — $\mathrm{P}^* \preceq \mathrm{R}^*$.
- Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}) : \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1]$, nous avons $s_0, \mathrm{A}_2, \mathrm{B}_2$ tels que
  — $\Gamma \vdash \mathrm{U} : s_0$
  — $\Gamma \vdash d : \Sigma u : \mathrm{A}_2\,.\,[\mathrm{B}_2]$
  — $\Gamma; z : \mathrm{T} \vdash g : \mathrm{U}$
  — $z \notin \mathsf{FV}(g^*)$
  — $(\mathrm{B}_2\,[u/\pi_1(d)])^* \preceq \mathrm{T}^*$
  — $\mathrm{U}^* \preceq (\Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1])^*$.

(2) Posons $f_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, g, \mathrm{P})$. Nous voulons donc montrer que $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}) : \mathrm{R}$. Pour cela nous allons appliquer deux fois la règle $(\Sigma_{\mathrm{SUB}}-\mathrm{E}-2)$ puis une fois la règle $(\mathrm{CUM_A})$.

- Montrons que $\Gamma; z : \mathrm{T} \vdash f_0 : \mathrm{P}$. Pour cela, montrons que nous pouvons dériver

$$\frac{\Gamma; z : \mathrm{T} \vdash \mathrm{P} : s \qquad \Gamma; z : \mathrm{T} \vdash g : \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1] \qquad \Gamma; z : \mathrm{T}; y : \mathrm{B} \vdash f : \mathrm{P} \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (\mathrm{B}_1\,[x/\pi_1(g)])^* \preceq \mathrm{B}^* \end{array}}{\Gamma; z : \mathrm{T} \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, g, \mathrm{P}) : \mathrm{P}}$$

— Les conditions de bord $y \notin \mathsf{FV}(f^*)$ et $(\mathrm{B}_1\,[x/\pi_1(g)])^* \preceq \mathrm{B}^*$ ont été démontrées plus haut.
— Puisque $\Gamma; z : \mathrm{T} \vdash$, nous montrons par affaiblissement sur $\Gamma \vdash \mathrm{P} : s$ que $\Gamma; z : \mathrm{T} \vdash \mathrm{P} : s$.
— Par cumulativité sur $\Gamma; z : \mathrm{T} \vdash g : \mathrm{U}$, nous montrons que $\Gamma; z : \mathrm{T} \vdash g : \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1]$.

— Nous montrons $\Gamma; z : \mathrm{T}; y : \mathrm{B} \vdash$ à partir de $\Gamma; z : \mathrm{T} \vdash$ et $\Gamma; y : \mathrm{B} \vdash$,[13] et en déduisons par affaiblissement sur $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$ que $\Gamma; z : \mathrm{T}; y : \mathrm{B} \vdash f : \mathrm{P}$.

- Montrons que $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}) : \mathrm{P}$. Pour cela montrons que nous pouvons appliquer $(\Sigma_{\mathrm{SUB}-\mathrm{E}-2})$ et dériver

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash d : \Sigma u : \mathrm{A}_2 . [\mathrm{B}_2] \qquad \Gamma; z : \mathrm{T} \vdash f_0 : \mathrm{P} \qquad \begin{array}{l} z \notin \mathsf{FV}(f_0^*) \\ (\mathrm{B}_2[u/\pi_1(d)])^* \preceq \mathrm{T}^* \end{array}}{\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}) : \mathrm{P}}$$

— Les trois prémisses et la condition de bord $(\mathrm{B}_2[u/\pi_1(d)])^* \preceq \mathrm{T}^*$ ont été démontrées plus haut.

— Montrons que $z \notin \mathsf{FV}(f_0^*)$.

— Par définition de f_0, $f_0^* = f^*$, d'où $\mathsf{FV}(f_0^*) = \mathsf{FV}(f^*)$.

— D'après le lemme 2.1.1 page 116, puisque f est typable, $\mathsf{FV}(f^*) \subset \mathsf{FV}(f)$, soit $\mathsf{FV}(f_0^*) \subset \mathsf{FV}(f)$.

— Le lemme de déclaration des variables libres (lemme 3.4.3 page 25) appliqué à $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$ nous indique que $\mathsf{FV}(f) \subset \mathsf{DV}(\Gamma; y : \mathrm{B})$, d'où $z \notin \mathsf{FV}(f)$, puis $z \notin \mathsf{FV}(f_0^*)$.

- Nous montrons finalement $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}) : \mathrm{R}$ par cumulativité sur $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([z{:}\mathrm{T}].f_0, d, \mathrm{P}) : \mathrm{P}$.

□

3.2.3. *Constructeurs de type et éliminateur simplifié.*

3.2.4. Lemme (Correction de $\rhd^{\mathrm{A}}_{\Gamma,\ \Box-\mathsf{Simpl}}$)**.** *Si*

$$\frac{\begin{cases} \Gamma^*; y : \mathrm{B}^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{T}} \\ \Gamma^*; y : \mathrm{B}^*; z : \mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{U}} \end{cases} \qquad \begin{cases} \mathrm{T}_0 = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{T}, c, \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{T}})) \\ \mathrm{U}_0 = \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{U}, c, \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{U}})) \end{cases}}{\mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\Box z : \mathrm{T}.\mathrm{U}, c, \mathrm{P}) \rhd^{\mathrm{A}}_{\Gamma,\ \Box-\mathsf{Simpl}} \Box z : \mathrm{T}_0.\mathrm{U}_0}$$

et si $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\Box z : \mathrm{T}.\mathrm{U}, c, \mathrm{P}) : \mathrm{R}$ *alors* $\Gamma \vdash \Box z : \mathrm{T}_0.\mathrm{U}_0 : \mathrm{R}$.

DÉMONSTRATION. Nous procédons une fois de plus en deux étapes.

(1) Déconstruisons le jugement de départ.

- Par inversion de $\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].\Box z : \mathrm{T}.\mathrm{U}, c, \mathrm{P}) : \mathrm{R}$, il existe $\mathrm{A}_0, \mathrm{B}_0$ tels que
  - $\Gamma \vdash c : \Sigma x : \mathrm{A}_0 . [\mathrm{B}_0]$
  - $\Gamma; y : \mathrm{B} \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{P}$
  - $y \notin \mathsf{FV}(\Box^* z : \mathrm{T}^*.\mathrm{U}^*)$
  - $(\mathrm{B}_0[x/\pi_1(c)])^* \preceq \mathrm{B}^*$
  - $\mathrm{P}^* \preceq \mathrm{R}^*$.
- Par inversion de $\Gamma; y : \mathrm{B} \vdash \Box z : \mathrm{T}.\mathrm{U} : \mathrm{P}$, il existe des sortes s_1, s_2, s_3 telles que
  - $\Gamma; y : \mathrm{B} \vdash \mathrm{T} : s_1$
  - $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash \mathrm{U} : s_2$

13. Preuve similaire à celle de $\Gamma; z : \mathrm{T}; x : \mathrm{A}; y : \mathrm{B} \vdash$ dans le cas $\rhd^{\mathrm{A}}_{\exists-\mathsf{Simpl}}$.

- $(s_1, s_2, s_3) \in \mathbf{Rule}_{\square}$
- $s_3 \preceq \mathrm{P}^*$.

(2) Construisons le jugement d'arrivée.

- Montrons que $\mathtt{NF\_BI}(\mathrm{S}'_\mathrm{T})$ et $\mathtt{NF\_BI}(\mathrm{S}'_\mathrm{U})$ existent et que ce sont des sortes telles que $\mathtt{NF\_BI}(\mathrm{S}'_\mathrm{T}) \preceq s_1$, $\mathtt{NF\_BI}(\mathrm{S}'_\mathrm{U}) \preceq s_2$.
  - Puisque $\Gamma; y : \mathrm{B} \vdash$ et $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash$, le point (2) du corollaire 3.3.3 page 159 prouve l'existence de $\mathtt{NF\_BI}(\mathrm{S}'_\mathrm{T})$ et $\mathtt{NF\_BI}(\mathrm{S}'_\mathrm{U})$.
  - Par complétude de l'inférence de type extrait (proposition 4.2.1 page 161) et par unicité du type extrait inféré, nous déduisons
    - de $\Gamma; y : \mathrm{B} \vdash \mathrm{T} : s_1, \Gamma^*; y : \mathrm{B}^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_\mathrm{T}$
    - et de $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash \mathrm{U} : s_2, \Gamma^*; y : \mathrm{B}^*; z : \mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_\mathrm{U}$,

    que $\mathrm{S}'_\mathrm{T} \preceq s_1$ et $\mathrm{S}'_\mathrm{U} \preceq s_2$.
  - Nous concluons grâce au point (1) du lemme 4.1.3 page 160.
- Nous pouvons donc poser $s'_\mathrm{T} = \mathtt{NF\_BI}(\mathrm{S}'_\mathrm{T}) \in \mathbf{Sort}$ et $s'_\mathrm{U} = \mathtt{NF\_BI}(\mathrm{S}'_\mathrm{U}) \in \mathbf{Sort}$. Montrons que $\Gamma; y : \mathrm{B} \vdash \mathrm{T} : s'_\mathrm{T}$ et $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash \mathrm{U} : s'_\mathrm{U}$.
  - Puisque $\Gamma; y : \mathrm{B} \vdash$ et $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash$, par correction de l'algorithme d'inférence de type extrait, il existe $\mathrm{S}_\mathrm{T}, \mathrm{S}_\mathrm{U}$ tels que
    - $\Gamma; y : \mathrm{B} \vdash \mathrm{T} : \mathrm{S}_\mathrm{T}$ et $\mathrm{S}^*_\mathrm{T} = \mathrm{S}'_\mathrm{T}$;
    - $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash \mathrm{U} : \mathrm{S}_\mathrm{U}$ et $\mathrm{S}^*_\mathrm{U} = \mathrm{S}'_\mathrm{U}$.
  - Nous obtenons par cumulativité les jugements souhaités.
- Montrons que $\Gamma \vdash \mathrm{T}_0 : s'_\mathrm{T}$. Puisque $\mathrm{T}_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{T}, c, s'_\mathrm{T})$, il suffit de dériver

$$\frac{\Gamma \vdash s'_\mathrm{T} : s_\mathrm{T} \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}_0 . [\mathrm{B}_0] \qquad \Gamma; y : \mathrm{B} \vdash \mathrm{T} : s'_\mathrm{T} \qquad \begin{array}{l} y \notin \mathsf{FV}(\mathrm{T}^*) \\ (\mathrm{B}_0\,[x/\pi_1(c)])^* \preceq \mathrm{B}^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{T}, c, s'_\mathrm{T}) : s'_\mathrm{T}}$$

  - Montrons que les conditions de bord sont satisfaites :
    - $(\mathrm{B}_0\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*$ est déjà prouvé ;
    - $y \notin \mathsf{FV}(\mathrm{T}^*)$ découle de $y \notin \mathsf{FV}(\square^* z : \mathrm{T}^* . \mathrm{U}^*)$.
  - Montrons que les prémisses sont dérivables :
    - $\Gamma \vdash c : \Sigma[x{:}\mathrm{A}_0] . \mathrm{B}_0$ et $\Gamma; y : \mathrm{B} \vdash \mathrm{T} : s'_\mathrm{T}$ sont déjà prouvées ;
    - puisque s'_T est une sorte, que Γ est bien formé et par absence de sorte maximale de **Axiom**, il existe une sorte s_T telle que $\Gamma \vdash s'_\mathrm{T} : s_\mathrm{T}$.
- Montrons que $\Gamma; z : \mathrm{T}_0 \vdash \mathrm{U}_0 : s'_\mathrm{U}$. En dépliant la définition de U_0, il suffit de montrer que nous pouvons dériver

$$\frac{\Gamma; z : \mathrm{T}_0 \vdash s'_\mathrm{U} : s_\mathrm{U} \qquad \Gamma; z : \mathrm{T}_0 \vdash c : \Sigma x : \mathrm{A}_0 . [\mathrm{B}_0] \qquad \Gamma; z : \mathrm{T}_0; y : \mathrm{B} \vdash \mathrm{U} : s'_\mathrm{U} \qquad \begin{array}{l} y \notin \mathsf{FV}(\mathrm{U}^*) \\ (\mathrm{B}_0\,[x/\pi_1(c)])^* \preceq \mathrm{B}^* \end{array}}{\Gamma; z : \mathrm{T}_0 \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{U}, c, s'_\mathrm{U}) : s'_\mathrm{U}}$$

  - Les conditions de bord sont satisfaites :
    - $(\mathrm{B}_0\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*$ est déjà prouvé ;

— $y \notin \mathsf{FV}(\mathrm{U}^*)$ découle de $y \notin \mathsf{FV}(\Box^* z : \mathrm{T}^* . \mathrm{U}^*)$, puisque $y \neq z$.

- Montrons qu'il existe une sorte s_U telle que $\Gamma; z : \mathrm{T}_0 \vdash s'_\mathrm{U} : s_\mathrm{U}$:
  — Par absence de sorte maximale dans **Axiom**, il existe une sorte s_U telle que $(s'_\mathrm{U}, s_\mathrm{U}) \in$ **Axiom**.
  — En appliquant (WF-S$_\mathrm{A}$) à $\Gamma \vdash \mathrm{T}_0 : s'_\mathrm{T}$, nous dérivons $\Gamma; z : \mathrm{T}_0 \vdash$.
  — Nous concluons alors en appliquant (SORT$_\mathrm{A}$).
- Par affaiblissement de $\Gamma \vdash c : \Sigma[x : \mathrm{A}_0] . \mathrm{B}_0$, et puisque nous avons vu que $\Gamma; z : \mathrm{T}_0 \vdash$, nous déduisons $\Gamma; z : \mathrm{T}_0 \vdash c : \Sigma[x : \mathrm{A}_0] . \mathrm{B}_0$.
- Montrons $\Gamma; z : \mathrm{T}_0; y : \mathrm{B} \vdash \mathrm{U} : s'_\mathrm{U}$.
  — Nous avons vu que $\Gamma; y : \mathrm{B}; z : \mathrm{T} \vdash \mathrm{U} : s'_\mathrm{U}$.
  — Par conversion de contexte, nous en déduisons $\Gamma; y : \mathrm{B}; z : \mathrm{T}_0 \vdash \mathrm{U} : s'_\mathrm{U}$ puisque :
    — $\mathrm{T}_0^* = \mathrm{T}^*$;
    — nous montrons, comme nous l'avons fait plus haut, $\Gamma; y : \mathrm{B}; z : \mathrm{T}_0 \vdash$ à partir de $\Gamma; y : \mathrm{B} \vdash$ et $\Gamma; z : \mathrm{T}_0 \vdash$.
  — Par échange de variables (lemme 2.3.12 page 125), puisque $y \notin \mathsf{FV}(\mathrm{T}_0)$, nous en déduisons $\Gamma; z : \mathrm{T}_0; y : \mathrm{B} \vdash \mathrm{U} : s'_\mathrm{U}$.

- Montrons que $\Gamma \vdash \Box z : \mathrm{T}_0 . \mathrm{U}_0 : \mathrm{R}$.
  - Par complétude de **Rule**$_\Box$, il existe une sorte s telle que $(s'_\mathrm{T}, s'_\mathrm{U}, s) \in$ **Rule**$_\Box$.
  - Par application de ($\Box_\mathrm{A}$-FORM), nous dérivons donc $\Gamma \vdash \Box z : \mathrm{T}_0 . \mathrm{U}_0 : s$.
  - Nous concluons en remplaçant s par R par cumulativité :
    — puisque $s'_\mathrm{T} \preceq s_1$ et $s'_\mathrm{U} \preceq s_2$, nous avons $s \preceq s_3$ par compatibilité de **Rule**$_\Box$ avec la cumulativité, d'où $s \preceq \mathrm{R}^*$ par transitivité.
    — d'après le lemme du type des types appliqué à $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y : \mathrm{B}].\Box z : \mathrm{T} . \mathrm{U}, c, \mathrm{P}) : \mathrm{R}$, il existe une sorte s_0 telle que $\Gamma \vdash \mathrm{R} : s_0$.

□

3.2.4. *Sortes et éliminateur simplifié.*

3.2.5. Lemme (Correction pour $\rhd^{\mathrm{A}}_{s-\mathsf{Simpl}}$)**.** *Si* $\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y : \mathrm{B}].s, c, \mathrm{P}) : \mathrm{R}$ *alors* $\Gamma \vdash s : \mathrm{R}$.

DÉMONSTRATION. Nous procédons comme précédemment par double inversion (π_2 puis la sorte) puis en appliquant (SORT$_\mathrm{A}$) et (CUM$_\mathrm{A}$). □

3.2.5. *Réunion des règles de réductions.*

3.2.6. Proposition (Correction pour le typage)**.** *Si* $\Gamma \vdash \mathrm{M}_0 : \mathrm{R}$ *et si* $\mathrm{M}_0 \rhd^{\mathrm{A}}_\Gamma \mathrm{M}_1$*, alors* $\Gamma \vdash \mathrm{M}_1 : \mathrm{R}$.

DÉMONSTRATION. Conséquence des lemmes 3.2.1, 3.2.2, 3.2.3, 3.2.4 et et 3.2.5. □

3.2.6. *Réduction dans un sous-terme.*

3.2.7. Proposition (Correction pour le typage)**.** *Si* $\Gamma \vdash \mathrm{N}_0 : \mathrm{R}$ *et* $\mathrm{N}_0 \to^h_\Gamma \mathrm{N}_1$ *alors* $\Gamma \vdash \mathrm{N}_1 : \mathrm{R}$.

DÉMONSTRATION. Par induction sur la dérivation de $\mathrm{N}_0 \to^h_\Gamma \mathrm{N}_1$. Le cas de base (HRED-BASE) se résout grâce à la proposition 3.2.6 page 199. Pour les cas inductifs, le schéma suivant s'applique :

(1) inversion du jugment $\Gamma \vdash \mathrm{N}_0 : \mathrm{R}$;

(2) application de l'hypothèse d'induction à la prémisse qui type le sous-terme de N_0 se réduisant ;

(3) application de la règle de typage permettant de typer N_1 ;

(4) application de la règle (CUM$_A$) pour changer le type et obtenir $\Gamma \vdash N_1 : R$.

Ce schéma est commun à tous les cas. Aussi nous ne traitons que deux cas en détails : celui de (HRED-∃-Elim), qui est le plus complexe, et celui de (HRED-Simpl-BRANCH), qui se distingue par le changement de contexte dans la réduction du sous-terme.

- (HRED-∃-Elim) Si $\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, M_0, P) : R$ et $M_0 \to^h_\Gamma M_1$, montrons que $\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, M_1, P) : R$.
  (1) Par inversion de $\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, M_0, P) : R$, il existe des sortes s, s' telles que
    - $\Gamma \vdash \Sigma[x{:}A]\,.\,B : s'$
    - $\Gamma \vdash P : \Sigma[x{:}A]\,.\,B \to s$
    - $\Gamma \vdash M_0 : \Sigma[x{:}A]\,.\,B$
    - $\Gamma; x : A; y : B \vdash f : P\,(x, y)_{\Sigma x:A.B}$
    - $x \notin \mathsf{FV}(f^*)$
    - $(P\,M_0)^* \preceq R^*$.
  (2) Par hypothèse d'induction, nous avons $\Gamma \vdash M_1 : \Sigma[x{:}A]\,.\,B$.
  (3) Nous pouvons donc appliquer ($\Sigma_\exists - E$) et dériver $\Gamma \vdash \mathsf{Elim}_{I_L}([x{:}A].(y{:}B).f, M_1, P) : P\,M_1$.
  (4) Pour conclure par cumulativité, il suffit de montrer que $(P\,M_1)^* \preceq R^*$.
    - D'après la proposition 3.1.2, nous avons $M_0^* \cong M_1^*$, d'où $(P\,M_0)^* \cong (P\,M_1)^*$.
    - Nous en déduisons $(P\,M_1)^* \preceq (P\,M_0)^* \preceq R^*$.
- (HRED-Simpl-BRANCH) Si $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_0, c, P) : R$ et $M_0 \to^h_{\Gamma;y:B} M_1$, montrons que $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_1, c, P) : R$.
  (1) Par inversion de $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_0, c, P) : R$, nous avons s, A_1, B_1 tels que
    - $\Gamma \vdash P : s$
    - $\Gamma \vdash c : \Sigma x : A_1\,.\,[B_1]$
    - $\Gamma; y : B \vdash M_0 : P$
    - $y \notin \mathsf{FV}(M_0^*)$
    - $(B_1\,[x/\pi_1(c)])^* \preceq B^*$
    - $P^* \preceq R^*$.
  (2) Puisque le contexte du jugement $\Gamma; y : B \vdash M_0 : P$ est bien le contexte de la réduction $M_0 \to^h_{\Gamma;y:B} M_1$, nous pouvons appliquer le résultat par induction, d'où $\Gamma; y : B \vdash M_1 : P$.
  (3) Montrons que nous pouvons appliquer ($\Sigma_{SUB}-E-2$) et dériver $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_1, c, P) : P$:

$$\frac{\Gamma \vdash P : s \qquad \Gamma \vdash c : \Sigma x : A_1\,.\,[B_1] \qquad \Gamma; y : B \vdash M_1 : P \qquad \begin{array}{l} y \notin \mathsf{FV}(M_1^*) \\ (B_1\,[x/\pi_1(c)])^* \preceq B^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_1, c, P) : P}$$

  Seule la condition de bord $y \notin \mathsf{FV}(M_1^*)$ reste à démontrer :
    - D'après la proposition 3.1.2, nous avons $M_0^* \twoheadrightarrow_{\beta\iota} M_1^*$.
    - D'après le lemme 2.2.2 page 12, nous avons $\mathsf{FV}(M_1^*) \subset \mathsf{FV}(M_0^*)$.
    - De $y \notin \mathsf{FV}(M_0^*)$, nous déduisons alors $y \notin \mathsf{FV}(M_1^*)$.
  (4) Enfin, par cumulativité, nous obtenons $\Gamma \vdash \mathsf{Elim}^s_{I_R}([y{:}B].M_1, c, P) : R$.

□

4. Algorithme de réduction

Nous définissons dans cette section l'algorithme de réduction. Nous montrons également que cet algorithme vérifie bien les spécifications souhaitées.

4.1. Conjecture. Pour tout contexte Γ, nous admettons la normalisation forte pour $\rightarrow_\Gamma^h$ des termes typés sous Γ.

4.1.1. Conjecture (Normalisation forte)**.** *Si* $\Gamma \vdash M : T$, *alors il n'existe pas de séquence infinie de réductions pour* $\rightarrow_\Gamma^h$ *commençant par* M.

4.2. Définitions.

4.2.1. DÉFINITION (Forme normale de tête faible). Soient Γ un contexte annoté et M un terme annoté. M est en *forme normale de tête faible* s'il est en forme normale pour $\rightarrow_\Gamma^h$.

Nous parlons de forme normale de tête faible car $\rightarrow_\Gamma^h$ est une réduction de tête faible.

Pour tout contexte Γ, d'après la conjecture 4.1.1, toute stratégie de réduction pour $\rightarrow_\Gamma^h$ termine pour les termes typés dans Γ (normalisation forte). Nous pouvons donc définir un algorithme de mise en forme normale de tête.

4.2.2. DÉFINITION (Algorithme de mise en forme normale). Soit Γ un contexte bien formé de AICC_Σ. Nous notons hnf_Γ un algorithme de mise en forme normale implémentant une stratégie d'appel par valeur externe droite (*right outermost*). Si $\Gamma \vdash M : T$, le résultat de l'algorithme appliqué à M est noté $\mathsf{hnf}_\Gamma(M)$.

4.2.3. REMARQUE. $\rightarrow_\Gamma^h$ n'étant pas confluente, le résultat de l'algorithme dépend de la stratégie de réduction choisie. Les propriétés de l'algorithme que nous allons montrer sont néanmoins valables quelle que soit la stratégie de réduction choisie.

4.3. Correction.

4.3.1. Proposition (Correction)**.** *Si* $\Gamma \vdash M : T$ *alors*

(1) $M^* \twoheadrightarrow_{\beta\iota} \mathsf{hnf}_\Gamma(M)^*$

(2) $\Gamma \vdash \mathsf{hnf}_\Gamma(M) : T$.

DÉMONSTRATION. Par définition de $\mathsf{hnf}_\Gamma(M)$, nous avons $M \twoheadrightarrow_\Gamma^h \mathsf{hnf}_\Gamma(M)$. Par une récurrence sur le nombre de pas de réduction, nous déduisons (1) de la proposition 3.1.2 et (2) de la proposition 3.2.7. □

Le corollaire suivant est utile pour la preuve de la correction de l'algorithme d'inférence.

4.3.2. Corollaire (Forme normale et changement de type)**.** *Si* $\Gamma \vdash M : T$ *alors* $\mathsf{hnf}_\Gamma(T)$ *existe et* $\Gamma \vdash M : \mathsf{hnf}_\Gamma(T)$.

DÉMONSTRATION.

- D'après la proposition du type des types, il existe une sorte s telle que $\Gamma \vdash T : s$. L'existence de $\mathsf{hnf}_\Gamma(T)$ en découle.
- D'après la proposition 4.3.1, $T^* \twoheadrightarrow_{\beta\iota} \mathsf{hnf}_\Gamma(T)^*$ et $\Gamma \vdash \mathsf{hnf}_\Gamma(T) : s$.
- Nous concluons alors par cumulativité.

□

4.4. Complétude.

4.4.1. Proposition. *Si* $\Gamma \vdash R : s_R$ *et s'il existe* T', U' *termes de ICC*$_\Sigma$ *tels que* $R^* \preceq \square^* x : T' . U'$ *ou* $\square^* x : T' . U' \preceq R^*$ *alors il existe* T, U *dans* AICC$_\Sigma$ *tels que* $\mathsf{hnf}_\Gamma(R) = \square x : T . U$.

DÉMONSTRATION. Montrons qu'il existe T, U dans AICC$_\Sigma$ tels que $\mathsf{hnf}_\Gamma(R) = \square x : T . U$.

- D'après la proposition 4.3.1, nous avons $\Gamma \vdash \mathsf{hnf}_\Gamma(R) : s_R$ et $\mathsf{hnf}_\Gamma(R)^* \preceq \square^* x : T' . U'$ ou $\square^* x : T' . U' \preceq \mathsf{hnf}_\Gamma(R)^*$.
- En appliquant les résultats de classification établis précédemment à $\Gamma \vdash \mathsf{hnf}_\Gamma(R) : s_R$ (corollaire 2.2.5 page 120 et lemme 2.3.1), $\mathsf{hnf}_\Gamma(R)$ est
  (1) un constructeur de type ou une sorte ;
  (2) ou un terme neutre (y compris une variable) ;
  (3) ou réductible pour $\rightarrow^h_\Gamma$;
  (4) ou une paire implicite projetée.
- $\mathsf{hnf}_\Gamma(R)$ est en forme normale pour $\rightarrow^h_\Gamma$ donc n'est pas réductible.
- $\mathsf{hnf}_\Gamma(R)$ n'est pas une paire implicite projetée car cela contredirait le lemme 2.2.3.
- $\mathsf{hnf}_\Gamma(R)$ n'est pas un terme neutre de AICC$_\Sigma$ car sinon $\mathsf{hnf}_\Gamma(R)^*$ serait un terme neutre de ICC$_\Sigma$ (lemme 2.1.5) et alors le point 5 du lemme 2.1.2 page 179 serait contredit.
- $\mathsf{hnf}_\Gamma(R)$ n'est pas une sorte ni un constructeur de type de nature analogue à celle de $\square^* x : T' . U'$ car sinon $\mathsf{hnf}_\Gamma(R)^*$ le serait également et alors le lemme 2.3.15 page 18 serait contredit.

□

5. Algorithme

5.1. Définitions.

5.1.1. DÉFINITION (Jugement d'inférence de type). Soient Γ un contexte de AICC$_\Sigma$, M, T des termes de AICC$_\Sigma$. Le *jugement d'inférence de type*, noté $\Gamma \vdash M \Uparrow T$, signifie que le terme T est le type inféré de M dans le contexte Γ.

5.1.2. DÉFINITION (Algorithme d'inférence de type (annoté)). Les règles d'inférence définissant l'algorithme d'inférence de type sont détaillées dans les figures 28 et 29.

De même que pour l'algorithme d'inférence de type extrait, les règles de l'algorithme d'inférence de type annoté sont dirigées par la syntaxe. Nous prouvons de même l'unicité du type inféré et la terminaison de l'algorithme.

5.2. Propriétés.

5.2.1. *Correction.* Tout d'abord un résultat utile.

5.2.1. Lemme. *Si* $\Gamma \vdash M : S$ *avec* $\mathtt{NF\_BI}(S^*) \in \textbf{\textit{Sort}}$ *alors* $\Gamma \vdash M : \mathtt{NF\_BI}(S^*)$.

DÉMONSTRATION. Par cumulativité sur $\Gamma \vdash M : S$. Conséquence de la correction de `NF_BI`. □

5.2.2. Proposition (Correction de l'inférence de type)**.** *Soient* Γ *un contexte,* M, T *des termes. Si* $\Gamma \vdash$ *et* $\Gamma \vdash M \Uparrow T$ *sont dérivables, alors* $\Gamma \vdash M : T$ *l'est également.*

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma \vdash M \Uparrow T$ en considèrant la dernière règle appliquée.

