# Théorie des types et réécriture

Frédéric Blanqui

N. D'ORDRE :

**UNIVERSITÉ PARIS XI**
**UFR SCIENTIFIQUE D'ORSAY**

THÈSE

Présentée

Pour obtenir

Le GRADE de DOCTEUR EN SCIENCES

DE L'UNIVERSITÉ PARIS XI ORSAY

PAR

**Frédéric BLANQUI**

Sujet :

# Théorie des Types et Récriture

Soutenue le 28 septembre 2001
devant la Commission d'examen composée de :

Mr Thierry COQUAND
Mr Gilles DOWEK
Mr Herman GEUVERS
Mr Jean-Pierre JOUANNAUD
Mme Christine PAULIN
Mr Miklos SANTHA

Le 19 mars 1998 est une date importante pour au moins deux raisons. La première est personnelle. La seconde est que, ce jour-là, Jean-Pierre Jouannaud accepta de me prendre comme stagiaire de DEA pour "étendre au Calcul des Constructions une nouvelle version du Schéma Général" qu'il avait ébauchée avec Mitsuhiro Okada. Cela ne signifiait pas encore grand chose pour moi. Cependant, l'idée d'avoir à étudier à la fois le $\lambda$-calcul et la récriture, et bien sûr leur interaction, m'enthousiasma. C'est de cet enthousiasme que résulte ce travail.

C'est pourquoi je commencerais par remercier Jean-Pierre Jouannaud, pour l'honneur qu'il m'a fait, la confiance, l'aide, les conseils et les encouragements qu'il m'a prodigués tout au long de ces trois années de thèse. Il m'a appris beaucoup et je lui en serai toujours reconnaissant.

Je remercie également Mitsuhiro Okada pour les discussions que nous avons eues ensemble et le soutien qu'il m'a apporté. Ce fût un grand honneur pour moi d'avoir eu la possibilité de travailler avec lui. J'espère que nous aurons de nombreuses autres collaborations fructueuses.

Je remercie également Maribel Fernández qui m'a beaucoup aidé au début de ma thèse en co-encadrant avec Jean-Pierre Jouannaud mon stage de DEA.

Je remercie également Gilles Dowek qui m'a encouragé dans mon travail et m'a aidé en plusieurs occasions importantes. Son travail a été pour moi (et est encore !) une source importante de réflexion et d'inspiration.

Je remercie également Daria Walukiewicz avec qui j'ai eu de nombreuses discussions très fructueuses. Je la remercie vivement pour avoir lu en détail une bonne partie de cette thèse et pour m'avoir aidé à corriger quelques erreurs ou imprécisions.

Je remercie également toutes les personnes de l'équipe DÉMONS du LRI et de l'équipe Coq de l'INRIA Rocquencourt (nouvellement baptisée LogiCal), en particulier Christine Paulin et Claude Marché qui m'ont aidé à plusieurs reprises. Ces deux équipes constituent un lieu privilégié de recherche dans une ambiance très agréable.

Je remercie également les rapporteurs de cette thèse, Thierry Coquand et Herman Geuvers, pour l'intérêt qu'ils ont porté à mon travail et les remarques qu'ils m'ont faites pour l'améliorer.

Enfin, je remercie également les membres du jury et le président du jury pour l'honneur qu'ils me font d'examiner ce travail.

# Table des matières

# Chapitre 1

# Introduction

Qu'est-ce que bien programmer ? Outre le fait d'écrire des programmes compréhensibles et exploitables par d'autres personnes, il s'agit avant tout de parvenir à écrire des programmes sans erreur. Mais comment savoir si un programme n'a pas d'erreur ? En le démontrant. Autrement dit, bien programmer nécessite de faire des mathématiques.

Mais comment savoir si la démonstration que le programme n'a pas d'erreur n'a elle-même pas d'erreur ? En écrivant une démonstration qui peut être vérifiée par une machine. Autrement dit, bien programmer nécessite de faire des mathématiques formelles.

C'est à cela que nous nous intéressons dans cette thèse : définir un système formel dans lequel on puisse programmer et montrer qu'un programme est correct.

Mais il ne faut pas croire que le travail se trouve multiplié par deux : programmer et démontrer. En fait, d'une démonstration qu'une spécification est correcte, on peut extraire un programme sans erreur ! Cela est dû à la terminaison de "l'élimination des coupures" en logique intuitionniste découverte par G. Gentzen en 1933 [55].

Plus précisément, nous allons considérer une classe particulière de systèmes formels, les systèmes de types. Nous allons étudier les propriétés des systèmes de types lorsqu'on y ajoute des définitions par récriture. La récriture est un paradigme de calcul simple et général à base de règles telles que $x + 0 \to x$, c'est-à-dire, si j'ai une expression de la forme $x + 0$, alors je peux la simplifier en $x$.

Mais pour qu'un tel système serve à prouver la correction de programmes, il faut s'assurer que lui-même est correct, c'est-à-dire, qu'on ne puisse pas y montrer n'importe quoi. C'est pourquoi nous allons donner des conditions sur les définitions et montrer que ces conditions assurent la correction du système.

Avant cela, revenons un peu en arrière et voyons comment sont apparus les systèmes de types, quels résultats ont déjà été obtenus et quelles sont nos contributions (résumées dans le Sous-chapitre 1.4).

## 1.1 Historique

Ce sous-chapitre n'a pas la prétention de fournir un aperçu historique absolument rigoureux. Nous souhaitons seulement rappeler les concepts de base sur lesquels repose notre travail (types, $\lambda$-calcul, etc.) et montrer comment celui-ci s'inscrit dans la continuation de travaux visant à introduire toujours plus de calcul dans la logique ou, dualement, à introduire toujours plus de logique dans la programmation. Nous prendrons donc certaines libertés avec le formalisme utilisé.

Le lecteur rompu à ces notions (en particulier le Calcul des Constructions et le Calcul des Constructions Inductives) peut aller directement au Sous-chapitre 1.2 suivant où nous présentons nos motivations pour l'ajout dans le Calcul des Constructions de récriture aussi bien au niveau des objets qu'au niveau des prédicats.

### La Théorie des ensembles

Un des premiers systèmes formels qui permettent de décrire l'ensemble des mathématiques est la *théorie des ensembles* de E. Zermelo (1908) étendue plus tard par A. Fraenkel (1922). Il a été suivi par la *théorie des types* de A. Whitehead et B. Russell (1911) [119], aussi appelée *logique d'ordre supérieur*. Ces deux systèmes formels ont été introduits pour répondre à l'incohérence de la *théorie des ensembles* de G. Cantor (1878).

En logique du premier ordre, dans laquelle est généralement exprimée la théorie des ensembles de E. Zermelo et A. Fraenkel, les objets du discours sont définis à partir de constantes et de symboles de fonctions $(0, +, \ldots)$. Ensuite, des symboles de prédicats $(\in, \ldots)$, les connecteurs logiques $(\vee, \wedge, \Rightarrow, \ldots)$ et les quantificateurs universels et existentiels $(\forall, \exists)$ permettent d'exprimer des propositions sur ces objets.

Un des axiomes de la théorie des ensembles de G. Cantor est l'*axiome de compréhension* qui dit que toute proposition définit un ensemble :

$$(\exists x)(\forall y)\ y \in x \Leftrightarrow P(y)$$

À partir de cet axiome, on peut exprimer le paradoxe de Russell (1902). En prenant $P(x) = x \notin x$, on peut définir l'ensemble $R$ des $x$ qui n'appartiennent pas à eux-mêmes. Alors, $R \in R \Leftrightarrow R \notin R$ et on peut déduire que toute proposition est vraie. Pour remédier à ce problème, E. Zermelo a proposé de restreindre l'axiome de compréhension de la manière suivante :

$$(\forall z)(\exists x)(\forall y)\ y \in x \Leftrightarrow y \in z \wedge P(y)$$

c'est-à-dire que, désormais, on ne pourra définir par compréhension que des sous-ensembles d'ensembles précédemment définis.

### La Théorie des types

En théorie des types, l'idée n'est pas de restreindre l'axiome de compréhension mais d'interdire des expressions comme $x \notin x$ ou $x \in x$, en restreignant l'application

d'un prédicat à un objet. Pour cela, on associe à chaque symbole de fonction et à chaque symbole de prédicat (sauf le symbole d'appartenance) un *type* de la manière suivante :

- à une constante, on associe le type $\iota$,
- à une fonction prenant un argument, on associe le type $\iota \to \iota$,
- à une fonction prenant deux arguments, on associe le type $\iota \to \iota \to \iota$,
- ...
- à une proposition, on associe le type $o$,
- à un prédicat prenant un argument, on associe le type $\iota \to o$,
- à un prédicat prenant deux arguments, on associe le type $\iota \to \iota \to o$,
- ...

Alors, on peut appliquer une fonction $f$ prenant $n$ arguments à des objets $t_1, \ldots, t_n$ si le type de $f$ est $\iota \to \ldots \to \iota \to \iota$ et chacun des $t_i$ est de type $\iota$. Et on peut écrire que des objets $t_1, \ldots, t_n$ vérifient un prédicat $P$ à $n$ arguments si le type de $P$ est $\iota \to \ldots \to \iota \to o$ et chacun des $t_i$ est de type $\iota$.

Enfin, on considère qu'un ensemble n'est plus un objet (une expression de type $\iota$) mais un prédicat (une expression de type $\iota \to o$). Et, pour représenter $x \in E$, qui signifie que $x$ vérifie $E$, on écrira $Ex$ (application de $E$ à $x$). Ainsi, on vérifie aisément qu'il n'est pas possible d'exprimer le paradoxe de Russell : on ne peut pas écrire $xx$ pour représenter $x \in x$ car il faudrait que $x$ soit à la fois de type $\iota \to o$ et de type $\iota$. Par la suite, on écrira $t : \tau$ pour signifier que $t$ est de type $\tau$.

Maintenant, pour représenter les entiers naturels, il y a plusieurs possibilités. Mais il est toujours nécessaire de poser un axiome d'infinité pour $\iota$ et de pouvoir exprimer l'ensemble des entiers naturels comme le plus petit ensemble contenant zéro et stable par incrémentation. Pour cela, il faut pouvoir quantifier sur des ensembles, c'est-à-dire sur des expressions de type $\iota \to o$ et non plus seulement sur des expressions de type $\iota$.

C'est là que le terme de logique d'*ordre supérieur* prend tout son sens. Désormais, on ne va plus se restreindre aux expressions d'objets et de prédicats décrites précédemment, on va considérer l'ensemble des expressions que l'on peut former par application en respectant les types, quels que soient ceux-ci :

- On définit l'*ensemble des types simples* comme le plus petit ensemble $T$ contenant $\iota$, $o$ et $\sigma \to \tau$ dès lors que $\sigma$ et $\tau$ appartiennent à $T$.
- On définit l'*ensemble des termes de type* $\tau$ comme le plus petit ensemble contenant les constantes de type $\tau$ et les applications $tu$ dès lors que $t$ est un terme de type $\sigma \to \tau$ et $u$ un terme de type $\sigma$.

Enfin, on introduit une notation explicite pour désigner les fonctions et les ensembles, la $\lambda$-*abstraction*, et on inclut dans les constantes les connecteurs et quantificateurs en leur assignant les types suivants : $\vee : o \to o \to o$, $\wedge : o \to o \to o$, $\forall_\tau : (\tau \to o) \to o$, ... Par exemple, si $\iota$ désigne l'ensemble des entiers naturels, on peut désigner le prédicat "est pair" (de type $\iota \to o$) par l'expression $pair = \lambda x : \iota . \exists_\iota (\lambda y : \iota . x = 2 \times y)$ qu'on abrégera par $\lambda x : \iota . \exists y : \iota . x = 2 \times y$. Le langage des termes ainsi obtenu s'appelle le $\lambda$-*calcul simplement typé*, noté $\lambda^{\to}$.

Mais que dire de $pair\,2$ et $\exists y : \iota.2 = 2 \times y$ ? La seconde expression s'obtient en substituant $x$ par 2 dans le corps de $pair$. Cette opération de substitution s'appelle la *$\beta$-réduction*. De manière générale, $\lambda x : \tau.\, t$ appliqué à $u$ se $\beta$-réduit en le terme $t$ où $x$ est substitué par $u$ : $\lambda x : \tau.\, t\ u \to_\beta t\{x \mapsto u\}$.

Il est assez naturel de considérer ces deux expressions comme dénotant la même proposition. C'est pourquoi on rajoute l'*axiome de conversion* suivant :

$$P \Leftrightarrow Q \quad \text{si} \quad P \to_\beta Q$$

On aboutit alors à la *théorie des types* de A. Church (1940) [28].

Dans une telle théorie, il est possible de quantifier sur l'univers des propositions lui-même : $\forall P : o.P \Rightarrow P$. Autrement dit, une proposition peut être définie en quantifiant sur l'ensemble des propositions, y compris elle-même. Si on autorise de telles quantifications, on dit que la théorie est *imprédicative*, sinon on dit qu'elle est *prédicative*.

### Les mathématiques comme langage de programmation

La $\beta$-réduction correspond au processus d'évaluation d'une fonction. Lorsqu'on a une fonction $f$ définie par une expression $f(x)$ et qu'on veut sa valeur en 5 par exemple, on va substituer $x$ par 5 dans $f(x)$ et simplifier l'expression obtenue jusqu'à obtenir la valeur de $f(5)$.

On peut alors se demander quelles fonctions on peut ainsi définir dans la théorie des types de A. Church. En fait, bien peu. Sur les entiers de Peano (*i.e.* en prenant $0 : \iota$ pour zéro et $s : \iota \to \iota$ pour la fonction successeur), on ne peut exprimer que des fonctions constantes et des fonctions qui ajoutent une constante à l'un de leurs arguments. Sur les entiers de A. Church, où $n$ est représenté par le terme $\lambda x : \iota.\lambda f : \iota \to \iota.f \ldots fx$ avec $n$ occurrences de $f$, H. Schwichtenberg [105] montra qu'on ne peut exprimer que des polynômes étendus (plus petit ensemble de fonctions clos par composition et contenant la fonction nulle, la fonction successeur, les projections, l'addition, la multiplication, la fonction caractéristique de $\{0\}$ et la fonction caractéristique de $\mathbb{N} \setminus \{0\}$).

Bien sûr, il est possible de montrer l'existence de nombreuses fonctions, c'est-à-dire de prouver une proposition de la forme $(\forall x)(\exists y)\, Pxy$ où le prédicat $P$ représente le graphe de la fonction. Dans la théorie des types intuitionniste par exemple (*i.e.* sans l'axiome du tiers-exclu $P \vee \neg P$), il est possible de montrer l'existence de toutes les fonctions primitives récursives. Mais on ne dispose pas de terme $f : \iota \to \iota$ qui nous permette de *calculer* les puissances de 2 par exemple, c'est-à-dire, tel que $fn \to_\beta \ldots \to_\beta 2^n$.

### La représentation des démonstrations

G. Frege et D. Hilbert ont proposé de représenter une démonstration d'une proposition $Q$ comme une suite $P_1, \ldots, P_n$ de propositions telles que $P_n = Q$ et,

pour chaque $i$, soit $P_i$ est un axiome, soit $P_i$ est une conséquence des propositions précédentes par *modus ponens* (de $P$ et $P \Rightarrow Q$ on peut déduire $Q$) ou par *généralisation* (de $P(x)$ avec $x$ quelconque on peut déduire $(\forall x)P(x)$). Cependant, pour faire de telles démonstrations, il est nécessaire de considérer de nombreux axiomes, indépendants de toute théorie, qui expriment le sens des connecteurs logiques.

Plus tard, en 1933, G. Gentzen [55] a proposé un nouveau système de déduction où les axiomes logiques sont remplacés par des *règles d'introduction* et *d'élimination* des connecteurs et quantificateurs :

$$\text{(axiome)}\ \frac{}{\Gamma, P, \Gamma' \vdash P}$$

$$(\wedge\text{-intro})\ \frac{\Gamma \vdash P \quad \Gamma \vdash Q}{\Gamma \vdash P \wedge Q} \qquad (\wedge\text{-élim1})\ \frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash P} \qquad (\wedge\text{-élim2})\ \frac{\Gamma \vdash P \wedge Q}{\Gamma \vdash Q}$$

$$(\Rightarrow\text{-intro})\ \frac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q} \qquad (\Rightarrow\text{-élim})\ \frac{\Gamma \vdash P \Rightarrow Q \quad \Gamma \vdash P}{\Gamma \vdash Q}$$

$$(\exists\text{-intro})\ \frac{\Gamma \vdash P(t)}{\Gamma \vdash (\exists x)P(x)} \qquad (\exists\text{-élim})\ ^{1}\ \frac{\Gamma \vdash (\exists x)P \quad \Gamma, P \vdash Q}{\Gamma \vdash Q} \qquad \ldots$$

où $\Gamma$ est un ensemble de propositions (les hypothèses). Un couple $\Gamma \vdash Q$ formé d'un ensemble de propositions $\Gamma$ et d'une proposition $Q$ s'appelle un *séquent*. Alors, une démonstration d'un séquent $\Gamma \vdash Q$ est un arbre dont la racine est $\Gamma \vdash Q$, les noeuds sont des instances des règles de déduction et les feuilles sont des applications de la règle (axiome).

### L'élimination des coupures

G. Gentzen a remarqué que certaines démonstrations peuvent être simplifiées. Par exemple, cette démonstration de $Q$ :

$$\frac{\dfrac{\Gamma, P \vdash Q}{\Gamma \vdash P \Rightarrow Q}\,(\Rightarrow\text{-intro}) \qquad \Gamma \vdash P}{\Gamma \vdash Q}\,(\Rightarrow\text{-élim})$$

fait un détour qui peut être éliminé. Il suffit de remplacer dans la preuve de $\Gamma, P \vdash Q$ les feuilles (axiome) donnant $\Gamma, P, \Gamma' \vdash P$ par la preuve de $\Gamma \vdash P$ où $\Gamma$ est remplacé par $\Gamma, \Gamma'$. En fait, à tout endroit où il y a une *coupure*, c'est-à-dire une règle d'introduction suivie d'une règle d'élimination du même connecteur, il est possible de simplifier la démonstration. Il montra alors le résultat remarquable suivant : le processus d'élimination des coupures termine.