(1) Variable

$$\frac{(x:\mathrm{T})\in\Gamma}{\Gamma\vdash x\Uparrow \mathrm{T}}\ (\textsc{Inf-Var})$$

(2) Sorte

$$\frac{}{\Gamma\vdash s_1\Uparrow \mathtt{Axiom}(s_1)}\ (\textsc{Inf-Sort})$$

(3) Produit explicite

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*)=s_1\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{U}\Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*)=s_2\in\mathsf{Sort}}{\Gamma\vdash \Pi x:\mathrm{T}.\,\mathrm{U}\Uparrow \mathtt{Rule}(s_1,s_2)}\ (\textsc{Inf-E-Prod})$$

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S} \qquad \mathtt{NF\_BI}(\mathrm{S}^*)\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{M}\Uparrow \mathrm{U}}{\Gamma\vdash \lambda x{:}\mathrm{T}.\,\mathrm{M}\Uparrow \Pi x:\mathrm{T}.\,\mathrm{U}}\ (\textsc{Inf-E-Lam})$$

$$\frac{\Gamma\vdash \mathrm{M}\Uparrow \mathrm{R}_1 \qquad \mathsf{hnf}_\Gamma(\mathrm{R}_1)=\Pi x:\mathrm{T}.\,\mathrm{U} \qquad \Gamma\vdash \mathrm{N}\Uparrow \mathrm{R}_2 \qquad \mathtt{cumu}(\mathrm{R}_2^*,\mathrm{T}^*)}{\Gamma\vdash \mathrm{M}\,\mathrm{N}\Uparrow \mathrm{U}[x/\mathrm{N}]}\ (\textsc{Inf-E-App})$$

(4) Produit implicite

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*)=s_1\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{U}\Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*)=s_2\in\mathsf{Sort}}{\Gamma\vdash \Pi[x:\mathrm{T}].\,\mathrm{U}\Uparrow \mathtt{Rule}(s_1,s_2)}\ (\textsc{Inf-I-Prod})$$

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S} \qquad \mathtt{NF\_BI}(\mathrm{S}^*)\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{M}\Uparrow \mathrm{U} \qquad x\notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma\vdash \lambda[x:\mathrm{T}].\,\mathrm{M}\Uparrow \Pi[x:\mathrm{T}].\,\mathrm{U}}\ (\textsc{Inf-I-Lam})$$

$$\frac{\Gamma\vdash \mathrm{M}\Uparrow \mathrm{R}_1 \qquad \mathsf{hnf}_\Gamma(\mathrm{R}_1)=\Pi[x:\mathrm{T}].\,\mathrm{U} \qquad \Gamma\vdash \mathrm{N}\Uparrow \mathrm{R}_2 \qquad \mathtt{cumu}(\mathrm{R}_2^*,\mathrm{T}^*)}{\Gamma\vdash \mathrm{M}\,[\mathrm{N}]\Uparrow \mathrm{U}[x/\mathrm{N}]}\ (\textsc{Inf-I-App})$$

FIGURE 28. Règles d'inférence du jugement d'inférence de type (Variable, sorte et produits)

- (Inf-Var) : immédiat en appliquant la règle ($\mathrm{Var_A}$).
- (Inf-Sort) : par correction de `Axiom`, nous avons $(s_1,\mathtt{Axiom}(s_1))\in\mathbf{Axiom}$ et pouvons donc appliquer ($\mathrm{Sort_A}$).
- Les règles pour les constructeurs de type sont similaires. Traitons la règle générique. Si

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*)=s_1\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{U}\Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*)=s_2\in\mathsf{Sort}}{\Gamma\vdash \Box x:\mathrm{T}.\,\mathrm{U}\Uparrow \mathtt{Rule}_\Box(s_1,s_2)}\ (\textsc{Inf-CstrTyp})$$

  Montrons que nous pouvons appliquer ($\Box_\mathrm{A}$-Form) et dériver

$$\frac{\Gamma\vdash \mathrm{T}:s_1 \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{U}:s_2 \qquad (s_1,s_2,\mathtt{Rule}_\Box(s_1,s_2))\in\mathbf{Rule}_\Box}{\Gamma\vdash \Box x:\mathrm{T}.\,\mathrm{U}:\mathtt{Rule}_\Box(s_1,s_2)}$$

  - Montrons que $\Gamma\vdash \mathrm{T}:s_1$.
    - Par induction sur $\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S}_1$, nous obtenons $\Gamma\vdash \mathrm{T}:\mathrm{S}_1$.
    - Nous concluons alors en appliquant le lemme 5.2.1 à $\Gamma\vdash \mathrm{T}:\mathrm{S}_1$ et à $s_1=\mathtt{NF\_BI}(\mathrm{S}_1^*)\in\mathsf{Sort}$.
  - Montrons que $\Gamma;x:\mathrm{T}\vdash \mathrm{U}:s_2$.
    - En appliquant ($\mathrm{Wf\text{-}S_A}$) à $\Gamma\vdash \mathrm{T}:s_1$, nous obtenons $\Gamma;x:\mathrm{T}\vdash$.

(5) Somme explicite

$$\frac{\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*) = s_1 \in \mathbf{Sort} \qquad \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*) = s_2 \in \mathbf{Sort}}{\Gamma \vdash \Sigma x:\mathrm{A}.\mathrm{B} \Uparrow \mathtt{RulePrime}(s_1, s_2)}\ (\textsc{Inf-E-}\Sigma)$$

$$\frac{\begin{cases} \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_\mathrm{A} \\ \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow^\mathrm{I} \mathrm{S}_\mathrm{B} \\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*), \mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*) \in \mathbf{Sort} \end{cases} \quad \begin{cases} \Gamma \vdash a \Uparrow \mathrm{R}_1 \\ \Gamma \vdash b \Uparrow \mathrm{R}_2 \end{cases} \quad \begin{cases} \mathtt{cumu}(\mathrm{R}_1^*, \mathrm{A}^*) \\ \mathtt{cumu}(\mathrm{R}_2^*, (\mathrm{B}[x/a])^*) \end{cases}}{\Gamma \vdash (a,b)_{\Sigma x:\mathrm{A}.\mathrm{B}} \Uparrow \Sigma x:\mathrm{A}.\mathrm{B}}\ (\textsc{Inf-E-Pair})$$

$$\frac{\begin{cases} \Gamma \vdash \mathrm{P} \Uparrow \mathrm{R}_1 \\ \Gamma \vdash c \Uparrow \mathrm{R}_2 \\ \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_\mathrm{A} \\ \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow^\mathrm{I} \mathrm{S}_\mathrm{B} \\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*), \mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*) \in \mathbf{Sort} \\ \Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash f \Uparrow \mathrm{R}_3 \end{cases} \quad \begin{cases} \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Pi x:\mathrm{T}.\mathrm{U} \\ \mathtt{NF\_BI}(\mathrm{U}^*) \in \mathbf{Sort} \\ \mathsf{hnf}_\Gamma(\mathrm{R}_2) = \Sigma x:\mathrm{A}_2.\mathrm{B}_2 \\ \mathtt{cumu}((\Sigma x:\mathrm{A}.\mathrm{B})^*, \mathrm{T}^*) \\ \mathtt{cumu}((\Sigma x:\mathrm{A}_2.\mathrm{B}_2)^*, (\Sigma x:\mathrm{A}.\mathrm{B})^*) \\ \mathtt{cumu}(\mathrm{R}_3^*, (\mathrm{P}(x,y)_{\Sigma x:\mathrm{A}.\mathrm{B}})^*) \end{cases}}{\Gamma \vdash \mathsf{Elim}_\mathrm{E}((x:\mathrm{A}).(y:\mathrm{B}).f, c, \mathrm{P}) \Uparrow \mathrm{P}\,c}\ (\textsc{Inf-E-Elim})$$

(6) Somme implicite gauche

$$\frac{\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*) = s_1 \in \mathbf{Sort} \qquad \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*) = s_2 \in \mathbf{Sort}}{\Gamma \vdash \Sigma[x:\mathrm{A}].\mathrm{B} \Uparrow \mathtt{RulePrime}(s_1, s_2)}\ (\textsc{Inf-I-}\Sigma\textsc{-L})$$

$$\frac{\begin{cases} \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_\mathrm{A} \\ \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow^\mathrm{I} \mathrm{S}_\mathrm{B} \\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*), \mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*) \in \mathbf{Sort} \end{cases} \quad \begin{cases} \Gamma \vdash a \Uparrow \mathrm{R}_1 \\ \Gamma \vdash b \Uparrow \mathrm{R}_2 \end{cases} \quad \begin{cases} \mathtt{cumu}(\mathrm{R}_1^*, \mathrm{A}^*) \\ \mathtt{cumu}(\mathrm{R}_2^*, (\mathrm{B}[x/a])^*) \end{cases}}{\Gamma \vdash ([a], b)_{\Sigma[x\mathrm{A}].\mathrm{B}} \Uparrow \Sigma[x:\mathrm{A}].\mathrm{B}}\ (\textsc{Inf-I-Pair-L})$$

$$\frac{\begin{cases} \Gamma \vdash \mathrm{P} \Uparrow \mathrm{R}_1 \\ \Gamma \vdash c \Uparrow \mathrm{R}_2 \\ \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_\mathrm{A} \\ \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow^\mathrm{I} \mathrm{S}_\mathrm{B} \\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*), \mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*) \in \mathbf{Sort} \\ \Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash f \Uparrow \mathrm{R}_3 \end{cases} \quad \begin{cases} \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Pi x:\mathrm{T}.\mathrm{U} \\ \mathtt{NF\_BI}(\mathrm{U}^*) \in \mathbf{Sort} \\ \mathsf{hnf}_\Gamma(\mathrm{R}_2) = \Sigma[x:\mathrm{A}_2].\mathrm{B}_2 \\ \mathtt{cumu}((\Sigma[x:\mathrm{A}].\mathrm{B})^*, \mathrm{T}^*) \\ \mathtt{cumu}((\Sigma[x:\mathrm{A}_2].\mathrm{B}_2)^*, (\Sigma[x:\mathrm{A}].\mathrm{B})^*) \\ \mathtt{cumu}(\mathrm{R}_3^*, (\mathrm{P}([x], y)_{\Sigma[x\mathrm{A}].\mathrm{B}})^*) \end{cases} \quad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{\mathrm{I}_\mathrm{L}}([x:\mathrm{A}].(y:\mathrm{B}).f, c, \mathrm{P}) \Uparrow \mathrm{P}\,c}\ (\textsc{Inf-I-Elim-L})$$

(7) Somme implicite droite

$$\frac{\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_1 \qquad \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*), \mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort}}{\Gamma \vdash \Sigma x:\mathrm{A}.[\mathrm{B}] \Uparrow \mathtt{NF\_BI}(\mathrm{S}_1^*)}\ (\textsc{Inf-I-}\Sigma\textsc{-R})$$

$$\frac{\begin{cases} \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_\mathrm{A} \\ \Gamma; x:\mathrm{A} \vdash \mathrm{B} \Uparrow^\mathrm{I} \mathrm{S}_\mathrm{B} \\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*), \mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*) \in \mathbf{Sort} \end{cases} \quad \begin{cases} \Gamma \vdash a \Uparrow \mathrm{R}_1 \\ \Gamma \vdash b \Uparrow \mathrm{R}_2 \end{cases} \quad \begin{cases} \mathtt{cumu}(\mathrm{R}_1^*, \mathrm{A}^*) \\ \mathtt{cumu}(\mathrm{R}_2^*, (\mathrm{B}[x/a])^*) \end{cases}}{\Gamma \vdash (a,[b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]} \Uparrow \Sigma x:\mathrm{A}.[\mathrm{B}]}\ (\textsc{Inf-I-Pair-R})$$

$$\frac{\Gamma \vdash c \Uparrow \mathrm{R} \qquad \mathsf{hnf}_\Gamma(\mathrm{R}) = \Sigma x:\mathrm{A}.[\mathrm{B}]}{\Gamma \vdash \pi_1(c) \Uparrow \mathrm{A}}\ (\textsc{Inf-I-Elim-R-1})$$

$$\frac{\begin{cases} \Gamma \vdash \mathrm{P} \Uparrow \mathrm{S}_1 \\ \Gamma \vdash c \Uparrow \mathrm{R}_1 \\ \Gamma \vdash \mathrm{B} \Uparrow \mathrm{S}_2 \end{cases} \quad \begin{cases} \mathtt{NF\_BI}(\mathrm{S}_1^*) \in \mathbf{Sort} \\ \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Sigma x:\mathrm{A}_1.[\mathrm{B}_1] \\ \mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort} \end{cases} \quad \begin{cases} \Gamma; y:\mathrm{B} \vdash f \Uparrow \mathrm{R}_2 \\ \mathtt{cumu}(\mathrm{R}_2^*, \mathrm{P}^*) \\ \mathtt{cumu}((\mathrm{B}_1[x/\pi_1(c)])^*, \mathrm{B}^*) \\ y \notin \mathsf{FV}(f^*) \end{cases}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I}_\mathrm{R}}([y:\mathrm{B}].f, c, \mathrm{P}) \Uparrow \mathrm{P}}\ (\textsc{Inf-I-Elim-R-2})$$

FIGURE 29. Règles d'inférence du jugement d'inférence de type (Sommes)

— Nous pouvons donc appliquer par induction le résultat à la prémisse $\Gamma; x : T \vdash U \Uparrow S_2$, puis en déduire, de même que précédemment, en appliquant le lemme 5.2.1 à $\Gamma; x : T \vdash U : S_2$, que $\Gamma; x : T \vdash U : s_2$.

- Par correction de $\texttt{Rule}_\square$, nous avons $(s_1, s_2, \texttt{Rule}_\square(s_1, s_2)) \in \mathbf{Rule}_\square$.

- (Inf-E-Lam) et (Inf-I-Lam) se traitent similairement. Traitons par exemple (Inf-I-Lam). Si

$$\frac{\Gamma \vdash T \Uparrow S \qquad \texttt{NF\_BI}(S^*) \in \mathbf{Sort} \qquad \Gamma; x : T \vdash M \Uparrow U \qquad x \notin \mathsf{FV}(M^*)}{\Gamma \vdash \lambda[x{:}T].M \Uparrow \Pi[x{:}T].U} \text{ (Inf-I-Lam)}$$

montrons qu'il existe une sorte s_0 telle que nous pouvons appliquer (I-Lam) et dériver

$$\frac{\Gamma; x : T \vdash M : U \qquad \Gamma \vdash \Pi[x{:}T].U : s_0 \qquad x \notin \mathsf{FV}(M^*)}{\Gamma \vdash \lambda[x{:}T].M : \Pi[x{:}T].U}$$

La condition de bord $x \notin \mathsf{FV}(M^*)$ est vraie par hypothèse. Montrons que les deux autres prémisses sont dérivables.

- Montrons que $\Gamma; x : T \vdash M : U$ est dérivable.
  - Par induction sur $\Gamma \vdash T \Uparrow S$ et d'après le lemme 5.2.1, nous déduisons $\Gamma \vdash T : s$ avec $s = \texttt{NF\_BI}(S^*)$, d'où $\Gamma; x : T \vdash$ en appliquant (Wf-S_A).
  - Puisque $\Gamma; x : T \vdash$, nous pouvons appliquer par induction le résultat à la prémisse $\Gamma; x : T \vdash M \Uparrow U$ et en déduire que $\Gamma; x : T \vdash M : U$ est dérivable.
- Montrons qu'il existe une sorte s_0 telle que $\Gamma \vdash \Pi[x{:}T].U : s_0$.
  - D'après le lemme du type des types appliqué à $\Gamma; x : T \vdash M : U$ il existe $s' \in \mathbf{Sort}$ telle que $\Gamma; x : T \vdash U : s'$.
  - Par complétude de **Rule**, il existe une sorte s_0 telle que $(s, s', s_0) \in \mathbf{Sort}$.
  - En appliquant (I-Prod), nous dérivons $\Gamma \vdash \Pi[x{:}T].U : s_0$.

- (Inf-E-App) et (Inf-I-App) sont semblables. Traitons par exemple (Inf-E-App). Si

$$\frac{\Gamma \vdash M \Uparrow R_1 \qquad \texttt{hnf}_\Gamma(R_1) = \Pi x : T.U \qquad \Gamma \vdash N \Uparrow R_2 \qquad \texttt{cumu}(R_2^*, T^*)}{\Gamma \vdash MN \Uparrow U[x/N]} \text{ (Inf-E-App)}$$

montrons que nous pouvons appliquer (E-App) et dériver

$$\frac{\Gamma \vdash M : \Pi x : T.U \qquad \Gamma \vdash N : T}{\Gamma \vdash MN : U[x/N]}$$

- Par induction sur $\Gamma \vdash M \Uparrow R_1$ et d'après le corollaire 4.3.2, nous avons $\Gamma \vdash M : \Pi x : T.U$.
- Par induction sur $\Gamma \vdash N \Uparrow R_2$ nous avons $\Gamma \vdash N : R_2$, puis, par cumulativité, $\Gamma \vdash N : T$. En effet :
  - par correction de cumu, nous avons $R_2^* \preceq T^*$;
  - d'après le lemme du type des types appliqué à $\Gamma \vdash M : \Pi x : T.U$ et par inversion, il existe une sorte s telle que $\Gamma \vdash T : s$.

- (Inf-E-Pair), (Inf-I-Pair-L) et (Inf-I-Pair-R) se traitent de manière similaire. Traitons le cas de la paire explicite. Si

$$\frac{\left\{\begin{array}{l}\Gamma \vdash A \Uparrow S_A \\ \Gamma; x : A \vdash B \Uparrow^I S_B \\ \texttt{NF\_BI}(S_A^*), \texttt{NF\_BI}(S_B^*) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma \vdash a \Uparrow R_1 \\ \Gamma \vdash b \Uparrow R_2\end{array}\right. \qquad \left\{\begin{array}{l}\texttt{cumu}(R_1^*, A^*) \\ \texttt{cumu}(R_2^*, (B[x/a])^*)\end{array}\right.}{\Gamma \vdash (a, b)_{\Sigma x:A.B} \Uparrow \Sigma x : A.B} \text{ (Inf-E-Pair)}$$

montrons qu'il existe une sorte s telle que que nous pouvons appliquer $(\Sigma_A\text{-I})$:

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash b : B[x/a] \qquad \Gamma \vdash \Sigma x : A.B : s}{\Gamma \vdash (a,b)_{\Sigma x:A.B} : \Sigma x : A.B}$$

Montrons tout d'abord deux résultats utiles.

- Par induction sur $\Gamma \vdash A \Uparrow S_A$ et d'après le lemme 5.2.1, nous obtenons $\Gamma \vdash A : s_A$ où $s_A = \texttt{NF\_BI}(S_A^*)$.
- Posons $s_B = \texttt{NF\_BI}(S_B^*)$. Montrons que $\Gamma; x : A \vdash B : s_B$.
  - — De $\Gamma \vdash A : s_A$, nous déduisons $\Gamma; x : A \vdash$ en appliquant (WF-S_A).
  - — Nous pouvons donc appliquer par induction le résultat à la prémisse $\Gamma; x : A \vdash B \Uparrow S_B$ et en déduire $\Gamma; x : A \vdash B : S_B$.
  - — Nous concluons grâce au lemme 5.2.1.

  Montrons maintenant que les trois prémisses sont dérivables.

(1) Montrons que $\Gamma \vdash a : A$.
  - Par induction sur $\Gamma \vdash a \Uparrow R_1$ nous déduisons $\Gamma \vdash a : R_1$.
  - Nous concluons par cumulativité sur $\Gamma \vdash a : R_1$ puisque
    - — $R_1^* \preceq A^*$ par correction de `cumu` appliquée à $\texttt{cumu}(R_1^*, A^*)$;
    - — nous avons vu plus haut que $\Gamma \vdash A : s_A$.

(2) Montrons que $\Gamma \vdash b : B[x/a]$.
  - Par induction sur $\Gamma \vdash b \Uparrow R_2$, nous obtenons $\Gamma \vdash b : R_2$.
  - Nous concluons par cumulativité sur $\Gamma \vdash b : R_2$ puisque
    - — $R_2^* \preceq (B[x/a])^*$ par correction de `cumu` appliquée à $\texttt{cumu}(R_2^*, (B[x/a])^*)$;
    - — de $\Gamma; x : A \vdash B : s_B$ et $\Gamma \vdash a : A$ nous déduisons par substitutivité que $\Gamma \vdash B[x/a] : s_B$.

(3) Montrons qu'il existe une sorte s telle que $\Gamma \vdash \Sigma x : A.B : s$.
  - Nous avons vu que $\Gamma \vdash A : s_A$ et $\Gamma; x : A \vdash B : s_B$.
  - Par complétude de $\mathbf{Rule}'$, il existe une sorte s telle que $(s_A, s_B, s) \in \mathbf{Rule}'$.
  - Nous concluons en appliquant (Σ_A).

- (INF-E-ELIM) et (INF-I-ELIM-L) : leurs démonstrations sont similaires ; traitons le cas (INF-I-ELIM-L).

  Si

$$\frac{\left\{\begin{array}{l}\Gamma \vdash P \Uparrow R_1\\ \Gamma \vdash c \Uparrow R_2\\ \Gamma \vdash A \Uparrow S_A\\ \Gamma; x : A \vdash B \Uparrow^I S_B\\ \texttt{NF\_BI}(S_A^*), \texttt{NF\_BI}(S_B^*) \in \mathbf{Sort}\\ \Gamma; x : A; y : B \vdash f \Uparrow R_3\end{array}\right. \quad \left\{\begin{array}{l}\mathsf{hnf}_\Gamma(R_1) = \Pi x : T.U\\ \texttt{NF\_BI}(U^*) \in \mathbf{Sort}\\ \mathsf{hnf}_\Gamma(R_2) = \Sigma[x:A_2].B_2\\ \texttt{cumu}((\Sigma[x:A].B)^*, T^*)\\ \texttt{cumu}((\Sigma[x:A_2].B_2)^*, (\Sigma[x:A].B)^*)\\ \texttt{cumu}(R_3^*, (P([x], y)_{\Sigma[x:A].B})^*)\end{array}\right. \quad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{I_L}([x:A].(y:B).f, c, P) \Uparrow P\,c}\ (\text{INF-I-ELIM-L})$$

Posons $s = \texttt{NF\_BI}(U^*)$ et montrons que nous pouvons appliquer $(\Sigma_\exists - E)$ et dériver :

$$\frac{\Gamma \vdash P : \Sigma[x:A].B \to s \qquad \Gamma \vdash c : \Sigma[x:A].B \qquad \Gamma; x : A; y : B \vdash f : P([x], y)_{\Sigma[x:A].B} \qquad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{I_L}([x:A].(y:B).f, c, P) : P\,c}$$

Avant de traiter les trois prémisses, montrons deux résultats utiles :

- Montrons qu'il existe s_0 tel que $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s_0$.
  - — Nous procédons comme plus haut dans le cas des règles de formation puisque les prémisses $\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}$, $\Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S_B}$ et $\mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_A^*}) \in \mathbf{Sort}$ sont exactement les prémisses de la règle (Inf-CstrTyp).
- Par correction de `cumu`, nous avons
  - — $(\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B})^* \preceq \mathrm{T}^*$,
  - — $(\Sigma[x{:}\mathrm{A_2}]\,.\,\mathrm{B_2})^* \preceq (\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B})^*$
  - — et $\mathrm{R_3^*} \preceq (\mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}})^*$.

Montrons maintenant les trois prémisses.

(1) Montrons que $\Gamma \vdash c : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$.
  - Par hypothèse d'induction, nous avons $\Gamma \vdash c : \mathrm{R_2}$.
  - D'après le corollaire 4.3.2 appliqué à $\Gamma \vdash c : \mathrm{R_2}$ et $\mathsf{hnf}_\Gamma(\mathrm{R_2})$, $\Gamma \vdash c : \Sigma[x{:}\mathrm{A_2}]\,.\,\mathrm{B_2}$.
  - D'après les résultats préliminaires, nous pouvons appliquer (Cum$_\mathrm{A}$) à $\Gamma \vdash c : \Sigma[x{:}\mathrm{A_2}]\,.\,\mathrm{B_2}$ et dériver $\Gamma \vdash c : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$.

(2) Montrons que $\Gamma \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s$.
  - Par induction sur $\Gamma \vdash \mathrm{P} \Uparrow \mathrm{R_1}$ nous obtenons $\Gamma \vdash \mathrm{P} : \mathrm{R_1}$.
  - Montrons qu'il existe une sorte s_1 telle que $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s : s_1$.
    - — Nous avons $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s_0$.
    - — Soit z une variable telle que $z \notin \mathsf{DV}(\Gamma)$. Montrons qu'il existe une sorte s' telle que $\Gamma; z : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \vdash s : s'$.
      - — Par absence de sorte maximale, il existe une sorte s' telle que $(s, s') \in$ **Axiom**.
      - — En appliquant (Sort$_\mathrm{A}$), nous déduisons $\Gamma; z : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \vdash s : s'$.
    - — Par complétude de **Rule**, il existe une sorte s_1 telle que $(s_0, s', s_1) \in$ **Rule**.
    - — En appliquant (E-Prod), et puisque $z \notin \mathsf{FV}(s)$, nous dérivons $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s : s_1$.
  - Montrons que $\mathrm{R_1^*} \twoheadrightarrow_{\beta\iota} \mathrm{T}^* \to s$.
    - — Montrons que $\mathrm{R_1^*} \twoheadrightarrow_{\beta\iota} \Pi x : \mathrm{T}^*\,.\,\mathrm{U}^*$.
      - — Puisque $\Gamma \vdash \mathrm{P} : \mathrm{R_1}$, il existe une sorte $s_{\mathrm{R_1}}$ telle que $\Gamma \vdash \mathrm{R_1} : s_{\mathrm{R_1}}$.
      - — Nous concluons en appliquant la proposition 4.3.1 à $\Gamma \vdash \mathrm{R_1} : s_{\mathrm{R_1}}$, puisque $\mathsf{hnf}_\Gamma(\mathrm{R_1}) = \Pi x : \mathrm{T}\,.\,\mathrm{U}$.
    - — Par correction de `NF_BI`, nous avons $\mathrm{U}^* \twoheadrightarrow_{\beta\iota} \mathtt{NF\_BI}(\mathrm{U}^*) = s$.
    - — Nous déduisons alors $\mathrm{R_1^*} \twoheadrightarrow_{\beta\iota} (\Sigma[x{:}\mathrm{A_1}]\,.\,\mathrm{B_1})^* \to s$ de $\mathrm{U}^* \twoheadrightarrow_{\beta\iota} s$ et $\mathrm{R_1^*} \twoheadrightarrow_{\beta\iota} \Pi x : \mathrm{T}^*\,.\,\mathrm{U}^*$.
  - Nous en déduisons $\mathrm{R_1^*} \preceq (\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s)^*$, puisque $(\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B})^* \preceq \mathrm{T_1^*}$.
  - Nous concluons alors en appliquant (Cum$_\mathrm{A}$) à $\Gamma \vdash \mathrm{P} : \mathrm{R_1}$.

(3) Montrons enfin que $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}}$.
  - Montrons que $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{R_3}$.
    - — Montrons que $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash$.

— Par inversion de $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s_0$, il existe une sorte s_B telle que $\Gamma; x:\mathrm{A} \vdash \mathrm{B} : s_\mathrm{B}$.

— Nous concluons en appliquant $(\mathrm{WF\text{-}S_A})$.

— Puisque $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash$, nous avons $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash f : \mathrm{R}_3$ comme hypothèse d'induction.

- Remplaçons par cumulativité le type de f.
  — Nous avons $\mathrm{R}_3^* \preceq (\mathrm{P}\,([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}})^*$.
  — Montrons que $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \mathrm{P}\,([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} : s$. Nous voulons pour cela appliquer (E-App) :

$$\frac{\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s \qquad \Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash ([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}}{\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \mathrm{P}\,([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} : s}$$

  — puisque $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash$, nous avons, par affaiblissement de $\Gamma \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s$ et $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s_0$, $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \mathrm{P} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \to s$ et $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s_0$;
  — nous dérivons $\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash ([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}$ en appliquant $(\Sigma_\exists - \mathrm{I})$:

$$\frac{\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash x:\mathrm{A} \qquad \Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash y:\mathrm{B} \qquad \Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s_0}{\Gamma; x:\mathrm{A}; y:\mathrm{B} \vdash ([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} : \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}}$$

- (Inf-I-Elim-R-1) : Si

$$\frac{\Gamma \vdash c \Uparrow \mathrm{R} \qquad \mathsf{hnf}_\Gamma(\mathrm{R}) = \Sigma x:\mathrm{A}\,.\,[\mathrm{B}]}{\Gamma \vdash \pi_1(c) \Uparrow \mathrm{A}}\ (\text{Inf-I-Elim-R-1})$$

montrons que $\Gamma \vdash \pi_1(c) : \mathrm{A}$.

- Par hypothèse d'induction et en appliquant le corollaire 4.3.2, nous avons $\Gamma \vdash c : \Sigma x : \mathrm{A}\,.\,[\mathrm{B}]$.
- Nous concluons alors en appliquant la règle $(\Sigma_{\mathrm{SUB}} - \mathrm{E} - 1)$.

- (Inf-I-Elim-R-2). Si

$$\frac{\left\{\begin{array}{l}\Gamma \vdash \mathrm{P} \Uparrow \mathrm{S}_1\\ \Gamma \vdash c \Uparrow \mathrm{R}_1\\ \Gamma \vdash \mathrm{B} \Uparrow \mathrm{S}_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{S}_1^*) \in \mathbf{Sort}\\ \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Sigma x:\mathrm{A}_1\,.\,[\mathrm{B}_1]\\ \mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma; y:\mathrm{B} \vdash f \Uparrow \mathrm{R}_2\\ \mathtt{cumu}(\mathrm{R}_2^*, \mathrm{P}^*)\\ \mathtt{cumu}((\mathrm{B}_1\,[x/\pi_1(c)])^*, \mathrm{B}^*)\\ y \notin \mathsf{FV}(f^*)\end{array}\right.}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P}) \Uparrow \mathrm{P}}\ (\text{Inf-I-Elim-R-2})$$

montrons que
$\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P}) : \mathrm{P}$.
Posons $s_1 = \mathtt{NF\_BI}(\mathrm{S}_1^*)$. Nous voulons pour cela appliquer la règle $(\Sigma_{\mathrm{SUB}} - \mathrm{E} - 2)$:

$$\frac{\Gamma \vdash \mathrm{P} : s_1 \qquad \Gamma \vdash c : \Sigma x:\mathrm{A}_1\,.\,[\mathrm{B}_1] \qquad \Gamma; y:\mathrm{B} \vdash f : \mathrm{P} \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*)\\ (\mathrm{B}_1\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*\end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P}) : \mathrm{P}}$$

La condition de bord $y \notin \mathsf{FV}(f^*)$ est vraie par hypothèse. Montrons les trois prémisses et la seconde condition de bord.

- Par hypothèse d'induction sur $\Gamma \vdash \mathrm{P} \Uparrow \mathrm{S}_1$ et en utilisant le lemme 5.2.1, nous montrons que $\Gamma \vdash \mathrm{P} : s_1$.

- Par hypothèse d'induction sur $\Gamma \vdash c \Uparrow \mathrm{R}_1$ et en appliquant le corollaire 4.3.2, nous avons $\Gamma \vdash c : \Sigma[x{:}\mathrm{A}_1].\,\mathrm{B}_1$.
- Montrons que $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$.
  - Montrons que $\Gamma; y : \mathrm{B} \vdash$.
    - Par hypothèse d'induction sur $\Gamma \vdash \mathrm{B} \Uparrow \mathrm{S}_2$ et en utilisant le lemme 5.2.1, nous montrons que $\Gamma \vdash \mathrm{B} : \mathtt{NF\_BI}(\mathrm{S}_2^*)$.
    - Puisque $\mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathsf{Sort}$, nous pouvons appliquer $(\mathrm{WF\text{-}S_A})$.
  - Puisque $\Gamma; y : \mathrm{B} \vdash$, nous avons, par hypothèse d'induction sur $\Gamma; y : \mathrm{B} \vdash f \Uparrow \mathrm{R}_2$, $\Gamma; y : \mathrm{B} \vdash f : \mathrm{R}_2$.
  - Montrons alors par cumulativité sur $\Gamma; y : \mathrm{B} \vdash f : \mathrm{R}_2$ que $\Gamma; y : \mathrm{B} \vdash f : \mathrm{P}$:
    - par correction de $\mathtt{cumu}$, puisque $\mathtt{cumu}(\mathrm{R}_2^*, \mathrm{P}^*)$, nous avons $\mathrm{R}_2^* \preceq \mathrm{P}^*$;
    - par affaiblissement de $\Gamma \vdash \mathrm{P} : s_1$, nous avons $\Gamma; y : \mathrm{B} \vdash \mathrm{P} : s_1$.
- Par correction de $\mathtt{cumu}$, $\mathtt{cumu}((\mathrm{B}_1\,[x/\pi_1(c)])^*, \mathrm{B}^*)$ implique $(\mathrm{B}_1\,[x/\pi_1(c)])^* \preceq \mathrm{B}^*$.

□

Le corollaire suivant, conséquence immédiate du lemme précédent et de la proposition du type des types, montre que tout type inféré est bien typé.

5.2.3. Corollaire. *Si* $\Gamma \vdash \mathrm{M} \Uparrow \mathrm{R}$ *alors il existe une sorte* s *telle que* $\Gamma \vdash \mathrm{R} : s$.

5.2.2. *Complétude.* Deux lemmes auxiliaires dont les énoncés correspondent aux besoins de la preuve de la complétude.

5.2.4. Lemme. *Si* $\Gamma \vdash \mathrm{M} \Uparrow \mathrm{T}$ *et si* $\mathrm{T}^* \preceq s$ *alors* $\mathtt{NF\_BI}(\mathrm{T}^*)$ *est défini et* $\mathtt{NF\_BI}(\mathrm{T}^*) \in \mathsf{Sort}$.

DÉMONSTRATION. Nous procédons en trois étapes.

- Montrons que T^* est bien typé dans ICC_Σ.
  - D'après le corollaire 5.2.3, il existe une sorte s_0 telle que $\Gamma \vdash \mathrm{T} : s_0$.
  - Par correction de l'extraction, nous avons $\Gamma^* \vdash \mathrm{T}^* : s_0$.
- Puisque T^* est bien typé et que $\mathrm{T}^* \preceq s$, le lemme 4.1.1 page 160 nous indique que $\mathsf{nf}_{\beta\iota}(\mathrm{T}^*) \in \mathsf{Sort}$.
- Puisque T^* est bien typé, l'algorithme $\mathtt{NF\_BI}$ appliqué à T^* termine et nous avons, par correction, $\mathtt{NF\_BI}(\mathrm{T}^*) = \mathsf{nf}_{\beta\iota}(\mathrm{T}^*) \in \mathsf{Sort}$.

□

5.2.5. Lemme. *Si* $\Gamma \vdash \mathrm{M} \Uparrow \mathrm{R}$ *et si* $\mathrm{R}^* \preceq \Box^* x : \mathrm{T}'.\,\mathrm{U}'$ *(resp.* $\Box^* x : \mathrm{T}'.\,\mathrm{U}' \preceq \mathrm{R}^*$*) alors*

(1) $\mathsf{hnf}_\Gamma(\mathrm{R})$ *est bien défini ;*

(2) il existe T, U *dans* AICC_Σ *tels que*

(a) $\mathsf{hnf}_\Gamma(\mathrm{R}) = \Box x : \mathrm{T}.\,\mathrm{U}$ *;*

(b) $\mathrm{R}^* \twoheadrightarrow_{\beta\iota} (\Box x : \mathrm{T}.\,\mathrm{U})^*$ *;*

(c) si de plus $\Box^* x : \mathrm{T}'.\,\mathrm{U}'$ *est bien typé dans* ICC_Σ *alors* $\mathtt{cumu}((\Box x : \mathrm{T}.\,\mathrm{U})^*, \Box^* x : \mathrm{T}'.\,\mathrm{U}')$ *(resp.* $\mathtt{cumu}(\Box^* x : \mathrm{T}'.\,\mathrm{U}', (\Box x : \mathrm{T}.\,\mathrm{U})^*)$*).*

DÉMONSTRATION. (1) Montrons que $\mathsf{hnf}_\Gamma(\mathrm{R})$ existe. D'après la conjecture 4.1.1,il suffit de montrer que R est bien typé dans AICC_Σ.

- Par correction de l'algorithme d'inférence de type, nous avons $\Gamma \vdash \mathrm{M} : \mathrm{R}$.

- Nous concluons en appliquant le lemme du type des types à $\Gamma \vdash M : R$.

(2) (a) D'après la proposition 4.4.1, il existe T, U tel que $\mathsf{hnf}_\Gamma(R) = \Box x : T.U$.

(b) Montrons que $\mathtt{cumu}((\Box x : T.U)^*, \Box^* x : T'.U')$ (resp. $\mathtt{cumu}(\Box^* x : T'.U', (\Box x : T.U)^*)$).

- Nous avons vu que $\Gamma \vdash R : s$ d'où, d'après la proposition 4.3.1, $R^* \twoheadrightarrow_{\beta\iota} (\mathsf{hnf}_\Gamma(R))^*$ et $\Gamma \vdash \mathsf{hnf}_\Gamma(R) : s$.
- Nous en déduisons $(\Box x : T.U)^* \preceq \Box^* x : T'.U'$ (resp. $\Box^* x : T'.U' \preceq (\Box x : T.U)^*$).
- Par correction de l'extraction sur $\Gamma \vdash \mathsf{hnf}_\Gamma(R) : s$, nous avons $\Gamma \vdash \mathsf{hnf}_\Gamma(R)^* : s$.
- Par complétude de cumu, puisque $\mathsf{hnf}_\Gamma(R)^*$ et $\Box^* x : T'.U'$ sont typables dans ICC_Σ, nous avons bien $\mathtt{cumu}((\Box x : T.U)^*, \Box^* x : T'.U')$ (resp. $\mathtt{cumu}(\Box^* x : T'.U', (\Box x : T.U)^*)$).

□

L'énoncé du lemme suivant correspond exactement à un résultat nécessaire pour la preuve de complétude.

5.2.6. Lemme. *Si* $\Gamma \vdash M_1 \Uparrow R_1$ *et* $\Gamma \vdash M_2 : R_2$ *et* $R_1^* \preceq R_2^*$ *alors* $\mathtt{cumu}(R_1^*, R_2^*)$.

DÉMONSTRATION.
- Par complétude de cumu sur $R_1^* \preceq R_2^*$, il suffit de montrer que R_1^* et R_2^* sont bien typés dans ICC_Σ.
- Par correction de l'extraction il suffit de montrer qu'ils sont bien typés dans AICC_Σ.
- Par correction de l'algorithme d'inférence de type, nous déduisons $\Gamma \vdash M_1 : R_1$.
- Nous concluons en appliquant le lemme du type des types à $\Gamma \vdash M_1 : R_1$ et $\Gamma \vdash M_2 : R_2$.

□

Nous pouvons maintenant montrer la complétude de l'algorithme d'inférence de type.

5.2.7. Proposition (Complétude de l'inférence de type)**.** *Si* $\Gamma \vdash M : T$ *alors il existe un terme* R *tel que* $\Gamma \vdash M \Uparrow R$ *et* $R^* \preceq T^*$.

DÉMONSTRATION. Par induction sur la dérivation de $\Gamma \vdash M : T$ en considérant la dernière règle appliquée.

- Immédiat pour (VAR$_A$) : nous appliquons directement la règle (INF-VAR).
- (SORT$_A$) : si $\Gamma \vdash s_1 : s_2$, montrons que $\Gamma \vdash s_1 \Uparrow s_2$.
  - Par inversion de $\Gamma \vdash s_1 : s_2$, $(s_1, s_2) \in \mathbf{Axiom}$.
  - Par complétude de Axiom, nous avons $s_2 = \mathtt{Axiom}(s_1)$.
  - Par application de (INF-SORT), nous avons bien $\Gamma \vdash s_1 \Uparrow s_2$.
- Les règles de formation (E-PROD), (I-PROD), (Σ_A), ($\Sigma_\exists$), (Σ_{SUB}) se prouvent de manière similaire. Traitons la règle générique.

  Si

$$\frac{\Gamma \vdash T : s_1' \qquad \Gamma; x : T \vdash U : s_2' \qquad (s_1', s_2', s_3') \in \mathbf{Rule}_\Box}{\Gamma \vdash \Box x : T.U : s_3'}$$

  montrons qu'il existe des termes S_1, S_2 et des sortes s_1, s_2 tels que

$$\frac{\Gamma \vdash T \Uparrow S_1 \qquad \mathtt{NF\_BI}(S_1^*) = s_1 \in \mathbf{Sort} \qquad \Gamma; x : T \vdash U \Uparrow S_2 \qquad \mathtt{NF\_BI}(S_2^*) = s_2 \in \mathbf{Sort}}{\Gamma \vdash \Box x : T.U \Uparrow \mathtt{Rule}_\Box(s_1, s_2)}\ (\text{INF-CSTRTYP})$$

et $\texttt{Rule}_\square(s_1, s_2) \preceq s_3'$.

- Par hypothèses d'induction, il existe $\mathrm{S}_1, \mathrm{S}_2$ tels que
  — $\Gamma \vdash \mathrm{T} \Uparrow \mathrm{S}_1$ et $\mathrm{S}_1^* \preceq s_1'$,
  — $\Gamma; x : \mathrm{T} \vdash \mathrm{U} \Uparrow \mathrm{S}_1$ et $\mathrm{S}_2^* \preceq s_2'$.
- D'après le lemme 5.2.4, il existe des sortes s_1 et s_2 telles que $\texttt{NF\_BI}(\mathrm{S}_1^*) = s_1, \texttt{NF\_BI}(\mathrm{S}_2^*) = s_2$.
- Montrons que $\texttt{Rule}_\square(s_1, s_2) \preceq s_3'$.
  — De $\mathrm{S}_1^* \preceq s_1'$ et $\mathrm{S}_2^* \preceq s_2'$, nous déduisons $\texttt{NF\_BI}(\mathrm{S}_1^*) = s_1 \preceq s_1'$ et $\texttt{NF\_BI}(\mathrm{S}_2^*) = s_2 \preceq s_2'$.
  — Par correction de $\texttt{Rule}_\square$, nous avons $(s_1, s_2, \texttt{Rule}_\square(s_1, s_2)) \in \mathbf{Rule}_\square$.
  — Par compatibilité de $\mathbf{Rule}_\square$ avec la cumulativité, nous déduisons $\texttt{Rule}_\square(s_1, s_2) \preceq s_3'$.

- Les règles d'introduction des types produit (E-Lam) et (I-Lam) sont similaires. Traitons le cas de (I-Lam).

  Si
  $$\frac{\Gamma; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{U} \quad \Gamma \vdash \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U} : s \quad x \notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma \vdash \lambda[x{:}\mathrm{T}]\,.\,\mathrm{M} : \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}}\,(\text{I-Lam})$$
  montrons qu'il existe S, U_0 tels que
  $$\frac{\Gamma \vdash \mathrm{T} \Uparrow \mathrm{S} \qquad \texttt{NF\_BI}(\mathrm{S}^*) \in \mathbf{Sort} \qquad \Gamma; x : \mathrm{T} \vdash \mathrm{M} \Uparrow \mathrm{U}_0 \qquad x \notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma \vdash \lambda[x{:}\mathrm{T}]\,.\,\mathrm{M} \Uparrow \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}_0}\,(\text{Inf-I-Lam})$$
  avec $(\Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}_0)^* \preceq (\Pi[x{:}\mathrm{T}]\,.\,\mathrm{U})^*$.
  - $x \notin \mathsf{FV}(\mathrm{M}^*)$ est vrai par hypothèse.
  - Par hypothèse d'induction, il existe U_0 tel que $\Gamma; x : \mathrm{T} \vdash \mathrm{M} \Uparrow \mathrm{U}_0$ et $\mathrm{U}_0^* \preceq \mathrm{U}^*$.
  - De $\mathrm{U}_0^* \preceq \mathrm{U}^*$ nous déduisons, par cumulativité, que $(\Pi[x{:}\mathrm{T}]\,.\,\mathrm{U}_0)^* \preceq (\Pi[x{:}\mathrm{T}]\,.\,\mathrm{U})^*$.
  - Montrons qu'il existe une sorte s_T telle qu'il existe une dérivation de $\Gamma \vdash \mathrm{T} : s_\mathrm{T}$ incluse dans celle de $\Gamma; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{U}$.
    — D'après le point 2 du lemme 2.3.1 page 120, la dérivation de $\Gamma; x : \mathrm{T} \vdash \mathrm{M} : \mathrm{U}$ contient une dérivation de $\Gamma; x : \mathrm{T} \vdash$.
    — Par inversion de (Wf-S$_\mathrm{A}$), il existe une sorte s_T telle que la prémisse de $\Gamma; x : \mathrm{T} \vdash$ soit $\Gamma \vdash \mathrm{T} : s_\mathrm{T}$.
  - Nous pouvons alors appliquer par induction le résultat sur le jugement $\Gamma \vdash \mathrm{T} : s_\mathrm{T}$ et en déduire qu'il existe S tel que $\Gamma \vdash \mathrm{T} \Uparrow \mathrm{S}$ et $\mathrm{S}^* \preceq s$.
  - Puisque $\Gamma \vdash \mathrm{T} \Uparrow \mathrm{S}$ et $\mathrm{S}^* \preceq s$, le lemme 5.2.4 nous indique que $\texttt{NF\_BI}(\mathrm{S}^*) \in \mathbf{Sort}$.

- Les règles d'élimination du produit (E-App) et (I-App) sont similaires. Traitons par exemple (I-App). Si
  $$\frac{\Gamma \vdash \mathrm{M} : \Pi[x{:}\mathrm{T}]\,.\,\mathrm{U} \qquad \Gamma \vdash \mathrm{N} : \mathrm{T}}{\Gamma \vdash \mathrm{M}\,[\mathrm{N}] : \mathrm{U}\,[x/\mathrm{N}]}$$
  montrons qu'il existe $\mathrm{R}_1, \mathrm{R}_2, \mathrm{T}_1, \mathrm{U}_1$ tels que
  $$\frac{\Gamma \vdash \mathrm{M} \Uparrow \mathrm{R}_1 \qquad \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Pi[x{:}\mathrm{T}_1]\,.\,\mathrm{U}_1 \qquad \Gamma \vdash \mathrm{N} \Uparrow \mathrm{R}_2 \qquad \texttt{cumu}(\mathrm{R}_2^*, \mathrm{T}_1^*)}{\Gamma \vdash \mathrm{M}\,[\mathrm{N}] \Uparrow \mathrm{U}_1\,[x/\mathrm{N}]}\,(\text{Inf-I-App})$$
  et $(\mathrm{U}_1\,[x/\mathrm{N}])^* \preceq \mathrm{U}\,[x/\mathrm{N}]$.
  - Par hypothèses d'induction, il existe $\mathrm{R}_1, \mathrm{R}_2$ tels que

— $\Gamma \vdash M \Uparrow R_1$ et $R_1^* \preceq (\Pi[x{:}T]\,.\,U)^*$,

— $\Gamma \vdash N \Uparrow R_2$ et $R_2^* \preceq T^*$.

- D'après le lemme 5.2.5 appliqué à $\Gamma \vdash M \Uparrow R_1$ et $R_1^* \preceq (\Pi[x{:}T]\,.\,U)^*$, il existe T_1, U_1 tels que $\mathsf{hnf}_\Gamma(R_1) = \Pi[x{:}T_1]\,.\,U_1$ et $R_1^* \twoheadrightarrow_{\beta\iota} (\Pi[x{:}T_1]\,.\,U_1)^*$.
- Montrons que $\mathtt{cumu}(R_2^*, T_1^*)$.
  - Montrons que $T^* \preceq T_1^*$.
    - Nous déduisons $\forall x : T_1^*\,.\,U_1^* \preceq \forall x : T^*\,.\,U^*$ de $R_1^* \preceq (\Pi[x:T]\,.\,U)^*$ et $R_1^* \twoheadrightarrow_{\beta\iota} (\Pi[x{:}T_1]\,.\,U_1)^*$.
    - Nous concluons par inversion de la cumulativité.
  - Par transitivité de $\preceq$, nous déduisons $R_2^* \preceq T_1^*$ de $R_2^* \preceq T^*$ et $T^* \preceq T_1^*$.
  - Montrons que R_2^* est bien typé dans ICC_Σ.
    - R_2 est bien typé d'après le corollaire 5.2.3 appliqué à $\Gamma \vdash N \Uparrow R_2$.
    - Nous concluons par correction de l'extraction.
  - Montrons que T_1^* est bien typé dans ICC_Σ.
    - D'après le corollaire 5.2.3 appliqué à $\Gamma \vdash M \Uparrow R_1$, il existe une sorte s telle que $\Gamma \vdash R_1 : s$.
    - D'après la proposition 4.3.1 appliquée à $\Gamma \vdash R_1 : s$, nous avons $\Gamma \vdash \Pi[x : T_1]\,.\,U_1 : s$.
    - Par inversion de $\Gamma \vdash \Pi[x{:}T_1]\,.\,U_1 : s$, il existe une sorte s_1 telle que $\Gamma \vdash T : s_1$.
    - Nous concluons par correction de l'extraction.
  - Par complétude de cumu, puisque R_2^* et T_1^* sont bien typés, nous déduisons $\mathtt{cumu}(R_2^*, T_1^*)$ de $R_2^* \preceq T_1^*$.

- Les règles d'introduction des types somme $(\Sigma_A\text{-I})$, $(\Sigma_\exists - I)$ et $(\Sigma_{SUB} - I)$ sont similaires. Traitons par exemple $(\Sigma_\exists - I)$. Si

$$\frac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[x/a] \quad \Gamma \vdash \Sigma[x{:}A]\,.\,B : s}{\Gamma \vdash ([a], b)_{\Sigma[x:A]\,.\,B} : \Sigma[x{:}A]\,.\,B}\,(\Sigma_\exists\text{-I})$$

Montrons qu'il existe S_A, S_B, R_1, R_2 tels que

$$\frac{\left\{\begin{array}{l}\Gamma \vdash A \Uparrow S_A\\ \Gamma; x : A \vdash B \Uparrow^I S_B\\ \mathtt{NF\_BI}(S_A^*), \mathtt{NF\_BI}(S_B^*) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma \vdash a \Uparrow R_1\\ \Gamma \vdash b \Uparrow R_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{cumu}(R_1^*, A^*)\\ \mathtt{cumu}(R_2^*, (B[x/a])^*)\end{array}\right.}{\Gamma \vdash ([a], b)_{\Sigma[x:A]\,.\,B} \Uparrow \Sigma[x{:}A]\,.\,B}\,(\textsc{Inf-I-Pair-L})$$

- Par hypothèses d'induction, il existe R_1, R_2 tels que
  - $\Gamma \vdash a \Uparrow R_1$ et $R_1^* \preceq A^*$;
  - $\Gamma \vdash b \Uparrow R_2$ et $R_2^* \preceq (B[x/a])^*$.
- D'après le lemme 5.2.6 appliqué à $\Gamma \vdash a \Uparrow R_1$, $\Gamma \vdash a : A$ et $R_1^* \preceq A^*$, nous déduisons $\mathtt{cumu}(R_1^*, A^*)$.
- De même nous montrons que $\mathtt{cumu}(R_2^*, (B[x/a])^*)$ en appliquant le lemme 5.2.6 à $\Gamma \vdash b \Uparrow R_2$, $\Gamma \vdash b : B[x/a]$ et $R_2^* \preceq (B[x/a])^*$.
- Montrons qu'il existe S_A, S_B tels que $\Gamma \vdash A \Uparrow S_A$, $\Gamma; x : A \vdash B \Uparrow S_B$, $\mathtt{NF\_BI}(S_A^*), \mathtt{NF\_BI}(S_B^*) \in \mathbf{Sort}$.

— Par inversion de $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s$, nous savons qu'il existe des sortes $s_\mathrm{A}, s_\mathrm{B}$ et des dérivations de $\Gamma \vdash \mathrm{A} : s_\mathrm{A}$ et de $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_\mathrm{B}$ incluses dans celle de $\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} : s$.

— Nous pouvons donc appliquer par induction le résultat à $\Gamma \vdash \mathrm{A} : s_\mathrm{A}$ et $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_\mathrm{B}$ et en déduire qu'il existe $\mathrm{S_A}, \mathrm{S_B}$ tels que

— $\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}$ et $\mathrm{S_A^*} \preceq s_\mathrm{A}$,

— $\Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S_B}$ et et $\mathrm{S_B^*} \preceq s_\mathrm{B}$.

— Nous déduisons que $\mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathsf{Sort}$ en appliquant le lemme 5.2.4.

- Les règles d'élimination des types somme $(\Sigma_\mathrm{A}\text{-E})$ et $(\Sigma_\exists - \mathrm{E})$ se traitent de la même façon. Prenons le cas de la somme explicite. Nous avons

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma x : \mathrm{A}.\mathrm{B} \to s \quad \Gamma \vdash c : \Sigma x : \mathrm{A}.\mathrm{B} \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}}}{\Gamma \vdash \mathsf{Elim}_\mathrm{E}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}\,(\Sigma_\mathrm{A}\text{-E})$$

Montrons qu'il existe $\mathrm{R_1}, \mathrm{R_2}, \mathrm{R_3}, \mathrm{S_A}, \mathrm{S_B}, \mathrm{T}, \mathrm{U}, \mathrm{A_2}, \mathrm{B_2}$ tels que

$$\frac{\left\{\begin{array}{l} \Gamma \vdash \mathrm{P} \Uparrow \mathrm{R_1} \\ \Gamma \vdash c \Uparrow \mathrm{R_2} \\ \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A} \\ \Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow^\mathrm{I} \mathrm{S_B} \\ \mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathsf{Sort} \\ \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f \Uparrow \mathrm{R_3} \end{array}\right. \quad \left\{\begin{array}{l} \mathsf{hnf}_\Gamma(\mathrm{R_1}) = \Pi x : \mathrm{T}\,.\,\mathrm{U} \\ \mathtt{NF\_BI}(\mathrm{U}^*) \in \mathsf{Sort} \\ \mathsf{hnf}_\Gamma(\mathrm{R_2}) = \Sigma x : \mathrm{A_2}\,.\,\mathrm{B_2} \\ \mathtt{cumu}((\Sigma x : \mathrm{A}.\mathrm{B})^*, \mathrm{T}^*) \\ \mathtt{cumu}((\Sigma x : \mathrm{A_2}\,.\,\mathrm{B_2})^*, (\Sigma x : \mathrm{A}.\mathrm{B})^*) \\ \mathtt{cumu}(\mathrm{R_3^*}, (\mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}})^*) \end{array}\right.}{\Gamma \vdash \mathsf{Elim}_\mathrm{E}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) \Uparrow \mathrm{P}\,c}\,(\textsc{Inf-E-Elim})$$

Nous procédons en quatre étapes.

(1) Montrons qu'il existe $\mathrm{R_3}$ tel que $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f \Uparrow \mathrm{R_3}$ et $\mathtt{cumu}(\mathrm{R_3^*}, (\mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}})^*)$.

- Par hypothèse d'induction sur $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}}$, il existe $\mathrm{R_3}$ tel que $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f \Uparrow \mathrm{R_3}$ et $\mathrm{R_3^*} \preceq (\mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}})^*$.
- Nous déduisons $\mathtt{cumu}(\mathrm{R_3^*}, (\mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}})^*)$ du lemme 5.2.6 appliqué à $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f \Uparrow \mathrm{R_3}$, $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}}$ et $\mathrm{R_3^*} \preceq (\mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}})^*$.

(2) Montrons qu'il existe $\mathrm{S_A}$ et $\mathrm{S_B}$ tels que $\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}$, $\Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S_B}$ et $\mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathsf{Sort}$.

- Montrons qu'il existe des sortes $s_\mathrm{A}, s_\mathrm{B}$ telles qu'une dérivation de $\Gamma \vdash \mathrm{A} : s_\mathrm{A}$ et une dérivation de $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_\mathrm{B}$ sont incluses dans celle de $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}}$.
  — D'après le lemme de bonne formation de contextes, la dérivation de $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma[x:\mathrm{A}]\,.\,\mathrm{B}}$, contient une dérivation de $\Gamma; x : \mathrm{A} \vdash$ et une dérivation de $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash$.
  — Nous concluons en inversant $(\mathrm{WF\text{-}S_A})$ dans $\Gamma; x : \mathrm{A} \vdash$ et $\Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash$.
- Par hypothèses d'induction sur $\Gamma \vdash \mathrm{A} : s_\mathrm{A}$ et $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_\mathrm{B}$, il existe $\mathrm{S_A}, \mathrm{S_B}$ tels que
  — $\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}$,
  — $\Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S_B}$,
  — $\mathrm{S_A^*} \preceq s_\mathrm{A}$ et $\mathrm{S_B^*} \preceq s_\mathrm{B}$.
- D'après le lemme 5.2.4 appliqué à $\mathrm{S_A}$ et $\mathrm{S_B}$, nous déduisons $\mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathsf{Sort}$.

(3) Montrons qu'il existe R_2, A_2, B_2 tels que $\Gamma \vdash c \Uparrow R_2$, $\mathsf{hnf}_\Gamma(R_2) = \Sigma x : A_2 . B_2$ et $\mathtt{cumu}((\Sigma x : A_2 . B_2)^*, (\Sigma x : A . B)^*)$.

- Par hypothèse d'induction sur $\Gamma \vdash c : \Sigma x : A . B$, il existe R_2 tel que $\Gamma \vdash c \Uparrow R_2$ et $R_2^* \preceq (\Sigma x : A . B)^*$.
- Nous pouvons alors appliquer le lemme 5.2.5 et en déduire qu'il existe A_2, B_2 tels que
  - $\mathsf{hnf}_\Gamma(R_2) = \Sigma x : A_2 . B_2$
  - $\mathtt{cumu}((\Sigma x : A_2 . B_2)^*, (\Sigma x : A . B)^*)$.

(4) Montrons enfin qu'il existe R_1, T, U tels que $\Gamma \vdash P \Uparrow R_1$, $\mathsf{hnf}_\Gamma(R_1) = \Pi x : T . U$, $\mathtt{NF\_BI}(U^*) \in \mathbf{Sort}$ et $\mathtt{cumu}((\Sigma x : A . B)^*, T^*)$.

- Par hypothèse d'induction sur $\Gamma \vdash P : \Sigma x : A . B \to s$ il existe R_1 tel que $\Gamma \vdash P \Uparrow R_1$ et $R_1^* \preceq (\Sigma x : A . B \to s)^*$.
- Nous pouvons alors appliquer le lemme 5.2.5 et en déduire qu'il existe T, U tels que
  - $\mathsf{hnf}_\Gamma(R_1) = \Pi x : T . U$,
  - $R_1^* \twoheadrightarrow_{\beta\iota} \mathsf{hnf}_\Gamma(R_1)^* = \Pi x : T^* . U^*$.
- Montrons que $(\Sigma x : A . B)^* \preceq T^*$ et $U^* \preceq s$.
  - D'après la proposition 4.3.1, nous avons $R_1^* \twoheadrightarrow_{\beta\iota\eta} \mathsf{hnf}_\Gamma(R_1)^* = (\Pi x : T . U)^*$.
  - De $R_1^* \preceq (\Sigma x : A . B \to s)^*$, nous déduisons $(\Pi x : T . U)^* \preceq (\Sigma x : A . B \to s)^*$.
  - Nous concluons par inversion de la cumulativité.
- Montrons qu'il existe des sortes s_T, s_U telles que $\Gamma \vdash T : s_T$ et $\Gamma; x : T \vdash U : s_U$.
  - Par correction de l'algorithme d'inférence de type, nous avons $\Gamma \vdash P : R_1$.
  - D'après le corollaire 4.3.2 appliqué à $\Gamma \vdash P : R_1$, nous déduisons $\Gamma \vdash P : \Pi x : T . U$.
  - Nous concluons grâce au lemme du type des types et après inversion de jugement de typage.
- Montrons que $\mathtt{NF\_BI}(U^*) \in \mathbf{Sort}$.
  - Par correction de l'extraction, nous avons $\Gamma^*; x : T^* \vdash U^* : s_U$.
  - Puisque U^* est bien typé et que $U^* \preceq s$, d'après le lemme 4.1.1 page 160 nous avons bien $\mathtt{NF\_BI}(U^*) \in \mathbf{Sort}$.
- Montrons enfin que $\mathtt{cumu}((\Sigma x : A . B)^*, T^*)$.
  - De $\Gamma \vdash c : \Sigma x : A . B$ nous déduisons que $\Sigma x : A . B$ est bien typé. Par correction de l'extraction, $(\Sigma x : A . B)^*$ est bien typé dans ICC_Σ.
  - De même, nous avons vu que T est bien typé dans AICC_Σ, donc T^* est bien typé dans ICC_Σ.
  - Par complétude de $\mathtt{cumu}$ et puisque $(\Sigma x : A . B)^* \preceq T^*$, nous déduisons bien $\mathtt{cumu}((\Sigma x : A . B)^*, T^*)$.

- $(\Sigma_{\mathrm{SUB}} - \mathrm{E} - 1)$:

  Si

$$\frac{\Gamma \vdash c : \Sigma x : A . [B]}{\Gamma \vdash \pi_1(c) : A}$$

  montrons qu'il existe R, A_0, B_0 tels que

$$\frac{\Gamma \vdash c \Uparrow \mathrm{R} \qquad \mathsf{hnf}_\Gamma(\mathrm{R}) = \Sigma x : \mathrm{A}_0 . [\mathrm{B}_0]}{\Gamma \vdash \pi_1(c) \Uparrow \mathrm{A}_0} \text{ (Inf-I-Elim-R-1)}$$

avec $\mathrm{A}_0^* \preceq \mathrm{A}^*$.