Ainsi, toute proposition démontrable admet une démonstration sans coupure. Or, en logique intuitionniste, toute preuve sans coupure d'une proposition $(\exists x)P(x)$ doit nécessairement se terminer par une règle d'introduction dont la prémisse est de la forme $P(t)$. Donc, le processus d'élimination des coupures fournit le témoin $t$

1. $x$ ne doit apparaître ni dans $\Gamma$ ni dans $Q$.

de la proposition d'existence. Autrement dit, toute fonction dont on peut prouver l'existence est calculable.

Si on peut exprimer les démonstrations elles-mêmes comme des objets de la théorie, il devient alors possible d'exprimer bien plus de fonctions que celles permises par le $\lambda$-calcul simplement typé.

### L'isomorphisme de Curry-de Bruijn-Howard

En 1958, Curry [38] a remarqué qu'aux types du $\lambda$-calcul simplement typé on peut faire correspondre les propositions formées à partir de l'implication seulement (il identifie $\to$ et $\Rightarrow$) et qu'à un terme de type $\tau$ on peut faire correspondre une preuve de la proposition correspondant à $\tau$. Autrement dit, le $\lambda$-calcul simplement typé permet de représenter les preuves de la logique propositionnelle minimale. Pour cela, à chaque proposition $P$ on associe une variable particulière $x_P$ de type $P$. Alors, on définit le $\lambda$-terme associé à une preuve par récurrence sur la taille de la preuve :

- une preuve de $\Gamma \vdash P$ obtenue par (axiome) est représentée par la variable $x_P$ ;
- une preuve de $\Gamma \vdash P \Rightarrow Q$ obtenue par ($\Rightarrow$-intro) à partir d'une preuve $\pi$ de $\Gamma, P \vdash Q$ est représentée par le terme $\lambda x_P : P.\, t$ où $t$ représente $\pi$ ;
- une preuve de $\Gamma \vdash Q$ obtenue par ($\Rightarrow$-élim) à partir d'une preuve $\pi$ de $\Gamma \vdash P \Rightarrow Q$ et d'une preuve $\pi'$ de $\Gamma \vdash P$ est représentée par le terme $tu$ où $t$ représente $\pi$ et $u$ représente $\pi'$.

L'ensemble de $\lambda$-termes ainsi obtenus peut être défini directement de la manière suivante. On appelle *environnement* un ensemble $\Gamma$ de couples $x : P$ formés d'une variable $x$ et d'un type $P$ (représentant une proposition). Alors, un terme $t$ est de type $P$ (une preuve de $P$) dans l'environnement $\Gamma$ si $\Gamma \vdash t : T$ peut être déduit des règles d'inférence suivantes :

$$\text{(axiome)}\ \frac{}{\Gamma, x : P, \Gamma' \vdash x : P}$$

$$(\Rightarrow\text{-intro})\ \frac{\Gamma, x : P \vdash t : Q}{\Gamma \vdash \lambda x : P.\, t : P \Rightarrow Q}$$

$$(\Rightarrow\text{-élim})\ \frac{\Gamma \vdash t : P \Rightarrow Q \quad \Gamma \vdash u : P}{\Gamma \vdash tu : Q}$$

En 1965, W. W. Tait [110] a remarqué que la $\beta$-réduction correspond à l'élimination des coupures. En effet, si on annote l'exemple de coupure donné précédemment, on obtient :

$$\frac{\dfrac{\Gamma, x : P \vdash t : Q}{\Gamma \vdash \lambda x : P.\, t : P \Rightarrow Q}\,(\Rightarrow\text{-intro}) \qquad \Gamma \vdash u : P}{\Gamma \vdash \lambda x : P.\, t\ u : Q}\,(\Rightarrow\text{-élim})$$

Si on $\beta$-réduit $\lambda x : P.\, t\ u$ en $t\{x \mapsto u\}$, on obtient exactement le terme correspondant à la preuve sans coupure de $\Gamma \vdash Q$. Ainsi, l'existence d'une preuve sans coupure correspond à la *normalisation faible* de la $\beta$-réduction, c'est-à-dire, à l'existence, pour tout $\lambda$-terme typable $t$, d'une séquence de $\beta$-réductions aboutissant à un terme $\beta$-irréductible (on dit encore en *forme normale*). C'est pourquoi la normalisation occupe une place si importante dans l'étude des systèmes de types.

En 1968, N. de Bruijn [39] a proposé un système de *types dépendants* étendant le $\lambda$-calcul simplement typé et dans lequel il est possible d'exprimer les propositions et les preuves de toute la logique intuitionniste du premier ordre. Ce système a été à la base d'un des tous premiers programmes informatiques de démonstration formelle : AUTOMATH. Un type dépendant est tout simplement une fonction qui, à un objet, associe une expression de type. Il permet de représenter les prédicats et les quantifications sur des objets. En 1969, W. A. Howard [67] a considéré un système similaire mais sans le considérer comme un système logique à part entière.

Dans un système avec types dépendants, la bonne formation des types dépend de la bonne formation des termes. Il est donc nécessaire de considérer des environnements avec des variables de types et de rajouter des règles assurant la bonne formation des types et des environnements (l'ordre des variables y étant désormais important). Enfin, il est nécessaire de rajouter une règle de conversion qui permette d'identifier les propositions $\beta$-équivalentes. On aboutit alors à un ensemble de règles de typage semblables à celles de la Figure 1.1 (il s'agit là d'une présentation moderne qui n'émergera qu'à la fin des années 80).

Dans ce système, $\star$ est le type des propositions et des ensembles du discours (entiers naturels, etc.) et $\Box$ celui des types de prédicat (dont fait partie $\star$). Par exemple, l'ensemble des entiers naturels $nat$ a le type $\star$, le prédicat $pair$ a le type $(n : nat)\star$ qu'on abrège par $nat \to \star$ car $n$ n'apparaît pas dans $\star$ (produit non dépendant) et $nat \to \star$ a le type $\Box$. Partant de la règle (ax), les règles (var) et (weak) permettent de construire des environnements bien formés. La règle (prod-$\lambda^{\to}$) permet de former des propositions et la règle (prod-$\lambda$P) des types de prédicats. Dans le cas d'une proposition, si le produit n'est pas dépendant ($x$ n'apparaît pas dans $U$), il s'agit d'une implication ; sinon il s'agit d'une quantification. Autrement dit, en l'absence de la règle (prod-$\lambda$P), on retrouve le $\lambda$-calcul simplement typé. La règle (abs) permet de former une fonction (si $s = \star$) ou un prédicat (si $s = \Box$). Enfin, la règle (app) permet l'application d'une fonction ou d'un prédicat à son argument. Autrement dit, les règles (abs) et (app) généralisent les règles ($\Rightarrow$-intro) et ($\Rightarrow$-élim) du $\lambda$-calcul simplement typé.

Du point de vue de la programmation, les types dépendants permettent d'accroître l'information sur les données et donc de réduire les erreurs. Par exemple, on peut définir le type $(list\ n)$ des listes d'entiers de longueur $n$ en déclarant $list : nat \to \star$. Alors, la liste vide $nil$ a le type $(list\ 0)$ et la fonction $cons$ qui ajoute un entier $x$ en tête d'une liste $\ell$ de longueur $n$ a le type $nat \to (n : nat)(list\ n) \to (list\ (s\ n))$. On peut ainsi chercher à vérifier si une liste ne dépasse pas une certaine longueur (dépassement d'indice dans un tableau).

centering

FIGURE 1.1 – Règles de typage de $\lambda$P

$$\text{(ax)} \quad \frac{}{\vdash \star : \Box}$$

$$\text{(var)} \quad \frac{\Gamma \vdash T : s \in \{\star, \Box\}}{\Gamma, x{:}T \vdash x : T}$$

$$\text{(weak)} \quad \frac{\Gamma \vdash t : T \quad \Gamma \vdash U : s \in \{\star, \Box\}}{\Gamma, x{:}U \vdash t : T}$$

$$\text{(prod-}\lambda^{\to}\text{)} \quad \frac{\Gamma \vdash T : \star \quad \Gamma, x{:}T \vdash U : \star}{\Gamma \vdash (x{:}T)U : \star}$$

$$\text{(prod-}\lambda P\text{)} \quad \frac{\Gamma \vdash T : \star \quad \Gamma, x{:}T \vdash U : \Box}{\Gamma \vdash (x{:}T)U : \Box}$$

$$\text{(abs)} \quad \frac{\Gamma, x{:}T \vdash u : U \quad \Gamma \vdash (x{:}T)U : s \in \{\star, \Box\}}{\Gamma \vdash \lambda x{:}T.\, u : (x{:}T)U}$$

$$\text{(app)} \quad \frac{\Gamma \vdash t : (x{:}U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$$

$$\text{(conv)} \quad \frac{\Gamma \vdash t : T \quad T \leftrightarrow^*_\beta T' \quad \Gamma \vdash T' : \star}{\Gamma \vdash t : T'}$$

### Les définitions inductives

En logique d'ordre supérieur, le principe de récurrence associé aux entiers naturels ne peut être démontré qu'à partir d'une définition imprédicative de ceux-ci. Autrement dit, si on souhaite rester dans un cadre prédicatif, il est nécessaire de poser le principe de récurrence comme un axiome.

C'est pourquoi, en 1971, P. Martin-Löf [83] a étendu le calcul de N. de Bruijn en y incluant des expressions pour représenter des types inductifs et leurs principes de récurrence associés. Par exemple, le type des entiers naturels est représenté par le symbole $nat : \star$, zéro par $0 : nat$, la fonction successeur par $s : nat \to nat$ et une preuve par récurrence d'un prédicat $P : nat \to o$ par $rec^P : P0 \to (\forall n : nat.Pn \to Ps(n)) \to \forall n : nat.Pn$. Dans la règle de conversion (conv), à la $\beta$-réduction, P. Martin-Löf ajoute la *$\iota$-réduction* qui correspond aux coupures de récurrence :

$$\text{(conv)} \quad \frac{\Gamma \vdash t : T \quad T \leftrightarrow^*_{\beta\iota} T' \quad \Gamma \vdash T' : \star}{\Gamma \vdash t : T'}$$

Dans le cas de $nat$, ces règles qui définissent la $\iota$-réduction sont celles du système

T de K. Gödel [63] :

$$\begin{array}{rcl} rec^P(p_0, p_s, 0) & \rightarrow_\iota & p_0 \\ rec^P(p_0, p_s, s(n)) & \rightarrow_\iota & p_s\ n\ rec^P(p_0, p_s, n) \end{array}$$

où $p_0$ est une preuve de $P0$ et $p_s$ une preuve de $\forall n : nat.Pn \rightarrow Ps(n)$. À partir de ces règles, en prenant $P = \lambda x : nat.nat$, il est possible de définir des fonctions sur les entiers naturels comme par exemple l'addition ou la multiplication :

$$\begin{array}{rcl} x + y & = & rec^P(y, \lambda u.\lambda v.s(v), x) \\ x \times y & = & rec^P(0, \lambda u.\lambda v.v + y, x) \end{array}$$

Pour s'en convaincre, posons $f = \lambda u.\lambda v.s(v)$ et montrons que $2 + 2$ se récrit en $4 : 2+2 = rec(2, f, 2) \rightarrow_\iota f\ 2\ rec(2, f, 1) \rightarrow_\beta s(rec(2, f, 1)) \rightarrow_\iota s(f\ 2\ rec(2, f, 0)) \rightarrow_\beta s(s(rec(2, f, 0))) \rightarrow_\iota s(s(2)) = 4$.

En fait, dans cette théorie, il est possible d'exprimer par un terme toute fonction dont l'existence est prouvable en arithmétique intuitionniste d'ordre supérieur prédicative (et ces fonctions sont également celles qui sont exprimables dans le système T de K. Gödel).

### Le polymorphisme

Le problème de l'élimination des coupures en arithmétique intuitionniste d'ordre supérieur imprédicative a été résolu par J.-Y. Girard [61] en 1971. Pour cela, il a introduit un système de *types polymorphes* $F\omega$ (J. Reynolds [103] a introduit indépendamment un système similaire restreint aux quantifications du second ordre). Un type polymorphe est une fonction qui, à un type, associe une expression de type. Et pour représenter les preuves des propositions imprédicatives, il faut également que les termes eux-mêmes soient polymorphes, c'est-à-dire qu'ils puissent être appliqués à des expressions de types. Formellement, pour les quantifications de second ordre (*i.e.* sur les propositions), cela se traduit par le remplacement dans les règles de la Figure 1.1 de la règle (prod-$\lambda$P) par la règle :

$$\text{(prod-F)} \quad \frac{\Gamma \vdash T : \Box \quad \Gamma, x{:}T \vdash U : \star}{\Gamma \vdash (x{:}T)U : \star}$$

qui permet par exemple de former le type $(P{:}\star)P \rightarrow P$ qui correspond à la proposition $\forall P : o.P \Rightarrow P$ en logique d'ordre supérieur. Pour les quantifications d'ordre supérieur, il faut rajouter la règle :

$$\text{(prod-F}\omega\text{)} \quad \frac{\Gamma \vdash T : \Box \quad \Gamma, x{:}T \vdash U : \Box}{\Gamma \vdash (x{:}T)U : \Box}$$

qui permet la formation de types de prédicats comme par exemple $\star \rightarrow \star$ qui correspond à $o \Rightarrow o$ en logique d'ordre supérieur.

Dans ce système, il est donc possible d'exprimer toute fonction dont l'existence est prouvable en arithmétique intuitionniste d'ordre supérieur imprédicative.

Par ailleurs, du point de vue de la programmation, le polymorphisme permet de formaliser les algorithmes génériques par rapport à des types de données. Par exemple, pour tout type $A$, on peut parler du type $(list\ A)$ des listes d'éléments de type $A$ en déclarant $list : \star \to \star$.

En 1984, T. Coquand et G. Huet [34] ont alors défini et étudié un système, le Calcul des Constructions (CC), qui fait la synthèse entre le système de N. de Bruijn et celui de J.-Y. Girard (il contient toutes les règles de formation du produit précédentes) et dans lequel il est possible d'exprimer l'ensemble de la théorie des types (mais il ne permet pas d'exprimer plus de fonctions que F$\omega$). Ce système a servi de base au programme de spécification et de démonstration formelle Coq [52].

### Les Systèmes de Types Purs (PTS)

À la fin des années 80, H. Barendregt [9] a remarqué que beaucoup des systèmes de types introduits dans les dernières décennies ($\lambda^{\to}$, $\lambda$P, F, F$\omega$, CC, etc.) se distinguent seulement par les règles de formation du produit qui y sont autorisées. C'est cela qui a conduit à la présentation des règles de typage que nous avons adoptées. En considérant la règle de typage générale suivante paramétrée par deux sortes $s_1, s_2 \in \{\star, \Box\}$ :

$$(s_1, s_2) \quad \frac{\Gamma \vdash T : s_1 \quad \Gamma, x{:}T \vdash U : s_2}{\Gamma \vdash (x{:}T)U : s_2}$$

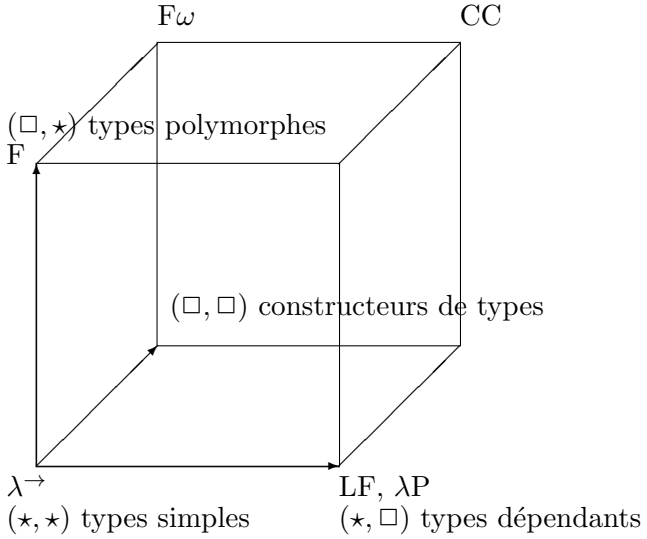
il est possible d'avoir 4 règles différentes ($(\star,\star)$ correspondant à (prod-$\lambda^{\to}$), $(\star,\Box)$ à (prod-$\lambda$P), $(\Box,\star)$ à (prod-F), et $(\Box,\Box)$ à (prod-F$\omega$)) et donc de former, à partir de $(\star,\star)$, 8 systèmes différents que l'on peut organiser en un cube dont chaque direction correspond à la présence ou non de types dépendants (règle $(\star,\Box)$), de types polymorphes (règle $(\Box,\star)$) ou de constructeurs de type (règle $(\Box,\Box)$) (voir Figure 1.2).

- $\lambda^{\to}$ désigne le $\lambda$-calcul simplement typé de A. Church [28],
- LF (Logical Framework) désigne le système de R. Harper, F. Honsell et G. Plotkin [66],
- $\lambda$P désigne le système AUTOMATH de N. de Bruijn [39],
- F et F$\omega$ désignent respectivement le $\lambda$-calcul polymorphe du second ordre et d'ordre supérieur de J.-Y. Girard [62],
- CC désigne le Calcul des Constructions de T. Coquand et G. Huet [35].