- Par hypothèse d'induction, il existe R tel que $\Gamma \vdash c \Uparrow \mathrm{R}$ et $\mathrm{R}^* \preceq \{x : \mathrm{A}^* \mid \mathrm{B}^*\}$.
- D'après le lemme 5.2.5 appliqué à R, $\mathsf{hnf}_\Gamma(\mathrm{R})$ est bien défini et il existe $\mathrm{A}_0, \mathrm{B}_0$ tels que $\mathsf{hnf}_\Gamma(\mathrm{R}) = \Sigma x : \mathrm{A}_0 . [\mathrm{B}_0]$ et $\{x : \mathrm{A}_0^* \mid \mathrm{B}_0^*\} \preceq \{x : \mathrm{A}^* \mid \mathrm{B}^*\}$.
- Nous en déduisons, par inversion de la cumulativité, $\mathrm{A}_0^* \preceq \mathrm{A}^*$.

- $(\Sigma_{\text{SUB}} - \text{E} - 2)$: Si

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A} . [\mathrm{B}] \qquad \Gamma ; y : \mathrm{B}_0 \vdash f : \mathrm{P} \qquad \begin{array}{l} y \notin \mathsf{FV}(f^*) \\ (\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \end{array}}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P}) : \mathrm{P}}$$

Montrons qu'il existe $\mathrm{R}_1, \mathrm{R}_2, \mathrm{S}_1, \mathrm{S}_2, \mathrm{A}_1, \mathrm{B}_1$ tels que

$$\frac{\left\{\begin{array}{l} \Gamma \vdash \mathrm{P} \Uparrow \mathrm{S}_1 \\ \Gamma \vdash c \Uparrow \mathrm{R}_1 \\ \Gamma \vdash \mathrm{B}_0 \Uparrow \mathrm{S}_2 \end{array}\right. \qquad \left\{\begin{array}{l} \mathtt{NF\_BI}(\mathrm{S}_1^*) \in \mathbf{Sort} \\ \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Sigma x : \mathrm{A}_1 . [\mathrm{B}_1] \\ \mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort} \end{array}\right. \qquad \left\{\begin{array}{l} \Gamma ; y : \mathrm{B}_0 \vdash f \Uparrow \mathrm{R}_2 \\ \mathtt{cumu}(\mathrm{R}_2^*, \mathrm{P}^*) \\ \mathtt{cumu}((\mathrm{B}_1\,[x/\pi_1(c)])^*, \mathrm{B}_0^*) \\ y \notin \mathsf{FV}(f^*) \end{array}\right.}{\Gamma \vdash \mathsf{Elim}^s_{\mathrm{I_R}}([y:\mathrm{B}_0].f, c, \mathrm{P}) \Uparrow \mathrm{P}}$$

- $y \notin \mathsf{FV}(f^*)$ est vrai par hypothèse.
- Par hypothèses d'induction, il existe $\mathrm{S}_1, \mathrm{R}_1, \mathrm{R}_2$ tels que
  (1) $\Gamma \vdash \mathrm{P} \Uparrow \mathrm{S}_1$ et $\mathrm{S}_1^* \preceq s$,
  (2) $\Gamma \vdash c \Uparrow \mathrm{R}_1$ et $\mathrm{R}_1^* \preceq (\Sigma x : \mathrm{A} . [\mathrm{B}])^*$,
  (3) $\Gamma ; y : \mathrm{B}_0 \vdash f \Uparrow \mathrm{R}_2$ et $\mathrm{R}_2^* \preceq \mathrm{P}^*$.
- Nous montrons que R_2^* et P^* sont bien typés dans ICC_Σ et en déduisons, par complétude de cumu sur $\mathrm{R}_2^* \preceq \mathrm{P}^*$, que $\mathtt{cumu}(\mathrm{R}_2^*, \mathrm{P}^*)$.
- D'après le lemme 5.2.4 appliqué à (1), nous avons $\mathtt{NF\_BI}(\mathrm{S}_1^*) \in \mathbf{Sort}$.
- D'après le lemme 5.2.5 appliqué à (2), il existe $\mathrm{A}_1, \mathrm{B}_1$ tels que $\mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Sigma x : \mathrm{A}_1 . [\mathrm{B}_1]$. Nous avons de plus $\mathrm{R}_1^* \twoheadrightarrow_{\beta\iota} (\Sigma x : \mathrm{A}_1 . [\mathrm{B}_1])^*$.
- Montrons qu'il existe un terme S_2 tel que $\Gamma \vdash \mathrm{B}_0 \Uparrow \mathrm{S}_2$ et $\mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort}$.
  — Nous montrons, comme dans le cas de (I-Lam) traité ci-dessus, qu'il existe une sorte s_B et une dérivation de $\Gamma \vdash \mathrm{B}_0 : s_\mathrm{B}$ incluse dans la dérivation de de $\Gamma ; y : \mathrm{B}_0 \vdash f : \mathrm{P}$.
  — Nous pouvons donc appliquer par induction la proposition sur le jugement $\Gamma \vdash \mathrm{B}_0 : s_\mathrm{B}$ et en déduire qu'il existe S_2 tel que $\Gamma \vdash \mathrm{B}\,[x/\pi_1(c)] \Uparrow \mathrm{S}_2$ et $\mathrm{S}_2^* \preceq s_\mathrm{B}$.
  — D'après le lemme 5.2.4 appliqué à $\Gamma \vdash \mathrm{B}\,[x/\pi_1(c)] \Uparrow \mathrm{S}_2$ et $\mathrm{S}_2^* \preceq s_\mathrm{B}$, nous prouvons bien que $\mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort}$.
- Montrons que $\mathtt{cumu}((\mathrm{B}_1\,[x/\pi_1(c)])^*, \mathrm{B}_0^*)$. Par complétude de cumu, il suffit de montrer que $(\mathrm{B}_1\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^*$ et que $(\mathrm{B}_1\,[x/\pi_1(c)])^*$ et B_0^* sont bien typés dans ICC_Σ.
  — Montrons que $(\mathrm{B}_1\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^*$.
    — De $\mathrm{R}_1^* \preceq (\Sigma x : \mathrm{A} . [\mathrm{B}])^*$ et $\mathrm{R}_1^* \twoheadrightarrow_{\beta\iota} (\Sigma x : \mathrm{A}_1 . [\mathrm{B}_1])^*$ nous déduisons $(\Sigma x : \mathrm{A}_1 . [\mathrm{B}_1])^* \preceq (\Sigma x : \mathrm{A} . [\mathrm{B}])^*$, puis $\mathrm{B}_1^* \preceq \mathrm{B}^*$ par inversion de la cumulativité.
    — Nous déduisons de $\mathrm{B}_1^* \preceq \mathrm{B}^*$ que $(\mathrm{B}_1\,[x/\pi_1(c)])^* \preceq (\mathrm{B}\,[x/\pi_1(c)])^*$.

— Nous concluons par transitivité puisque $(B[x/\pi_1(c)])^* \preceq B_0^*$.

— Nous montrons que B_0^* est bien typé dans ICC_Σ par correction de l'extraction sur $\Gamma \vdash B_0 : s_B$.

— Montrons que $(B_1[x/\pi_1(c)])^*$ est bien typé dans ICC_Σ. Par correction de l'extraction, il suffit de montrer que $B_1[x/\pi_1(c)]$ est bien typé dans $AICC_\Sigma$.

— Par correction de l'algorithme d'inférence, nous obtenons $\Gamma \vdash c : R_1$.

— D'après la proposition 4.3.1 appliquée à $\Gamma \vdash c : R_1$, nous déduisons $\Gamma \vdash c : \Sigma x : A_1 . [B_1]$.

— Par inversion de $\Gamma \vdash c : \Sigma x : A_1 . [B_1]$ il existe une sorte s_1 telle que $\Gamma; x : A_1 \vdash B_1 : s_1$.

— Par application de $(\Sigma_{SUB}-E-1)$ à $\Gamma \vdash c : \Sigma x : A_1 . [B_1]$, nous dérivons $\Gamma \vdash \pi_1(c) : A_1$.

— Nous pouvons donc substituer x par $\pi_1(c)$ dans $\Gamma; x : A_1 \vdash B_1 : s_1$ et en déduire $\Gamma \vdash B_1[x/\pi_1(c)] : s_1$.

□

Le corollaire suivant explique comment les deux algorithmes d'inférence sont liés.

5.2.8. Corollaire. *Soient* Γ *un contexte et* M *un terme de* $AICC_\Sigma$. *Alors il existe* R *dans* $AICC_\Sigma$ *tel que* $\Gamma \vdash M \Uparrow R$ *si et seulement s'il existe un terme* R' *dans* ICC_Σ *tel que* $\Gamma^* \vdash M \Uparrow^I R'$. *Dans ce cas, nous avons* $R^* \twoheadrightarrow_{\beta\iota} R'$.

DÉMONSTRATION.
- Si $\Gamma \vdash M \Uparrow R$ alors, par correction, $\Gamma \vdash M : R$. Par complétude de l'algorithme d'inférence de type extrait, il existe R' tel que $\Gamma^* \vdash M \Uparrow^I R'$ et $R' \preceq R^*$.
- Si $\Gamma^* \vdash M \Uparrow^I R'$ alors, par correction, il existe R_0 annoté tel que $\Gamma \vdash M : R_0$ et $R_0^* = R'$. Par complétude de l'algorithme d'inférence de type, il existe R tel que $\Gamma \vdash M \Uparrow R$ et $R^* \preceq R_0^* = R'$.
- Nous avons donc $R^* \cong R'$ si $\Gamma \vdash M \Uparrow R$ et $\Gamma^* \vdash M \Uparrow^I R'$.
- Une induction sur la structure de M permet de montrer que $R^* \twoheadrightarrow_{\beta\iota} R'$.

□

Type principal. La complétude de l'algorithme permet de montrer l'existence d'un type principal pour les termes bien typés de $AICC_\Sigma$. Comme dans ECC [**Luo94**], nous avons une règle de cumulativité, ce qui fait que la propriété d'unicité des types ne peut être vérifiée et que nous définissons une notion de type principal comme le type le plus petit au sens de la relation de cumulativité. La particularité de $AICC_\Sigma$ est que la règle (CUM_A) compare les *extractions* des termes, ce qui nous impose de définir le type principal comme le type dont l'*extraction* est le type le plus petit.

5.2.9. DÉFINITION (Type principal). Soient M un terme et Γ un contexte. T est un *type principal de* M *pour* Γ si $\Gamma \vdash M : T$ est dérivable et si pour tout terme R tel que $\Gamma \vdash M : R$, $T^* \preceq R^*$.

5.2.10. Corollaire (Existence du type principal)**.** *Tout terme bien typé admet un type principal.*

DÉMONSTRATION. La complétude de l'algorithme d'inférence de type prouve que le type inféré est un type principal. □

5.2.11. REMARQUES.

(1) La notion de type principal dépend du contexte dans lequel nous nous plaçons. Toutefois, s'il n'y a pas d'ambiguïté, nous ne le mentionnerons pas.

(2) Le type principal est défini de manière unique à convertibilité près de son extraction. Ainsi, si T_0, T_1 sont des types principaux de M sous Γ, nous avons par antisymétrie $T_0^* \cong T_1^*$.

Perspectives

Ce chapitre présente et étudie un algorithme d'inférence de type pour AICC_Σ. Afin d'en compléter l'étude, il conviendra de montrer la conjecture 4.1.1 concernant la normalisation forte de la réduction $\rightarrow_\Gamma^h$.

Établir un algorithme d'inférence de type nous a obligé à enrichir AICC_Σ de règles de réduction. Il serait intéressant de savoir si ces règles pourraient permettre de simuler dans AICC_Σ toute réduction dans ICC_Σ. Pour cela il faudrait considérer une réduction $\rightarrow_\Gamma$ dans un contexte plus large que dans $\rightarrow_\Gamma^h$, où seuls certains sous-termes sont réduits. Cette réduction aurait la spécification suivante : si $\Gamma \vdash M : T$ et si $M^* \rightarrow_{\beta\iota} N'$ alors il existe un terme N de AICC_Σ tel que $N^* = N'$ et $M \rightarrow_\Gamma N$. AICC_Σ muni de cette réduction pourrait alors être défini indépendamment de ICC_Σ.

Conclusion

Nous avons présenté deux calculs des constructions avec sommes dépendantes et arguments implicites.

Le premier, ICC_Σ, étend le Calcul des Constructions Implicite de Miquel avec des types somme. De même que ICC contient deux types produit (le produit dépendant habituel et un produit dépendant implicite qui généralise le polymorphisme paramétrique à tous les termes du calcul), ICC_Σ contient trois types somme : la somme dépendante habituelle, et deux variantes implicites, le type existentiel et le type sous-ensemble. Sa syntaxe est à la Curry. ICC_Σ possède de bonnes propriétés métathéoriques, Même si l'établissement des lemmes d'inversion usuels pose problème, nous sommes parvenus à montrer que le typage était préservé par $\beta\iota$-réduction ; cela ne semble pas être le cas pour la η-réduction. La cohérence logique de ICC_Σ a été montrée par l'établissement d'un modèle. Ce modèle est une adaptation à ICC_Σ du modèle établi par Miquel pour ICC.

Le second système, $AICC_\Sigma$, est un Calcul des Constructions à la Church avec deux types produit et trois types somme. Il est muni d'un mécanisme interne d'extraction qui transforme tout terme de $AICC_\Sigma$ en un terme de ICC_Σ. Une caractéristique essentielle de ce système est que la règle de cumulativité compare des termes extraits et non des termes de $AICC_\Sigma$: le calcul se fait donc dans ICC_Σ. Ainsi nous pouvons montrer que les termes bien typés de $AICC_\Sigma$ correspondent exactement aux termes bien typés de ICC_Σ.

Nous définissons deux algorithmes d'inférence de type pour $AICC_\Sigma$. Le premier extrait un type de ICC_Σ et permet de définir un algorithme de vérification de type de $AICC_\Sigma$. Le second algorithme infère un type de $AICC_\Sigma$. L'établissement de ce dernier a nécessité de rajouter des règles de réduction dans $AICC_\Sigma$ afin d'émuler les réductions ayant lieu au niveau de ICC_Σ. La correction et la complétude de ces algorithmes repose sur deux conjectures : la normalisation forte de la $\beta\iota\eta$-réduction dans ICC_Σ et, pour le deuxième algorithme uniquement, la normalisation forte de la relation de réduction dans $AICC_\Sigma$.

Perspectives

Démontrer les conjectures que nous avons posées serait une première extension de nos travaux de thèse. Plus généralement, il serait intéressant de prouver certains résultats mentionnés au cours de cette thèse qui nous semblent très probables. Parmi ceux-là se trouve en première place l'indécidabilité du typage dans ICC_Σ. Nous avons également des résultats moins généraux et dont la démonstration serait sans doute plus simple : le fait qu'un type existentiel purement implicite invalide la préservation du typage, que la η-réduction ne préserve pas le typage. Établir solidement ces derniers résultats semble nécessiter d'établir des lemmes d'inversion plus précis pour l'application et la variable. Peut-être que des techniques similaires à celles utilisées dans le chapitre 2 (préservation du typage par $\beta\iota$-réduction) et dans le chapitre 7 (relèvement) permettraient de conclure.

Un autre sujet d'étude intéressant est la possibilité d'avoir un type existentiel vraiment implicite. Comme nous l'avons vu, il semble probable que la préservation du typage soit perdue. Mais

il est possible qu'en restreignant les βι-réductions possibles, comme dans [**Xi06, SF14**], le typage serait préservé.

Enfin, notre projet à long terme est d'ajouter les types inductifs. Les prochaines étapes pourraient être l'étude de l'égalité et des W-types de Martin-Löf [**ML85**].

Travaux connexes

Pfenning a proposé un Calcul des Constructions inspiré de la logique modale avec des annotations permettant de distinguer les preuves des expressions *a priori* calculatoires [**Pfe01**]. Dans ce système, les variables considérées comme logiques ne peuvent avoir de rôle calculatoire ni dans les termes (comme dans AICC_Σ), ni dans les types (contrairement à AICC_Σ). Ce système semble donc être moins expressif car le produit dépendant implicite n'est pas exprimable. Mishra-Linger et Sheard [**ML08**] ont repris l'approche de Pfenning en ajoutant au système une version à la Church du produit dépendant implicite de Miquel et un mécanisme d'effacement tout à fait similaire à l'extraction de ICC_Σ. Une différence majeure avec ICC_Σ est que la conversion se fait entre termes annotés. Ainsi leur système ne parvient pas à représenter complètement ICC et reste sémantiquement un système à la Church. Mishra-Linger propose ensuite dans sa thèse [**ML08**] une variante où la conversion compare, comme dans AICC_Σ, des termes extraits. Notons également que dans sa thèse, Mishra-Linger considère des PTS plutôt qu'un Calcul des Constructions. Étendre nos travaux aux PTS pourrait être une extension intéressante et relativement rapide de notre travail.[14] Nous pourrions en fait même nous intéresser aux CTS définis par Barras [**Bar99**], qui sont une généralisation des PTS avec deux paramètres supplémentaires : une relation de conversion et une relation de cumulativité.

Enfin, plusieurs langages de programmation à types dépendants ont été proposés. Cayenne [**Aug98**] ajoute les types dépendants à Haskell ; le typage y est indécidable. DML [**XP99**] enrichit ML avec des types qui peuvent dépendre d'expressions arithmétiques. Son successeur, ATS [**CDX05**], permet de mieux séparer l'élaboration du programme (purement calculatoire) de celle des preuves. Des langages plus récents (Agda [**Nor07**], Epigram [**McB04**] et Idris [**Bra13**]) proposent les types dépendants dans toute leur généralité tout en gardant un typage décidable. Chacun d'entre eux a ses particularités mais, à nos yeux, deux éléments les relient entre eux et les distinguent de notre approche. Tout d'abord ce sont des systèmes à la Church alors que AICC_Σ se comporte comme un système à la Curry. Enfin, leur démarche est de chercher à améliorer les langages fonctionnels (ML, Haskell,...), très pratiques pour les programmeurs mais avec un système de types limité ; la nôtre est de vouloir rendre un assistant de preuves, avec un langage de types très riche, plus intuitif d'utilisation. Nous partageons cependant un but commun : définir un langage permettant d'élaborer intuitivement des programmes sûrs.

14. Les algorithmes d'inférence ne pourraient toutefois être définis que pour des PTS fonctionnels.

Annexes

Annexe A

Confluence des $\beta\iota$- et $\beta\iota\eta$-réductions

Le concept de réduction parallèle est habituellement utilisé dans les preuves de confluence de la β-réduction pour le λ-calcul pur [**Bar84**] ou pour des λ-calculs typés [**MLS84**]. Son invention a été attribuée à Tait par Martin-Löf.

Takahashi utilise dans [**Tak95**] des réductions parallèles pour démontrer élégamment d'autres résultats classiques des λ-calculs, dont le report de la η-réduction. Il présente également une nouvelle preuve pour la confluence, plus courte que celle de Tait-Martin-Löf.

Nous commençons par définir les $\beta\iota$-,η- et $\beta\iota\eta$-réductions parallèles et montrer leurs caractéristiques intéressantes. Puis dans une deuxième section, nous montrons la confluence des $\beta\iota$- et $\beta\iota\eta$-réductions.

La η-réduction parallèle, inutilisée dans les preuves de confluence présentées ici, sera utilisée pour la preuve du report de la η-réduction présentée dans l'annexe C.

1. Réductions parallèles

Pour une relation de réduction R donnée, la réduction R-parallèle, notée $\Longrightarrow_R$, consiste à réduire *simultanément* plusieurs R-redex présents dans un terme.

La réduction R-parallèle a deux propriétés intéressantes : elle est close par substitution [1] et, surtout, sa clôture réflexive et transitive est identique à $\twoheadrightarrow_R$.

1.1. Définitions.

1.1.1. DÉFINITION (Réductions parallèles).

- La $\beta\iota$-*réduction parallèle*, notée $\Longrightarrow_{\beta\iota}$, est définie inductivement par la réunion des règles de la figure 30 (avec $R = \beta\iota$) et de celles de la figure 31.
- La η-*réduction parallèle*, notée $\Longrightarrow_\eta$, est définie inductivement par la réunion des règles de la figure 30 (avec $R = \eta$) et de celle de la figure 32.
- La $\beta\iota\eta$-*réduction parallèle*, notée $\Longrightarrow_{\beta\iota}$, est définie inductivement par la réunion des règles de la figure 30 (avec $R = \beta\iota\eta$) et de celles de la figure 33.

1.1.2. DÉFINITION (Relation symétrique et clôture réflexive et transitive). Soit $R \in \{\beta\iota\eta, \beta\iota, \eta\}$.

- La *relation symétrique* de $\Longrightarrow_R$ est notée $\Longleftarrow_R$.
- La *clôture réflexive et transitive* de $\Longrightarrow_R$ est notée $\Longrightarrow_R^*$.

1. Si $M_0 \Longrightarrow_R M_1$ et $N_0 \Longrightarrow_R N_1$ alors $M_0[x/N_0] \Longrightarrow_R M_1[x/N_1]$.

$$\frac{}{x \Longrightarrow_{\mathrm{R}} x}\ (\text{R-VAR})$$

$$\frac{}{s \Longrightarrow_{\mathrm{R}} s}\ (\text{R-SORT})$$

$$\frac{\mathrm{T}_0 \Longrightarrow_{\mathrm{R}} \mathrm{T}_1 \qquad \mathrm{U}_0 \Longrightarrow_{\mathrm{R}} \mathrm{U}_1}{\Box x:\mathrm{T}_0.\mathrm{U}_0 \Longrightarrow_{\mathrm{R}} \Box x:\mathrm{T}_1.\mathrm{U}_1}\ (\text{R-}\Box)$$

$$\frac{\mathrm{M}_0 \Longrightarrow_{\mathrm{R}} \mathrm{M}_1}{\lambda x.\mathrm{M}_0 \Longrightarrow_{\mathrm{R}} \lambda x.\mathrm{M}_1}\ (\text{R-LAM})$$

$$\frac{\mathrm{M}_0 \Longrightarrow_{\mathrm{R}} \mathrm{M}_1 \qquad \mathrm{N}_0 \Longrightarrow_{\mathrm{R}} \mathrm{N}_1}{\mathrm{M}_0\,\mathrm{N}_0 \Longrightarrow_{\mathrm{R}} \mathrm{M}_1\,\mathrm{N}_1}\ (\text{R-APP})$$

$$\frac{a_0 \Longrightarrow_{\mathrm{R}} a_1 \qquad b_0 \Longrightarrow_{\mathrm{R}} b_1}{(a_0, b_0) \Longrightarrow_{\mathrm{R}} (a_1, b_1)}\ (\text{R-PAIR})$$

$$\frac{f_0 \Longrightarrow_{\mathrm{R}} f_1 \qquad c_0 \Longrightarrow_{\mathrm{R}} c_1}{\mathsf{Elim}_\Sigma(xy.f_0, c_0) \Longrightarrow_{\mathrm{R}} \mathsf{Elim}_\Sigma(xy.f_1, c_1)}\ (\text{R-Elim-}\Sigma)$$

$$\frac{b_0 \Longrightarrow_{\mathrm{R}} b_1}{(\diamond, b_0) \Longrightarrow_{\mathrm{R}} (\diamond, b_1)}\ (\text{R-}\exists)$$

$$\frac{f_0 \Longrightarrow_{\mathrm{R}} f_1 \qquad c_0 \Longrightarrow_{\mathrm{R}} c_1}{\mathsf{Elim}_\exists(y.f_0, c_0) \Longrightarrow_{\mathrm{R}} \mathsf{Elim}_\exists(y.f_1, c_1)}\ (\text{R-Elim-}\exists)$$

FIGURE 30. Règles communes aux réductions parallèles

$$\frac{\mathrm{M}_0 \Longrightarrow_{\beta\iota} \mathrm{M}_1 \qquad \mathrm{N}_0 \Longrightarrow_{\beta\iota} \mathrm{N}_1}{(\lambda x.\mathrm{M}_0)\,\mathrm{N}_0 \Longrightarrow_{\beta\iota} \mathrm{M}_1[x/\mathrm{N}_1]}\ (\beta\iota\text{-REDEX-}\beta)$$

$$\frac{f_0 \Longrightarrow_{\beta\iota} f_1 \qquad a_0 \Longrightarrow_{\beta\iota} a_1 \qquad b_0 \Longrightarrow_{\beta\iota} b_1}{\mathsf{Elim}_\Sigma(xy.f_0, (a_0, b_0)) \Longrightarrow_{\beta\iota} f_1[x/a_1][y/b_1]}\ (\beta\iota\text{-REDEX-}\iota_\Sigma)$$

$$\frac{f_0 \Longrightarrow_{\beta\iota} f_1 \qquad b_0 \Longrightarrow_{\beta\iota} b_1}{\mathsf{Elim}_\exists(y.f_0, (\diamond, b_0)) \Longrightarrow_{\beta\iota} f_1[y/b_1]}\ (\beta\iota\text{-REDEX-}\iota_\exists)$$

FIGURE 31. Règles propres à la βι-réduction parallèle

$$\frac{\mathrm{M}_0 \Longrightarrow_{\eta} \mathrm{M}_1 \qquad x \notin \mathsf{FV}(\mathrm{M}_0)}{\lambda x.\mathrm{M}_0\,x \Longrightarrow_{\eta} \mathrm{M}_1}\ (\eta\text{-REDEX-}\eta)$$

FIGURE 32. Règle propre à la η-réduction parallèle

1.2. Propriétés des réductions parallèles. Soit $\mathrm{R} \in \{\beta\iota\eta, \beta\iota, \eta\}$.

1.2.1. Lemme. *Soient* $\mathrm{M}_0, \mathrm{M}_1, \mathrm{N}_0, \mathrm{N}_1$ *des termes de ICC*$_\Sigma$.

(1) Si $\mathrm{M}_0 \to_{\mathrm{R}} \mathrm{M}_1$ *alors* $\mathrm{M}_0 \Longrightarrow_{\mathrm{R}} \mathrm{M}_1$.

(2) Si $\mathrm{M}_0 \Longrightarrow_{\mathrm{R}} \mathrm{M}_1$ *alors* $\mathrm{M}_0 \twoheadrightarrow_{\mathrm{R}} \mathrm{M}_1$.

(3) Si $\mathrm{M}_0 \Longrightarrow_{\mathrm{R}} \mathrm{M}_1$ *et* $\mathrm{N}_0 \Longrightarrow_{\mathrm{R}} \mathrm{N}_1$ *alors* $\mathrm{M}_0[x/\mathrm{N}_0] \Longrightarrow_{\mathrm{R}} \mathrm{M}_1[x/\mathrm{N}_1]$.

DÉMONSTRATION. Les trois points se montrent par induction sur M_0. Pour le troisième point, il faut également utiliser le lemme des substitutions multiples (lemme 1.2.13 page 10). □

$$\frac{M_0 \Longrightarrow_{\beta\iota\eta} M_1 \qquad N_0 \Longrightarrow_{\beta\iota\eta} N_1}{(\lambda x.M_0)\,N_0 \Longrightarrow_{\beta\iota\eta} M_1[x/N_1]}\ (\beta\iota\eta\text{-REDEX-}\beta)$$

$$\frac{f_0 \Longrightarrow_{\beta\iota\eta} f_1 \qquad a_0 \Longrightarrow_{\beta\iota\eta} a_1 \qquad b_0 \Longrightarrow_{\beta\iota\eta} b_1}{\mathsf{Elim}_\Sigma(xy.f_0,(a_0,b_0)) \Longrightarrow_{\beta\iota\eta} f_1[x/a_1][y/b_1]}\ (\beta\iota\eta\text{-REDEX-}\iota_\Sigma)$$

$$\frac{f_0 \Longrightarrow_{\beta\iota\eta} f_1 \qquad b_0 \Longrightarrow_{\beta\iota\eta} b_1}{\mathsf{Elim}_\exists(y.f_0,(\diamond,b_0)) \Longrightarrow_{\beta\iota\eta} f_1[y/b_1]}\ (\beta\iota\eta\text{-REDEX-}\iota_\exists)$$

$$\frac{M_0 \Longrightarrow_{\beta\iota\eta} M_1 \qquad x \notin \mathsf{FV}(M_0)}{\lambda x.M_0\,x \Longrightarrow_{\beta\iota\eta} M_1}\ (\beta\iota\eta\text{-REDEX-}\eta)$$

FIGURE 33. Règles propres à la βιη-réduction parallèle

1.2.2. Corollaire. *$\Longrightarrow_R^*$ est égale à $\twoheadrightarrow_R$.*

DÉMONSTRATION.

- Du point (1) du lemme précédent, nous déduisons par une récurrence immédiate que la relation $\twoheadrightarrow_R$ est incluse dans $\Longrightarrow_R^*$.
- $\twoheadrightarrow_R$ étant close par réflexivité et transitivité, nous déduisons du point (2) du lemme précédent, par une simple récurrence, que $\Longrightarrow_R^*$ est incluse dans $\twoheadrightarrow_R$.

□

2. Confluence

2.1. Présentation informelle de la preuve. Soit $R \in \{\beta\iota, \beta\iota\eta\}$. La preuve peut se décomposer en trois étapes.

(1) Définir une fonction de réduction simultanée intégrale qui réduit *tous* les R-redex d'un terme ;

(2) Utiliser cette fonction pour montrer que la relation parallèle associée $\Longrightarrow_R$ vérifie la propriété du diamant : si $M_1 \Longleftarrow_R M \Longrightarrow_R M_2$ alors il existe M_0 tel que $M_1 \Longrightarrow_R M_0 \Longleftarrow_R M_2$;

(3) En déduire par double récurrence la confluence de $\rightarrow_R$, puisque $\twoheadrightarrow_R$ est la clôture réflexive et transitive de $\Longrightarrow_R$.