Cela va conduire S. Berardi [18] et J. Terlouw [112] à une étude plus systématique des systèmes de types en fonction des types exprimables, jusqu'à la définition par H. Geuvers et M.-J. Nederhof des *Systèmes de Types Purs* (PTS) [57] qui sont des systèmes de types paramétrés par :

- un ensemble de *sortes* $\mathcal{S}$ représentant les différents univers du discours ($\{\star,\Box\}$ dans le Cube),
- un ensemble d'*axiomes* $\mathcal{A} \subseteq \mathcal{S}^2$ représentant comment ces univers sont inclus les uns dans les autres ($\{(\star,\Box)\}$ dans le Cube) et la règle :

FIGURE 1.2 – Cube de Barendregt




$$\text{(ax)} \quad \frac{}{\vdash s_1 : s_2} \quad ((s_1, s_2) \in \mathcal{A})$$

– un ensemble de *règles* de formation du produit $\mathcal{B} \subseteq \mathcal{S}^3$ représentant les quantifications possibles ($\{(\star,\star,\star), (\star,\Box,\Box), (\Box,\star,\star), (\Box,\Box,\Box)\}$ dans le Cube) et la règle :

$$\text{(prod)} \quad \frac{\Gamma \vdash T : s_1 \quad \Gamma, x{:}T \vdash U : s_2}{\Gamma \vdash (x{:}T)U : s_3} \quad ((s_1, s_2, s_3) \in \mathcal{B})$$

### Le Calcul des Constructions Inductives (CIC)

On a vu que le Calcul des Constructions est un système très puissant dans lequel il est possible d'exprimer beaucoup de fonctions. Cependant, ces fonctions ne peuvent pas toujours être définies de la manière que l'on veut. Par exemple, il ne semble pas possible d'y programmer la fonction prédécesseur sur les entiers naturels de façon à ce que son évaluation se fasse toujours en temps constant [62]. Ce n'est pas le cas dans le système de P. Martin-Löf où les entiers naturels et leur principe de récurrence sont des objets à part entière du système alors que, dans le Calcul des Constructions, ils sont définis de manière imprédicative.

C'est pourquoi, en 1988, T. Coquand et C. Paulin ont proposé le Calcul des Constructions Inductives (CIC) [36] qui fait la synthèse entre le Calcul des Constructions et la théorie des types de P. Martin-Löf et permet ainsi de programmer plus efficacement. En 1994, B. Werner [118] a montré la terminaison de l'élimination des coupures dans ce système. (En 1993, T. Altenkirch [2] a également montré cette propriété mais pour une présentation du calcul avec jugements d'égalité.)

Mais, même dans ce système, certains algorithmes ne sont toujours pas exprimables. L. Colson [29] a montré par exemple que, si on utilise une stratégie

d'évaluation d'appel par valeur (réduction des arguments d'abord), la fonction minimum de deux entiers naturels ne peut pas être implémentée par un programme dont le temps d'évaluation est proportionnel au minimum des deux entiers.

## 1.2 Motivations

Nous avons dit au tout début que la récriture est un paradigme de calcul simple et général basé sur la donnée de *règles de récriture*. Cette notion est bien sûr très ancienne mais elle a vraiment commencé à être étudiée dans les années 70 à partir des travaux de D. Knuth et D. Bendix [17]. Ces derniers ont étudiés la récriture pour déterminer si, dans une théorie équationnelle donnée, une équation est valide ou non. Ensuite, la récriture a été rapidement utilisée comme paradigme de programmation [96, 65, 53, 69, 90] puisque toute fonction semi-calculable peut être définie par des règles de récriture.

Voyons l'exemple des opérations arithmétiques sur les entiers naturels définis à partir de 0 pour zéro et $s$ pour la fonction successeur :

$$\begin{array}{rcl} 0+x & \to & x \\ s(x)+y & \to & s(x+y) \\ 0\times x & \to & 0 \\ s(x)\times y & \to & (x\times y)+y \end{array}$$

Ces règles définissent complètement l'addition et la multiplication : partant de deux entiers quelconques $p$ et $q$ (exprimés à partir de 0 et $s$), $p+q$ et $p\times q$ se récrivent en un nombre fini d'étapes en un terme qui ne peut plus se récrire, c'est-à-dire en un entier représentant la valeur de $p+q$ et $p\times q$ respectivement.

**Récriture d'ordre supérieur**

On peut également envisager des définitions par récriture qui utilisent des paramètres de fonctions ou des abstractions : c'est la récriture dite d'*ordre supérieur* par opposition à la récriture du *premier ordre* qui n'utilise pas de paramètres fonctionnels ou d'abstractions. Par exemple, la fonction *map* qui à une fonction $f$ et une liste d'entiers $(a_1,\ldots,a_n)$ associe la liste $(f(a_1),\ldots,f(a_n))$ peut être définie par les règles suivantes :

$$\begin{array}{rcl} map(f,nil) & \to & nil \\ map(f,cons(x,\ell)) & \to & cons(fx,map(f,\ell)) \end{array}$$

où *nil* est la liste vide et *cons* la fonction qui ajoute un élément à la tête d'une liste.

Ainsi, les règles qui définissent les récurseurs des types inductifs (la $\iota$-réduction) sont un cas particulier de récriture d'ordre supérieur.

**Définitions plus faciles**

On peut voir que de telles définitions sont bien plus naturelles et faciles à écrire que celles à base de récurseurs de la théorie des types de P. Martin-Löf ou du Calcul des Constructions Inductives. Par exemple, la définition de la fonction $\leq$ sur les entiers naturels nécessite deux niveaux de récursion :

$$\lambda x.rec(x, \lambda y.true, \lambda nzy.rec(y, false, \lambda n'z'.zn', y))$$

tandis que la définition par récriture est :

$$\begin{array}{rcl} 0 \leq y & \rightarrow & true \\ s(x) \leq 0 & \rightarrow & false \\ s(x) \leq s(y) & \rightarrow & x \leq y \end{array}$$

**Définitions plus efficaces**

D'un point de vue calculatoire, les définitions par récriture peuvent être rendues plus efficaces en ajoutant des règles. Par exemple, avec une définition par récurrence sur son premier argument, $n + 0$ nécessite $n + 1$ étapes de calcul. En ajoutant la règle $n + 0 \rightarrow n$, cela ne prend plus qu'une étape.

Cependant, il peut devenir plus difficile de s'assurer que, pour toute séquence d'arguments, la définition conduit toujours, en un nombre fini d'étapes (propriété de *normalisation forte*), à un résultat unique (propriété de *confluence*), qu'on appelle la *forme normale* de l'expression de départ.

**Types quotients**

Jusqu'à maintenant nous avons toujours parlé des entiers naturels mais jamais des entiers relatifs qui, pourtant, occupent une place tout aussi importante. Une manière de représenter les entiers relatifs est d'ajouter aux côtés de 0 et $s$ une fonction prédécesseur $p$. Ainsi, $p(p(0))$ représente $-2$. Malheureusement, dans ce cas, un nombre peut avoir plusieurs représentations : $p(s(0))$ ou $s(p(0))$ représentent tous les deux 0. En fait, les entiers relatifs sont les classes d'équivalences modulo les équations $p(s(x)) = x$ et $s(p(x)) = x$.

Cependant, il est possible d'orienter ces équations en un système de récriture confluent et fortement normalisant : $p(s(x)) \rightarrow x$ et $s(p(x)) \rightarrow x$, chaque nombre ayant ainsi une forme normale unique. On voit donc que la récriture nous permet de modéliser des types quotients de manière interne sans avoir recours à des extensions supplémentaires [13].

**Plus de termes typables**

L'introduction de récriture dans un système de types dépendants peut permettre de typer davantage de termes, donc de formaliser davantage de propositions. Dans le Calcul des Constructions Inductives, considérons le type $listn : nat \rightarrow \star$ des listes d'entiers naturels de longueur $n$ avec les constructeurs $niln : listn(0)$ pour la liste vide et $cons : nat \rightarrow (n : nat)listn(n) \rightarrow listn(s(n))$ pour ajouter un élément en tête d'une liste. Soit alors la fonction $appn : (n{:}nat)listn(n) \rightarrow (n'{:}nat)listn(n') \rightarrow listn(n + n')$ de concaténation de deux listes. Tout comme $+$ a été défini à l'aide du récurseur associé à $nat$ (par récurrence sur son premier argument), $appn$ peut être défini à l'aide du récurseur associé à $listn$ (par récurrence sur son premier argument). Mais alors les propositions suivantes ne sont pas typables alors qu'elles sont *a priori* tout à fait acceptables :

$$\begin{array}{rcl} appn(n, \ell, 0, \ell') & = & \ell \\ appn(n + n', appn(n, \ell, n', \ell'), n'', \ell'') & = & appn(n, \ell, n' + n'', appn(n', \ell', n'', \ell'')) \end{array}$$

Dans la première règle, le membre gauche est de type $listn(n+0)$ et le membre droit de type $listn(n)$. On peut montrer par récurrence que $n+0$ est égal à $n$ mais $n+0$ n'est pas $\beta\iota$-convertible à $n$ car $+$ a été défini par récurrence sur son premier argument. On ne peut donc pas appliquer la règle de conversion (conv) pour typer l'égalité.

Dans la seconde règle, le membre gauche est de type $listn((n+n')+n'')$ et le membre droit de type $listn(n+(n'+n''))$. De même, bien qu'on puisse montrer que $(n+n')+n''$ soit égal à $n+(n'+n'')$ (associativité de $+$), ces deux termes ne sont pas $\beta\iota$-convertibles. On ne peut donc pas appliquer la règle de conversion pour typer l'égalité.

Cela nous montre les limitations des définitions à l'aide de récurseurs. L'utilisation de la récriture, et en particulier de la récriture associative et commutative, c'est-à-dire en remplaçant dans la règle de conversion (conv) la $\iota$-réduction par une relation de réduction $\to_{\mathcal{R}}$ engendrée par un ensemble $\mathcal{R}$ *a priori* quelconque de règles de récriture définies par l'utilisateur :

$$\text{(conv)} \quad \frac{\Gamma \vdash t : T \quad T \leftrightarrow^*_{\beta\mathcal{R}} T' \quad \Gamma \vdash T' : \star}{\Gamma \vdash t : T'},$$

les égalités précédentes deviennent typables.

**Preuves équationnelles automatiques**

Une autre motivation pour vouloir introduire de la récriture dans les systèmes de types est que cela facilite considérablement les preuves équationnelles, ce pourquoi a été introduite la récriture. En effet, dans le cas d'un système confluent et fortement normalisant, pour vérifier que deux termes sont égaux, il suffit de vérifier qu'ils ont la même forme normale.

De plus, dans la mesure où l'égalité de deux termes est décidable, il n'est pas nécessaire de garder la trace des étapes de récriture qui ont mené à l'égalité puisqu'on peut refaire le calcul à volonté. Cela permet de réduire la taille des preuves et donc de pouvoir traiter des preuves de tailles plus importantes.

**Intégration de procédures de décision**

On peut également imaginer définir des prédicats par récriture ou avoir des règles de simplification sur les propositions, généralisant ainsi les définitions par *élimination forte* du Calcul des Constructions Inductives [99]. Par exemple, on peut envisager l'ensemble de règles de la Figure 1.3 [68] où xor ("ou" exclusif) et $\wedge$ sont considérés comme des symboles commutatifs et associatifs, et où $\bot$ représente la proposition toujours fausse et $\top$ la proposition toujours vraie (en considérant une constante $I$ de type $\top$).

J. Hsiang [68] a montré que ce système est confluent et fortement normalisant, et qu'une proposition $P$ est une tautologie (*i.e.* est toujours vraie) si $P$ se réduit sur $\top$. Cet ensemble de règles constitue donc une procédure de décision pour les tautologies propositionnelles classiques.

Ainsi, la récriture (au niveau type) permet la formalisation et l'intégration de procédures de décision. En effet, du fait de la règle de conversion (conv), si $P$ est une

FIGURE 1.3 – Procédure de décision pour les tautologies propositionnelles classiques

$$
\begin{array}{rcl}
P \operatorname{xor} \bot & \to & P \\
P \operatorname{xor} P & \to & \bot \\
P \wedge \top & \to & P \\
P \wedge \bot & \to & \bot \\
P \wedge P & \to & P \\
P \wedge (Q \operatorname{xor} R) & \to & (P \wedge Q) \operatorname{xor} (P \wedge R)
\end{array}
\qquad
\begin{array}{rcl}
\neg P & \to & P \operatorname{xor} \top \\
P \vee Q & \to & (P \wedge Q) \operatorname{xor} P \operatorname{xor} Q \\
P \Rightarrow Q & \to & (P \wedge Q) \operatorname{xor} P \operatorname{xor} \top \\
P \Leftrightarrow Q & \to & (P \operatorname{xor} Q) \operatorname{xor} \top
\end{array}
$$

tautologie alors $I$, la preuve canonique de $\top$, en est une preuve. Autrement dit, si la relation de typage est décidable, pour savoir si une proposition $P$ est une tautologie, il suffit de proposer $I$ comme preuve au programme de vérification.

Enfin, on peut également envisager des règles de simplification de l'égalité sur les entiers naturels comme celles de la Figure 1.4 où $+$ et $\times$ sont associatifs et commutatifs, et $=$ est commutatif.

FIGURE 1.4 – Règles de simplification de l'égalité sur les entiers naturels

$$
\begin{array}{rcl}
x + 0 & \to & x \\
x + s(y) & \to & s(x + y) \\
x \times 0 & \to & 0 \\
x \times s(y) & \to & (x \times y) + x \\
x \times (y + z) & \to & (x \times y) + (x \times z)
\end{array}
\qquad
\begin{array}{rcl}
x = x & \to & \top \\
s(x) = s(y) & \to & x = y \\
s(x) = 0 & \to & \bot \\
x + y = 0 & \to & x = 0 \wedge y = 0 \\
x \times y = 0 & \to & x = 0 \vee y = 0
\end{array}
$$

## 1.3 Travaux antérieurs

Les premiers travaux sur la combinaison $\lambda$-calcul typé et récriture (du premier ordre) sont dûs à V. Breazu-Tannen en 1988 [23] qui a montré que la combinaison du $\lambda$-calcul simplement typé avec de la récriture du premier ordre est confluente si la récriture est confluente. En 1989, V. Breazu-Tannen et J. Gallier [24], et M. Okada [97] indépendamment, ont montré que la normalisation forte également est préservée. Ces résultats ont été étendus par D. Dougherty [44] à tout ensemble "stable" de $\lambda$-termes non typés.

En 1991, J.-P. Jouannaud et M. Okada [72] ont étendu le résultat de V. Breazu-Tannen et J. Gallier à la récriture d'ordre supérieur sous la condition que les règles d'ordre supérieur vérifient le *Schéma Général*, une généralisation du schéma de récursion primitive. Avec de la récriture d'ordre supérieur, la normalisation forte devient plus difficile à montrer car, dans ce cas, il y a interaction entre récriture et $\beta$-réduction, ce qui n'est pas le cas avec de la récriture du premier ordre.

En 1993, M. Fernández [51] a étendu la méthode de J.-P. Jouannaud et M. Okada au Calcul des Constructions en se restreignant à de la récriture au niveau objet et à des symboles de fonction simplement typés. Les méthodes utilisées pour la récriture du premier ordre et les systèmes de types non dépendants [24, 44] ne

peuvent pas s'appliquer car, dans ce cas-là, la récriture n'est pas seulement une addition syntaxique : en étant rajouté à la $\beta$-réduction dans la règle de conversion (conv), la récriture est une composante à part entière du typage ; en particulier, elle permet de typer davantage de termes.

D'autres méthodes pour montrer la normalisation forte des systèmes de récriture d'ordre supérieur ont également vu le jour. En 1996, J. van de Pol [115] a étendu à l'ordre supérieur l'utilisation d'interprétations strictement décroissantes sur un domaine bien ordonné. Et, en 1999, J.-P. Jouannaud et A. Rubio [74] ont étendu à l'ordre supérieur la méthode RPO (Recursive Path Ordering) [100, 41]. Cette méthode (HORPO) est plus puissante que le Schéma Général puisqu'elle consiste à définir un ordre sur les termes de manière récursive.

Dans tous ces travaux, mêmes ceux concernant le Calcul des Constructions, les symboles de fonction sont toujours simplement typés. C'est T. Coquand [31] en 1992 qui a initié l'étude de la récriture avec des symboles dépendants et polymorphes. Il a étudié la complétude des définitions par récriture en présence de types dépendants. Pour la normalisation forte, il a proposé un schéma plus général que celui de J.-P. Jouannaud et M. Okada, car permettant des définitions par récursion sur des types inductifs strictement positifs [36]), sans que toutefois ce schéma implique la normalisation. En 1996, E. Giménez [60] a défini une restriction de ce schéma pour laquelle il a montré la normalisation forte. En 1999, J.-P. Jouannaud, M. Okada et moi-même [22, 21] avons étendu le Schéma Général afin de traiter les types strictement positifs mais en conservant des symboles simplement typés. Enfin, en 2000, D. Walukiewicz [117] a étendu au Calcul des Constructions le HORPO de J.-P. Jouannaud et A. Rubio avec des symboles dépendants et polymorphes.

Mais il reste toujours un point commun entre tous ces travaux : la récriture est toujours restreinte au niveau objet.

En 1998, G. Dowek, T. Hardin et C. Kirchner [47] ont proposé une nouvelle approche de la déduction pour la logique du premier ordre : la *Déduction Naturelle Modulo* une congruence $\equiv$ sur les propositions. Ce système de déduction consiste à remplacer les règles de la Déduction Naturelle par des règles équivalentes modulo $\equiv$. Par exemple, la règle ($\Rightarrow$-élim) est remplacée par :

$$(\Rightarrow\text{-élim-modulo}) \quad \frac{\Gamma \vdash R \quad \Gamma \vdash P}{\Gamma \vdash Q} \quad \text{si } R \equiv P \Rightarrow Q$$

Ils ont montré que la théorie des types simples et la théorie des ensembles skolémisée peuvent être vues comme des théories du premier ordre modulo une certaine congruence utilisant des *substitutions explicites* [1]. Et G. Dowek et B. Werner [48] ont donné différentes conditions assurant l'élimination des coupures dans un tel système.

## 1.4 Contributions

Notre principale contribution est d'établir un ensemble de conditions très générales pour assurer la normalisation forte du Calcul des Constructions étendus avec de

la récriture au niveau des types [19]. À titre d'exemples, nous montrons que nos conditions sont vérifiées par un important sous-système du Calcul des Constructions Inductives (CIC) et par la Déduction Naturelle Modulo une large classe de théories équationnelles.