2.2. Preuve.

2.2.1. DÉFINITION (Réduction simultanée intégrale).

- La $\beta\iota$-*réduction simultanée intégrale* d'un terme M, notée $|M|_{\beta\iota}$, est définie inductivement par la réunion des cas de la figure 34 (avec $R = \beta\iota$) et de celui de la figure 35.
- La $\beta\iota\eta$-*réduction simultanée intégrale* d'un terme M, notée $|M|_{\beta\iota\eta}$, est définie inductivement par la réunion des cas de la figure 34 (avec $R = \beta\iota\eta$) et de ceux de la figure 36.

Le résultat suivant est assez intuitif : il indique que si N est obtenu en réduisant *certains* R-redex de M, alors il est possible en réduisant *certains* R-redex de N d'obtenir $|M|_R$, obtenu en réduisant *tous* les R-redex de M. Puisque cela est vrai pour tout N et que $|M|_R$ est indépendant de N, nous pourrons utiliser $|M|_R$ pour prouver la propriété du diamant pour $\Longrightarrow_R$.

2.2.2. Lemme. *Pour $R \in \{\beta\iota, \beta\iota\eta\}$, pour tous termes M, N si $M \Longrightarrow_R N$, alors $N \Longrightarrow_R |M|_R$.*

$$
\begin{array}{rcll}
|x|_{\mathrm{R}} & = & x & \\
|s|_{\mathrm{R}} & = & s & \\
|\Box x:\mathrm{T}.\mathrm{U}|_{\mathrm{R}} & = & \Box x:|\mathrm{T}|_{\mathrm{R}}.|\mathrm{U}|_{\mathrm{R}} & \\
|\mathrm{M}\,\mathrm{N}|_{\mathrm{R}} & = & |\mathrm{M}|_{\mathrm{R}}\,|\mathrm{N}|_{\mathrm{R}} & \text{(si MN n'est pas un } \beta\text{-redex)} \\
|(\lambda x.\mathrm{M})\,\mathrm{N}|_{\mathrm{R}} & = & |\mathrm{M}|_{\mathrm{R}}\,[x/|\mathrm{N}|_{\mathrm{R}}] & \\
|(a,b)|_{\mathrm{R}} & = & (|a|_{\mathrm{R}},|b|_{\mathrm{R}}) & \\
|\mathsf{Elim}_{\Sigma}(xy.f,c)|_{\mathrm{R}} & = & \mathsf{Elim}_{\Sigma}(xy.|f|_{\mathrm{R}},|c|_{\mathrm{R}}) & \text{(si } \mathsf{Elim}_{\Sigma}(xy.f,c) \text{ n'est pas un } \iota_{\Sigma}\text{-redex)} \\
|\mathsf{Elim}_{\Sigma}(xy.f,(a,b))|_{\mathrm{R}} & = & |f|_{\mathrm{R}}\,[x/|a|_{\mathrm{R}}]\,[y/|b|_{\mathrm{R}}] & \\
|(\diamond,b)|_{\mathrm{R}} & = & (\diamond,|b|_{\mathrm{R}}) & \\
|\mathsf{Elim}_{\exists}(y.f,c)|_{\mathrm{R}} & = & \mathsf{Elim}_{\exists}(y.|f|_{\mathrm{R}},|c|_{\mathrm{R}}) & \text{(si } \mathsf{Elim}_{\exists}(y.f,c) \text{ n'est pas un } \iota_{\exists}\text{-redex)} \\
|\mathsf{Elim}_{\exists}(y.f,(\diamond,b))|_{\mathrm{R}} & = & |f|_{\mathrm{R}}\,[y/|b|_{\mathrm{R}}] &
\end{array}
$$

FIGURE 34. Cas communs à βι et βιη

$$|\lambda x.\mathrm{M}|_{\beta\iota} \;=\; \lambda x.|\mathrm{M}|_{\beta\iota}$$

FIGURE 35. Cas de l'abstraction pour βι

$$
\begin{array}{rcll}
|\lambda x.\mathrm{M}|_{\beta\iota\eta} & = & \lambda x.|\mathrm{M}|_{\beta\iota\eta} & \text{(si } \lambda x.\mathrm{M} \text{ n'est pas un } \eta\text{-redex)} \\
|\lambda x.\mathrm{M}\,x|_{\beta\iota\eta} & = & |\mathrm{M}|_{\beta\iota\eta} & \text{(si } x \notin \mathsf{FV}(\mathrm{M}))
\end{array}
$$

FIGURE 36. Cas de l'abstraction pour βιη

DÉMONSTRATION. Nous ne traitons que le cas de la βιη-réduction. Celui de la βι-réduction est similaire en plus simple.

Si $\mathrm{M} \Longrightarrow_{\beta\iota\eta} \mathrm{N}$, montrons par induction sur la structure de M que $\mathrm{N} \Longrightarrow_{\beta\iota\eta} |\mathrm{M}|_{\beta\iota\eta}$.

- Les cas de la variable et de la sorte sont triviaux ($\mathrm{M} = \mathrm{N} = |\mathrm{M}|_{\beta\iota\eta}$).
- Les cas où M n'est pas un βιη-redex se traitent immédiatement par induction car $\mathrm{M}, |\mathrm{M}|_{\beta\iota\eta}$ et N sont nécessairement de même nature.
- Les cas restants sont ceux où M est un βιη-redex. Nous ne traitons que le cas du η-redex, les autres cas sont analogues.

  Si $\mathrm{M} = \lambda x.\mathrm{M}_0\,x$ est un η-redex, alors $|\mathrm{M}|_{\beta\iota\eta} = |\mathrm{M}_0|_{\beta\iota\eta}$. Montrons que $\mathrm{N} \Longrightarrow_{\beta\iota\eta} |\mathrm{M}_0|_{\beta\iota\eta}$. Deux cas sont possibles selon la dernière règle appliquée dans $\mathrm{M} \Longrightarrow_{\beta\iota\eta} \mathrm{N}$.

  (1) La dernière règle est (βιη-REDEX-η).

  $$\frac{\mathrm{M}_0 \Longrightarrow_{\beta\iota\eta} \mathrm{N} \qquad x \notin \mathsf{FV}(\mathrm{M}_0)}{\lambda x.\mathrm{M}_0\,x \Longrightarrow_{\beta\iota\eta} \mathrm{N}}\ (\beta\iota\eta\text{-REDEX-}\eta)$$

  Nous concluons directement par hypothèse d'induction.

  (2) La dernière règle est (βιη-LAM). Il existe N_0 tel que $\mathrm{N} = \lambda x.\mathrm{N}_0$ et

  $$\frac{\mathrm{M}_0\,x \Longrightarrow_{\beta\iota\eta} \mathrm{N}_0}{\lambda x.\mathrm{M}_0\,x \Longrightarrow_{\beta\iota\eta} \lambda x.\mathrm{N}_0}\ (\beta\iota\eta\text{-LAM})$$

  Deux sous-cas sont possibles selon la dernière règle appliquée dans $\mathrm{M}_0\,x \Longrightarrow_{\beta\iota\eta} \mathrm{N}_0$.

  (a) $\mathrm{M}_0\,x$ est un β-redex et cette règle est (βιη-REDEX-β).

  Alors il existe une variable z et des termes $\mathrm{M}_1, \mathrm{N}_1$ tels que $\mathrm{M}_0 = \lambda z.\mathrm{M}_1$, $\mathrm{N}_0 = \mathrm{N}_1\,[z/x]$ et

$$\dfrac{\dfrac{M_1 \Longrightarrow_{\beta\iota\eta} N_1 \qquad x \Longrightarrow_{\beta\iota\eta} x}{(\lambda z.M_1)\,x \Longrightarrow_{\beta\iota\eta} N_1[z/x]}\ (\beta\iota\eta\text{-Redex-}\beta)}{\lambda x.(\lambda z.M_1)\,x \Longrightarrow_{\beta\iota\eta} \lambda x.(N_1[z/x])}\ (\beta\iota\eta\text{-Lam})$$

- Par hypothèse d'induction, nous avons $N_1 \Longrightarrow_{\beta\iota\eta} |M_1|_{\beta\iota\eta}$
- Nous avons $|M_0|_{\beta\iota\eta} = \lambda z.|M_1|_{\beta\iota\eta}$ et, par α-conversion, $N = \lambda x.(N_1[z/x]) = \lambda z.N_1$.
- Nous pouvons donc conclure en appliquant (βιη-Lam).

$$\dfrac{N_1 \Longrightarrow_{\beta\iota\eta} |M_1|_{\beta\iota\eta}}{\lambda z.N_1 \Longrightarrow_{\beta\iota\eta} \lambda z.|M_1|_{\beta\iota\eta}}\ (\beta\iota\eta\text{-Lam})$$

.

(b) La dernière règle appliquée est (βιη-App). Il existe N_1 tel que $N_0 = N_1\,x$ et nous avons la dérivation suivante.

$$\dfrac{\dfrac{M_0 \Longrightarrow_{\beta\iota\eta} N_0 \qquad x \Longrightarrow_{\beta\iota\eta} x}{M_0\,x \Longrightarrow_{\beta\iota\eta} N_0\,x}\ (\beta\iota\eta\text{-App})}{\lambda x.M_0\,x \Longrightarrow_{\beta\iota\eta} \lambda x.N_0\,x}\ (\beta\iota\eta\text{-Lam})$$

Montrons que $\lambda x.N_0\,x \Longrightarrow_{\beta\iota\eta} |M_0|_{\beta\iota\eta}$.

- Par hypothèse d'induction, $N_0 \Longrightarrow_{\beta\iota\eta} |M_0|_{\beta\iota\eta}$.
- Puisque $M_0 \twoheadrightarrow_{\beta\iota\eta} N_0$, nous avons $\mathsf{FV}(N_0) \subset \mathsf{FV}(M_0)$ d'où $x \notin \mathsf{FV}(N_0)$.
- Nous pouvons donc conclure en appliquant (βιη-Redex-η).

$$\dfrac{N_0 \Longrightarrow_{\beta\iota\eta} |M_0|_{\beta\iota\eta} \qquad x \notin \mathsf{FV}(N_0)}{\lambda x.N_0\,x \Longrightarrow_{\beta\iota\eta} |M_0|_{\beta\iota\eta}}\ (\beta\iota\eta\text{-Redex-}\eta)$$

□

2.2.3. Corollaire (Propriété du diamant pour $\Longrightarrow_R$)**.** *Soient* $R \in \{\beta\iota, \beta\iota\eta\}$ *et* M, M_1, M_2 *des termes de* ICC_Σ*. Si* $M_1 \Longleftarrow_R M \Longrightarrow_R M_2$ *alors il existe* M_0 *tel que* $M_1 \Longrightarrow_R M_0 \Longleftarrow_R M_2$.

Démonstration. D'après le lemme précédent, il suffit de prendre $M_0 = |M|_R$. □

2.2.4. Corollaire (Confluence de R)**.** *Soient* $R \in \{\beta\iota, \beta\iota\eta\}$ *et* M, M_1, M_2 *des termes de* ICC_Σ*. Si* $M_1 \twoheadleftarrow_R M \twoheadrightarrow_R M_2$ *alors il existe* M_0 *tel que* $M_1 \twoheadrightarrow_R M_0 \twoheadleftarrow_R M_2$.

Démonstration. Soit $R \in \{\beta\iota, \beta\iota\eta\}$.

- D'après le corollaire précédent, et par une récurrence sur le nombre de pas de réduction, nous montrons pour tous termes de ICC_Σ M, M_1, M_2 que si $M_1 \Longleftarrow_R M \Longrightarrow_R^* M_2$ alors il existe M_0 tel que $M_1 \Longrightarrow_R^* M_0 \Longleftarrow_R M_2$.
- Par une nouvelle récurrence, nous montrons que si $M_1 \Longleftarrow_R^* M \Longrightarrow_R^* M_2$ alors il existe M_0 tel que $M_1 \Longrightarrow_R^* M_0 \Longleftarrow_R^* M_2$.
- Nous concluons alors en remarquant que $\twoheadrightarrow_R$ est identique à $\Longrightarrow_R^*$.

□

Annexe B

Type sous-ensemble : règle d'élimination dépendante alternative

Nous considérons un système ICC_Σ' identique en tous points (syntaxe, réduction,...) à ICC_Σ mais où les règles de typage (Sub-E-1) et (Sub-E-2) sont remplacées par la règle d'élimination dépendante du type sous-ensemble :

$$\frac{\Gamma \vdash \mathrm{P} : \{x : \mathrm{A} \mid \mathrm{B}\} \to s \qquad \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} \qquad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\, x \qquad y \notin \mathsf{FV}(f)}{\Gamma \vdash f\,[x/c] : \mathrm{P}\, c}\ (\textsc{Sub-E})$$

Rappelons les règles (Sub-E-1) et (Sub-E-2).

$$\frac{\Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\}}{\Gamma \vdash c : \mathrm{A}}\ (\textsc{Sub-E-1})$$

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} \qquad \Gamma; y : \mathrm{B}\,[x/c] \vdash f : \mathrm{P} \qquad y \notin \mathsf{FV}(f)}{\Gamma \vdash f : \mathrm{P}}\ (\textsc{Sub-E-2})$$

Le but de cette annexe est d'étudier les liens entre ICC_Σ' et ICC_Σ. Nous montrons les résultats suivants :

(1) (Sub-E) est admissible dans ICC_Σ ;

(2) si ICC_Σ' vérifie certaines propriétés usuelles : inversion du type sous-ensemble et lemme du type des types, alors

   (a) (Sub-E-2) est admissible dans ICC_Σ' ;

   (b) (Sub-E-1) est admissible dans ICC_Σ'.

Nous déduisons du premier point que tout jugement dérivable de ICC_Σ' est dérivable dans ICC_Σ. Réciproquement, les deux points suivants montrent que, sous les conditions citées, tout jugement dérivable de ICC_Σ' est dérivable dans ICC_Σ.

Admissibilité de (Sub-E) dans ICC_Σ

Proposition. *La règle* (Sub-E) *est admissible dans* ICC_Σ.

Démonstration. La preuve est détaillée dans la figure 37. Intuitivement, nous substituons x par c dans la prémisse typant f pour nous ramener aux prémisses de (Sub-E-2). Notons que les deux règles (Sub-E-1) et (Sub-E-2) sont utilisées dans la dérivation. □

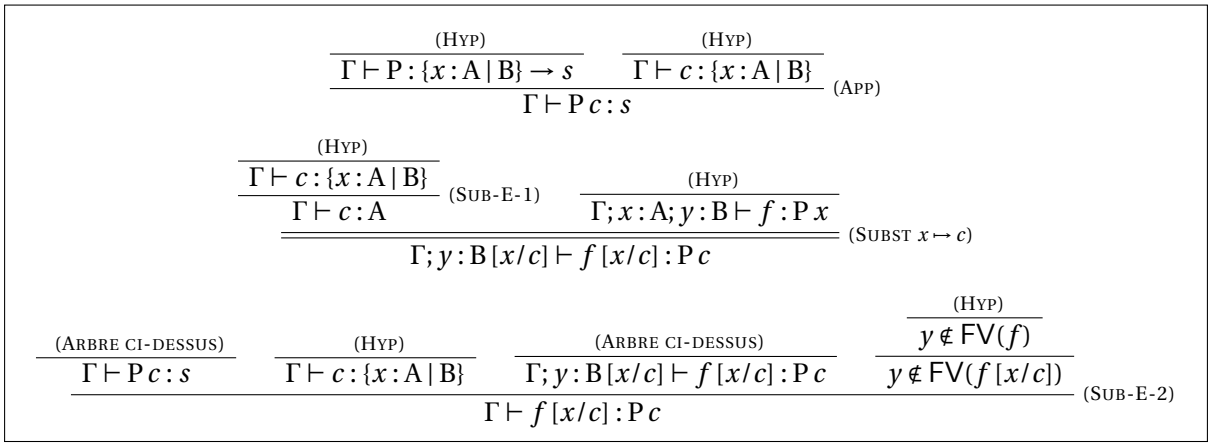

FIGURE 37. Admissibilité de (SUB-E) dans ICC_Σ

Admissibilité de (SUB-E-1) et (SUB-E-2) dans ICC_Σ'

Proposition. *Si les deux résultats suivants sont valables dans ICC_Σ' :*

- *inversion du type sous-ensemble : si* $\Gamma \vdash \{x : \mathrm{A} \mid \mathrm{B}\} : s$ *alors il existe des sortes* $s_\mathrm{A}, s_\mathrm{B}$ *telles que* $\Gamma \vdash \mathrm{A} : s_\mathrm{A}$ *et* $\Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_\mathrm{B}$
- *type des types : si* $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *alors il existe une sorte* s *telle que* $\Gamma \vdash \mathrm{T} : s$.

Alors les règles (SUB-E-1) *et* (SUB-E-2) *sont admissibles dans ICC_Σ'.*

DÉMONSTRATION.

- La dérivation prouvant le résultat se trouve dans la figure 38. [1] Intuitivement, il s'agit d'appliquer la règle (SUB-E) avec $f = x$ et $\mathrm{P} = \lambda z.\mathrm{A}$, où z est une variable fraîche.
- Nous voulons appliquer la règle (SUB-E)

$$\frac{\Gamma \vdash \mathrm{P}_0 : \{x : \mathrm{A} \mid \mathrm{B}\} \to s \qquad \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} \qquad \Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : \mathrm{P}_0\, x \qquad z \notin \mathsf{FV}(f)}{\Gamma \vdash f\,[x/c] : \mathrm{P}_0\, c}\ (\text{SUB-E})$$

  où z est une variable fraîche, $\mathrm{P}_0 = \lambda t.\mathrm{P}_1\, t$ avec $\mathrm{P}_1 = \lambda x.\mathrm{P}_2$ et $\mathrm{P}_2 = [\mathrm{B} \to \mathrm{B}\,[x/c]] \to \mathrm{P}$ et, par complétude de **Rule**, s_0 une sorte telle que $(s_\mathrm{B}, s, s_0) \in$ **Rule**.

  $z \notin \mathsf{FV}(f)$ est vraie car z est fraîche. $\Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\}$ est une des prémisses de (SUB-E-2). Il nous faut donc montrer les prémisses $\Gamma \vdash \mathrm{P}_0 : \{x : \mathrm{A} \mid \mathrm{B}\} \to s$ et $\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : \mathrm{P}_0\, x$. Les dérivations de ces prémisses sont détaillées dans les figures 39 et 40 ci-dessous. ((SUB-E-1) est admissible d'après le point précédent.) La troisième étape, détaillée dans la figure 41, consiste à appliquer (SUB-E), avec $f\,[x/c] = f$, et à en déduire $\Gamma \vdash f : \mathrm{P}$ par cumulativité et par une instanciation d'argument implicite.

□

1. La notation COND indique l'utilisation des deux résultats supposés vrais dans ICC_Σ'.

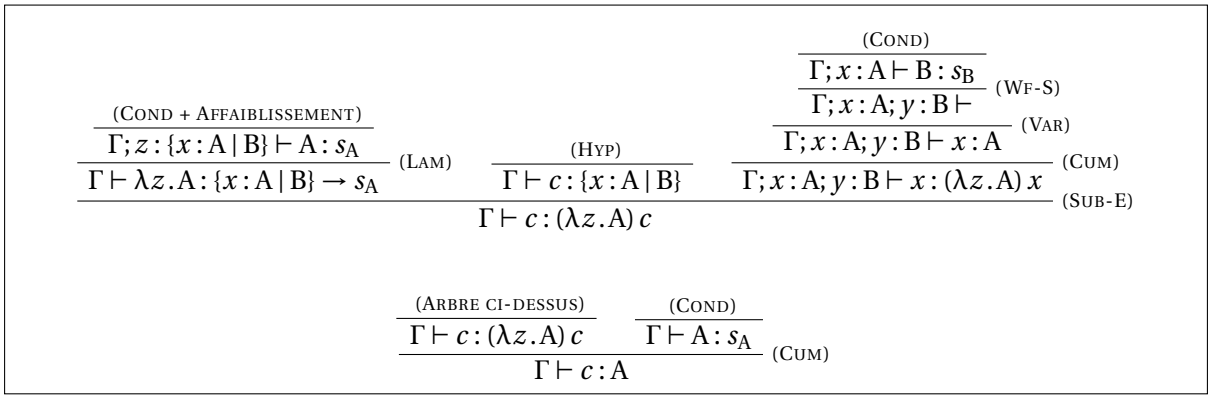

FIGURE 38. Admissibilité de (SUB-E-1) dans ICC$_\Sigma$'

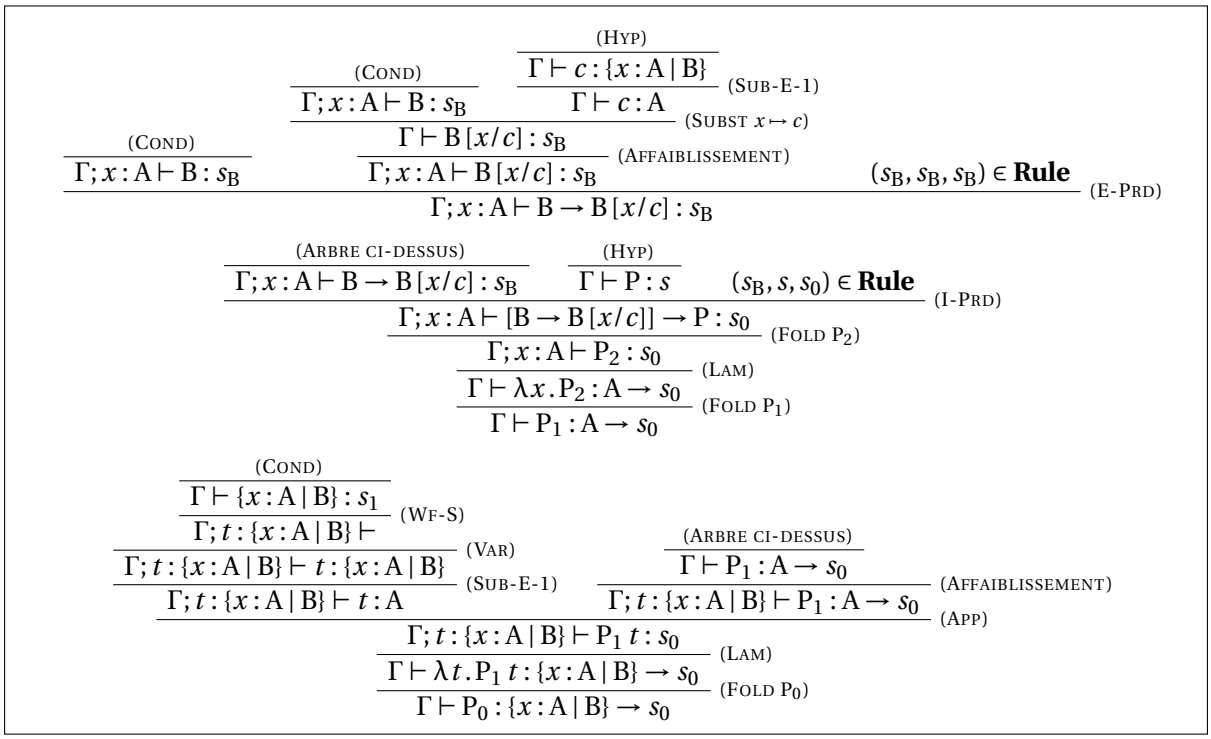

FIGURE 39. Admissibilité de (SUB-E-2) dans ICC$_\Sigma$' : étape 1

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma; y : \mathrm{B}\,[x/c] \vdash f : \mathrm{P}}\,(\textsc{Hyp}) \qquad \dfrac{}{y \notin \mathsf{FV}(f)}\,(\textsc{Hyp})}{\Gamma \vdash f : [\mathrm{B}\,[x/c]] \to \mathrm{P}}\,(\textsc{Gen}) \qquad \dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B}; \mathrm{H} : \mathrm{B} \to \mathrm{B}\,[x/c] \vdash}\,(\textsc{Cond} + \textsc{Wf-S} + \ldots)}{\Gamma; x : \mathrm{A}; z : \mathrm{B}; \mathrm{H} : \mathrm{B} \to \mathrm{B}\,[x/c] \vdash f : [\mathrm{B}\,[x/c]] \to \mathrm{P}}\,(\textsc{Aff.})}{}
$$

$$
\dfrac{\dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B}; \mathrm{H} : \mathrm{B} \to \mathrm{B}\,[x/c] \vdash f : [\mathrm{B}\,[x/c]] \to \mathrm{P}}\,(\textsc{Arbre ci-dessus}) \qquad \dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B}; \mathrm{H} : \mathrm{B} \to \mathrm{B}\,[x/c] \vdash \mathrm{H}\,z : \mathrm{B}\,[x/c]}\,(\textsc{Cond} + \textsc{Wf-S} + \textsc{Var} + \ldots)}{\Gamma; x : \mathrm{A}; z : \mathrm{B}; \mathrm{H} : \mathrm{B} \to \mathrm{B}\,[x/c] \vdash f : \mathrm{P}}\,(\textsc{App})\,(\textsc{Inst})
$$

$$
\dfrac{\dfrac{\dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B}; \mathrm{H} : \mathrm{B} \to \mathrm{B}\,[x/c] \vdash f : \mathrm{P}}\,(\textsc{Arbre ci-dessus}) \qquad \dfrac{}{\mathrm{H} \notin \mathsf{FV}(f)}\,(\textsc{H fraîche})}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : [\mathrm{B} \to \mathrm{B}\,[x/c]] \to \mathrm{P}}\,(\textsc{Gen})}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : \mathrm{P}_2}\,(\textsc{Fold P}_2)
$$

$$
\dfrac{\dfrac{\dfrac{}{\Gamma; x : \mathrm{A} \vdash \mathrm{P}_0 : \{x : \mathrm{A} \mid \mathrm{B}\} \to s_0}\,(\textsc{Étape 1}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \mathrm{A} : s_{\mathrm{A}}}\,(\textsc{Cond})}{\Gamma; x : \mathrm{A} \vdash}\,(\textsc{Wf-S})}{\Gamma; x : \mathrm{A} \vdash x : \mathrm{A}}\,(\textsc{Var})}{\Gamma; x : \mathrm{A} \vdash x : \{x : \mathrm{A} \mid \mathrm{B}\}}\,(\textsc{Sub-I})}{\Gamma; x : \mathrm{A} \vdash \mathrm{P}_0\,x : s_0}\,(\textsc{App})}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash \mathrm{P}_0\,x : s_0}\,(\textsc{Aff.})
$$

$$
\dfrac{\dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : \mathrm{P}_2}\,(\textsc{Arbre ci-dessus}) \qquad \dfrac{\mathrm{P}_0\,x \twoheadrightarrow_\beta \mathrm{P}_2}{\mathrm{P}_2 \preceq \mathrm{P}_0\,x} \qquad \dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash \mathrm{P}_0\,x : s_0}\,(\textsc{Arbre ci-dessus})}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : \mathrm{P}_0\,x}\,(\textsc{Cum})
$$

FIGURE 40. Admissibilité de (SUB-E-2) dans ICC_Σ' : étape 2

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \mathrm{P}_0 : \{x : \mathrm{A} \mid \mathrm{B}\} \to s}\,(\textsc{Étape 1}) \quad \dfrac{}{\Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\}}\,(\textsc{Hyp}) \quad \dfrac{}{\Gamma; x : \mathrm{A}; z : \mathrm{B} \vdash f : \mathrm{P}_0\,x}\,(\textsc{Étape 2}) \quad \dfrac{}{z \notin \mathsf{FV}(f)}\,(z\ \textsc{fraîche})}{\Gamma \vdash f : \mathrm{P}_0\,c}\,(\textsc{Sub-E})}{\Gamma \vdash f : (\lambda t\,.\,(\lambda x\,.\,[\mathrm{B} \to \mathrm{B}\,[x/c]] \to \mathrm{P})\;t)\;c}\,(\textsc{Unfold P}_0)}{\Gamma \vdash f : [\mathrm{B}\,[x/c] \to \mathrm{B}\,[x/c]] \to \mathrm{P}}\,(\textsc{Cum})
$$

$$
\dfrac{\dfrac{}{\Gamma \vdash f : [\mathrm{B}\,[x/c] \to \mathrm{B}\,[x/c]] \to \mathrm{P}}\,(\textsc{Arbre ci-dessus}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \mathrm{B}\,[x/c] : s_{\mathrm{B}}}\,(\textsc{Déja fait dans étape 1})}{\Gamma; y : \mathrm{B}\,[x/c] \vdash}\,(\textsc{Wf-S})}{\Gamma; y : \mathrm{B}\,[x/c] \vdash y : \mathrm{B}\,[x/c]}\,(\textsc{Var})}{\Gamma \vdash \lambda y\,.\,y : \mathrm{B}\,[x/c] \to \mathrm{B}\,[x/c]}\,(\textsc{Lam})}{\Gamma \vdash f : \mathrm{P}}\,(\textsc{Inst})
$$

FIGURE 41. Admissibilité de (SUB-E-2) dans ICC_Σ' : étape 3

Annexe C

Report de η

Nous adaptons ici la preuve faite par Takahashi [**Tak95**]. Nous l'avons remarqué plus haut, le report de η est faux en général. Nous montrons donc un résultat plus faible : le report de η pour les termes bien typés. Pour cela nous utilisons notamment la préservation du typage par βι-réduction.

1. η-expansion d'ordre k

1.1. Définition.

1.1.1. DÉFINITION (η-expansion d'ordre k). Soit k un entier positif et M un terme de ICC_Σ, et $z_1, \ldots, z_k \notin \mathsf{FV}(\mathrm{M})$ si k est non-nul. L'η-*expansion d'ordre k* de M, notée $(\mathrm{M})_k$, est définie par

$$(\mathrm{M})_k = \lambda z_1 . (\lambda z_2 . (\ldots (\lambda z_k . \mathrm{M}\, z_k) \ldots z_2))\, z_1$$

si k est non-nul et $(\mathrm{M})_0 = \mathrm{M}$ sinon.

1.1.2. REMARQUE. Par une récurrence immédiate sur k, nous montrons, en appliquant (η-REDEX-η), que si $\mathrm{M} \Longrightarrow_\eta \mathrm{N}$ alors $(\mathrm{M})_k \Longrightarrow_\eta \mathrm{N}$. En particulier, nous avons $(\mathrm{M})_k \Longrightarrow_\eta \mathrm{M}$.

1.2. Propriétés.

1.2.1. Lemme (Inversion de la η-réduction parallèle)**.** *Soient* $\mathrm{R}_1, \mathrm{R}_2$ *des termes de* ICC_Σ *tels que* $\mathrm{R}_1 \Longrightarrow_\eta \mathrm{R}_2$.