D'une certaine manière, notre travail peut être vu comme une extension de la Déduction Naturelle Modulo au Calcul des Constructions, où la congruence considérée non seulement inclut de la récriture du premier ordre mais également de la récriture d'ordre supérieur puisque, dans le Calcul des Constructions, fonctions et prédicats peuvent être appliqués à des fonctions et à des prédicats.

Notre travail peut donc servir de base à une puissante extension des systèmes de spécification et d'aide à la démonstration actuels comme Coq [52] ou LEGO [81] qui ne permettent que des définitions par récursion. En effet, la normalisation forte non seulement assure la cohérence logique du système (si les symboles considérés sont cohérents) mais aussi la décidabilité de la vérification de type, c'est-à-dire, du problème consistant à savoir si un terme est une preuve d'une proposition ou non.

Pour réaliser une telle extension, plutôt que de récrire complètement le noyau de ces systèmes pour prendre en compte la récriture, il peut être intéressant de faire appel à des applications spécialisées comme CiME [30], ELAN [77] ou Maude [90].

Enfin, en terme d'extraction de programmes à partir de preuves de correction [98], on peut envisager d'extraire les programmes vers des langages à base de récriture et ainsi de parvenir à des programmes extraits plus efficaces.

Considérer une forme de récriture au niveau des types n'est pas tout à fait nouveau : c'est l'"élimination forte" dans le Calcul des Constructions Inductives, c'est-à-dire la définition de prédicats par récurrence sur des types inductifs. La nouveauté ici est de ne pas se restreindre à des définitions par récurrence mais de considérer un ensemble *a priori* quelconque de règles de récriture fournit par l'utilisateur. Or, les preuves de normalisation avec élimination forte de B. Werner [118] ou T. Altenkirch [2] utilisent de manière essentielle le fait qu'il s'agisse de définitions par récurrence.

Par ailleurs, aucune des méthodes utilisées dans le cas de récriture du premier ordre [24, 4, 44] ne peut s'appliquer. Cela tient à plusieurs raisons. Premièrement, nous considérons des récritures d'ordre supérieur et celles-ci, contrairement aux récritures du premier ordre, interagissent avec la $\beta$-réduction. Deuxièmement, la récriture est intégrée à la règle de conversion du système de types ce qui fait que davantage de termes peuvent être typés.

Pour établir nos conditions et montrer la normalisation forte, nous avons adapté la méthode des candidats de réductibilité de Tait et Girard [62] également utilisée par F. Barbanera, M. Fernández et H. Geuvers [7, 6, 5] pour la récriture au niveau objet, et par B. Werner et T. Altenkirch pour l'élimination forte. Cependant, tous utilisent comme candidats des ensembles de $\lambda$-termes purs (non typés) et, mis à part T. Altenkirch, ils utilisent tous des langages ou des systèmes de types intermédiaires. En nous appuyant sur un travail de T. Coquand et J. Gallier [33], nous utilisons des candidats faits de termes bien typés et n'utilisons aucun langage intermédiaire. Nous obtenons ainsi une preuve plus simple et plus courte pour un résultat plus général.

Signalons également trois autres contributions.

Pour permettre le filtrage sur des symboles définis et la définition de types quotients à l'aide de règles de récriture sur les constructeurs [70] (ce qui n'est pas possible dans le Calcul des Constructions Inductives), nous considérons une notion de "constructeur" plus générale que celle qui est généralement retenue pour les types inductifs (voir le Sous-chapitre 6.2).

Nous donnons des conditions pour assurer qu'une règle de récriture préserve le typage dans un PTS quelconque (voir le Chapitre 5). Ces conditions sont bien plus générales que celles employées jusqu'à maintenant. En particulier, elles permettent d'éliminer de nombreuses non-linéarités dûes au typage, ce qui est un gain très important en terme d'efficacité d'une part et de facilité à montrer la confluence d'autre part.

Nous établissons un corpus de propriétés (Chapitres 3 et 4) pour les Systèmes de Types Purs (PTS) où, dans la règle de conversion (conv), la $\beta$-conversion est remplacée par une relation *a priori* quelconque. Nous factorisons ainsi de nombreuses démontrations faîtes avec des relations particulières et établissons quelques propriétés nouvelles.

## 1.5 Plan de la thèse

**Chapitre 3 :** Nous étudions les propriétés de base des Systèmes de Types Purs dont la relation de conversion au niveau des types est abstraite. Nous appelons de tels systèmes des Systèmes de Types Modulo (TSM).

**Chapitre 4 :** Nous étudions les propriétés d'une classe particulière de TSM, ceux dont la relation de conversion est engendrée par une relation de réduction et que nous appelons les Systèmes de Types à Réduction (RTS). Un problème essentiel dans ces systèmes est de s'assurer que la relation de réduction préserve le typage.

**Chapitre 5 :** Nous donnons des conditions suffisantes pour assurer la préservation du typage dans les RTS dont la relation de réduction est engendrée par des règles de récriture et que nous appelons les Systèmes de Types Algébriques (ATS).

**Chapitre 6 :** À partir de ce chapitre, nous considérons un Système de Types Algébrique particulier, le Calcul des Constructions Algébriques (CAC). Nous donnons des conditions suffisantes pour assurer la normalisation forte d'un tel calcul.

**Chapitre 7 :** Nous donnons quelques exemples importants de systèmes de types vérifiant nos conditions de normalisation forte. Parmi ces systèmes, on trouve un sous-système avec élimination forte du Calcul des Constructions Inductives qui est à la base du système Coq [52] et la Déduction Naturelle Modulo une large classe de théories équationnelles.

**Chapitre 8 :** Nous établissons la correction de nos conditions de normalisation forte en prenant tout particulièrement soin d'indiquer les conditions utilisées. Un

index permet de retrouver les endroits précis où sont utilisées chacune des conditions.

**Chapitre 9 :** Nous terminons par l'évocation de plusieurs directions de recherche qui pourrqient permettre d'améliorer nos conditions de normalisation forte.

# Chapitre 2

# Préliminaires

Dans ce chapitre, nous définissons la syntaxe des systèmes que nous étudierons ensuite et rappelons quelques notions élémentaires sur le $\lambda$-calcul, les Systèmes de Types Purs (PTS) (voir [10] pour plus de détails) et les relations. Cette syntaxe étend simplement celle des PTS en y ajoutant des symboles ($nat$, 0, $+$, $\geq$, ...) devant être appliqués à autant d'arguments que leur *arité* le spécifie (voir la Remarque 10 pour une discussion à propos de cette notion).

**Définition 1 ($\lambda$-système sorté)** Un *$\lambda$-système sorté* est donné par :

- un ensemble de *sortes* $\mathcal{S}$,
- une famille $\mathcal{F} = (\mathcal{F}^s_n)^{s\in\mathcal{S}}_{n\geq 0}$ d'ensembles de *symboles*,
- une famille $\mathcal{X} = (\mathcal{X}^s)^{s\in\mathcal{S}}$ d'ensembles infinis dénombrables de *variables*,

tels que ces ensembles soient tous disjoints les uns des autres. Un symbole $f \in \mathcal{F}^s_n$ est dit d'*arité* $\alpha_f = n$ et de *sorte* $s$. Par la suite, on notera l'ensemble des symboles de sorte $s$ par $\mathcal{F}^s$ et l'ensemble des symboles d'arité $n$ par $\mathcal{F}_n$.

**Définition 2 (Termes)** L'ensemble $\mathcal{T}$ des *termes* d'un $\lambda$-système sorté est le plus petit ensemble tel que :

- les sortes et les variables sont des termes ;
- si $x$ est une variable et $t, u$ sont des termes alors le *produit dépendant* $(x:t)u$ et l'*abstraction* $[x:t]u$ sont des termes ;
- si $t$ et $u$ sont des termes alors l'*application* $tu$ est un terme ;
- si $f$ est un symbole d'arité $n$ et $t_1, \ldots, t_n$ sont des termes alors $f(t_1, \ldots, t_n)$ est un terme (certains symboles comme $+$, $\times$, ... pourront être notés de manière infixe).

**Variables libres et variables liées**

Une variable $x$ qui se trouve dans le champs d'une abstraction $[x:T]$ ou d'un produit $(x:T)$ est dite *liée*. Comme il est de coutume, elle peut-être remplacée par toute autre variable, mais à condition que celle-ci soit de même sorte. C'est ce qu'on appelle l'*$\alpha$-équivalence*. Une variable qui n'est pas liée est dite *libre*. Nous désignons par $\mathrm{FV}(t)$ l'ensemble des variables libres d'un terme $t$ et par $\mathrm{FV}^s(t)$ l'ensemble des variables libres de sorte $s$. Un terme ne contenant aucune variable libre est dit *clos*.

Nous utiliserons parfois la notation $U \to V$ pour désigner un produit $(x:U)V$ où $x \notin \mathrm{FV}(V)$ (produit non-dépendant).

**Vecteurs**

Par la suite, nous utiliserons souvent des vecteurs $(\vec{t}, \vec{u}, \ldots)$ pour désigner des séquences de termes (ou d'autres choses). La taille d'un vecteur $\vec{t}$ est notée $|\vec{t}|$. Par exemple, $[\vec{x}:\vec{T}]u$ désigne le terme $[x_1:T_1]\ldots[x_n:T_n]u$ où $n = |\vec{x}|$.

**Définition 3 (Positions dans un terme)** Pour désigner les sous-termes d'un terme, on utilise un système de *positions*. Formellement, l'ensemble $\mathrm{Pos}(t)$ des positions d'un terme $t$ est le plus petit ensemble de mots sur l'alphabet des entiers positifs tel que :

- $\mathrm{Pos}(s) = \mathrm{Pos}(x) = \{\varepsilon\}$,
- $\mathrm{Pos}((x:t)u) = \mathrm{Pos}([x:t]u) = \mathrm{Pos}(tu) = 1 \cdot \mathrm{Pos}(t) \cup 2 \cdot \mathrm{Pos}(u)$,
- $\mathrm{Pos}(f(\vec{t})) = \{\varepsilon\} \cup \bigcup \{i \cdot \mathrm{Pos}(t_i) \mid 1 \leq i \leq \alpha_f\}$,

où $\varepsilon$ désigne le mot vide et $\cdot$ la concaténation. Alors, on désigne par $t|_p$ le sous-terme de $t$ à la position $p$ et par $t[u]_p$ le terme obtenu en remplaçant $t|_p$ par $u$ dans $t$. La relation "est sous-terme de" est notée $\lhd$.

Étant donnés un terme $t$ et un symbole $f$, on notera par $\mathrm{Pos}(f,t)$ l'ensemble des positions $p$ où $t|_p$ est de la forme $f(\vec{t})$. De même, si $x$ est une variable, on notera par $\mathrm{Pos}(x,t)$ l'ensemble des positions $p$ telles que $t|_p$ est une occurrence libre de $x$.

**Définition 4 (Substitution)** Une *substitution* $\theta$ est une application de $\mathcal{X}$ dans $\mathcal{T}$ de *domaine* $\mathrm{dom}(\theta) = \{x \in \mathcal{X} \mid x\theta \neq x\}$ fini. Appliquer une substitution $\theta$ à un terme $t$ consiste à remplacer les variables libres de $t$ par leur image (en prenant soin que les variables liées soient distinctes des variables libres). Le résultat est noté $t\theta$. On désigne par $\mathrm{dom}^s(\theta)$ les variables du domaine qui sont de sorte $s$, par $\{\vec{x} \mapsto \vec{t}\}$ la substitution qui à $x_i$ associe $t_i$, et par $\theta \cup \{x \mapsto t\}$ la substitution qui à $x$ associe $t$ et à $y \neq x$ associe $y\theta$.

**Relations**

Nous rappelons maintenant quelques définitions élémentaires à propos des relations. Étant donné une relation binaire $\to$ sur les termes :

- $\leftarrow$ est l'inverse de $\to$,
- $\to^+$ est la plus petite relation transitive contenant $\to$,
- $\to^*$ est la plus petite relation réflexive et transitive contenant $\to$,
- $\leftrightarrow^*$ est la plus petite relation réflexive, transitive et symétrique contenant $\to$,
- $\downarrow$ est la relation binaire $\to^* {}^*\!\leftarrow$ ($t \downarrow u$ s'il existe $v$ tel que $t \to^* v$ et $u \to^* v$).

Si $t \to t'$, on dit que $t$ se *récrit* en $t'$. Si $t \to^* t'$, on dit que $t$ se *réduit* en $t'$.

La relation $\to$ est *stable par contexte* si $u \to u'$ implique $t[u]_p \to t[u']_p$ pour tout terme $t$ et position $p \in \mathrm{Pos}(t)$.

La relation $\to$ est *stable par substitution* si $t \to t'$ implique $t\theta \to t'\theta$ pour toute substitution $\theta$.

La *$\beta$-réduction* (resp. *$\eta$-réduction*) est la plus petite relation stable par contexte et substitution contenant $[x:U]t\ u \to_\beta t\{x \mapsto u\}$ (resp. $[x:U]tx \to_\eta t$ si $x \notin \mathrm{FV}(t)$).

Un terme de la forme $[x:U]t\ u$ (resp. $[x:U]tx$ avec $x \notin \mathrm{FV}(t)$) est un *$\beta$-radical* (resp. *$\eta$-radical*).

**Normalisation**

La relation $\to$ est *faiblement normalisante* si, pour tout terme $t$, il existe un terme irréductible $t'$ en lequel $t$ peut se réduire. On dit alors que $t'$ est une *forme normale* de $t$. La relation $\to$ est *fortement normalisante* (bien fondée, nœthérienne) si, quelque soit $t$, toutes les réductions issues de $t$ sont finies.

**Confluence**

La relation $\to$ est *localement confluente* si, dès qu'un terme $t$ se récrit en deux termes distincts $u$ et $v$, alors $u \downarrow v$. Enfin, $\to$ est *confluente* si, dès qu'un terme $t$ se réduit en deux termes distincts $u$ et $v$, alors $u \downarrow v$.

Il est assez facile de voir que si $\to$ est localement confluente et fortement normalisante alors $\to$ est confluente [94], et que si $\to$ est confluente et faiblement normalisante alors chaque terme $t$ admet une forme normale unique notée $t\downarrow$. (Le symbol $\downarrow$ désigne ici une relation unaire postfixée, ce qui permet de le différencier de son autre utilisation.)

**Ordres lexicographiques et multi-ensembles**

Étant donné des ordres $>_1, \ldots, >_n$ sur des ensembles $E_1, \ldots, E_n$, nous désignons par $(>_1, \ldots, >_n)_{\mathrm{lex}}$ l'ordre *lexicographique* construit sur $E_1 \times \ldots \times E_n$ à partir de $>_1, \ldots, >_n$. Rappelons que, pour $n = 2$, nous avons $(x, y)(>_1, >_2)_{\mathrm{lex}}(x', y')$ si $x >_1 x'$ ou, $x =_1 x'$ et $y >_2 y'$.

Soit $E$ un ensemble. Un *multi-ensemble* $M$ sur $E$ est une application de $E$ dans $\mathbb{N}$, $M(x)$ désignant le nombre de fois où $x$ "apparaît" dans $M$. Nous noterons par $\mathcal{M}(E)$ l'ensemble des multi-ensembles finis sur $E$ (*i.e.* à domaine fini).

Soit $>$ un ordre sur $E$. L'*extension multi-ensemble* de $>$ est l'ordre $>_{\mathrm{mul}}$ sur $\mathcal{M}(E)$ égal à la fermeture transitive de la relation $\succ$ définie par : $M \cup \{x\} \succ N \cup \{y_1, \ldots, y_n\}$ $(n \geq 0)$ si, pour tout $i$, $x > y_i$ ($x$ est remplacé par un nombre fini, éventuellement nul, d'éléments plus petits).

Une propriété importante de ces extensions est qu'elles préservent la bonne fondation. Pour plus de détails sur ces notions, le lecteur est invité à consulter [43, 3].

# Chapitre 3

# Systèmes de Types Modulo (TSM)

Dans ce chapitre, nous considérons une extension des PTS avec des symboles de fonctions et une règle de conversion (conv) où $\leftrightarrow^*_\beta$ est remplacée par une relation de conversion $\mathcal{C}$ *a priori* quelconque.

Il y a déjà eu différentes extensions des PTS, en particulier :

- En 1989, Z. Luo [80] a étudié une extension du Calcul des Constructions avec une hiérarchie cumulative de sortes ($\star \prec \Box = \Box_0 \prec \Box_1 \prec \ldots$), le Calcul des Constructions Étendu (ECC) : $\mathcal{C}$ est le plus petit pré-ordre contenant $\leftrightarrow^*_\beta$, $\prec$ et compatible avec le produit ($U' \ \mathcal{C} \ U$ et $V \ \mathcal{C} \ V'$ implique $(x{:}U)V \ \mathcal{C} \ (x{:}U')V'$).
- En 1993, H. Geuvers [56] a étudié les PTS avec $\eta$-réduction : $\mathcal{C} = \leftrightarrow^*_{\beta\eta}$.
- En 1993, M. Fernández [51] a étudié une extension du Calcul des Constructions avec récriture d'ordre supérieur à la Jouannaud-Okada [72], le $\lambda R$-cube : $\mathcal{C} = \rightarrow^*_{\beta\mathcal{R}} \cup \ {}^*_{\beta R}\leftarrow$.
- En 1994, E. Poll et P. Severi [101] ont étudié les PTS avec abréviations (`let x= ... in ...`) : $\mathcal{C} = \leftrightarrow^*_\beta \cup \leftrightarrow^*_\delta$ où $\rightarrow_\delta$ est le remplacement d'une abréviation par sa définition, celle-ci pouvant être donnée dans un environnement quelconque.
- En 1994, B. Werner [118] a étudié une extension du Calcul des Constructions avec types inductifs, le Calcul des Constructions Inductives (CIC) : $\mathcal{C} = \leftrightarrow^*_{\beta\eta\iota}$ où $\rightarrow_\iota$ est la réduction associée aux schémas d'élimination des types inductifs.
- Entre 1995 et 1998, G. Barthe et ses coauteurs [14, 15, 16, 12] considèrent différentes extensions du Calcul des Constructions ou des PTS avec des relations de conversion plus ou moins abstraites souvent basées sur la récriture à la Jouannaud-Okada [72], étendant ainsi les travaux de M. Fernández [51].

Dans ces travaux, des propriétés de base bien connues des PTS sont redémontrées du fait de l'introduction de nouvelles constructions ou d'une nouvelle relation de conversion $\mathcal{C}$. C'est pourquoi il nous a paru utile d'étudier les propriétés des PTS dotés d'une relation de conversion $\mathcal{C}$ abstraite.

Le besoin n'est pas nouveau puisque cela a déjà été partiellement entreprit dans

des développements formels :

- En 1994, R. Pollack [102] prouve formellement dans LEGO [81], une implémentation de ECC avec types inductifs, que la vérification de type dans ECC (sans type inductif) est décidable (en supposant bien sûr que le calcul est fortement normalisant). La vérification de type consiste à dire si, dans un environnement $\Gamma$ donné, un terme $t$ est de type $T$ (*i.e.* une preuve de $T$). Ce faisant, il montre de nombreuses propriétés des PTS en considérant une relation de conversion $\mathcal{C} = \leq$ réflexive, transitive et stable par substitution et contexte.
- En 1999, B. Barras [11] prouve formellement dans Coq [52], une autre implémentation de ECC avec types inductifs, que la vérification de type dans Coq (avec types inductifs cette fois-ci) est décidable (là encore en supposant que le calcul est fortement normalisant). Pour cela, il montre également de nombreuses propriétés des PTS en considérant une relation de conversion $\mathcal{C} = \leq$ réflexive, transitive et stable par substitution et contexte. En fait, il considère une extension des PTS avec un schéma de règles de typage pour pouvoir introduire de nouvelles constructions de manière générique (abréviations, types inductifs, schémas d'élimination).

Ainsi, d'un côté, nous faisons moins d'hypothèses sur la relation de conversion $\mathcal{C}$ que n'en font R. Pollack ou B. Barras. Cela est justifié par le fait que, dans le travail de M. Fernández [51] par exemple, pour pouvoir établir que la réduction préserve le typage, il est indispensable que la relation de conversion ne soit pas transitive. D'un autre côté, le typage de nos symboles de fonction n'est pas aussi général que celui de B. Barras.

## 3.1 Définition

**Définition 5 (Environnement)** Un *environnement* $\Gamma$ est une liste de couples $x_i : T_i$ constitués d'une variable $x_i$ et d'un terme $T_i$. On notera par $\emptyset$ l'environnement vide, par $\mathcal{E}$ l'ensemble des environnements et par $x_i\Gamma$ le terme $T_i$ associé à la variable $x_i$ dans $\Gamma$. L'ensemble des *variables libres* d'un environnement $\Gamma$ est $\mathrm{FV}(\Gamma) = \bigcup\{\mathrm{FV}(x\Gamma) \mid x \in \mathrm{dom}(\Gamma)\}$. Le *domaine* d'un environnement $\Gamma$ est l'ensemble des variables auxquelles $\Gamma$ associe un terme. Étant donnés deux environnements $\Gamma$ et $\Gamma'$, $\Gamma$ est *inclus* dans $\Gamma'$, noté $\Gamma \subseteq \Gamma'$, si tous les éléments de $\Gamma$ apparaissent dans $\Gamma'$ dans le même ordre.

**Définition 6 (Assignement de type)** Un *assignement de type* est une application $\tau$ de $\mathcal{F}$ dans $\mathcal{T}$ qui, à un symbole $f$ d'arité $n$, associe un terme clos $\tau_f$ de la forme $(\vec{x} : \vec{T})U$ où $|\vec{x}| = n$. Nous noterons par $\Gamma_f$ l'environnement $\vec{x} : \vec{T}$.

**Définition 7 (TSM)** Un *Système de Types Modulo* (TSM) est un $\lambda$-système sorté $(\mathcal{S}, \mathcal{F}, \mathcal{X})$ avec :

- un ensemble d'*axiomes* $\mathcal{A} \subseteq \mathcal{S}^2$,
- un ensembles de *règles* de formation du produit $\mathcal{B} \subseteq \mathcal{S}^3$,
- un assignement de type $\tau$,
- une *relation de conversion* $\mathcal{C} \subseteq \mathcal{T}^2$.

Un $\beta$TSM (resp. $\eta$TSM) est un TSM tel que $\downarrow_\beta \subseteq \mathcal{C}$ (resp. $\downarrow_\eta \subseteq \mathcal{C}$).

**Définition 8 (Typage)** La relation de typage d'un TSM est la plus petite relation ternaire $\vdash \subseteq \mathcal{E} \times \mathcal{T} \times \mathcal{T}$ définie par les règles d'inférence de la Figure 3.1. Par rapport aux PTS, nous avons une nouvelle règle, (symb), pour typer les symboles et, dans la règle de conversion (conv), nous considérons la relation abstraite $\mathcal{C}$ au lieu de la $\beta$-conversion. Un terme $t$ est *typable* s'il existe un environnement $\Gamma$ et un terme $T$ tels que $\Gamma \vdash t : T$ ($T$ est le *type* de $t$ dans $\Gamma$). Un environnement $\Gamma$ est *valide* s'il existe un terme typable dans $\Gamma$. Dans la règle (symb), la prémice "$\Gamma$ valide" n'est donc utile que si $f$ est d'arité nulle ($n = 0$).

- $\mathbb{T} = \{t \in \mathcal{T} \mid \exists \Gamma \in \mathcal{E}, \exists T \in \mathcal{T},\ \Gamma \vdash t : T\}$ est l'ensemble des termes typables,
- $\mathbb{T}_0^s = \{T \in \mathcal{T} \mid \exists \Gamma \in \mathcal{E},\ \Gamma \vdash T : s\}$ est l'ensemble des *prédicats* de sorte $s$,
- $\mathbb{T}_1^s = \{t \in \mathcal{T} \mid \exists \Gamma \in \mathcal{E}, \exists T \in \mathcal{T},\ \Gamma \vdash t : T \text{ et } \Gamma \vdash T : s\}$ est l'ensemble des *objets* de sorte $s$,
- $\mathbb{E} = \{\Gamma \in \mathcal{E} \mid \exists t, T \in \mathcal{T},\ \Gamma \vdash t : T\}$ est l'ensemble des environnements valides.

FIGURE 3.1 – Règles de typage des TSM

$$\text{(ax)} \quad \frac{}{\vdash s_1 : s_2} \qquad ((s_1, s_2) \in \mathcal{A})$$

$$\text{(symb)} \quad \frac{\begin{array}{c}\vdash \tau_f : s \quad \Gamma \text{ valide} \\ \Gamma \vdash t_1 : T_1\gamma \quad \ldots \quad \Gamma \vdash t_n : T_n\gamma\end{array}}{\Gamma \vdash f(\vec{t}) : U\gamma} \qquad \begin{array}{c}(f \in \mathcal{F}_n^s, \\ \tau_f = (\vec{x} : \vec{T})U, \\ \gamma = \{\vec{x} \mapsto \vec{t}\})\end{array}$$

$$\text{(var)} \quad \frac{\Gamma \vdash T : s}{\Gamma, x{:}T \vdash x : T} \qquad (x \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma))$$

$$\text{(weak)} \quad \frac{\Gamma \vdash t : T \quad \Gamma \vdash U : s}{\Gamma, x{:}U \vdash t : T} \qquad (x \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma))$$

$$\text{(prod)} \quad \frac{\Gamma \vdash T : s_1 \quad \Gamma, x{:}T \vdash U : s_2}{\Gamma \vdash (x{:}T)U : s_3} \qquad ((s_1, s_2, s_3) \in \mathcal{B})$$

$$\text{(abs)} \quad \frac{\Gamma, x{:}T \vdash u : U \quad \Gamma \vdash (x{:}T)U : s}{\Gamma \vdash [x{:}T]u : (x{:}T)U}$$

$$\text{(app)} \quad \frac{\Gamma \vdash t : (x{:}U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$$

$$\text{(conv)} \quad \frac{\Gamma \vdash t : T \quad \Gamma \vdash T' : s'}{\Gamma \vdash t : T'} \qquad (T\ \mathcal{C}\ T')$$

**Remarque 9 (Conversion)**

Il pourrait paraître plus naturel de définir (conv) de manière symétrique en rajoutant dans les prémices, $\Gamma \vdash T : s$ ou $\Gamma \vdash T : s'$. Nous avons choisi de considérer cette règle-là pour deux raisons. Premièrement, c'est ainsi qu'elle est définie dans les textes de référence sur les PTS [57, 10]. Deuxièmement, d'un point de vue pratique, cela évite un test supplémentaire. Toutefois, nous verrons dans le Lemme 37 que, pour beaucoup de TSM, nous avons $\Gamma \vdash T : s'$. Nous désignerons par $\vdash_s$ la relation de typage définie par les mêmes règles d'inférence que $\vdash$ excepté (conv) que nous remplaçons par :

$$\text{(conv')} \quad \frac{\Gamma \vdash t : T \quad \Gamma \vdash T : s \quad \Gamma \vdash T' : s}{\Gamma \vdash t : T'} \quad (T \ \mathcal{C} \ T')$$

Nous montrons l'équivalence entre $\vdash_s$ et $\vdash$ dans le Lemme 43.

Dans le même ordre d'idée, dans les TSM dits *complets* $(\forall s_1, s_2 \in \mathcal{S}, \exists s_3 \in \mathcal{S}, (s_1, s_2, s_3) \in \mathcal{B})$, la règle (abs) peut être remplacée par :

$$\text{(abs')} \quad \frac{\Gamma, x{:}T \vdash u : U}{\Gamma \vdash [x{:}T]u : (x{:}T)U} \quad (U \notin \mathcal{S} \text{ ou } \exists s{\in}\mathcal{S}, (U, s) \in \mathcal{A})$$

Enfin, nous désignerons par $\vdash_w$ la relation de typage définie par les mêmes règles d'inférence que $\vdash$ excepté (weak) que nous remplaçons par :

$$\text{(weak')} \quad \frac{\Gamma \vdash t : T \quad \Gamma \vdash U : s}{\Gamma, x{:}U \vdash t : T} \quad (x \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma), t \in \mathcal{X} \cup \mathcal{S})$$

c'est-à-dire, où l'affaiblissement est restreint aux variables et aux sortes $(t \in \mathcal{X} \cup \mathcal{S})$. Nous montrerons l'équivalence entre $\vdash_w$ et $\vdash$ dans le Lemme 19.

**Remarque 10 (Arité)**

Enfin, on peut se demander pourquoi les symboles sont dotés d'une arité fixe alors qu'en général, dans le $\lambda$-calcul, il est d'usage de considérer des constantes d'ordre supérieur. Bien sûr, cela ne constitue pas une restriction puisqu'à un symbole $f$ d'arité $n$ et de type $(\vec{x} : \vec{T})U$, on peut toujours associer un "double" curryfié $f^c$ de même type mais d'arité nulle défini par la règle $f^c \to [\vec{x} : \vec{T}]f(\vec{x})$. De plus, en pratique, on peut toujours dissimuler à l'utilisateur l'existence de cette arité en $\eta$-expansant si $f$ n'est pas suffisamment appliqué, et en complétant par des applications si $f$ est trop appliqué. Cependant, on aurait une présentation plus simple où la règle (symb) serait réduite à :

$$\text{(symb')} \quad \frac{\vdash \tau_f : s}{\vdash f : \tau_f} \quad (f \in \mathcal{F}^s)$$

À notre connaissance, mis à part les travaux de Jouannaud et Okada [73] et de G. Barthe et ses coauteurs [14, 15, 16, 12], la plupart des autres travaux sur la combinaison $\lambda$-calcul typé et récriture [23, 24, 51] ne font pas usage d'une arité pour le typage des symboles. Nous avons choisi d'utiliser une telle notion pour des raisons purement techniques. Avec la méthode que nous employons pour montrer la

normalisation forte, nous avons besoin qu'une application $uv$ ne soit pas un radical de récriture (voir le Chapitre 5 pour l'explication de ces notions et le Lemme 121, cas $T = (x : U)V$, (b), (R3) pour l'utilisation de cette propriété). L'introduction d'une arité nous permet de distinguer syntaxiquement entre l'application du $\lambda$-calcul et l'application d'un symbole. On peut se demander si cette notion est vraiment indispensable.

**Définition 11 (Substitution bien typée)** Étant donnés deux environnements valides $\Gamma$ et $\Delta$, une substitution $\theta$ est *bien typée entre* $\Gamma$ *et* $\Delta$ , $\theta : \Gamma \to \Delta$, si, pour tout $x \in \mathrm{dom}(\Gamma)$, $\Delta \vdash x\theta : x\Gamma\theta$.

Par exemple, dans la règle (symb), nous avons $\gamma : \Gamma_f \to \Gamma$ où, rappelons-le, $\Gamma_f = \vec{x} : \vec{T}$.

## 3.2 Propriétés

Dans ce sous-chapitre et le suivant, on passe en revue un certain nombre de propriétés bien connues des PTS (sauf le Lemme 22) qui sont vraies pour tous les TSM. Mis à part le nouveau cas (symb) que nous détaillerons à chaque fois, les preuves sont identiques aux preuves habituelles dans le cas des PTS. Le fait que dans (conv), $\leftrightarrow^*_\beta$ soit remplacé par une relation quelconque $\mathcal{C}$ n'a aucune importance. Pour plus de détails, nous renvoyons le lecteur à [57, 10, 56].

**Lemme 12 (Variables libres)** Soit un environnement $\Gamma = \vec{x} : \vec{T}$. Si $\Gamma \vdash t : T$ alors :
(a) les $x_i$ sont des variables deux à deux distinctes,
(b) pour tout $i$, $\mathrm{FV}(T_i) \subseteq \{x_1, \ldots, x_{i-1}\}$,
(c) $\mathrm{FV}(t) \cup \mathrm{FV}(T) \subseteq \mathrm{dom}(\Gamma)$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. Nous ne détaillons que le nouveau cas (symb). (a) et (b) sont satisfaits par hypothèse de récurrence. Voyons (c) maintenant. Par hypothèse de récurrence, $\mathrm{FV}(\tau_f) = \emptyset$ et, pour tout $i$, $\mathrm{FV}(t_i) \subseteq \mathrm{dom}(\Gamma)$. Donc $\mathrm{FV}(f(\vec{t})) \subseteq \mathrm{dom}(\Gamma)$. De même, $\mathrm{FV}(U\gamma) \subseteq \mathrm{dom}(\Gamma)$ car $\mathrm{FV}(U) \subseteq \{\vec{x}\}$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$.

■

**Lemme 13 (Sous-termes)** Si un terme est typable alors tous ses sous-termes sont typables.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. Dans le cas (symb), par hypothèse de récurrence, pour tout $i$, tous les sous-termes de $t_i$ sont typables. Donc, tous les sous-termes de $f(\vec{t})$ sont typables. ■

**Lemme 14 (Environnement)** Soit $\Gamma = \vec{x} : \vec{T}$ un environnement valide.
(a) Si $x_i$ est de sorte $s$ alors $x_1{:}T_1, \ldots, x_{i-1}{:}T_{i-1} \vdash T_i : s$.
(b) Pour tout $i$, $x_1{:}T_1, \ldots, x_i{:}T_i \vdash x_i{:}T_i$.

**Preuve.** Par (var), (b) est une conséquence immédiate de (a). On montre (a) par récurrence sur $\Gamma \vdash t : T$. Dans le cas (symb), comme $\Gamma$ est valide, il existe $v$ et $V$ tel que $\Gamma \vdash v : V$. Donc, par hypothèse de récurrence, (a) est vérifié. ■

Le lemme suivant correspond à une forme d'$\alpha$-équivalence sur les variables d'un environnement.

**Lemme 15 (Remplacement)** Si $\Gamma, y{:}W, \Gamma' \vdash t : T$, $y \in \mathcal{X}^s$ et $z \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma, y{:}W, \Gamma')$ alors $\Gamma, z{:}W, \Gamma'\{y \mapsto z\} \vdash t\{y \mapsto z\} : T\{y \mapsto z\}$.

**Preuve.** Par récurrence sur $\Gamma, y{:}W, \Gamma' \vdash t : T$. Soit $\theta = \{y \mapsto z\}$, $\Delta = \Gamma, y{:}W, \Gamma'$ et $\Delta' = \Gamma, z{:}W, \Gamma'\theta$. Dans le cas (symb), par hypothèse de récurrence, nous avons $\Delta'$ valide et, pour tout $i$, $\Delta' \vdash t_i\theta : T_i\gamma\theta$. Donc, par (symb), $\Delta' \vdash f(\vec{t\theta}) : U\gamma\theta$. ■

**Lemme 16 (Affaiblissement)** Si $\Gamma \vdash t : T$ et $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $\Gamma' \vdash t : T$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. Dans le cas (symb), par hypothèse de récurrence, nous avons $\Gamma'$ valide et, pour tout $i$, $\Gamma' \vdash t_i : T_i\gamma$. Donc, par (symb), $\Gamma' \vdash f(\vec{t}) : U\gamma$. ■

**Lemme 17 (Transitivité)** Soient $\Gamma$ et $\Delta$ deux environnements valides. Si $\Gamma \vdash t : T$ et, pour tout $x \in \mathrm{dom}(\Gamma)$, $\Delta \vdash x : x\Gamma$, ce qu'on notera par $\Delta \vdash \Gamma$, alors $\Delta \vdash t : T$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. Dans le cas (symb), par hypothèse de récurrence, nous avons $\Delta$ valide et, pour tout $i$, $\Delta \vdash t_i : T_i\gamma$. Donc, par (symb), $\Delta \vdash f(\vec{t}) : U\gamma$. ■

**Lemme 18 (Permutation faible)** Si $\Gamma, y : A, z : B, \Gamma' \vdash t : T$ et $\Gamma \vdash B : s$ alors $\Gamma, z{:}B, y{:}A, \Gamma' \vdash t : T$.