(1) $\mathrm{R}_2 = x$ *si et seulement si il existe* $k \geq 0$ *tel que* $\mathrm{R}_1 = (x)_k$.

(2) $\mathrm{R}_2 = s$ *si et seulement si il existe* $k \geq 0$ *tel que* $\mathrm{R}_1 = (s)_k$.

(3) $\mathrm{R}_2 = \Box x : \mathrm{T}_2 . \mathrm{U}_2$ *si et seulement si il existe des termes* $\mathrm{T}_1, \mathrm{U}_1$ *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = (\Box x : \mathrm{T}_1 . \mathrm{U}_1)_k$, $\mathrm{T}_1 \Longrightarrow_\eta \mathrm{T}_2$ *et* $\mathrm{U}_1 \Longrightarrow_\eta \mathrm{U}_2$.

(4) $\mathrm{R}_2 = \lambda x . \mathrm{M}_2$ *si et seulement si il existe un terme* M_1 *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = (\lambda x . \mathrm{M}_1)_k$ *et* $\mathrm{M}_1 \Longrightarrow_\eta \mathrm{M}_2$.

(5) $\mathrm{R}_2 = \mathrm{M}_2\, \mathrm{N}_2$ *si et seulement si il existe des termes* $\mathrm{M}_1, \mathrm{N}_1$ *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = (\mathrm{M}_1\, \mathrm{N}_1)_k$, $\mathrm{M}_1 \Longrightarrow_\eta \mathrm{M}_2$ *et* $\mathrm{N}_1 \Longrightarrow_\eta \mathrm{N}_2$.

(6) $\mathrm{R}_2 = (a_2\,, b_2)$ *si et seulement si il existe des termes* a_1, b_1 *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = ((a_1\,, b_1))_k$, $a_1 \Longrightarrow_\eta a_2$ *et* $b_1 \Longrightarrow_\eta b_2$.

(7) $\mathrm{R}_2 = \mathsf{Elim}_\Sigma(xy.f_2, c_2)$ *si et seulement si il existe des termes* c_1, f_1 *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = (\mathsf{Elim}_\Sigma(xy.f_1, c_1))_k$, $f_1 \Longrightarrow_\eta f_2$ *et* $c_1 \Longrightarrow_\eta c_2$.

(8) $\mathrm{R}_2 = (\diamond, b_2)$ *si et seulement si il existe un terme* b_1 *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = ((\diamond, b_1))_k$ *et* $b_1 \Longrightarrow_\eta b_2$.

(9) $\mathrm{R}_2 = \mathsf{Elim}_\exists(y.f_2, c_2)$ *si et seulement si il existe des termes* c_1, f_1 *et un entier* $k \geq 0$ *tels que* $\mathrm{R}_1 = (\mathsf{Elim}_\exists(y.f_1, c_1))_k$, $f_1 \Longrightarrow_\eta f_2$ *et* $c_1 \Longrightarrow_\eta c_2$.

DÉMONSTRATION. Les réciproques sont immédiates : il suffit d'appliquer la remarque 1.1.2. Dans le sens direct, les preuves sont similaires. Traitons le cas de l'éliminateur de somme dépendante.

Si $\mathrm{R}_1 \Longrightarrow_\eta \mathsf{Elim}_\Sigma(xy.f_2, c_2)$, montrons qu'il existe un entier $k \geq 0$ et des termes c_1, f_1 tels que $\mathrm{R}_1 = (\mathsf{Elim}_\Sigma(xy.f_1, c_1))_k$, $f_1 \Longrightarrow_\eta f_2$ et $c_1 \Longrightarrow_\eta c_2$.

- La dernière règle appliquée dans la dérivation de $\mathrm{R}_1 \Longrightarrow_\eta \mathsf{Elim}_\Sigma(xy.f_2, c_2)$ est (η-Elim-Σ) ou (η-Redex-η).
- Soit k le nombre d'applications de la règle (η-Redex-η) lorsque nous remontons la dérivation. Nous pouvons construire une suite de termes $(\mathrm{R}_1^i)_{0\leq i\leq k}$ tels que $\mathrm{R}_1^0 = \mathrm{R}_1$, et, pour tout i, $\mathrm{R}_1^i = (\mathrm{R}^{i+1})_1$ et $\mathrm{R}_1^i \Longrightarrow_\eta \mathsf{Elim}_\Sigma(xy.f_2, c_2)$.
- La dernière règle de $\mathrm{R}_1^k \Longrightarrow_\eta \mathsf{Elim}_\Sigma(xy.f_2, c_2)$ est (η-Elim-Σ), d'où l'existence de f_1 et c_1 tels que $\mathrm{R}_1^k = \mathsf{Elim}_\Sigma(xy.f_1, c_1)$ avec $\mathrm{R}_1 = (\mathrm{R}_1^k)_k = (\mathsf{Elim}_\Sigma(xy.f_1, c_1))_k$, $f_1 \Longrightarrow_\eta f_2$ et $c_1 \Longrightarrow_\eta c_2$.

□

1.2.2. Lemme (η-expansion et βι-réduction parallèle)**.** *Soient k un entier, x, y des variables et $\mathrm{M}, \mathrm{M}', \mathrm{N}, \mathrm{N}'$ des termes annotés tels que $\mathrm{M} \Longrightarrow_{\beta\iota} \mathrm{M}'$, et $\mathrm{N} \Longrightarrow_{\beta\iota} \mathrm{N}'$. Nous avons :*

(1) $(\mathrm{M})_k \Longrightarrow_{\beta\iota} (\mathrm{M}')_k$

(2) $(\lambda x.\mathrm{M})_k \mathrm{N} \Longrightarrow_{\beta\iota} \mathrm{M}'[x/\mathrm{N}']$

Démonstration.

(1) Par récurrence sur k.
- Le cas $k = 0$ est immédiat.
- Supposons $(\mathrm{M})_k \Longrightarrow_{\beta\iota} (\mathrm{M}')_k$ et montrons que $(\mathrm{M})_{k+1} \Longrightarrow_{\beta\iota} (\mathrm{M}')_{k+1}$.
  - Soit z une variable telle que $z \notin \mathsf{FV}(\mathrm{M}) \cup \mathsf{FV}(\mathrm{M}')$. Nous avons $(\mathrm{M})_{k+1} = \lambda z.(\mathrm{M})_k z$ et $(\mathrm{M}')_{k+1} = \lambda z.(\mathrm{M}')_k z$.
  - L'arbre de dérivation suivant montre que $\lambda z.(\mathrm{M})_k z \Longrightarrow_{\beta\iota} \lambda z.(\mathrm{M}')_k z$.

$$\dfrac{\dfrac{\dfrac{}{(\mathrm{M})_k \Longrightarrow_{\beta\iota} (\mathrm{M}')_k}\,\textsc{Hyp} \qquad \dfrac{}{z \Longrightarrow_{\beta\iota} z}\,(\beta\iota\text{-Var})}{(\mathrm{M})_k z \Longrightarrow_{\beta\iota} (\mathrm{M}')_k z}\,(\beta\iota\text{-App})}{\lambda z.(\mathrm{M})_k z \Longrightarrow_{\beta\iota} \lambda z.(\mathrm{M}')_k z}\,(\beta\iota\text{-Lam})$$

(2) Par récurrence sur k.
- Si $k = 0$, il suffit d'appliquer la règle (βι-Redex-β).
- Si le résultat est vrai pour k, montrons-le pour $k+1$. Montrons que $(\lambda x.\mathrm{M})_{k+1}\mathrm{N} \Longrightarrow_{\beta\iota} \mathrm{M}'[x/\mathrm{N}']$.
  - Soit z une variable fraîche n'apparaissant pas dans $\mathrm{M}, \mathrm{M}', \mathrm{N}, \mathrm{N}'$. Nous avons $(\lambda x.\mathrm{M})_{k+1} = \lambda z.(\lambda x.\mathrm{M})_k z$.
  - Par hypothèse de récurrence avec $\mathrm{M} \Longrightarrow_{\beta\iota} \mathrm{M}'$ et $z \Longrightarrow_{\beta\iota} z$ nous avons $(\lambda x.\mathrm{M})_k z \Longrightarrow_{\beta\iota} \mathrm{M}'[x/z]$.
  - Nous déduisons donc, en appliquant (βι-Redex-β),

$$\dfrac{(\lambda x.\mathrm{M})_k z \Longrightarrow_{\beta\iota} \mathrm{M}'[x/z] \qquad \mathrm{N} \Longrightarrow_{\beta\iota} \mathrm{N}'}{(\lambda z.(\lambda x.\mathrm{M})_k z)\,\mathrm{N} \Longrightarrow_{\beta\iota} \mathrm{M}'[x/z][z/\mathrm{N}']}\,(\beta\iota\text{-Redex-}\beta)$$

  - D'après le lemme des substitutions multiples, nous avons, puisque $z \notin \mathsf{FV}(\mathrm{M}')$,

$$\mathrm{M}'[x/z][z/\mathrm{N}'] = \mathrm{M}'[z/\mathrm{N}'][x/z[z/\mathrm{N}']] = \mathrm{M}'[x/\mathrm{N}]$$

    Nous avons donc bien $(\lambda x.\mathrm{M})_{k+1}\mathrm{N} \Longrightarrow_{\beta\iota} \mathrm{M}'[x/\mathrm{N}']$.

□

1.2.3. Remarque. Il n'y a pas d'équivalent du point (2) pour les éliminateurs des types somme. Ainsi $\mathsf{Elim}_\Sigma(xy.f, ((\mathrm{M}, \mathrm{N}))_k) \Longrightarrow_{\beta\iota} f'[x/\mathrm{M}'][y/\mathrm{N}']$ n'est pas vérifié. Il suffit de prendre le contre-exemple du report de η $\mathsf{Elim}_\Sigma(xy.(x\,y), ((x, y))_1) = \mathsf{Elim}_\Sigma(xy.(x\,y), \lambda z.(x, y)\,z)$. De même nous n'avons pas $\mathsf{Elim}_\exists(y.f, ((\diamond, \mathrm{N}))_k) \Longrightarrow_{\beta\iota} f'[y/\mathrm{N}']$.

2. Preuve principale

2.1. Condition de report de η.

2.1.1. DÉFINITION. Soit M un terme de ICC_Σ. M vérifie la *condition de report de* η si M ne contient pas de sous-terme (non-strict) de la forme $\mathsf{Elim}_\Sigma(xy.f, \lambda x.\mathrm{R})$ ou $\mathsf{Elim}_\exists(y.f, \lambda x.\mathrm{R})$.

2.1.2. Lemme. *La condition de report de* η *est stable par* $\Longrightarrow_\eta$.

DÉMONSTRATION. Si $\mathrm{M} \Longrightarrow_\eta \mathrm{N}$ et si M vérifie la condition de report, montrons que N la vérifie également. Nous procédons par induction en considérant la dernière règle appliquée. Tous les cas sont immédiats sauf (η-REDEX-ι_Σ) et (η-REDEX-$\iota_\exists$). Dans ces cas-là si l'objet éliminé c_1 est une application, alors par inversion c_0 l'est également ce qui contredit le fait que M vérifie la condition de report. □

2.1.3. REMARQUE. La condition de report n'est pas préservée par $\rightarrow_{\beta\iota}$, donc *a fortiori* pas préservée non plus par $\Longrightarrow_{\beta\iota}$. $\mathrm{M} = \mathsf{Elim}_\Sigma(xy.f, (\lambda z.(\lambda x.\mathrm{R})\, z)$ est un contre-exemple.

2.1.4. Lemme. *Si* $\Gamma \vdash \mathrm{M} : \mathrm{T}$ *alors* M *vérifie la condition de report de* η.

DÉMONSTRATION. Tout sous-terme d'un terme bien typé est bien typé. Or, d'après les lemmes d'inversion faible, si $\Delta \vdash \mathsf{Elim}_\Sigma(xy.f, \lambda x.\mathrm{M}) : \mathrm{R}$ (resp. $\Delta \vdash \mathsf{Elim}_\exists(y.f, \lambda x.\mathrm{M}) : \mathrm{R}$) alors il existe $\Delta_0, z, \mathrm{A}, \mathrm{B}$ tels que $\Delta; \Delta_0 \vdash \lambda x.\mathrm{M} : \Sigma z : \mathrm{A}.\mathrm{B}$ (resp. $\Delta; \Delta_0 \vdash \lambda x.\mathrm{M} : \exists z : \mathrm{A}.\mathrm{B}$), ce qui contredit le lemme 3.4.18 page 31. □

2.2. Lemme auxiliaire.

2.2.1. Lemme (Lemme principal)**.** *Soient* $\mathrm{M}, \mathrm{N}, \mathrm{P}$ *des termes tels que* $\mathrm{M} \Longrightarrow_\eta \mathrm{P} \Longrightarrow_{\beta\iota} \mathrm{N}$. *Si* M *vérifie la condition de report de* η*, alors il existe un terme* P' *tel que* $\mathrm{M} \Longrightarrow_{\beta\iota} \mathrm{P}' \Longrightarrow_\eta \mathrm{N}$.

DÉMONSTRATION. Par induction sur la structure de P. Nous suivons la définition de $\Longrightarrow_{\beta\iota}$ et considérons douze cas que nous regroupons en trois groupes.

(1) P est une variable ou une sorte (n'a pas de sous-termes stricts) : alors P = N et $\mathrm{P}' = \mathrm{M}$ convient.

(2) P a des sous-termes stricts et la dernière règle appliquée dans $\mathrm{P} \Longrightarrow_{\beta\iota} \mathrm{N}$ n'est pas celle d'un βι-redex. Les hypothèses d'induction liées aux sous-termes de P permettent de conclure facilement. Traitons par exemple le cas du constructeur de type : $\mathrm{M} \Longrightarrow_\eta \Box x : \mathrm{P}_0.\mathrm{P}_1 \Longrightarrow_{\beta\iota} \mathrm{N}$.

- La dernière règle de $\Box x : \mathrm{P}_0.\mathrm{P}_1 \Longrightarrow_{\beta\iota} \mathrm{N}$ est nécessairement (βι-□). Il existe donc $\mathrm{N}_0, \mathrm{N}_1$ tels que $\mathrm{N} = \Box x : \mathrm{N}_0.\mathrm{N}_1$ et $\mathrm{P}_i \Longrightarrow_{\beta\iota} \mathrm{N}_i$ pour $i \in \{0,1\}$.
- Par inversion de $\mathrm{M} \Longrightarrow_\eta \Box x : \mathrm{P}_0.\mathrm{P}_1$ (point (3) du lemme 1.2.1), il existe un entier k et des termes $\mathrm{M}_0, \mathrm{M}_1$ tels que $\mathrm{M} = (\Box x : \mathrm{M}_0.\mathrm{M}_1)_k$ et $\mathrm{M}_i \Longrightarrow_\eta \mathrm{P}_i$ pour $i \in \{0,1\}$.
- Nous avons donc, pour $i \in \{0,1\}$, $\mathrm{M}_i \Longrightarrow_\eta \mathrm{P}_i \Longrightarrow_{\beta\iota} \mathrm{N}_i$. Par hypothèses d'induction, il existe P'_i tel que $\mathrm{M}_i \Longrightarrow_{\beta\iota} \mathrm{P}'_i \Longrightarrow_\eta \mathrm{N}_i$.
- Posons $\mathrm{P}' = (\Box x : \mathrm{P}'_0.\mathrm{P}'_1)_k$ et montrons que $\mathrm{M} \Longrightarrow_{\beta\iota} \mathrm{P}' \Longrightarrow_\eta \mathrm{N}$.
  - Montrons, en utilisant le point (1) du lemme 1.2.2, que $\mathrm{M} \Longrightarrow_{\beta\iota} \mathrm{P}'$.

$$\dfrac{\dfrac{\mathrm{M}_0 \Longrightarrow_{\beta\iota} \mathrm{P}'_0 \qquad \mathrm{M}_1 \Longrightarrow_{\beta\iota} \mathrm{P}'_1}{\Box x : \mathrm{M}_0.\mathrm{M}_1 \Longrightarrow_{\beta\iota} \Box x : \mathrm{P}'_0.\mathrm{P}'_1}\ (\beta\iota\text{-}\Box)}{(\Box x : \mathrm{M}_0.\mathrm{M}_1)_k \Longrightarrow_{\beta\iota} (\Box x : \mathrm{P}'_0.\mathrm{P}'_1)_k}\ 1.2.2.(1)$$

  - Montrons, en appliquant la remarque 1.1.2, que $\mathrm{P}' \Longrightarrow_\eta \mathrm{N}$.

$$\dfrac{\dfrac{P'_0 \Longrightarrow_\eta N_0 \qquad P'_1 \Longrightarrow_\eta N_1}{\Box x : P'_0 . P'_1 \Longrightarrow_\eta \Box x : N_0 . N_1}\ (\eta\text{-}\Box)}{(\Box x : P'_0 . P'_1)_k \Longrightarrow_\eta \Box x : N_0 . N_1}\ 1.1.2$$

(3) La dernière règle appliquée dans $P \Longrightarrow_{\beta\iota} N$ est celle d'un βι-redex.

(a) P est un β-redex.

- Il existe P_0, P_1, N_0, N_1 tels que $P = (\lambda x . P_0)\, P_1$, $N = N_0\,[x/N_1]$ et

$$\dfrac{P_0 \Longrightarrow_{\beta\iota} N_0 \qquad P_1 \Longrightarrow_{\beta\iota} N_1}{(\lambda x . P_0)\, P_1 \Longrightarrow_{\beta\iota} N_0\,[x/N_1]}\ (\beta\iota\text{-Redex-}\beta)$$

- Par inversion de $M \Longrightarrow_\eta (\lambda x . P_0)\, P_1$ (point (5) du lemme 1.2.1) il existe un entier l et des termes M'_0, M_1 tels que $M = (M'_0\, M_1)_l$ $M'_0 \Longrightarrow_\eta \lambda x . P_0$ et $M_1 \Longrightarrow_\eta P_1$.
- Par inversion de $M'_0 \Longrightarrow_\eta \lambda x . P_0$ (point (4) du lemme 1.2.1), il existe un entier k et un terme M_0 tel que $M'_0 = (\lambda x . M_0)_k$ et $M_0 \Longrightarrow_\eta P_0$.
- Nous avons donc $M = ((\lambda x . M_0)_k\, M_1)_l$ et, pour $i \in \{0,1\}$, $M_i \Longrightarrow_\eta P_i \Longrightarrow_{\beta\iota} N_i$.
- Par hypothèses d'induction, il existe P'_i tel que $M_i \Longrightarrow_{\beta\iota} P'_i \Longrightarrow_\eta N_i$, pour $i \in \{0,1\}$.
- Posons $P' = (P'_0\,[x/P'_1])_l$ et montrons que $M \Longrightarrow_{\beta\iota} P' \Longrightarrow_\eta N$.
  - Montrons, grâce aux points (2) et (1) du lemme 1.2.2, que $M \Longrightarrow_{\beta\iota} P'$.

$$\dfrac{\dfrac{M_0 \Longrightarrow_{\beta\iota} P'_0 \qquad M_1 \Longrightarrow_{\beta\iota} P'_1}{(\lambda x . M_0)_k\, M_1 \Longrightarrow_{\beta\iota} P'_0\,[x/P'_1]}\ 1.2.2.(2)}{((\lambda x . M_0)_k\, M_1)_l \Longrightarrow_{\beta\iota} (P'_0\,[x/P'_1])_l}\ 1.2.2.(1)$$

  - Montrons, grâce à la clôture par substitution de $\Longrightarrow_\eta$ et la remarque 1.1.2, que $P' \Longrightarrow_\eta N$.

$$\dfrac{\dfrac{P'_0 \Longrightarrow_\eta N_0 \qquad P'_1 \Longrightarrow_\eta N_1}{P'_0\,[x/P'_1] \Longrightarrow_\eta N_0\,[x/N_1]}\ \text{Clôt. Subst.}}{(P'_0\,[x/P'_1])_l \Longrightarrow_\eta N_0\,[x/N_1]}\ 1.1.2$$

(b) P est un ι_Σ-redex. Ce cas se traite presque comme celui du β-redex. Une différence importante est due à l'absence d'équivalent du point (2) du lemme 1.2.2 pour les éliminateurs des types somme (cf. remarque 1.2.3).

- Il existe des termes $P_0, P_1, P_2, N_0, N_1, N_2$ tels que $P = \mathsf{Elim}_\Sigma(xy.P_0, (P_1\,,\, P_2))$ et $N = N_0\,[x/N_1]\,[y/N_2]$.
- Par double inversion de $M \Longrightarrow_\eta P$ (points (7) puis (6) du lemme 1.2.1), nous avons des entiers k, l et des termes M_0, M_1, M_2 tels que $M = (\mathsf{Elim}_\Sigma(xy.M_0, ((M_1\,,\, M_2))_k))_l$ et $M_i \Longrightarrow_\eta P_i$ pour $i \in \{0,1,2\}$.
- Par hypothèses d'induction, nous avons, pour $i \in \{0,1,2\}$, P'_i tel que $M_i \Longrightarrow_{\beta\iota} P'_i \Longrightarrow_\eta N_i$. Nous posons alors $P' = (P'_0\,[x/P'_1]\,[y/P'_2])_l$.
- De même que pour le cas du β-redex, nous montrons, grâce à la clôture par substitution de $\Longrightarrow_\eta$ et la remarque 1.1.2, que $P' \Longrightarrow_\eta N$.
- Puisque M vérifie la condition de report de η, nous avons $k = 0$.
- Nous montrons grâce au point (1) du lemme 1.2.2 que $M \Longrightarrow_{\beta\iota} P'$.

$$\dfrac{\dfrac{M_0 \Longrightarrow_{\beta\iota} P'_0 \qquad M_1 \Longrightarrow_{\beta\iota} P'_1 \qquad M_2 \Longrightarrow_{\beta\iota} P'_2}{\mathsf{Elim}_\Sigma(xy.M_0, (M_1\,,\, M_2)) \Longrightarrow_{\beta\iota} P'_0\,[x/P'_1]\,[y/P'_2]}\ (\beta\iota\text{-Redex-}\iota_\Sigma)}{(\mathsf{Elim}_\Sigma(xy.M_0, (M_1\,,\, M_2)))_l \Longrightarrow_{\beta\iota} (P'_0\,[x/P'_1]\,[y/P'_2])_l}\ 1.2.2.(1)$$

(c) P est un $\iota_\exists$-redex. Similaire, en plus simple, au cas précédent.

□

2.2.2. Corollaire.

(1) Soient M, N, P *des termes tels que* $M \Longrightarrow_\eta^* P \Longrightarrow_{\beta\iota} N$. *Si* M *vérifie la condition de report de* η, *alors il existe un terme* P' *tel que* $M \Longrightarrow_{\beta\iota} P' \Longrightarrow_\eta^* N$.

(2) Soient M, N, P *des termes tels que* $M \Longrightarrow_\eta^* P \Longrightarrow_{\beta\iota}^* N$. *Si* M *vérifie la condition de report de* η, *alors il existe un terme* P' *tel que* $M \Longrightarrow_{\beta\iota}^* P' \Longrightarrow_\eta^* N$.

DÉMONSTRATION.

(1) Par récurrence sur le nombre de pas de $\Longrightarrow_\eta$.
- Si $M = P$, $P' = N$ convient.
- Si $M \Longrightarrow_\eta M_0 \Longrightarrow_\eta^* P \Longrightarrow_{\beta\iota} N$, alors, puisque M_0 vérifie la condition de report (lemme 2.1.2), nous avons par induction P_0' tel que $M \Longrightarrow_\eta M_0 \Longrightarrow_{\beta\iota} P_0' \Longrightarrow_\eta^* N$. D'après le lemme 2.2.1, il existe P' tel que $M \Longrightarrow_{\beta\iota} P' \Longrightarrow_\eta P_0'$, d'où $M \Longrightarrow_{\beta\iota} P' \Longrightarrow_\eta^* N$.

(2) Par récurrence sur le nombre de pas de $\Longrightarrow_{\beta\iota}$ en utilisant le point précédent.

□

2.3. Report de η.

2.3.1. Proposition (Report de la règle η)**.** *Soient* M, N, T *des termes et* Γ *un contexte. Si* $M \twoheadrightarrow_{\beta\iota\eta} N$ *et si* $\Gamma \vdash M : T$ *alors il existe un terme* P *tel que* $M \twoheadrightarrow_{\beta\iota} P \twoheadrightarrow_\eta N$.

DÉMONSTRATION. Par récurrence sur le nombre de pas de réduction. Immédiat si $M = N$. Si $M \twoheadrightarrow_{\beta\iota\eta} N_0 \to_{\beta\iota\eta} N$, par hypothèse de récurrence, il existe P_0 tel que $M \twoheadrightarrow_{\beta\iota} P_0 \twoheadrightarrow_\eta N_0 \to_{\beta\iota\eta} N$.
- Si $N_0 \to_\eta N$, alors P_0 convient.
- Sinon, $N_0 \to_{\beta\iota} N$. Par préservation du typage, P_0 est bien typé, donc vérifie la condition du report (lemme 2.1.4). Nous pouvons donc appliquer le point (1) du corollaire précédent à $P_0 \twoheadrightarrow_\eta N_0 \to_{\beta\iota} N$ et en déduire l'existence de P tel que $P_0 \twoheadrightarrow_{\beta\iota} P \twoheadrightarrow_\eta N$ (car $\Longrightarrow_R$ contient $\to_R$ et $\Longrightarrow_R^*$ est identique à $\twoheadrightarrow_R$).

□

Annexe D

Récapitulatif des systèmes d'inférence et du modèle abstrait

Pour la commodité du lecteur, nous regroupons ici

- les règles de typage de ICC_Σ,
- les paramètres et les axiomes des modèles abstraits de ICC_Σ
- les règles de typage de AICC_Σ,
- les règles de l'algorithme d'inférence de type extrait,
- les règles de réduction ajoutées dans AICC_Σ, pour définir le deuxième algorithme d'inférence de type,
- les règles de l'algorithme d'inférence de type annoté.

1. Règles de typage de ICC_Σ

Calcul des Constructions.

$$\frac{}{\bullet \vdash}\,(\textsc{Wf}\text{-E}) \qquad \frac{\Gamma \vdash T : s \quad x \notin \mathsf{DV}(\Gamma)}{\Gamma; x : T \vdash}\,(\textsc{Wf}\text{-S})$$

$$\frac{\Gamma \vdash \quad (x : T) \in \Gamma}{\Gamma \vdash x : T}\,(\textsc{Var}) \qquad \frac{\Gamma \vdash \quad (s_1, s_2) \in \mathbf{Axiom}}{\Gamma \vdash s_1 : s_2}\,(\textsc{Sort})$$

$$\frac{\Gamma \vdash T : s_1 \quad \Gamma; x : T \vdash U : s_2 \quad (s_1, s_2, s_3) \in \mathbf{Rule}}{\Gamma \vdash \Pi x : T.U : s_3}\,(\text{E-}\textsc{Prd})$$

$$\frac{\Gamma; x : T \vdash M : U \quad \Gamma \vdash \Pi x : T.U : s}{\Gamma \vdash \lambda x.M : \Pi x : T.U}\,(\textsc{Lam})$$

$$\frac{\Gamma \vdash M : \Pi x : T.U \quad \Gamma \vdash N : T}{\Gamma \vdash MN : U\,[x/N]}\,(\textsc{App})$$

$$\frac{\Gamma \vdash T : s_1 \quad \Gamma; x : T \vdash U : s_2 \quad (s_1, s_2, s_3) \in \mathbf{Rule}}{\Gamma \vdash \forall x : T.U : s_3}\,(\text{I-}\textsc{Prd})$$

$$\frac{\Gamma; x : T \vdash M : U \quad \Gamma \vdash \forall x : T.U : s \quad x \notin \mathsf{FV}(M)}{\Gamma \vdash M : \forall x : T.U}\,(\textsc{Gen})$$

$$\frac{\Gamma \vdash M : \forall x : T.U \quad \Gamma \vdash N : T}{\Gamma \vdash M : U\,[x/N]}\,(\textsc{Inst})$$

$$\frac{\Gamma \vdash M : T \quad \Gamma \vdash T' : s \quad T \preceq T'}{\Gamma \vdash M : T'}\,(\textsc{Cum})$$

Types somme.

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s_3}\,(\Sigma)$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}[x/a] \quad \Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash (a, b) : \Sigma x : \mathrm{A}.\mathrm{B}}\,(\Sigma\text{-I})$$

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma x : \mathrm{A}.\mathrm{B} \to s \quad \Gamma \vdash c : \Sigma x : \mathrm{A}.\mathrm{B} \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)}{\Gamma \vdash \mathsf{Elim}_\Sigma(x y.f, c) : \mathrm{P}\,c}\,(\Sigma\text{-E})$$

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \exists x : \mathrm{A}.\mathrm{B} : s_3}\,(\exists)$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}[x/a] \quad \Gamma \vdash \exists x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash (\diamond, b) : \exists x : \mathrm{A}.\mathrm{B}}\,(\exists\text{-I})$$

$$\frac{\begin{array}{l}\Gamma \vdash \mathrm{P} : \exists x : \mathrm{A}.\mathrm{B} \to s \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(\diamond, y) \\ \Gamma \vdash c : \exists x : \mathrm{A}.\mathrm{B} \qquad x \notin \mathsf{FV}(f)\end{array}}{\Gamma \vdash \mathsf{Elim}_\exists(y.f, c) : \mathrm{P}\,c}\,(\exists\text{-E})$$

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2}{\Gamma \vdash \{x : \mathrm{A} \mid \mathrm{B}\} : s_1}\,(\textsc{Sub})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}[x/a] \quad \Gamma \vdash \{x : \mathrm{A} \mid \mathrm{B}\} : s}{\Gamma \vdash a : \{x : \mathrm{A} \mid \mathrm{B}\}}\,(\textsc{Sub}\text{-I})$$

$$\frac{\Gamma \vdash a : \{x : \mathrm{A} \mid \mathrm{B}\}}{\Gamma \vdash a : \mathrm{A}}\,(\textsc{Sub}\text{-E-1})$$

$$\frac{\begin{array}{ll}\Gamma \vdash \mathrm{P} : s & \Gamma; y : \mathrm{B}[x/c] \vdash f : \mathrm{P} \\ \Gamma \vdash c : \{x : \mathrm{A} \mid \mathrm{B}\} & y \notin \mathsf{FV}(f)\end{array}}{\Gamma \vdash f : \mathrm{P}}\,(\textsc{Sub}\text{-E-2})$$

2. Paramètres et les axiomes des modèles abstraits de ICC_Σ

Paramètres des modèles abstraits du calcul implicite avec sommes

- un ensemble $\mathscr{M}$
- une opération binaire $(\cdot) : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$
- une application $[\![]\!] : \Lambda_{\mathrm{ICC}_\Sigma} \times \mathbf{Val}_{\mathscr{M}} \to \mathscr{M}$
- une relation d'ordre $(\leq) \subset \mathscr{M} \times \mathscr{M}$
- un sous-ensemble $\mathscr{T} \subset \mathscr{M}$
- une application $\mathsf{El} : \mathscr{T} \to \mathbb{P}(\mathscr{M})$
- cinq constantes $\Pi, \forall, \Sigma, \exists, \sigma \in \mathscr{M}$
- une suite de constantes $(\mathsf{u}_i)_{i\in\omega} \in \mathscr{T}^\omega$
- deux constantes $\mathbf{1}, \mathbf{2} \in \mathscr{M}$
- une opération binaire $\mathsf{Pair} : \mathscr{M} \times \mathscr{M} \to \mathscr{M}$
- une application $\mathsf{Gen} : \mathbb{P}(\mathscr{M}) \to \mathscr{M}$

Axiomes de $[\![]\!]$,$\leq$ et de Pair

(a) si $a \leq a'$ et $b \leq b'$ alors $ab \leq a'b'$

(b) pour tout $a, b \in \mathscr{M}$, $\mathsf{Pair}(a,b)\cdot\mathbf{1} = a$ et $\mathsf{Pair}(a,b)\cdot\mathbf{2} = b$

(c) $[\![x]\!]_\rho = \rho(x)$

(d) $[\![\mathsf{Prop}]\!]_\rho = \mathsf{u}_0$

(e) $[\![\mathsf{Type}_i]\!]_\rho = \mathsf{u}_i$, pour $i > 0$

(f) $[\![\Pi x : \mathrm{T}.\mathrm{U}]\!]_\rho = (\Pi \cdot [\![\mathrm{T}]\!]_\rho) \cdot [\![\lambda x.\mathrm{U}]\!]_\rho$

(g) $[\![\lambda x.\mathrm{M}]\!]_\rho \cdot a = [\![\mathrm{M}]\!]_{\rho;x\leftarrow a}$

(h) si $[\![\mathrm{M}]\!]_{\rho;x\leftarrow a} = [\![\mathrm{M}']\!]_{\rho';x\leftarrow a}$ pour tout $a \in \mathscr{M}$, alors $[\![\lambda x.\mathrm{M}]\!]_\rho = [\![\lambda x.\mathrm{M}']\!]_{\rho'}$

(i) $[\![\mathrm{M}\,\mathrm{N}]\!]_\rho = [\![\mathrm{M}]\!]_\rho \cdot [\![\mathrm{N}]\!]_\rho$

(j) $[\![\forall x : \mathrm{T}.\mathrm{U}]\!]_\rho = (\forall \cdot [\![\mathrm{T}]\!]_\rho) \cdot [\![\lambda x.\mathrm{U}]\!]_\rho$

(k) $[\![\Sigma x : \mathrm{A}.\mathrm{B}]\!]_\rho = (\Sigma \cdot [\![\mathrm{A}]\!]_\rho) \cdot [\![\lambda x.\mathrm{B}]\!]_\rho$

(l) $[\![(\mathrm{M},\mathrm{N})]\!]_\rho = \mathsf{Pair}([\![\mathrm{M}]\!]_\rho, [\![\mathrm{N}]\!]_\rho)$

(m) $[\![\mathsf{Elim}_\Sigma(xy.f, c)]\!]_\rho = [\![f]\!]_{\rho;x\leftarrow([\![c]\!]_\rho\cdot\mathbf{1});y\leftarrow([\![c]\!]_\rho\cdot\mathbf{2})}$

(n) $[\![\exists x : \mathrm{A}.\mathrm{B}]\!]_\rho = (\exists \cdot [\![\mathrm{A}]\!]_\rho) \cdot [\![\lambda x.\mathrm{B}]\!]_\rho$

(o) $[\![(\diamond, b)]\!]_\rho = [\![b]\!]_\rho$

(p) $[\![\mathsf{Elim}_\exists(y.f, c)]\!]_\rho = [\![f]\!]_{\rho;y\leftarrow[\![c]\!]_\rho}$

(q) $[\![\{x : \mathrm{A} \mid \mathrm{B}\}]\!]_\rho = (\sigma \cdot [\![\mathrm{A}]\!]_\rho) \cdot [\![\lambda x.\mathrm{B}]\!]_\rho$

(r) $[\![\lambda f x.\, f\, x]\!]_\rho \leq [\![\lambda x.\, x]\!]_\rho$

Axiomes de $\mathscr{T}$, El

(1) si $t \in \mathscr{T}$ et $t \leq t'$ alors $t' \in \mathscr{T}$

(2) si $t \in \mathscr{T}$ et $t \leq t'$ alors $\mathsf{El}(t) = \mathsf{El}(t')$

(3) si $t \in \mathscr{T}$, $a \in \mathsf{El}(t)$ et $a \leq a'$ alors $a' \in \mathsf{El}(t)$

Axiomes des opérateurs de types

(4) si $\Pi tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\mathsf{El}(\Pi tu) = \{f \in \mathscr{M} \mid \forall a \in \mathsf{El}(t),\ fa \in \mathsf{El}(ua)\}$

(5) si $\forall tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\mathsf{El}(\forall tu) = \{b \in \mathscr{M} \mid \forall a \in \mathsf{El}(t),\ b \in \mathsf{El}(ua)\} = \bigcap_{a \in \mathsf{El}(t)} \mathsf{El}(ua)$

(6) si $\Sigma tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\Sigma tu = \mathsf{Gen}(\{\mathsf{Pair}(a,b) \in \mathscr{M} \mid a \in \mathsf{El}(t), b \in \mathsf{El}(ua)\})$

(7) si $\exists tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\exists tu = \mathsf{Gen}(\bigcup_{a \in \mathsf{El}(t)} \mathsf{El}(ua))$

(8) si $\sigma tu \in \mathscr{T}$ alors
  (a) $t \in \mathscr{T}$ et $ua \in \mathscr{T}$ pour tout $a \in \mathsf{El}(t)$
  (b) $\mathsf{El}(\sigma tu) = \{a \in \mathscr{M} \mid a \in \mathsf{El}(t) \wedge \mathsf{El}(ua) \neq \emptyset\}$

Axiomes de Gen

(9) pour tout $\mathrm{X} \subset \mathscr{M}$, si $\mathsf{Gen}(\mathrm{X}) \in \mathscr{T}$ alors $\mathrm{X} \subset \mathsf{El}(\mathsf{Gen}(\mathrm{X}))$

(10) si $\mathrm{X} \subset \mathrm{X}'$ et si $\mathsf{Gen}(\mathrm{X}), \mathsf{Gen}(\mathrm{X}') \in \mathscr{T}$, alors $\mathsf{El}(\mathsf{Gen}(\mathrm{X})) \subset \mathsf{El}(\mathsf{Gen}(\mathrm{X}'))$.

(11) si $\Pi(\mathsf{Gen}(\mathrm{X}))u \in \mathscr{T}$ alors

$$\{f \in \mathscr{M} \mid \forall a \in \mathrm{X}, fa \in \mathsf{El}(ua)\} \subset \mathsf{El}(\Pi(\mathsf{Gen}(\mathrm{X}))u)$$

Axiomes des codes des univers

(12) $\mathsf{El}(\mathsf{u}_i) \subset \mathscr{T}$ pour tout $i \in \omega$

(13) $\mathsf{u}_i \in \mathsf{El}(\mathsf{u}_{i+1})$ pour tout $i \in \omega$

(14) $\mathsf{El}(\mathsf{u}_i) \subset \mathsf{El}(\mathsf{u}_{i+1})$ pour tout $i \in \omega$

(15) si $t \in \mathsf{El}(\mathsf{u}_i)$ et $ua \in \mathsf{El}(\mathsf{u}_i)$ pour tout $a \in \mathsf{El}(t)$, alors $\Pi tu, \forall tu, \Sigma tu, \exists tu \in \mathsf{El}(\mathsf{u}_i)$, pour tout $i \in \omega$

(16) si $t \in \mathsf{El}(\mathsf{u}_i)$ et $ua \in \mathsf{El}(\mathsf{u}_0)$ pour tout $a \in \mathsf{El}(t)$, alors $\Pi tu, \forall tu \in \mathsf{El}(\mathsf{u}_0)$ pour tout $i \in \omega$.

(17) si $t \in \mathsf{El}(\mathsf{u}_i)$ et $ua \in \mathsf{El}(\mathsf{u}_j)$ pour tout $a \in \mathsf{El}(t)$, alors $\sigma tu \in \mathsf{El}(\mathsf{u}_i)$

3. Règles de typage de AICC$_\Sigma$

Calcul des Constructions.

$$\frac{}{\bullet \vdash}(\text{WF-E}_\text{A}) \qquad \frac{\Gamma \vdash \text{T} : s \quad x \notin \mathsf{DV}(\Gamma)}{\Gamma; x : \text{T} \vdash}(\text{WF-S}_\text{A})$$

$$\frac{\Gamma \vdash \quad (x : \text{T}) \in \Gamma}{\Gamma \vdash x : \text{T}}(\text{VAR}_\text{A}) \qquad \frac{\Gamma \vdash \quad (s_1, s_2) \in \textbf{Axiom}}{\Gamma \vdash s_1 : s_2}(\text{SORT}_\text{A})$$

$$\frac{\Gamma \vdash \text{T} : s_1 \quad \Gamma; x : \text{T} \vdash \text{U} : s_2 \quad (s_1, s_2, s_3) \in \textbf{Rule}}{\Gamma \vdash \Pi x : \text{T}.\,\text{U} : s_3}(\text{E-PROD})$$

$$\frac{\Gamma; x : \text{T} \vdash \text{M} : \text{U} \quad \Gamma \vdash \Pi x : \text{T}.\,\text{U} : s}{\Gamma \vdash \lambda x{:}\text{T}.\,\text{M} : \Pi x : \text{T}.\,\text{U}}(\text{E-LAM})$$

$$\frac{\Gamma \vdash \text{M} : \Pi x : \text{T}.\,\text{U} \quad \Gamma \vdash \text{N} : \text{T}}{\Gamma \vdash \text{MN} : \text{U}\,[x/\text{N}]}(\text{E-APP})$$

$$\frac{\Gamma \vdash \text{T} : s_1 \quad \Gamma; x : \text{T} \vdash \text{U} : s_2 \quad (s_1, s_2, s_3) \in \textbf{Rule}}{\Gamma \vdash \Pi[x{:}\text{T}].\,\text{U} : s_3}(\text{I-PROD})$$

$$\frac{\Gamma; x : \text{T} \vdash \text{M} : \text{U} \quad \Gamma \vdash \Pi[x{:}\text{T}].\,\text{U} : s \quad x \notin \mathsf{FV}(\text{M}^*)}{\Gamma \vdash \lambda[x{:}\text{T}].\,\text{M} : \Pi[x{:}\text{T}].\,\text{U}}(\text{I-LAM})$$

$$\frac{\Gamma \vdash \text{M} : \Pi[x{:}\text{T}].\,\text{U} \quad \Gamma \vdash \text{N} : \text{T}}{\Gamma \vdash \text{M}\,[\text{N}] : \text{U}\,[x/\text{N}]}(\text{I-APP})$$

$$\frac{\Gamma \vdash \text{M} : \text{T} \quad \Gamma \vdash \text{T}' : s \quad \text{T}^* \preceq \text{T}'^*}{\Gamma \vdash \text{M} : \text{T}'}(\text{CUM}_\text{A})$$

Types somme.

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s_3}\,(\Sigma_{\mathrm{A}})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}\,[x/a] \quad \Gamma \vdash \Sigma x : \mathrm{A}.\mathrm{B} : s}{\Gamma \vdash (a, b)_{\Sigma x:\mathrm{A}.\mathrm{B}} : \Sigma x : \mathrm{A}.\mathrm{B}}\,(\Sigma_{\mathrm{A}}\text{-I})$$

$$\frac{\Gamma \vdash \mathrm{P} : \Sigma x : \mathrm{A}.\mathrm{B} \to s \quad \Gamma \vdash c : \Sigma x : \mathrm{A}.\mathrm{B} \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,(x, y)_{\Sigma x:\mathrm{A}.\mathrm{B}}}{\Gamma \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}\,(\Sigma_{\mathrm{A}}\text{-E})$$

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2 \qquad (s_1, s_2, s_3) \in \mathbf{Rule}'}{\Gamma \vdash \Sigma [x{:}\mathrm{A}]\,.\mathrm{B} : s_3}\,(\Sigma_{\exists})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}\,[x/a] \quad \Gamma \vdash \Sigma [x{:}\mathrm{A}]\,.\mathrm{B} : s}{\Gamma \vdash ([a], b)_{\Sigma [x\mathrm{A}]\,.\mathrm{B}} : \Sigma [x{:}\mathrm{A}]\,.\mathrm{B}}\,(\Sigma_{\exists}\text{-I})$$

$$\frac{\begin{array}{l}\Gamma \vdash \mathrm{P} : \Sigma [x{:}\mathrm{A}]\,.\mathrm{B} \to s \quad \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f : \mathrm{P}\,([x], y)_{\Sigma [x\mathrm{A}]\,.\mathrm{B}} \\ \Gamma \vdash c : \Sigma [x{:}\mathrm{A}]\,.\mathrm{B} \qquad x \notin \mathsf{FV}(f^*)\end{array}}{\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) : \mathrm{P}\,c}\,(\Sigma_{\exists}\text{-E})$$

$$\frac{\Gamma \vdash \mathrm{A} : s_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} : s_2}{\Gamma \vdash \Sigma x : \mathrm{A}.[\mathrm{B}] : s_1}\,(\Sigma_{\mathrm{SUB}})$$

$$\frac{\Gamma \vdash a : \mathrm{A} \quad \Gamma \vdash b : \mathrm{B}\,[x/a] \quad \Gamma \vdash \Sigma x : \mathrm{A}.[\mathrm{B}] : s}{\Gamma \vdash (a, [b])_{\Sigma x:\mathrm{A}.[\mathrm{B}]} : \Sigma x : \mathrm{A}.[\mathrm{B}]}\,(\Sigma_{\mathrm{SUB}}\text{-I})$$

$$\frac{\Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}]}{\Gamma \vdash \pi_1(c) : \mathrm{A}}\,(\Sigma_{\mathrm{SUB}}\text{-E-1})$$

$$\frac{\Gamma \vdash \mathrm{P} : s \qquad \Gamma \vdash c : \Sigma x : \mathrm{A}.[\mathrm{B}] \quad \Gamma; y : \mathrm{B}_0 \vdash f : \mathrm{P} \quad (\mathrm{B}\,[x/\pi_1(c)])^* \preceq \mathrm{B}_0^* \quad y \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}_0].f, c, \mathrm{P}) : \mathrm{P}}\,(\Sigma_{\mathrm{SUB}}\text{-E-2})$$

4. Algorithme d'inférence de type extrait

(1) Variable

$$\frac{(x : \mathrm{T}') \in \Gamma'}{\Gamma' \vdash x \Uparrow^{\mathrm{I}} \mathrm{T}'} \text{ (Inf-Var)}$$

(2) Sorte

$$\frac{}{\Gamma' \vdash s_1 \Uparrow^{\mathrm{I}} \mathtt{Axiom}(s_1)} \text{ (Inf-Sort)}$$

(3) Produit explicite

$$\frac{\Gamma' \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Pi x : \mathrm{T}.\, \mathrm{U} \Uparrow^{\mathrm{I}} \mathtt{Rule}(\mathtt{NF\_BI}(\mathrm{S}'_1), \mathtt{NF\_BI}(\mathrm{S}'_2))} \text{ (Inf-E-Prd)}$$

$$\frac{\Gamma' \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}') \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{T}^* \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{U}'}{\Gamma' \vdash \lambda x{:}\mathrm{T}.\, \mathrm{M} \Uparrow^{\mathrm{I}} \Pi x : \mathrm{T}^*.\, \mathrm{U}'} \text{ (Inf-E-Lam)}$$

$$\frac{\Gamma' \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}') = \Pi x : \mathrm{T}'.\, \mathrm{U}' \qquad \Gamma' \vdash \mathrm{N} \Uparrow^{\mathrm{I}} \mathrm{T}'_0 \qquad \mathtt{cumu}(\mathrm{T}'_0, \mathrm{T}')}{\Gamma' \vdash \mathrm{M}\,\mathrm{N} \Uparrow^{\mathrm{I}} \mathrm{U}'[x/\mathrm{N}^*]} \text{ (Inf-E-App)}$$

(4) Produit implicite

$$\frac{\Gamma' \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Pi[x{:}\mathrm{T}].\, \mathrm{U} \Uparrow^{\mathrm{I}} \mathtt{Rule}(\mathtt{NF\_BI}(\mathrm{S}'_1), \mathtt{NF\_BI}(\mathrm{S}'_2))} \text{ (Inf-I-Prd)}$$

$$\frac{\Gamma' \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}' \qquad \mathtt{NF\_BI}(\mathrm{S}') \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{T}^* \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{U}' \qquad x \notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma' \vdash \lambda[x{:}\mathrm{T}].\, \mathrm{M} \Uparrow^{\mathrm{I}} \forall x : \mathrm{T}^*.\, \mathrm{U}'} \text{ (Inf-I-Lam)}$$

$$\frac{\Gamma' \vdash \mathrm{M} \Uparrow^{\mathrm{I}} \mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}') = \forall x : \mathrm{T}'.\, \mathrm{U}' \qquad \Gamma' \vdash \mathrm{N} \Uparrow^{\mathrm{I}} \mathrm{T}'_0 \qquad \mathtt{cumu}(\mathrm{T}'_0, \mathrm{T}')}{\Gamma' \vdash \mathrm{M}\,[\mathrm{N}] \Uparrow^{\mathrm{I}} \mathrm{U}'[x/\mathrm{N}^*]} \text{ (Inf-I-App)}$$

(5) Somme explicite

$$\frac{\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Sigma x : \mathrm{A}.\, \mathrm{B} \Uparrow^{\mathrm{I}} \mathtt{RulePrime}(\mathtt{NF\_BI}(\mathrm{S}'_1), \mathtt{NF\_BI}(\mathrm{S}'_2))} \text{ (Inf-Σ)}$$

$$\frac{\begin{cases} \Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort} \end{cases} \qquad \begin{cases} \Gamma' \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*) \end{cases} \qquad \begin{cases} \Gamma' \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}[x/a])^*) \end{cases}}{\Gamma' \vdash (a, b)_{\Sigma x:\mathrm{A}.\mathrm{B}} \Uparrow^{\mathrm{I}} \Sigma x : \mathrm{A}^*.\, \mathrm{B}^*} \text{ (Inf-Σ-Pair)}$$

$$\frac{\begin{cases} \Gamma' \vdash \mathrm{P} \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \end{cases} \qquad \begin{cases} \mathtt{NF\_BI}(\mathrm{R}'_1) = \Sigma x : \mathrm{A}'_1.\, \mathrm{B}'_1 \to s_1 \\ \mathtt{NF\_BI}(\mathrm{R}'_2) = \Sigma x : \mathrm{A}'_2.\, \mathrm{B}'_2 \\ \Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort} \end{cases} \qquad \begin{cases} \mathtt{cumu}(\Sigma x : \mathrm{A}^*.\, \mathrm{B}^*, \Sigma x : \mathrm{A}'_1.\, \mathrm{B}'_1) \\ \mathtt{cumu}(\Sigma x : \mathrm{A}'_2.\, \mathrm{B}'_2, \Sigma x : \mathrm{A}^*.\, \mathrm{B}^*) \\ \Gamma'; x : \mathrm{A}^*; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_3 \\ \mathtt{cumu}(\mathrm{R}'_3, \mathrm{P}^*(x, y)) \end{cases}}{\Gamma' \vdash \mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, c, \mathrm{P}) \Uparrow^{\mathrm{I}} \mathrm{P}^*\, c^*} \text{ (Inf-Σ-Elim)}$$

(6) Somme implicite droite

$$\frac{\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \Uparrow^{\mathrm{I}} \mathtt{RulePrime}(\mathtt{NF\_BI}(\mathrm{S}'_1), \mathtt{NF\_BI}(\mathrm{S}'_2))}\ (\textsc{Inf-}\exists)$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma' \vdash ([a], b)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} \Uparrow^{\mathrm{I}} \exists x : \mathrm{A}^*\,.\,\mathrm{B}^*}\ (\textsc{Inf-}\exists\textsc{-Pair})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{P} \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}'_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{R}'_1) = \exists x : \mathrm{A}'_1\,.\,\mathrm{B}'_1 \to s_1 \\ \mathtt{NF\_BI}(\mathrm{R}'_2) = \exists x : \mathrm{A}'_2\,.\,\mathrm{B}'_2 \\ \Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{cumu}(\exists x : \mathrm{A}^*\,.\,\mathrm{B}^*, \exists x : \mathrm{A}'_1\,.\,\mathrm{B}'_1) \\ \mathtt{cumu}(\exists x : \mathrm{A}'_2\,.\,\mathrm{B}'_2, \exists x : \mathrm{A}^*\,.\,\mathrm{B}^*) \\ \Gamma'; x : \mathrm{A}^*; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_3 \\ x \notin \mathsf{FV}(f^*) \\ \mathtt{cumu}(\mathrm{R}'_3, \mathrm{P}^*\,(\diamond, y))\end{array}\right.}{\Gamma' \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) \Uparrow^{\mathrm{I}} \mathrm{P}^*\,c^*}\ (\textsc{Inf-}\exists\textsc{-Elim})$$

(7) Somme implicite gauche

$$\frac{\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \qquad \mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \qquad \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2 \qquad \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}}{\Gamma' \vdash \Sigma x : \mathrm{A}\,.\,[\mathrm{B}] \Uparrow^{\mathrm{I}} \mathtt{NF\_BI}(\mathrm{S}'_1)}\ (\textsc{Inf-Sub})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{A} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{A}} \\ \Gamma'; x : \mathrm{A}^* \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{B}} \\ \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{A}}), \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{B}}) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash a \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \mathtt{cumu}(\mathrm{R}'_1, \mathrm{A}^*)\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma' \vdash b \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma' \vdash (a, [b])_{\Sigma x:\mathrm{A}.\,[\mathrm{B}]} \Uparrow^{\mathrm{I}} \{x : \mathrm{A}^* \mid \mathrm{B}^*\}}\ (\textsc{Inf-Sub-Pair})$$

$$\frac{\Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}' \qquad \mathtt{NF\_BI}(\mathrm{R}') = \{x : \mathrm{A}' \mid \mathrm{B}'\}}{\Gamma' \vdash \pi_1(c) \Uparrow^{\mathrm{I}} \mathrm{A}'}\ (\textsc{Inf-Sub-Elim-1})$$

$$\frac{\left\{\begin{array}{l}\Gamma' \vdash \mathrm{P} \Uparrow^{\mathrm{I}} \mathrm{S}'_1 \\ \Gamma' \vdash c \Uparrow^{\mathrm{I}} \mathrm{R}'_1 \\ \Gamma' \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S}'_2\end{array}\right. \qquad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{S}'_1) \in \mathbf{Sort} \\ \mathtt{NF\_BI}(\mathrm{R}'_1) = \{x : \mathrm{A}'_1 \mid \mathrm{B}'_1\} \\ \mathtt{NF\_BI}(\mathrm{S}'_2) \in \mathbf{Sort}\end{array}\right. \qquad \left\{\begin{array}{l}\Gamma'; y : \mathrm{B}^* \vdash f \Uparrow^{\mathrm{I}} \mathrm{R}'_2 \\ \mathtt{cumu}(\mathrm{R}'_2, \mathrm{P}^*) \\ \mathtt{cumu}(\mathrm{B}'_1\,[x/c^*], \mathrm{B}^*) \\ y \notin \mathsf{FV}(f^*)\end{array}\right.}{\Gamma' \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P}) \Uparrow^{\mathrm{I}} \mathrm{P}^*}\ (\textsc{Inf-Sub-Elim-2})$$

5. Règles de réduction dans AICC$_\Sigma$

Réduction à la racine.

- $\beta\iota$-réduction $\rhd^{\mathrm{A}}_{\beta\iota}$

$$
\begin{array}{rcl}
(\lambda x{:}\mathrm{T}\,.\,\mathrm{M})\,\mathrm{N} & \rhd^{\mathrm{A}}_{\beta^e} & \mathrm{M}\,[x/\mathrm{N}] \\
(\lambda [x{:}\mathrm{T}]\,.\,\mathrm{M})\,[\mathrm{N}] & \rhd^{\mathrm{A}}_{\beta^i} & \mathrm{M}\,[x/\mathrm{N}] \\
\mathsf{Elim}_{\mathrm{E}}((x{:}\mathrm{T}).(y{:}\mathrm{U}).f, (a,b)_{\Sigma x:\mathrm{A}\,.\,\mathrm{B}}, \mathrm{P}) & \rhd^{\mathrm{A}}_{\iota_\Sigma} & f\,[x/a]\,[y/b] \\
\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{T}].(y{:}\mathrm{U}).f, ([a],b)_{\Sigma [x:\mathrm{A}]\,.\,\mathrm{B}}, \mathrm{P}) & \rhd^{\mathrm{A}}_{\iota_\exists} & f\,[x/a]\,[y/b] \\
\pi_1((a,[b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]}) & \rhd^{\mathrm{A}}_{\iota^1_\sigma} & a \\
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{T}].f, (a,[b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]}, \mathrm{P}) & \rhd^{\mathrm{A}}_{\iota^2_\sigma} & f\,[y/b]
\end{array}
$$

- Éliminateur simplifié et constructeurs $\rhd^{\mathrm{A}}_{\mathsf{Constr-Simpl}}$

$$
\begin{array}{rcl}
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\lambda z{:}\mathrm{T}\,.\,\mathrm{M}, c, \Pi z:\mathrm{T_0}\,.\,\mathrm{U_0}) & \rhd^{\mathrm{A}}_{\lambda-\mathsf{Simpl}} & \lambda z{:}\mathrm{T_0}\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\mathrm{M}, c, \mathrm{U_0}) \\
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\lambda [z{:}\mathrm{T}]\,.\,\mathrm{M}, c, \Pi [z:\mathrm{T_0}]\,.\,\mathrm{U_0}) & \rhd^{\mathrm{A}}_{[\lambda]-\mathsf{Simpl}} & \lambda [z{:}\mathrm{T_0}]\,.\,\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B_0}].\mathrm{M}, c, \mathrm{U_0})
\end{array}
$$

$$
\frac{a_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U_0}].a, c, \mathrm{A_0}) \qquad b_0 = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U_0}].b, c, \mathrm{B_0}\,[z/a_0])}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{U_0}].(a,b)_{\Sigma z:\mathrm{A}\,.\,\mathrm{B}}, c, \Sigma z:\mathrm{A_0}\,.\,\mathrm{B_0}) \rhd^{\mathrm{A}}_{(\cdot,\cdot)-\mathsf{Simpl}} (a_0,b_0)_{\Sigma z:\mathrm{A_0}\,.\,\mathrm{B_0}}}
$$

- Éliminateur simplifié et autres éliminateurs $\rhd^{\mathrm{A}}_{\mathsf{Elim-Simpl}}$

$$
\frac{c = \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U})}{\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) \rhd^{\mathrm{A}}_{\exists-\mathsf{Simpl}} \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, g, \mathrm{P}), d, \mathrm{P}\,c)}
$$

$$
\begin{array}{rcl}
\pi_1(\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \Sigma x:\mathrm{T}\,.\,[\mathrm{U}])) & \rhd^{\mathrm{A}}_{\sigma^1-\mathsf{Simpl}} & \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\pi_1(f), c, \mathrm{T}) \\
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].g, d, \mathrm{U}), \mathrm{P}) & \rhd^{\mathrm{A}}_{\sigma^2-\mathsf{Simpl}} & \mathsf{Elim}^s_{\mathrm{I_R}}([z{:}\mathrm{T}].\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, g, \mathrm{P}), d, \mathrm{P})
\end{array}
$$

- Éliminateur simplifié et constructeurs de type $\rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}}$

$$
\frac{\left\{\begin{array}{l}\Gamma^*; y:\mathrm{B}^* \vdash \mathrm{T} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{T}} \\ \Gamma^*; y:\mathrm{B}^*; z:\mathrm{T}^* \vdash \mathrm{U} \Uparrow^{\mathrm{I}} \mathrm{S}'_{\mathrm{U}}\end{array}\right. \qquad \left\{\begin{array}{l}\mathrm{T_0} = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{T}, c, \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{T}})) \\ \mathrm{U_0} = \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{U}, c, \mathtt{NF\_BI}(\mathrm{S}'_{\mathrm{U}}))\end{array}\right.}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\square z:\mathrm{T}\,.\,\mathrm{U}, c, \mathrm{P}) \rhd^{\mathrm{A}}_{\Gamma,\ \square-\mathsf{Simpl}} \square z:\mathrm{T_0}\,.\,\mathrm{U_0}}
$$

- Éliminateur simplifié et sortes $\rhd^{\mathrm{A}}_{s-\mathsf{Simpl}}$

$$
\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].s, c, \mathrm{P}) \quad \rhd^{\mathrm{A}}_{s-\mathsf{Simpl}} \quad s
$$

Réduction dans un sous-terme.