**Preuve.** Soit $\Delta = \Gamma, y{:}A, z{:}B, \Gamma'$ et $\Delta' = \Gamma, z{:}B, y{:}A, \Gamma'$. D'après le lemme de transitivité, il suffit de montrer que $\Delta'$ est valide et que $\Delta' \vdash \Delta$. Et pour cela, il suffit de montrer que $\Delta'$ est valide. Par le lemme d'environnement, nous avons $\Gamma \vdash A : s'$ et, par hypothèse, nous avons $\Gamma \vdash B : s$. Donc, par affaiblissement, $\Gamma, z{:}B, y{:}A$ est valide. Supposons que $\Gamma' = \vec{x} : \vec{T}$ et posons $\Delta_i = \Gamma, y : A, z : B, x_1 : T_1, \ldots, x_i : T_i$ et $\Delta'_i = \Gamma, z{:}B, y{:}A, x_1{:}T_1, \ldots, x_i{:}T_i$. Montrons par récurrence sur $i$ que $\Delta'_i$ est valide. Nous avons montré que $\Delta'_0$ est valide. Supposons que $\Delta'_i$ soit valide. Par le lemme d'environnement, $\Delta_i \vdash T_{i+1} : s_{i+1}$. Comme $\Delta'_i \vdash \Delta_i$, $\Delta'_i \vdash T_{i+1} : s_{i+1}$ et $\Delta'_{i+1}$ est valide. Donc, $\Delta'$ est valide et $\Delta' \vdash t : T$. ■

**Lemme 19 (Équivalence de $\vdash_w$ et $\vdash$)** $\vdash_w = \vdash$.

**Preuve.** Tout d'abord, il est clair que $\vdash_w \subseteq \vdash$. Il suffit donc de montrer la réciproque, par récurrence sur $\Gamma \vdash t : T$. Le seul cas un peu ennuyeux est bien sûr celui de (weak). De $\Gamma \vdash t : T$ et $\Gamma \vdash U : s$, nous déduisons $\Gamma, x : U \vdash t : T$. Par hypothèse de récurrence, nous avons $\Gamma \vdash_w t : T$ et $\Gamma \vdash_w U : s$. Il faut alors modifier la preuve de $\Gamma \vdash_w t : T$ en rajoutant $x{:}U$ aux endroits appropriés de façon à obtenir une preuve de $\Gamma, x{:}U \vdash_w t : T$. Voir Lemme 4.4.21 page 102 dans [56]. ■

Maintenant, voyons ce que l'on peut dire des dérivations de $\Gamma \vdash t : T$ et de la forme de $T$ en fonction de $t$. Pour cela, nous allons introduire des relations liées à la règle (conv).

**Définition 20 (Relations de conversion)**
- $T\ \mathcal{C}_\Gamma\ T'$ ssi $T\ \mathcal{C}\ T'$ et il existe $s'$ tel que $\Gamma \vdash T' : s'$,
- $T\ \mathbb{C}_\Gamma\ T'$ ssi $T\ \mathcal{C}\ T'$ et il existe $s$ et $s'$ tels que $\Gamma \vdash T : s$ et $\Gamma \vdash T' : s'$,
- $\Gamma\ \mathbb{C}\ \Gamma'$ ssi $\Gamma = \vec{x} : \vec{T}$, $\Gamma' = \vec{x} : \vec{T'}$ et, soit $|\vec{x}| = 0$, soit il existe $j$ tel que $T_j\ \mathbb{C}_{x_1:T_1,\ldots,x_{j-1}:T_{j-1}}\ T'_j$ et, pour tout $i \neq j$, $T_i = T'_i$.

Nous avons $\mathbb{C}_\Gamma \subseteq \mathcal{C}_\Gamma$ mais, contrairement à $\mathbb{C}_\Gamma$, $\mathcal{C}_\Gamma$ n'est pas défini de manière symétrique. Cela est dû à l'asymétrie de la règle (conv) qui demande $\Gamma \vdash T' : s'$ mais pas $\Gamma \vdash T : s$. Toutefois, nous verrons dans le Lemme 37 que, pour beaucoup de TSM, ces deux relations sont égales.

**Lemme 21 (Inversion)** Supposons que $\Gamma \vdash t : T$.
- Si $t = s$ alors il existe $s'$ tel que $(s, s') \in \mathcal{A}$ et $s'\ \mathcal{C}^*_\Gamma\ T$.
- Si $t = f(\vec{t})$, $f \in \mathcal{F}^s$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$ alors $\vdash \tau_f : s$, $\gamma : (\vec{x} : \vec{T}) \to \Gamma$ et $U\gamma\ \mathcal{C}^*_\Gamma\ T$.
- Si $t = x \in \mathcal{X}^s$ alors $\Gamma \vdash x\Gamma : s$ et $x\Gamma\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = (x{:}U)V$ alors il existe $(s_1, s_2, s_3) \in \mathcal{B}$ tel que $\Gamma \vdash U : s_1$, $\Gamma, x{:}U \vdash V : s_2$ et $s_3\ \mathcal{C}^*_\Gamma\ T$.
- Si $t = [x{:}U]v$ alors il existe $V$ tel que $\Gamma, x{:}U \vdash v : V$ et $(x{:}U)V\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = uv$ alors il existe $V$ et $W$ tels que $\Gamma \vdash u : (x : V)W$, $\Gamma \vdash v : V$ et $W\{x \mapsto v\}\ \mathcal{C}^*_\Gamma\ T$.

**Preuve.** Une dérivation de typage se termine toujours par une règle distincte de (weak) et (conv) suivie d'une séquence éventuellement nulle d'applications de (weak) ou (conv). On obtient ainsi le terme à quoi $T$ doit être convertible. Pour les jugements de typage, il suffit de faire un affaiblissement pour les exprimer dans $\Gamma$.■

**Lemme 22 (Conversion pour les environnements)** Si $\Gamma \vdash t : T$ et $\Gamma\ \mathbb{C}\ \Gamma'$ alors $\Gamma' \vdash t : T$.

**Preuve.** Nous avons $\Gamma = \vec{x} : \vec{T}$, $\Gamma' = \vec{x} : \vec{T'}$, $j$ tel que $T_j\ \mathbb{C}_{x_1:T_1,\ldots,x_{j-1}:T_{j-1}}\ T'_j$ et, pour tout $i \neq j$, $T_i = T'_i$. Par transitivité, il suffit de prouver que, pour tout $i$, $\Gamma' \vdash x_i : T_i$. Soit $n = |\vec{x}|$, $\Gamma_1 = x_1{:}T_1, \ldots, x_{j-1}{:}T_{j-1}$ et $\Gamma_2 = x_{j+1}{:}T_{j+1}, \ldots, x_n{:}T_n$. Nous procédons par récurrence sur la taille de $\Gamma_2$.

Si $\Gamma_2$ est vide, $\Gamma = \Gamma_1, x_j : T_j$ et $\Gamma' = \Gamma_1, x_j : T'_j$. Puisque $\Gamma$ est valide, $\Gamma_1$ est valide et, pour tout $i < j$, $\Gamma_1 \vdash x_i : T_i$. Puisque $T_j\ \mathbb{C}_{\Gamma_1}\ T'_j$, il existe $s$ et $s'$ tels que $\Gamma_1 \vdash T_j : s$ et $\Gamma_1 \vdash T'_j : s'$. Par (weak), on obtient, pour tout $i < j$, $\Gamma' \vdash x_i : T_i$, et par (var), on obtient $\Gamma' \vdash x_j : T'_j$. De $\Gamma_1 \vdash T_j : s$, on obtient aussi, par (weak), $\Gamma' \vdash T_j : s$. Donc, par (conv), $\Gamma' \vdash x_j : T_j$.

Supposons maintenant que $\Gamma_2 = \Gamma_3, x_n : T_n$. Posons $\Delta = \Gamma_1, x_j : T_j, \Gamma_3$ et $\Delta' = \Gamma_1, x_j : T'_j, \Gamma_3$. Par hypothèse de récurrence, pour tout $i < n$, $\Delta' \vdash x_i : T_i$. Puisque $\Gamma$ est valide, il existe $s$ tel que $\Delta \vdash T_n : s$. Par transitivité, on obtient $\Delta' \vdash T_n : s$. Donc, par (var), on obtient $\Gamma' \vdash x_n : T_n$, et par (weak), $\Gamma' \vdash x_i : T_i$. ■

## 3.3 TSM stables par substitution

**Définition 23 (TSM stable par substitution)** Un TSM est stable par substitution si sa relation de conversion $\mathcal{C}$ est stable par substitution.

**Lemme 24 (Substitution)** Si $\mathcal{C}$ est stable par substitution, $\Gamma \vdash t : T$ et $\theta : \Gamma \to \Delta$ alors $\Delta \vdash t\theta : T\theta$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. Dans le cas (symb), par hypothèse de récurrence, nous avons $\Delta \vdash t_i\theta : T_i\gamma\theta$. Donc, par (symb), $\Delta \vdash f(\vec{t\theta}) : U\gamma\theta$. ■

**Corollaire 25** Si $\mathcal{C}$ est stable par substitution, $\Gamma, x{:}U, \Gamma' \vdash t : T$ et $\Gamma \vdash u : U$ alors $\Gamma, \Gamma'\{x \mapsto u\} \vdash t\{x \mapsto u\} : T\{x \mapsto u\}$.

**Preuve.** Cela revient à montrer que $\theta = \{x \mapsto u\}$ est une substitution bien typée de $\Gamma, x : U, \Gamma'$ à $\Gamma, \Gamma'\theta$. Procédons par récurrence sur la taille de $\Gamma'$. Si $\Gamma'$ est vide, c'est immédiat car $\Gamma$ est valide et $\Gamma \vdash u : U$. Supposons maintenant que $\Gamma' = \Gamma'', y{:}V$. Posons $\Delta = \Gamma, x{:}U, \Gamma''$ et $\Delta' = \Gamma, \Gamma''\theta$. Par hypothèse de récurrence, $\theta : \Delta \to \Delta'$. Puisque $\Delta \vdash V : s$, par substitution, on obtient $\Delta' \vdash V\theta : s$. Donc, par (var), $\Delta', y : V\theta \vdash y : V\theta$. Maintenant, soit $z \in \mathrm{dom}(\Delta)$. Comme $\Delta \vdash z : z\Delta$, par substitution, $\Delta' \vdash z : z\Delta\theta$. Alors, par (weak), $\Delta', y{:}V\theta \vdash z : z\Delta\theta$. ■

**Corollaire 26** Si $\mathcal{C}$ est stable par substitution, $\theta_1 : \Gamma_0 \to \Gamma_1$ et $\theta_2 : \Gamma_1 \to \Gamma_2$ alors $\theta_1\theta_2 : \Gamma_0 \to \Gamma_2$.

**Preuve.** Soit $x \in \mathrm{dom}(\Gamma_0)$. Puisque $\theta_1 : \Gamma_0 \to \Gamma_1$, par substitution, on obtient $\Gamma_1 x \vdash x\theta_1 : x\Gamma_0\theta_1$, et puisque $\theta_2 : \Gamma_1 \to \Gamma_2$, par substitution encore, on obtient $\Gamma_2 \vdash x\theta_1\theta_2 : x\Gamma_0\theta_1\theta_2$. ■

**Définition 27 (Sorte maximale)** Une sorte $s$ est *maximale* si aucune sorte $s'$ vérifie $(s, s') \in \mathcal{A}$.

**Lemme 28 (Correction des types)** Si $\mathcal{C}$ est stable par substitution et $\Gamma \vdash t : T$ alors, soit $T$ est une sorte maximale, soit il existe une sorte $s$ telle que $\Gamma \vdash T : s$. Autrement dit, $\mathbb{T} = \bigcup\{\mathbb{T}^s_0 \cup \mathbb{T}^s_1 \mid s \in \mathcal{S}\}$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. Dans le cas (symb), nous avons $\vdash \tau_f : s$. Par inversion, il existe $s'$ tel que $\Gamma_f \vdash U : s'$. Comme $\gamma : \Gamma_f \to \Gamma$, par substitution, $\Gamma \vdash U\gamma : s'$. ■

**Lemme 29 (Inversion pour les TSM stables par substitution)** Supposons que $\Gamma \vdash t : T$.

- Si $t = s$ alors il existe $s'$ tel que $(s, s') \in \mathcal{A}$ et $s'\ \mathcal{C}^*_\Gamma\ T$.
- Si $t = f(\vec{t})$, $f \in \mathcal{F}^s$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$ alors $\vdash \tau_f : s$, $\gamma : (\vec{x} : \vec{T}) \to \Gamma$ et $U\gamma\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = x \in \mathcal{X}^s$ alors $\Gamma \vdash x\Gamma : s$ et $x\Gamma\ \mathbb{C}^*_\Gamma\ T$.

– Si $t = (x{:}U)V$ alors il existe $(s_1, s_2, s_3) \in \mathcal{B}$ tel que $\Gamma \vdash U : s_1$, $\Gamma, x{:}U \vdash V : s_2$ et $s_3\ \mathcal{C}^*_\Gamma\ T$.
– Si $t = [x{:}U]v$ alors il existe $V$ tel que $\Gamma, x{:}U \vdash v : V$ et $(x{:}U)V\ \mathbb{C}^*_\Gamma\ T$.
– Si $t = uv$ alors il existe $V$ et $W$ tels que $\Gamma \vdash u : (x : V)W$, $\Gamma \vdash v : V$ et $W\{x \mapsto v\}\ \mathbb{C}^*_\Gamma\ T$.

**Preuve.** Seuls les cas $t = f(\vec{t})$ et $t = uv$ ont été modifiés, en remplçant $\mathcal{C}_\Gamma$ par $\mathbb{C}_\Gamma$.
– $t = f(\vec{t})$. Par inversion, $\vdash \tau_f : s$, $\gamma : (\vec{x} : \vec{T}) \to \Gamma$ et $U\gamma\ \mathcal{C}^*_\Gamma\ T$. Par inversion encore, il existe $s'$ tel que $\vec{x} : \vec{T} \vdash U : s'$. Donc, par substitution, $\Gamma \vdash U\gamma : s'$ et $U\gamma\ \mathbb{C}^*_\Gamma\ T$.
– $t = uv$. Par inversion, il existe $V$ et $W$ tel que $\Gamma \vdash u : (x{:}V)W$, $\Gamma \vdash v : V$ et $W\{x \mapsto v\}\ \mathcal{C}^*_\Gamma\ T$. Par correction des types, il existe $s$ tel que $\Gamma \vdash (x{:}V)W : s$. Par inversion, il existe $s'$ tel que $\Gamma, x{:}V \vdash W : s'$. Donc, par substitution, $\Gamma \vdash W\{x \mapsto v\} : s'$ et $W\{x \mapsto v\}\ \mathbb{C}^*_\Gamma\ T$. ■

## 3.4 TSM logiques

Nous introduisons maintenant une importante classe de TSM pour laquelle la $\beta$-réduction préserve le typage.

**Définition 30 (TSM logique)** Un TSM est *logique* si sa relation de conversion est *compatible avec le produit* :

$$(x{:}T)U\ \mathbb{C}^*_\Gamma\ (x'{:}T')U' \text{ implique } T\ \mathbb{C}^*_\Gamma\ T' \text{ et } U\ \mathbb{C}^*_{\Gamma,x:T}\ U'\{x' \mapsto x\}.$$

$T\ \mathbb{C}^*_\Gamma\ T'$ signifie qu'il existe une séquence de termes $\vec{T}$ telle que $T = T_0\ \mathbb{C}_\Gamma\ T_1\ \ldots T_{n-1}\ \mathbb{C}_\Gamma\ T_n = T'$. Ainsi, il n'y a aucune raison *a priori* de prendre $U\ \mathbb{C}^*_{\Gamma,x:T}\ U'\{x' \mapsto x\}$ plutôt que $U\ \mathbb{C}^*_{\Gamma,x:T_i}\ U'\{x' \mapsto x\}$ avec $i \neq 0$. En effet, comme $T\ \mathbb{C}^*_\Gamma\ T'$, d'après le Lemme de conversion pour les environnements, ce choix n'a pas d'importance.

La compatibilité avec le produit n'est pas une condition nouvelle et apparaît dans tous les travaux cités précédemment mais, à notre connaissance, elle n'a jamais fait l'objet d'une dénomination particulière.

Tous les TSM cités au début de ce chapitre sont logiques. Dans le cas où $\mathcal{C} = \leftrightarrow^*$ avec $\to$ une relation de réduction confluente, il est clair que $\mathcal{C}$ est compatible avec le produit. Montrer cette propriété quand on ne dispose pas de la confluence est plus délicat. C'est le cas des PTS avec $\eta$-réduction [56] ou du $\lambda R$-cube [51], une extension du Calcul des Constructions avec récriture d'ordre supérieur à la Jouannaud-Okada au niveau objet.

**Lemme 31 (Correction de la $\beta$-réduction)** Dans un $\beta$TSM logique, si $\Gamma \vdash t : T$ et $t \to_\beta t'$ alors $\Gamma \vdash t' : T$.

**Preuve.** Nous dirons qu'un environnement $\vec{x} : \vec{T}$ se $\beta$-récrit en un environnement $\vec{x}' : \vec{T}'$, noté $\vec{x} : \vec{T} \to_\beta \vec{x}' : \vec{T}'$, si $\vec{x} = \vec{x}'$ et, soit $|\vec{x}| = 0$, soit il existe $j$ tel que $T_j \to_\beta T'_j$ et, pour tout $i \neq j$, $T_i = T'_i$. Nous prouvons simultanément que :

(a) si $t \to_\beta t'$ alors $\Gamma \vdash t' : T$,
(b) si $\Gamma \to_\beta \Gamma'$ alors $\Gamma' \vdash t : T$,

par récurrence sur $\Gamma \vdash t : T$.

**(ax)** $\vdash s_1 : s_2 \quad ((s_1, s_2) \in \mathcal{A})$
Aucune $\beta$-réduction n'est possible dans $s_1$ ou dans $\Gamma = \emptyset$.