$$
\frac{\mathrm{M_0} \rhd^{\mathrm{A}}_{\Gamma} \mathrm{M_1}}{\mathrm{M_0} \to^h_{\Gamma} \mathrm{M_1}}\ (\textsc{HRed-Base})
$$

$$
\frac{\mathrm{M_0} \to^h_{\Gamma} \mathrm{M_1}}{\mathrm{M_0\,N} \to^h_{\Gamma} \mathrm{M_1\,N}}\ (\textsc{HRed-E-App})
$$

$$
\frac{\mathrm{M_0} \to^h_{\Gamma} \mathrm{M_1}}{\mathrm{M_0}\,[\mathrm{N}] \to^h_{\Gamma} \mathrm{M_1}\,[\mathrm{N}]}\ (\textsc{HRed-I-App})
$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}_\mathrm{E}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, \mathrm{M}_0, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}_\mathrm{E}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f, \mathrm{M}_1, \mathrm{P})} \text{(HRed-}\Sigma\text{-Elim)}$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathrm{M}_0, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, \mathrm{M}_1, \mathrm{P})} \text{(HRed-}\exists\text{-Elim)}$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\pi_1(\mathrm{M}_0) \to^h_\Gamma \pi_1(\mathrm{M}_1)} \text{(HRed-}\pi_1\text{)}$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathrm{M}_0, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, \mathrm{M}_1, \mathrm{P})} \text{(HRed-Simpl-Object)}$$

$$\frac{\mathrm{M}_0 \to^h_{\Gamma;y:\mathrm{B}} \mathrm{M}_1}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{M}_0, c, \mathrm{P}) \to^h_\Gamma \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].\mathrm{M}_1, c, \mathrm{P})} \text{(HRed-Simpl-Branch)}$$

$$\frac{\mathrm{M}_0 \to^h_\Gamma \mathrm{M}_1}{\mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{M}_0) \to^h_\Gamma \mathsf{Elim}^s_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{M}_1)} \text{(HRed-Simpl-Type)}$$

6. Algorithme d'inférence de type annoté

(1) Variable

$$\frac{(x:\mathrm{T})\in\Gamma}{\Gamma\vdash x\Uparrow \mathrm{T}}\ (\textsc{Inf-Var})$$

(2) Sorte

$$\frac{}{\Gamma\vdash s_1\Uparrow \mathtt{Axiom}(s_1)}\ (\textsc{Inf-Sort})$$

(3) Produit explicite

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*)=s_1\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{U}\Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*)=s_2\in\mathsf{Sort}}{\Gamma\vdash \Pi x:\mathrm{T}.\,\mathrm{U}\Uparrow \mathtt{Rule}(s_1,s_2)}\ (\textsc{Inf-E-Prod})$$

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S} \qquad \mathtt{NF\_BI}(\mathrm{S}^*)\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{M}\Uparrow \mathrm{U}}{\Gamma\vdash \lambda x{:}\mathrm{T}.\,\mathrm{M}\Uparrow \Pi x:\mathrm{T}.\,\mathrm{U}}\ (\textsc{Inf-E-Lam})$$

$$\frac{\Gamma\vdash \mathrm{M}\Uparrow \mathrm{R}_1 \qquad \mathsf{hnf}_\Gamma(\mathrm{R}_1)=\Pi x:\mathrm{T}.\,\mathrm{U} \qquad \Gamma\vdash \mathrm{N}\Uparrow \mathrm{R}_2 \qquad \mathtt{cumu}(\mathrm{R}_2^*,\mathrm{T}^*)}{\Gamma\vdash \mathrm{M}\,\mathrm{N}\Uparrow \mathrm{U}\,[x/\mathrm{N}]}\ (\textsc{Inf-E-App})$$

(4) Produit implicite

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*)=s_1\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{U}\Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*)=s_2\in\mathsf{Sort}}{\Gamma\vdash \Pi[x{:}\mathrm{T}].\,\mathrm{U}\Uparrow \mathtt{Rule}(s_1,s_2)}\ (\textsc{Inf-I-Prod})$$

$$\frac{\Gamma\vdash \mathrm{T}\Uparrow \mathrm{S} \qquad \mathtt{NF\_BI}(\mathrm{S}^*)\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{T}\vdash \mathrm{M}\Uparrow \mathrm{U} \qquad x\notin \mathsf{FV}(\mathrm{M}^*)}{\Gamma\vdash \lambda[x{:}\mathrm{T}].\,\mathrm{M}\Uparrow \Pi[x{:}\mathrm{T}].\,\mathrm{U}}\ (\textsc{Inf-I-Lam})$$

$$\frac{\Gamma\vdash \mathrm{M}\Uparrow \mathrm{R}_1 \qquad \mathsf{hnf}_\Gamma(\mathrm{R}_1)=\Pi[x{:}\mathrm{T}].\,\mathrm{U} \qquad \Gamma\vdash \mathrm{N}\Uparrow \mathrm{R}_2 \qquad \mathtt{cumu}(\mathrm{R}_2^*,\mathrm{T}^*)}{\Gamma\vdash \mathrm{M}\,[\mathrm{N}]\Uparrow \mathrm{U}\,[x/\mathrm{N}]}\ (\textsc{Inf-I-App})$$

(5) Somme explicite

$$\frac{\Gamma\vdash \mathrm{A}\Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*)=s_1\in\mathsf{Sort} \qquad \Gamma;x:\mathrm{A}\vdash \mathrm{B}\Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*)=s_2\in\mathsf{Sort}}{\Gamma\vdash \Sigma x:\mathrm{A}.\,\mathrm{B}\Uparrow \mathtt{RulePrime}(s_1,s_2)}\ (\textsc{Inf-E-}\Sigma)$$

$$\frac{\begin{cases}\Gamma\vdash \mathrm{A}\Uparrow \mathrm{S}_\mathrm{A}\\ \Gamma;x:\mathrm{A}\vdash \mathrm{B}\Uparrow^{\mathrm{I}} \mathrm{S}_\mathrm{B}\\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*),\mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*)\in\mathsf{Sort}\end{cases} \qquad \begin{cases}\Gamma\vdash a\Uparrow \mathrm{R}_1\\ \Gamma\vdash b\Uparrow \mathrm{R}_2\end{cases} \qquad \begin{cases}\mathtt{cumu}(\mathrm{R}_1^*,\mathrm{A}^*)\\ \mathtt{cumu}(\mathrm{R}_2^*,(\mathrm{B}\,[x/a])^*)\end{cases}}{\Gamma\vdash (a,b)_{\Sigma x:\mathrm{A}.\,\mathrm{B}}\Uparrow \Sigma x:\mathrm{A}.\,\mathrm{B}}\ (\textsc{Inf-E-Pair})$$

$$\frac{\begin{cases}\Gamma\vdash \mathrm{P}\Uparrow \mathrm{R}_1\\ \Gamma\vdash c\Uparrow \mathrm{R}_2\\ \Gamma\vdash \mathrm{A}\Uparrow \mathrm{S}_\mathrm{A}\\ \Gamma;x:\mathrm{A}\vdash \mathrm{B}\Uparrow^{\mathrm{I}} \mathrm{S}_\mathrm{B}\\ \mathtt{NF\_BI}(\mathrm{S}_\mathrm{A}^*),\mathtt{NF\_BI}(\mathrm{S}_\mathrm{B}^*)\in\mathsf{Sort}\\ \Gamma;x:\mathrm{A};y:\mathrm{B}\vdash f\Uparrow \mathrm{R}_3\end{cases} \qquad \begin{cases}\mathsf{hnf}_\Gamma(\mathrm{R}_1)=\Pi x:\mathrm{T}.\,\mathrm{U}\\ \mathtt{NF\_BI}(\mathrm{U}^*)\in\mathsf{Sort}\\ \mathsf{hnf}_\Gamma(\mathrm{R}_2)=\Sigma x:\mathrm{A}_2.\,\mathrm{B}_2\\ \mathtt{cumu}((\Sigma x:\mathrm{A}.\,\mathrm{B})^*,\mathrm{T}^*)\\ \mathtt{cumu}((\Sigma x:\mathrm{A}_2.\,\mathrm{B}_2)^*,(\Sigma x:\mathrm{A}.\,\mathrm{B})^*)\\ \mathtt{cumu}(\mathrm{R}_3^*,(\mathrm{P}\,(x,y)_{\Sigma x:\mathrm{A}.\,\mathrm{B}})^*)\end{cases}}{\Gamma\vdash \mathsf{Elim}_\mathrm{E}((x{:}\mathrm{A}).(y{:}\mathrm{B}).f,c,\mathrm{P})\Uparrow \mathrm{P}\,c}\ (\textsc{Inf-E-Elim})$$

(6) Somme implicite gauche

$$\frac{\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_1 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*) = s_1 \in \mathbf{Sort} \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_2^*) = s_2 \in \mathbf{Sort}}{\Gamma \vdash \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B} \Uparrow \mathtt{RulePrime}(s_1, s_2)}\ (\textsc{Inf-I-}\Sigma\textsc{-L})$$

$$\frac{\left\{\begin{array}{l}\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}\\ \Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S_B}\\ \mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathbf{Sort}\end{array}\right. \quad \left\{\begin{array}{l}\Gamma \vdash a \Uparrow \mathrm{R}_1\\ \Gamma \vdash b \Uparrow \mathrm{R}_2\end{array}\right. \quad \left\{\begin{array}{l}\mathtt{cumu}(\mathrm{R}_1^*, \mathrm{A}^*)\\ \mathtt{cumu}(\mathrm{R}_2^*, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma \vdash ([a], b)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}} \Uparrow \Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B}}\ (\textsc{Inf-I-Pair-L})$$

$$\frac{\left\{\begin{array}{l}\Gamma \vdash \mathrm{P} \Uparrow \mathrm{R}_1\\ \Gamma \vdash c \Uparrow \mathrm{R}_2\\ \Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}\\ \Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S_B}\\ \mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathbf{Sort}\\ \Gamma; x : \mathrm{A}; y : \mathrm{B} \vdash f \Uparrow \mathrm{R}_3\end{array}\right. \quad \left\{\begin{array}{l}\mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Pi x : \mathrm{T}\,.\,\mathrm{U}\\ \mathtt{NF\_BI}(\mathrm{U}^*) \in \mathbf{Sort}\\ \mathsf{hnf}_\Gamma(\mathrm{R}_2) = \Sigma[x{:}\mathrm{A}_2]\,.\,\mathrm{B}_2\\ \mathtt{cumu}((\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B})^*, \mathrm{T}^*)\\ \mathtt{cumu}((\Sigma[x{:}\mathrm{A}_2]\,.\,\mathrm{B}_2)^*, (\Sigma[x{:}\mathrm{A}]\,.\,\mathrm{B})^*)\\ \mathtt{cumu}(\mathrm{R}_3^*, (\mathrm{P}\,([x], y)_{\Sigma[x\mathrm{A}]\,.\,\mathrm{B}})^*)\end{array}\right. \quad x \notin \mathsf{FV}(f^*)}{\Gamma \vdash \mathsf{Elim}_{\mathrm{I_L}}([x{:}\mathrm{A}].(y{:}\mathrm{B}).f, c, \mathrm{P}) \Uparrow \mathrm{P}\,c}\ (\textsc{Inf-I-Elim-L})$$

(7) Somme implicite droite

$$\frac{\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S}_1 \qquad \Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow \mathrm{S}_2 \qquad \mathtt{NF\_BI}(\mathrm{S}_1^*), \mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort}}{\Gamma \vdash \Sigma x : \mathrm{A}\,.\,[\mathrm{B}] \Uparrow \mathtt{NF\_BI}(\mathrm{S}_1^*)}\ (\textsc{Inf-I-}\Sigma\textsc{-R})$$

$$\frac{\left\{\begin{array}{l}\Gamma \vdash \mathrm{A} \Uparrow \mathrm{S_A}\\ \Gamma; x : \mathrm{A} \vdash \mathrm{B} \Uparrow^{\mathrm{I}} \mathrm{S_B}\\ \mathtt{NF\_BI}(\mathrm{S_A^*}), \mathtt{NF\_BI}(\mathrm{S_B^*}) \in \mathbf{Sort}\end{array}\right. \quad \left\{\begin{array}{l}\Gamma \vdash a \Uparrow \mathrm{R}_1\\ \Gamma \vdash b \Uparrow \mathrm{R}_2\end{array}\right. \quad \left\{\begin{array}{l}\mathtt{cumu}(\mathrm{R}_1^*, \mathrm{A}^*)\\ \mathtt{cumu}(\mathrm{R}_2^*, (\mathrm{B}\,[x/a])^*)\end{array}\right.}{\Gamma \vdash (a, [b])_{\Sigma x:\mathrm{A}\,.\,[\mathrm{B}]} \Uparrow \Sigma x : \mathrm{A}\,.\,[\mathrm{B}]}\ (\textsc{Inf-I-Pair-R})$$

$$\frac{\Gamma \vdash c \Uparrow \mathrm{R} \qquad \mathsf{hnf}_\Gamma(\mathrm{R}) = \Sigma x : \mathrm{A}\,.\,[\mathrm{B}]}{\Gamma \vdash \pi_1(c) \Uparrow \mathrm{A}}\ (\textsc{Inf-I-Elim-R-1})$$

$$\frac{\left\{\begin{array}{l}\Gamma \vdash \mathrm{P} \Uparrow \mathrm{S}_1\\ \Gamma \vdash c \Uparrow \mathrm{R}_1\\ \Gamma \vdash \mathrm{B} \Uparrow \mathrm{S}_2\end{array}\right. \quad \left\{\begin{array}{l}\mathtt{NF\_BI}(\mathrm{S}_1^*) \in \mathbf{Sort}\\ \mathsf{hnf}_\Gamma(\mathrm{R}_1) = \Sigma x : \mathrm{A}_1\,.\,[\mathrm{B}_1]\\ \mathtt{NF\_BI}(\mathrm{S}_2^*) \in \mathbf{Sort}\end{array}\right. \quad \left\{\begin{array}{l}\Gamma; y : \mathrm{B} \vdash f \Uparrow \mathrm{R}_2\\ \mathtt{cumu}(\mathrm{R}_2^*, \mathrm{P}^*)\\ \mathtt{cumu}((\mathrm{B}_1\,[x/\pi_1(c)])^*, \mathrm{B}^*)\\ y \notin \mathsf{FV}(f^*)\end{array}\right.}{\Gamma \vdash \mathsf{Elim}^{s}_{\mathrm{I_R}}([y{:}\mathrm{B}].f, c, \mathrm{P}) \Uparrow \mathrm{P}}\ (\textsc{Inf-I-Elim-R-2})$$

Bibliographie

[Aug98] Lennart Augustsson. Cayenne - A language with dependent types. In *Advanced Functional Programming*, pages 240–267, 1998. 220

[Bar84] Henk P Barendregt. *The Lambda Calculus : Its Syntax and Semantics*. North-Holland, Amsterdam, revised edition, 1984. 9, 98, 223

[Bar91] H. Barendregt. Lambda Calculi with Types. Technical Report 91-19, Catholic University Nijmegen, 1991. In Handbook of Logic in Computer Science, Vol II. 7, 20

[Bar99] Bruno Barras. *Auto-validation d'un système de preuves avec familles inductives*. PhD thesis, Université Paris 7, 1999. 220

[BB08] Bruno Barras and Bruno Bernardo. The implicit calculus of constructions as a programming language with dependent types. In Roberto Amadio, editor, *Foundations of Software Science and Computational Structures*, volume 4962 of *Lecture Notes in Computer Science*, pages 365–379. Springer Berlin Heidelberg, 2008. 3

[BC00] Gilles Barthe and Thierry Coquand. An introduction to dependent type theory. In Gilles Barthe, Peter Dybjer, Luis Pinto, and João Saraiva, editors, *Applied Semantics, International Summer School, APPSEM 2000, Caminha, Portugal, September 9-15, 2000, Advanced Lectures*, volume 2395 of *Lecture Notes in Computer Science*, pages 1–41. Springer, 2000. 7, 23

[BDd95] Franco Barbanera, Mariangiola Dezani-Ciancaglini, and Ugo de'Liguoro. Intersection and union types : Syntax and semantics. *Inf. Comput.*, 119(2) :202–230, 1995. 24

[Ber78] Gérard Berry. Stable models of typed lambda-calculi. In *Proceedings of the Fifth Colloquium on Automata, Languages and Programming*, pages 72–89, London, UK, 1978. Springer-Verlag. 73

[Ber09] Bruno Bernardo. Towards an Implicit Calculus of Inductive Constructions. Extending the Implicit Calculus of Constructions with Union and Subset Types. In *TPHOLs (Emerging trends)*, Munich, Germany, August 2009. TU München. 24

[BG93] Marc Bezem and Jan Friso Groote, editors. *Typed Lambda Calculi and Applications, International Conference on Typed Lambda Calculi and Applications, TLCA '93, Utrecht, The Netherlands, March 16-18, 1993, Proceedings*, volume 664 of *Lecture Notes in Computer Science*. Springer, 1993. 252, 253

[Bra13] Edwin Brady. Idris, a general-purpose dependently typed programming language : Design and implementation. *J. Funct. Program.*, 23(5) :552–593, 2013. 220

[BS00] G. Barthe and M.H. Sørensen. Domain-free pure type systems. *In Journal of Functional Programming*, 10(5) :412–452, September 2000. 7

[CDX05] Sa Cui, Kevin Donnelly, and Hongwei Xi. ATS : A language that combines programming with theorem proving. In Bernhard Gramlich, editor, *Frontiers of Combining Systems, 5th International Workshop, FroCoS 2005, Vienna, Austria, September 19-21, 2005, Proceedings*, volume 3717 of *Lecture Notes in Computer Science*, pages 310–320. Springer, 2005. 220

[CF58] Haskell B Curry and Robert Feys. Combinatory logic, volume i of studies in logic and the foundations of mathematics, 1958. 1, 18

[CH88] Thierry Coquand and Gerard Huet. The calculus of constructions. *Information and computation*, 76(2) :95–120, 1988. 7

[Chu32] Alonzo Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33(2) :pp. 346–366, 1932. 1

[Chu36] Alonzo Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2) :pp. 345–363, 1936. 1

[Chu40] Alonzo Church. A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5 :56–68, 6 1940. 1

[Coq85] Thierry Coquand. *Une théorie des constructions*. PhD thesis, 1985. 7

[Coq86] Thierry Coquand. An analysis of girard's paradox. In *Logic in Computer Science*, pages 227–236, 1986. 7

[Dow93] Gilles Dowek. The undecidability of typability in the lambda-pi-calculus. In Bezem and Groote [**BG93**], pages 139–145. 32

[Fri02] Daniel Fridlender. A proof-irrelevant model of martin-löf's logical framework. *Mathematical Structures in Computer Science*, 12(6) :771–795, 2002. 61

[GAA+13] Georges Gonthier, Andrea Asperti, Jeremy Avigad, Yves Bertot, Cyril Cohen, François Garillot, Stéphane Le Roux, Assia Mahboubi, Russell O'Connor, Sidi Ould Biha, Ioana Pasca, Laurence Rideau, Alexey Solovyev, Enrico Tassi, and Laurent Théry. A machine-checked proof of the odd order theorem. In Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie, editors, *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 163–179. Springer, 2013. 2

[Gen35] Gerhard Gentzen. Untersuchungen über das logische schließen. i. *Mathematische Zeitschrift*, 39(1) :176–210, 1935. 21

[Geu92] Herman Geuvers. The church-rosser property for-reduction in typed-calculi. In *Seventh Annual IEEE Symposium on Logic in Computer Science*, pages 453–460. Citeseer, 1992. 33, 34

[GHRDR93] Paola Giannini, Furio Honsell, and Simona Ronchi Della Rocca. Type inference : some results, some problems. *Fundamenta Informaticæ*, 19(1-2) :87 – 125, 1993. 32

[Gir72] Jean-Yves Girard. Interprétation fonctionelle et élimination des coupures de l'arithmétique d'ordre supérieur. 1972. 7

[Gir86] Jean-Yves Girard. The system f of variable types, fifteen years later. *Theor. Comput. Sci.*, 45(2) :159–192, September 1986. 73

[Gir06] J.Y. Girard. *Le point aveugle : cours de logique. Vers la perfection*. Le point aveugle : cours de logique. Hermann, 2006. 73

[Glo12] Stéphane Glondu. *Vers une certification de l'extraction de Coq*. PhD thesis, Université Paris 7, June 2012. 2

[GN91] Herman Geuvers and Mark-Jan Nederhof. Modular proof of strong normalization for the calculus of constructions. *J. Funct. Program.*, 1(2) :155–189, 1991. 20

[Gon07] Georges Gonthier. The four colour theorem : Engineering of a formal proof. In Deepak Kapur, editor, *Computer Mathematics, 8th Asian Symposium, ASCM 2007, Singapore, December 15-17, 2007. Revised and Invited Papers*, volume 5081 of *Lecture Notes in Computer Science*, page 333. Springer, 2007. 2

[GTL89] Jean-Yves Girard, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge University Press, New York, NY, USA, 1989. 73

[Hof97] Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997. 38

[How80] W Howard. The formulae-as-type notion of construction. to hb curry : Essays in combinatory logic, lambda calculus and formalism, seldin j, hindley j, 1980. 1, 18

[KR35] S. C. Kleene and J. B. Rosser. The inconsistency of certain formal logics. *Annals of Mathematics*, 36(3) :pp. 630–636, 1935. 1

[Lei83] Daniel Leivant. Polymorphic type inference. In *Proceedings of the 10th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, POPL '83, pages 88–98, New York, NY, USA, 1983. ACM. 7

[Ler06] Xavier Leroy. Formal certification of a compiler back-end or : programming a compiler with a proof assistant. In *Principles of Programming Languages (POPL)*, pages 42–54, January 2006. 2

[Let04] Pierre Letouzey. *Programmation fonctionnelle certifiée – L'extraction de programmes dans l'assistant Coq*. PhD thesis, Université Paris-Sud, July 2004. 2

[Luo90] Z. Luo. *An Extended Calculus of Constructions*. PhD thesis, University of Edinburgh, 1990. 21

[Luo94] Zhaohui Luo. *Computation and Reasoning : A Type Theory for Computer Science*. Oxford University Press, Inc., New York, NY, USA, 1994. 13, 216

[McB04] Conor McBride. Epigram : Practical programming with dependent types. In Varmo Vene and Tarmo Uustalu, editors, *Advanced Functional Programming, 5th International School, AFP 2004, Tartu, Estonia, August 14-21, 2004, Revised Lectures*, volume 3622 of *Lecture Notes in Computer Science*, pages 130–170. Springer, 2004. 220

[Men88] Paul Francis Mendler. *Inductive Definition in Type Theory*. PhD thesis, Cornell University, Ithaca, NY, USA, January 1988. 3

[Miq01] Alexandre Miquel. *Le Calcul des Constructions implicite : syntaxe et sémantique*. PhD thesis, Université Paris 7, December 2001. 7, 34, 61, 73

[Mit88] John C. Mitchell. Polymorphic type inference and containment. *Information and Computation*, 76(2–3) :211 – 249, 1988. 7

[ML75] Per Martin-Löf. An intuitionistic theory of types : Predicative part. *Studies in Logic and the Foundations of Mathematics*, 80 :73–118, 1975. 23

[ML85] P. Martin-Löf. Constructive mathematics and computer programming. In *Proc. Of a Discussion Meeting of the Royal Society of London on Mathematical Logic and Programming Languages*, pages 167–184, Upper Saddle River, NJ, USA, 1985. Prentice-Hall, Inc. 3, 220

[ML98] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer-Verlag, 2nd edition, September 1998. 73

[ML08] Richard Nathan Mishra-Linger. *Irrelevance, polymorphism, and erasure in type theory*. PhD thesis, Portland State University, November 2008. 220

[MLS84] Per Martin-Lof and Giovanni Sambin. *Intuitionistic type theory*, volume 17. Bibliopolis Naples„ Italy, 1984. 23, 223

[MS82] D. B. MacQueen and Ravi Sethi. A semantic model of types for applicative languages. In *Proceedings of the 1982 ACM Symposium on LISP and Functional Programming*, LFP '82, pages 243–252, New York, NY, USA, 1982. ACM. 7

[Nor07] Ulf Norell. *Towards a practical programming language based on dependent type theory*. PhD thesis, Department of Computer Science and Engineering, Chalmers University of Technology, SE-412 96 Göteborg, Sweden, September 2007. 220

[Pfe01] Frank Pfenning. Intensionality, extensionality, and proof irrelevance in modal type theory. In J. Halpern, editor, *Proceedings of the 16th Annual Symposium on Logic in Computer Science (LICS'01)*, pages 221–230, Boston, Massachusetts, June 2001. IEEE Computer Society Press. 220

[PM89] Christine Paulin-Mohring. Extracting F_ω's programs from proofs in the calculus of constructions. In *Principles of Programming Languages (POPL)*, pages 89–104, January 1989. 2

[PM96] C. Paulin-Mohring. *Définitions Inductives en Théorie des Types d'Ordre Supérieur*. Habilitation à diriger les recherches, Université Claude Bernard Lyon I, December 1996. 2

[PPM90] F. Pfenning and C. Paulin-Mohring. Inductively defined types in the Calculus of Constructions. In *Proceedings of Mathematical Foundations of Programmi ng Semantics*, volume 442 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990. technical report CMU-CS-89-209. 2

[Pra65] Dag Prawitz. *Natural Deduction : a Proof-Theoretical Study*. Number 3 in Stockholm Studies in Philosophy. Almquist and Wiskell, 1965. 21

[Rey74] John C Reynolds. Towards a theory of type structure. In *Programming Symposium*, pages 408–425. Springer, 1974. 7

[Sco76] Dana S. Scott. Data types as lattices. *SIAM J. Comput.*, 5(3) :522–587, 1976. 73

[Sco82] Dana S. Scott. Domains for denotational semantics. In Mogens Nielsen and Erik Meineche Schmidt, editors, *Automata, Languages and Programming, 9th Colloquium, Aarhus, Denmark, July 12-16, 1982, Proceedings*, volume 140 of *Lecture Notes in Computer Science*, pages 577–613. Springer, 1982. 73

[SF14] Aleksy Schubert and Ken-etsu Fujita. A note on subject reduction in $(\rightarrow, \exists)$-curry with respect to complete developments. *Inf. Process. Lett.*, 114(1-2) :72–75, 2014. 220

[SU98] Morten Heine B. Sørensen and Pawel Urzyczyn. Lectures on the curry-howard isomorphism, 1998. 24

[Tak95] Masako Takahashi. Parallel reductions in λ-calculus. *Information and computation*, 118(1) :120–127, 1995. 223, 233

[Tat07] Makoto Tatsuta. Simple saturated sets for disjunction and second-order existential quantification. In Simona Ronchi Della Rocca, editor, *TLCA*, volume 4583 of *Lecture Notes in Computer Science*, pages 366–380. Springer, 2007. 24

[The15] The Coq development team. The coq proof assistant reference manual v8.4pl6. Technical report, INRIA, France, April 2015. `http://coq.inria.fr/`. 2

[Urz93] Pawel Urzyczyn. Type reconstruction in f-omega is undecidable. In Bezem and Groote [**BG93**], pages 418–432. 32

[Urz97] Pawel Urzyczyn. Type reconstruction in f_{omega}. *Mathematical Structures in Computer Science*, 7(4) :329–358, 1997. 32

[Wel96] Joseph Brian Wells. *Type Inference for System F with and without the Eta Rule*. PhD thesis, Boston University, 1996. 33

[Wel99] J. B. Wells. Typability and type checking in system F are equivalent and undecidable. *Ann. Pure Appl. Logic*, 98(1-3) :111–156, 1999. 32

[Wer94] Benjamin Werner. *Une Théorie des Constructions Inductives*. Theses, Université Paris-Diderot - Paris VII, May 1994. 2

[Xi06] Hongwei Xi. Development separation in lambda-calculus. *Electr. Notes Theor. Comput. Sci.*, 143 :207–221, 2006. 220

[XP99] Hongwei Xi and Frank Pfenning. Dependent types in practical programming. In *Proceedings of the 26th ACM SIGPLAN Symposium on Principles of Programming Languages*, pages 214–227, San Antonio, January 1999. 220

Table des figures

Résumé

Cette thèse propose un langage permettant de développer des programmes certifiés purement calculatoires. Pour cela deux systèmes de types sont présentés. Le premier, ICC_Σ, est une extension du Calcul des Constructions Implicite de Miquel (ICC), enrichi avec des sommes dépendantes. ICC_Σ est un système à la Curry avec des opérateurs de type implicite. Ces opérateurs permettent aux indications logiques de ne pas apparaître explicitement dans les termes, rendant possible l'écriture de de programmes certifiés purement calculatoires. L'ajout des sommes dépendantes donne plus d'expressivité au système et est un premier pas en vue de l'ajout des types inductifs, déjà présents dans l'assistant de preuves Coq. Une limitation de ICC_Σ est que l'inférence de type est vraisemblablement indécidable. Pour contourner ce problème, nous définisssons un deuxième système : le Calcul des Constructions Implicite Annoté avec sommes dépendantes, noté $AICC_\Sigma$. $AICC_\Sigma$ est un système de types équivalent à ICC_Σ, mais avec une syntaxe bicolore à la Church. Les indications logiques des programmes apparaissent explicitement, rendant l'inférence de type décidable, mais peuvent être marquées. $AICC_\Sigma$ est muni d'un mécanisme interne d'effacement qui transforme un terme de $AICC_\Sigma$ avec indications logiques en un terme de ICC_Σ purement calculatoire. Nous prouvons que cet effacement est correct et complet, ce qui signifie que les programmes valides de $AICC_\Sigma$ correspondent exactement à ceux de ICC_Σ. Nous définissons également un algorithme d'inférence de type correct et complet. La combinaison de cet algorithme et du mécanisme d'effacement rend possible la programmation certifiée dans ICC_Σ.

Mots clés : Théorie des types, arguments implicites, programmation certifiée, sommes dépendantes, langages fonctionnels, inférence de type.

Abstract

This thesis presents a programming language for developing purely computational certified programs with no explicit logical information. We present two type systems. The first one, ICC_Σ, extends Miquel's Implicit Calculus of Constructions (ICC) with dependent sums. ICC_Σ is a Curry-style system with implicit type operators. With these operators, logical information may remain implicit, which makes it possible to write purely computational certified programs. Adding dependent sums gives more expressive power to the system and is a first step towards adding inductive types that the Coq proof assistant already has. One drawback of ICC_Σ is that type inference is likely to be undecidable. To solve that issue, we define a second system : the Annotated Implicit Calculus of Constructions, $AICC_\Sigma$. $AICC_\Sigma$ is a type system equivalent to ICC_Σ but with a bicolor Church-style syntax. Logical information may appear explicitly, thus making type inference decidable, but is flagged. A built-in erasure procedure transforms an annotated $AICC_\Sigma$ term into a purely computational ICC_Σ term. We prove that this procedure is correct and complete, which implies that valid $AICC_\Sigma$ programs correspond exactly to valid ICC_Σ ones. We also define a correct and complete type inference algorithm. This algorithm, combined with the erasure procedure makes certified programming in ICC_Σ possible.

Keywords : Type theory, implicit arguments, certified programming, dependent sums, functional languages, type inference.