**(symb)** $\dfrac{\vdash \tau_f : s \quad \Gamma \text{ valide} \quad \Gamma \vdash t_1 : T_1\gamma \ldots \Gamma \vdash t_n : T_n\gamma}{\Gamma \vdash f(\vec{t}) : U\gamma} \quad \begin{array}{c}(f \in \mathcal{F}^s_n,\\ \tau_f = (\vec{x} : \vec{T})U,\\ \gamma = \{\vec{x} \mapsto \vec{t}\})\end{array}$

(a) Si $f(\vec{t}) \to_\beta t'$ alors $t' = f(\vec{t'})$ avec $j$ tel que $t_j \to_\beta t'_j$ et, pour tout $i \neq j$, $t_i = t'_i$. Par hypothèse de récurrence, nous avons, pour tout $i$, $\Gamma \vdash t'_i : T_i\gamma$. Soit $\gamma' = \{\vec{x} \mapsto \vec{t'}\}$. Nous avons $U\gamma'^* \leftarrow \beta U\gamma$ et, pour tout $i$, $T_i\gamma \to^*_\beta T_i\gamma'$. Comme $\downarrow_\beta \subseteq \mathcal{C}$, $U\gamma' \; \mathcal{C} \; U\gamma$ et $T_i\gamma \; \mathcal{C} \; T_i\gamma'$. Si nous prouvons que chaque $T_i\gamma'$ est typable par une sorte dans $\Gamma$ alors, par (conv), nous avons $\Gamma \vdash t'_i : T_i\gamma'$ et, par (symb), $\Gamma \vdash t' : U\gamma'$. Il suffit alors de montrer que $U\gamma$ est typable par une sorte dans $\Gamma$ pour pouvoir appliquer à nouveau (conv) et conclure que $\Gamma \vdash t' : U\gamma$.

Commençons par vérifier que $U\gamma$ est typable par une sorte. Nous avons $\vdash \tau_f : s$. Par inversion, $\gamma : (\vec{x} : \vec{T}) \to \Gamma$ et il existe $s'$ tel que $\vec{x} : \vec{T} \vdash U : s'$. Par substitution, il vient donc $\Gamma \vdash U\gamma : s'$.

Maintenant, nous allons montrer que chaque $T_i\gamma'$ est typable par une sorte. Pour cela, il suffit de montrer que $\gamma' : (\vec{x} : \vec{T}) \to \Gamma$. En effet, puisque $\vdash \tau_f : s$, par inversion, chaque $T_i$ est typable par une sorte dans $\Gamma_{i-1} = x_1 : T_1, \ldots, x_{i-1} : T_{i-1}$. Montrons donc par récurrence sur $i$ que $\gamma' : \Gamma_i \to \Gamma$.

Pour $i = 0$, il n'y a rien à prouver. Supposons donc que $\gamma' : \Gamma_i \to \Gamma$. Alors $\gamma' : \Gamma_{i+1} \to \Gamma$ si $\Gamma \vdash t'_{i+1} : T_{i+1}\gamma'$. Nous savons que $\Gamma \vdash t'_{i+1} : T_{i+1}\gamma$, que $T_{i+1}\gamma \to^*_\beta T_{i+1}\gamma'$ et qu'il existe $s$ tel que $\Gamma_i \vdash T_{i+1} : s$. Donc, par substitution, $\Gamma \vdash T_{i+1}\gamma' : s$ et, par (conv), $\Gamma \vdash t'_{i+1} : T_{i+1}\gamma'$.

(b) Si $\Gamma \to_\beta \Gamma'$ alors, par hypothèse de récurrence, $\Gamma'$ est valide et $\Gamma' \vdash t_i : T_i\gamma$. Donc, par (symb), $\Gamma' \vdash f(\vec{t}) : U\gamma$.

**(var)** $\dfrac{\Gamma \vdash T : s}{\Gamma, x : T \vdash x : T}$

(a) Aucune $\beta$-réduction n'est possible dans $x$.
(b) Il y a deux cas, selon que la $\beta$-réduction ait lieu dans $\Gamma$ ou dans $T$ :
- $\Gamma \to_\beta \Gamma'$. Par hypothèse de récurrence, $\Gamma' \vdash T : s$. Donc, par (var), $\Gamma', x : T \vdash x : T$.
- $T \to_\beta T'$. Par hypothèse de récurrence, $\Gamma \vdash T' : s$. Donc, par (var), $\Gamma, x : T' \vdash x : T'$. Comme $\downarrow_\beta \subseteq \mathcal{C}$, $T' \; \mathcal{C} \; T$. Comme $\Gamma \vdash T : s$, par (conv), $\Gamma, x : T' \vdash x : T$.

**(weak)** $\dfrac{\Gamma \vdash t : T \quad \Gamma \vdash U : s}{\Gamma, x : U \vdash t : T}$

(a) Si $t \to_\beta t'$ alors, par hypothèse de récurrence, $\Gamma \vdash t' : T$. Comme $\Gamma \vdash U : s$, par (weak), $\Gamma, x : U \vdash t' : T$.
(b) Il y a deux cas, selon que la $\beta$-réduction a lieu dans $\Gamma$ ou dans $U$ :
- $\Gamma \to_\beta \Gamma'$. Par hypothèse de récurrence, $\Gamma' \vdash t : T$ et $\Gamma' \vdash U : s$. Donc, par

(weak), $\Gamma', x{:}U \vdash t : T$.

– $U \to_\beta U'$. Par hypothèse de récurrence, $\Gamma \vdash U' : s$. Donc, par (weak), $\Gamma, x{:}U' \vdash t : T$.

**(prod)** $\dfrac{\Gamma \vdash T : s_1 \quad \Gamma, x{:}T \vdash U : s_2}{\Gamma \vdash (x{:}T)U : s_3} \quad ((s_1, s_2, s_3) \in \mathcal{B})$

(a) Il y a deux cas selon que la $\beta$-réduction ait lieu dans $T$ ou $U$ :

– $T \to_\beta T'$. Par hypothèse de récurrence, $\Gamma \vdash T' : s_1$ et $\Gamma, x : T' \vdash U : s_2$. Donc, par (prod), on obtient $\Gamma \vdash (x{:}T')U : s_3$.

– $U \to_\beta U'$. Par hypothèse de récurrence, $\Gamma, x{:}T \vdash U' : s_2$. Donc, par (prod), $\Gamma \vdash (x{:}T)U' : s_3$.

(b) Si $\Gamma \to_\beta \Gamma'$ alors, par hypothèse de récurrence, $\Gamma' \vdash T : s_1$ et $\Gamma', x{:}T \vdash U : s_2$. Donc, par (prod), $\Gamma' \vdash (x{:}T)U : s_3$.

**(abs)** $\dfrac{\Gamma, x{:}T \vdash u : U \quad \Gamma \vdash (x{:}T)U : s}{\Gamma \vdash [x{:}T]u : (x{:}T)U}$

(a) Il y a deux cas selon que la $\beta$-réduction ait lieu dans $T$ ou dans $u$ :

– $T \to_\beta T'$. Par hypothèse de récurrence, $\Gamma, x{:}T' \vdash u : U$ et $\Gamma \vdash (x{:}T')U : s$. Par (abs), $\Gamma \vdash [x{:}T']u : (x{:}T')U$. Comme $(x{:}T')U \leftarrow \beta(x{:}T)U$ et $\downarrow_\beta \subseteq \mathcal{C}$, $(x{:}T')U \ \mathcal{C} \ (x{:}T)U$. Comme $\Gamma \vdash (x{:}T)U : s$, par (conv), $\Gamma \vdash [x{:}T']u : (x{:}T)U$.

– $u \to_\beta u'$. Par hypothèse de récurrence, $\Gamma, x : T \vdash u' : U$. Comme $\Gamma \vdash (x{:}T)U : s$, par (abs), $\Gamma \vdash [x{:}T]u' : (x{:}T)U$.

(b) Si $\Gamma \to_\beta \Gamma'$ alors, par hypothèse de récurrence, $\Gamma', x : T \vdash u : U$ et $\Gamma' \vdash (x{:}T)U : s$. Donc, par (abs), $\Gamma' \vdash [x{:}T]u : (x{:}T)U$.

**(app)** $\dfrac{\Gamma \vdash t : (x{:}U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$

(a) Il y a trois cas selon que la $\beta$-réduction ait lieu dans $t$, dans $u$ ou au sommet de $tu$ :

– $t \to_\beta t'$. Par hypothèse de récurrence, $\Gamma \vdash t' : (x{:}U)V$. Comme $\Gamma \vdash u : U$, par (app), $\Gamma \vdash t'u : V\{x \mapsto u\}$.

– $u \to_\beta u'$. Par hypothèse de récurrence, $\Gamma \vdash u' : U$. Par (app), $\Gamma \vdash tu' : V\{x \mapsto u'\}$. Comme $V\{x \mapsto u'\}^* \leftarrow \beta V\{x \mapsto u\}$ et $\downarrow_\beta \subseteq \mathcal{C}$, $V\{x \mapsto u'\} \ \mathcal{C} \ V\{x \mapsto u\}$. Par inversion, il existe $s$ tel que $\Gamma \vdash V\{x \mapsto u\} : s$. Donc, par (conv), $\Gamma \vdash tu' : V\{x \mapsto u\}$.

– $t = [x{:}U']v$ et $tu \to_\beta v\{x \mapsto u\}$. Par inversion, il existe $V'$ tel que $\Gamma, x{:}U' \vdash v : V'$ et $(x{:}U')V' \ \mathbb{C}^*_\Gamma \ (x{:}U)V$. Par compatibilité avec le produit, $U' \ \mathbb{C}^*_\Gamma \ U$ et $V' \ \mathbb{C}^*_{\Gamma, x:U} \ V$. Par conversion de l'environnement, $\Gamma, x{:}U \vdash v : V'$ et, par (conv), $\Gamma, x{:}U \vdash v : V$.

(b) Si $\Gamma \to_\beta \Gamma'$ alors, par hypothèse de récurrence, $\Gamma' \vdash t : (x{:}U)V \quad \Gamma' \vdash u : U$. Donc, par (app), $\Gamma' \vdash tu : V\{x \mapsto u\}$.

**(conv)** $\dfrac{\Gamma \vdash t : T \quad T \ \mathcal{C} \ T' \quad \Gamma \vdash T' : s'}{\Gamma \vdash t : T'}$

(a) Si $t \to_\beta t'$ alors, par hypothèse de récurrence, $\Gamma \vdash t' : T$. Comme $T \ \mathcal{C} \ T'$ et $\Gamma \vdash T' : s'$, par (conv), $\Gamma \vdash t' : T'$.

(b) Si $\Gamma \to_\beta \Gamma'$ alors, par hypothèse de récurrence, $\Gamma' \vdash t : T$ et $\Gamma' \vdash T' : s'$. Donc, par (conv), $\Gamma' \vdash t : T'$. ■

# Chapitre 4

# Systèmes de Types à Réduction (RTS)

Maintenant, nous allons étudier le cas des TSM dont la relation de conversion $\mathcal{C}$ est de la forme $\downarrow$ avec $\rightarrow$ une relation de réduction. Nous appellerons de tels systèmes des Systèmes de Types à Réduction (RTS). Excepté ECC [80], semble-t-il, qui utilise une notion de sous-typage, les systèmes cités précédemment sont des RTS, soit par ce qu'ils sont définis comme tels [51, 14], soit parce qu'ils sont définis avec $\mathcal{C} = \leftrightarrow^*$ et $\rightarrow$ confluente, ce qui est équivalent [99, 56, 118, 101]. L'étude générale de tels systèmes est justifiée par le fait que, dans [51], la preuve que la réduction préserve le typage repose sur le fait que $\mathcal{C}$ soit de la forme $\downarrow$. Enfin, pour ECC, on peut se demander s'il n'y a pas un RTS équivalent.

Les preuves des Lemmes 41, 47, 50 et 52 sont largement inspirées de celles données par H. Geuvers et M.-J. Nederhof [57] ou H. Geuvers [56].

## 4.1 Définition

**Définition 32 (RTS)** Un *pré-RTS* est un TSM dont la relation de conversion $\mathcal{C}$ est de la forme $\downarrow$ où $\rightarrow$ désigne une relation stable par substitution et contexte. La relation $\rightarrow$ est appelée *relation de réduction* du pré-RTS. Un pré-RTS est *confluent* si sa relation de réduction est confluente. Un pré-RTS est *admissible* si sa relation de réduction *préserve le typage* : $\Gamma \vdash t : T$ et $t \rightarrow t'$ impliquent $\Gamma \vdash t' : T$. On dit alors que c'est un *RTS*.

Tout pré-RTS vérifie les propriétés élémentaires suivantes :

**Lemme 33** La relation $\mathcal{C}$ est :
– stable par substitution : $T\ \mathcal{C}\ T'$ implique $T\theta\ \mathcal{C}\ T'\theta$.
– stable par contexte : $T\ \mathcal{C}\ T'$ implique $C[T]_p\ \mathcal{C}\ C[T']_p$.
– symétrique : $T\ \mathcal{C}\ T'$ implique $T'\ \mathcal{C}\ T$.
– préserve les sortes : $s\ \mathcal{C}\ s'$ implique $s = s'$.

Dans ECC, la relation de conversion $\mathcal{C}$ n'est pas symétrique et ne préserve pas les sortes. Il serait intéressant de chercher à reformuler certaines des propriétés ci-après

dans le cadre plus général des Systèmes de Types purs Cumulatifs (CTS) dont fait partie ECC. Pour cela, nous renvoyons le lecteur aux travaux de Z. Luo [80], R. Pollack [102] et Barras [11].

La préservation du typage peut être étendue aux types, aux environnements et aux substitutions :

**Définition 34** Une substitution $\theta$ *se récrit* en une substitution $\theta'$, $\theta \to \theta'$, s'il existe $x$ tel que $x\theta \to x\theta'$ et, pour tout $y \neq x$, $y\theta = y\theta'$. Un environnement $\Gamma = \vec{x} : \vec{T}$ *se récrit* en un environnement $\Gamma'$, $\Gamma \to \Gamma'$, si $\Gamma' = \vec{x} : \vec{T'}$ et il existe $i$ tel que $T_i \to T'_i$ et, pour tout $j \neq i$, $T_j = T'_j$.

**Lemme 35** Dans un RTS :

(a) si $\Gamma \vdash t : T$ et $T \to T'$ alors $\Gamma \vdash t : T'$,
(b) si $\theta : \Gamma \to \Delta$ et $\theta \to \theta'$ alors $\theta' : \Gamma \to \Delta$,
(c) si $\Gamma \vdash t : T$ et $\Gamma \to \Gamma'$ alors $\Gamma' \vdash t : T$.

**Preuve.**

(a) Par correction des types, soit $T = s$ soit $\Gamma \vdash T : s$. Le cas $T = s$ n'est pas possible puisque $s$ n'est pas réductible. Donc, $\Gamma \vdash T : s$ et, par préservation du typage, $\Gamma \vdash T' : s$. Donc, par (conv), $\Gamma \vdash t : T'$.
(b) Par récurrence sur la taille de $\Gamma$. Si $\Gamma$ est vide, c'est immédiat. Supposons alors que $\Gamma = \Gamma', x{:}T$. Puisque $\theta : \Gamma' \to \Delta$, par hypothèse de récurrence, $\theta' : \Gamma' \to \Delta$. Il suffit alors de prouver que $\Delta \vdash x\theta' : T\theta'$. Comme $\theta : \Gamma \to \Delta$, nous avons $\Delta \vdash x\theta : T\theta$. Par préservation du typage, $\Delta \vdash x\theta' : T\theta$. D'après le lemme d'environnement, il existe $s$ tel que $\Gamma \vdash T : s$. Par substitution, $\Delta \vdash T\theta : s$. Puisque $T\theta \to^* T\theta'$, $T\theta \; \mathcal{C} \; T\theta'$ et, par préservation du typage, $\Delta \vdash T\theta' : s$. Donc, par (conv), $\Delta \vdash x\theta' : T\theta'$.
(c) Supposons que $\Gamma = \Gamma_1, x{:}T, \Gamma_2$ et $\Gamma' = \Gamma_1, x{:}T', \Gamma_2$. Par le lemme d'environnement, $\Gamma_1 \vdash T : s$. Par préservation du typage $\Gamma_1 \vdash T' : s$. Donc $\Gamma \; \mathbb{C} \; \Gamma'$ et, par le lemme de conversion des environnements, $\Gamma' \vdash t : T$. ■

**Lemme 36 (Inconvertibilité des sortes maximales)** Dans un RTS, si $s \; \mathcal{C}^*_\Gamma \; T$ alors $s \; \mathbb{C}^*_\Gamma \; T$. Donc, si $s$ est maximale alors $T = s$.

**Preuve.** Par cas sur le nombre de conversions entre $s$ et $T$. Si $s = T$, c'est immédiat. Supposons alors que $s \; \mathcal{C}_\Gamma \; T' \; \mathcal{C}^*_\Gamma \; T$. Par définition de $\mathcal{C}_\Gamma$, il existe $s'$ tel que $\Gamma \vdash T' : s'$. Comme $\mathcal{C} = \downarrow$ et $s$ est irréductible, $T' \to^* s$. Par préservation du typage, $\Gamma \vdash s : s'$ et $s \; \mathbb{C}^*_\Gamma \; T$. ■

On obtient ainsi l'équivalence entre les relations de conversion $\mathcal{C}_\Gamma$ et $\mathbb{C}_\Gamma$, et un raffinement du lemme d'inversion pour les RTS.

**Lemme 37 (Équivalence de $\mathcal{C}_\Gamma$ et $\mathbb{C}_\Gamma$)** Dans un RTS, $\mathcal{C}_\Gamma = \mathbb{C}_\Gamma$.

**Preuve.** Tout d'abord, nous avons $\mathbb{C}_\Gamma \subseteq \mathcal{C}_\Gamma$. Montrons la réciproque. Supposons que $T \; \mathcal{C}_\Gamma \; T'$. Comme il existe $t$ tel que $\Gamma \vdash t : T$, par correction des types, soit $T$ est

une sorte maximale, soit il existe $s$ tel que $\Gamma \vdash T : s$. D'après le lemme précédent, $T$ ne peut pas être une sorte maximale. Donc, il existe $s$ tel que $\Gamma \vdash T : s$ et $T\ \mathbb{C}_\Gamma\ T'$.

■

**Définition 38 (Sorte régulière)** Une sorte $s$ est *régulière* si, pour tout $(s_1, s_2, s) \in \mathcal{B}$, $s_2 = s$. Un TSM est *régulier* si toutes ses sortes sont régulières.

La plupart des PTS que l'on peut rencontrer dans la littérature sont réguliers. Pour ces systèmes, il est souvent fait usage de l'abréviation $(s_1, s_2) \in \mathcal{B}$ pour $(s_1, s_2, s_2) \in \mathcal{B}$ [57, 10].

**Lemme 39 (Inversion pour les RTS)** Supposons que $\Gamma \vdash t : T$.
- Si $t = s$ alors il existe $s'$ tel que $(s, s') \in \mathcal{A}$ et $s'\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = f(\vec{t})$, $f \in \mathcal{F}^s$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$ alors $\vdash \tau_f : s$, $\gamma : (\vec{x} : \vec{T}) \to \Gamma$ et $U\gamma\ \mathbb{C}^*_\Gamma\ T$. De plus, si $s$ est régulière alors $\Gamma \vdash U\gamma : s$.
- Si $t = x \in \mathcal{X}^s$ alors $\Gamma \vdash x\Gamma : s$ et $x\Gamma\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = (x{:}U)V$ alors il existe $(s_1, s_2, s_3) \in \mathcal{B}$ tel que $\Gamma \vdash U : s_1$, $\Gamma, x{:}U \vdash V : s_2$ et $s_3\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = [x{:}U]v$ alors il existe $V$ tel que $\Gamma, x{:}U \vdash v : V$ et $(x{:}U)V\ \mathbb{C}^*_\Gamma\ T$.
- Si $t = uv$ alors il existe $V$ et $W$ tels que $\Gamma \vdash u : (x{:}V)W$, $\Gamma \vdash v : V$ et $W\{x \mapsto v\}\ \mathbb{C}^*_\Gamma\ T$. De plus, si $\Gamma \vdash (x{:}V)W : s$ et $s$ est régulière alors $\Gamma \vdash W\{x \mapsto v\} : s$.

**Preuve.** Les modifications des cas $t = s$ et $t = (x{:}U)V$ découlent de l'inconvertibilité des sortes maximales. Il ne nous reste donc plus qu'à montrer les propriétés supplémentaires dans le cas où on a une sorte régulière. La propriété pour $t = f(\vec{t})$ découle, par itération, de celle pour $t = uv$.

Supposons que $\Gamma \vdash (x : V)W : s$. Par inversion, il existe $(s_1, s_2, s_3) \in \mathcal{B}$ tel que $\Gamma, x : V \vdash W : s_2$ et $s_3\ \mathbb{C}^*_\Gamma\ s$. Par préservation des sortes, $s_3 = s$. Par régularité, $s_2 = s_3$. Donc, $\Gamma, x{:}V \vdash W : s$ et, par substitution, $\Gamma \vdash W\{x \mapsto v\} : s$. ■

## 4.2 RTS logiques et fonctionnels

**Définition 40 (TSM fonctionnel)** Un ensemble de règles $\mathcal{B}$ est *fonctionnel* si $(s_1, s_2, s_3) \in \mathcal{B}$ et $(s_1, s_2, s_3') \in \mathcal{B}$ impliquent $s_3 = s_3'$. Un TSM est *fonctionnel* si $\mathcal{A}$ est une relation fonctionnelle et $\mathcal{B}$ est fonctionnel.

La plupart des PTS que l'on peut rencontrer dans la littérature sont fonctionnels. Par ailleurs, dans un TSM régulier, $\mathcal{B}$ est fonctionnel. Donc, pour qu'un TSM régulier soit fonctionnel, il suffit que $\mathcal{A}$ soit une relation fonctionnelle.

**Lemme 41 (Convertibilité des types)** Dans un RTS logique et fonctionnel, si $\Gamma \vdash t : T$ et $\Gamma \vdash t : T'$ alors $T\ \mathbb{C}^*_\Gamma\ T'$.

**Preuve.** Par récurrence sur $t$. Nous suivons les notations du lemme d'inversion.
- $t = s$. Par inversion, il existe $s_1'$ et $s_2'$ tels que $(s, s_1') \in \mathcal{A}$, $(s, s_2') \in \mathcal{A}$, $s_1'\ \mathbb{C}^*_\Gamma\ T$ et $s_2'\ \mathbb{C}^*_\Gamma\ T'$. Par fonctionnalité, $s_1' = s_2'$. Donc, par symétrie, $T\ \mathbb{C}^*_\Gamma\ T'$.

– $t = f(\vec{t})$. Par inversion, $U\gamma\ \mathbb{C}^*_\Gamma\ T$ et $U\gamma\ \mathbb{C}^*_\Gamma\ T'$. Donc, symétrie, $T\ \mathbb{C}^*_\Gamma\ T'$.
– $t = x$. Par inversion, $x\Gamma\ \mathbb{C}^*_\Gamma\ T$ et $x\Gamma\ \mathbb{C}^*_\Gamma\ T'$. Donc, par symétrie, $T\ \mathbb{C}^*_\Gamma\ T'$.
– $t = (x:U)V$. Par inversion, il existe $(s_1, s_2, s_3) \in \mathcal{B}$ et $(s'_1, s'_2, s'_3) \in \mathcal{B}$ tels que $\Gamma \vdash U : s_1$, $\Gamma \vdash U : s'_1$, $\Gamma, x:U \vdash V : s_2$, $\Gamma, x:U \vdash V : s'_2$, $s_3\ \mathbb{C}^*_\Gamma\ T$ et $s'_3\ \mathbb{C}^*_\Gamma\ T'$. Par hypothèse de récurrence, $s_1\ \mathbb{C}^*_\Gamma\ s'_1$ et $s_2\ \mathbb{C}^*_{\Gamma,x:U}\ s'_2$. Par préservation des sortes, $s_1 = s'_1$ et $s_2 = s'_2$. Donc, par fonctionnalité, $s_3 = s'_3$ et, par symétrie, $T\ \mathbb{C}^*_\Gamma\ T'$.
– $t = [x:U]v$. Par inversion, il existe $V$ et $V'$ tels que $\Gamma, x:U \vdash v : V$, $\Gamma, x:U \vdash v : V'$, $(x:U)V\ \mathbb{C}^*_\Gamma\ T$ et $(x:U)V'\ \mathbb{C}^*_\Gamma\ T'$. Par hypothèse de récurrence, $V\ \mathbb{C}^*_{\Gamma,x:U}\ V'$. Par stabilité par contexte, $(x:U)V\ \mathbb{C}^*_\Gamma\ (x:U)V'$. Donc, par symétrie, $T\ \mathbb{C}^*_\Gamma\ T'$.
– $t = uv$. Par inversion, il existe $V$, $V'$, $W$ et $W'$ tels que $\Gamma \vdash u : (x:V)W$, $\Gamma \vdash u : (x:V')W'$, $W\{x \mapsto v\}\ \mathbb{C}^*_\Gamma\ T$ et $W'\{x \mapsto v\}\ \mathbb{C}^*_\Gamma\ T'$. Par hypothèse de récurrence, $(x:V)W\ \mathbb{C}^*_\Gamma\ (x:V')W'$. Par compatibilité avec le produit, $W\ \mathbb{C}^*_{\Gamma,x:V}\ W'$. Par substitution et stabilité par substitution, $W\{x \mapsto v\}\ \mathbb{C}^*_\Gamma\ W'\{x \mapsto v\}$. Donc, par symétrie, $T\ \mathbb{C}^*_\Gamma\ T'$. ■

**Lemme 42 (Correction de la conversion)** Dans un RTS logique et fonctionnel, si $\Gamma \vdash T : s$ et $T\ \mathbb{C}_\Gamma\ T'$ alors $\Gamma \vdash T' : s$.

**Preuve.** Par définition de $\mathbb{C}_\Gamma$, il existe $s'$ tel que $\Gamma \vdash T' : s'$. Comme $\mathcal{C} = \downarrow$, il existe $U$ tel que $T \to^* U$ et $T' \to^* U$. Par préservation du typage, $\Gamma \vdash U : s$ et $\Gamma \vdash U : s'$. Par convertibilité des types et préservation des sortes, $s = s'$ et $\Gamma \vdash T' : s$.

■

**Lemme 43 (Équivalence entre $\vdash_s$ et $\vdash$)** Dans un RTS, $\vdash_s = \vdash$.

**Preuve.** Tout d'abord, il est immédiat que $\vdash_s \subseteq \vdash$. Montrons la réciproque par récurrence sur $\Gamma \vdash t : T$. Le seul cas ennuyeux est bien sûr (conv). Par hypothèse de récurrence, nous avons $\Gamma \vdash_s t : T$ et $\Gamma \vdash_s T' : s'$. Il est facile de vérifier que les lemmes de substitution et de correction des types sont aussi valables pour $\vdash_s$. Ainsi, par correction des types, soit $T$ est une sorte maximale, soit il existe $s'$ tel que $\Gamma \vdash_s T : s'$. Si $s'$ est une sorte maximale alors $T' \to^* s'$ et $s'$ est typable, ce qui est exclu. Donc, $\Gamma \vdash_s T : s'$. Par convertibilité des types et préservation des sortes, $s = s'$ et, par (conv'), $\Gamma \vdash_s t : T'$. ■

**Lemme 44 ($\alpha$-Équivalence)** Dans un RTS logique et fonctionnel, si $(x:T)U\ \mathbb{C}_\Gamma\ (x' : T')U'$ alors $x$ et $x'$ sont de la même sorte et $(x' : T')U'$ est $\alpha$-équivalent à $(x:T')U'\{x' \mapsto x\}$.

**Preuve.** Supposons que $x$ soit de sorte $s$ et $x'$ de sorte $s'$. Par définition de $\mathbb{C}_\Gamma$, nous avons $\Gamma \vdash (x:T)U : s_3$ et $\Gamma \vdash (x' : T')U' : s'_3$. Par inversion, nous avons $\Gamma, x : T \vdash U : s_1$ et $\Gamma, x' : T' \vdash U' : s'_1$. Par le Lemme d'environnement, nous avons $\Gamma \vdash T : s$ et $\Gamma \vdash T' : s'$. Par correction de la conversion et préservation des sortes, $s = s'$. Donc $x$ et $x'$ sont de même sorte et $(x' : T')U'$ est $\alpha$-équivalent à $(x:T')U'\{x' \mapsto x\}$. ■

**Lemme 45 (Sorte maximale)** Dans un RTS logique et fonctionnel, si $s$ est une sorte maximale et $\Gamma \vdash t : s$ alors $t$ est de la forme $(\vec{x} : \vec{t})s'$. De plus, si $t\ \mathbb{C}^*_\Gamma\ t'$ alors $t'$ est de la forme $(\vec{y} : \vec{t'})s'$ avec $|\vec{y}| = |\vec{x}|$.

**Preuve.** Montrons la première assertion par cas sur $t$. Remarquons tout d'abord qu'il n'existe aucun $s'$ tel que $\Gamma \vdash s : s'$. Sinon, par inversion, il existerait $s''$ tel que $(s, s'') \in \mathcal{A}$ et $s'' \ \mathbb{C}^*_\Gamma \ s'$, ce qui est exclu puisque $s$ est maximale.

- $t = f(\vec{t})$. Soit $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$. Par inversion, il existe $s'$ tel que $\Gamma \vdash U\gamma : s'$ et $U\gamma \ \mathbb{C}^*_\Gamma \ s$. Par correction de la conversion, $\Gamma \vdash s : s'$. Ce cas est donc impossible.
- $t = x \in \mathcal{X}^{s'}$. Par inversion, $\Gamma \vdash x\Gamma : s'$ et $x\Gamma \ \mathbb{C}^*_\Gamma \ s$. Par correction de la conversion, $\Gamma \vdash s : s'$. Ce cas est donc impossible.
- $t = [x{:}U]v$. Par inversion, il existe $V$ et $s'$ tel que $\Gamma, x{:}U \vdash v : V$, $\Gamma \vdash (x{:}U)V : s'$ et $(x : U)V \ \mathbb{C}^*_\Gamma \ s$. Par correction de la conversion, $\Gamma \vdash s : s'$. Ce cas est donc impossible.
- $t = uv$. Par inversion, il existe $V$, $W$ et $s'$ tel que $\Gamma \vdash W\{x \mapsto u\} : s'$ et $W\{x \mapsto u\} \ \mathbb{C}^*_\Gamma \ s$. Par correction de la conversion, $\Gamma \vdash s : s'$. Ce cas est donc impossible.

Il ne reste plus que les cas $t = (x{:}U)V$ et $t = s'$. Donc $t$ doit être de la forme $(\vec{x} : \vec{t})s'$.

Montrons maintenant la deuxième assertion. Par correction de la conversion, $\Gamma \vdash t' : s$. D'après la première assertion, $t'$ est de la forme $(\vec{y} : \vec{t'})s''$. Par compatibilité avec le produit et $\alpha$-équivalence, et quitte à échanger les rôles de $t$ et $t'$, on peut supposer que $\vec{y} = \vec{x}\vec{z}$ et $\vec{t'} = \vec{t}\vec{u}$. Ainsi, $s' \ \mathbb{C}^*_{\Gamma'} \ (\vec{z} : \vec{u})s''$ où $\Gamma' = \vec{x} : \vec{t}$. Prouvons alors par récurrence sur le nombre de conversions entre $s'$ et $(\vec{z} : \vec{u})s''$ que $|\vec{z}| = 0$ et $s' = s''$. Si $s' = (\vec{z} : \vec{u})s''$, c'est immédiat. Supposons alors que $s' \ \mathbb{C}'_\Gamma \ v \ \mathbb{C}^*_{\Gamma'} \ (\vec{z} : \vec{u})s''$. Par correction de la conversion, $\Gamma' \vdash v : s$. Donc, d'après la première assertion, $v$ est de la forme $(\vec{z'} : \vec{u'})s'''$. Comme $\mathcal{C} = \downarrow$ et $s'$ est irréductible, $v \to^* s'$. Donc $|\vec{z'}| = 0$ et $s''' = s'$. Ainsi, $v = s'$ et, par hypothèse de récurrence, $|\vec{z}| = 0$ et $s'' = s'$. ■

## 4.3 RTS logiques et injectifs

**Définition 46 (TSM injectif)** Un ensemble de règles $\mathcal{B}$ est *injectif* si $(s_1, s_2, s_3) \in \mathcal{B}$ et $(s_1, s'_2, s_3) \in \mathcal{B}$ implique $s_2 = s'_2$. Un TSM est *injectif* si $\mathcal{A}$ est une relation fonctionnelle et injective et $\mathcal{B}$ est fonctionnel et injectif.

Dans un TSM régulier, $\mathcal{B}$ est fonctionnel et injectif. Donc, pour qu'un TSM régulier soit fonctionnel, il suffit que $\mathcal{A}$ soit une relation fonctionnelle et injective.

**Lemme 47 (Séparation)** Dans un RTS logique et injectif, si $s_1 \neq s_2$ alors, pour tout $i \in \{0, 1\}$, $\mathbb{T}_i^{s_1} \cap \mathbb{T}_i^{s_2} = \emptyset$.

**Preuve.** Montrons que $t \in \mathbb{T}_i^{s_1} \cap \mathbb{T}_i^{s_2}$ implique $s_1 = s_2$, par récurrence sur $t$.

**Cas i = 0**. Il existe $\Gamma_j$ tel que $\Gamma_j \vdash t : s_j$.

- $t = s$. Par inversion, il existe $s'_j$ tel que $(s, s'_j) \in \mathcal{A}$ et $s'_j \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Par fonctionnalité, $s'_1 = s'_2$. Notons $s'$ par la sorte $s'_1 = s'_2$. Alors $s' \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Donc, par préservation des sortes, $s_1 = s_2 = s'$.
- $t = f(\vec{t})$, $f \in \mathcal{F}^s$ et $\tau_f = (\vec{x} : \vec{T})U$. Soit $\Gamma = \vec{x} : \vec{T}$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$. Par inversion, $\vdash \tau_f : s$, $\gamma : \Gamma \to \Gamma_j$ et $U\gamma \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Par inversion encore, il existe $s'$ tel

que $\Gamma \vdash U : s'$. Par substitution, $\Gamma_j \vdash U\gamma : s'$. Par correction de la conversion, $\Gamma_j \vdash s_j : s'$. Par inversion et préservation des sortes, $(s_j, s') \in \mathcal{A}$. Donc, par injectivité, $s_1 = s_2$.

- $t = x \in \mathcal{X}^s$. Par inversion, $\Gamma_j \vdash x\Gamma_j : s$ et $x\Gamma_j \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Par correction de la conversion, $\Gamma_j \vdash s_j : s$. Par inversion et préservation des sortes, $(s_j, s) \in \mathcal{A}$. Donc, par injectivité, $s_1 = s_2$.
- $t = (x : U)V$. Par inversion, il existe $(s_j^1, s_j^2, s_j^3) \in \mathcal{B}$ tel que $\Gamma_j \vdash U : s_j^1$, $\Gamma_j, x{:}U \vdash V : s_j^2$ et $s_j^3 \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Par hypothèse de récurrence, $s_1^1 = s_2^1$ et $s_1^2 = s_2^2$. Donc, par fonctionnalité, $s_1^3 = s_2^3$. Notons $s$ la sorte $s_1^3 = s_2^3$. Alors, $s \ \mathbb{C}^*_{\Gamma_j} \ s_j$ et, par préservation des sortes, $s_1 = s_2 = s$.
- $t = [x : U]v$. Par inversion, il existe $V_j$ et $s_j^4$ tel que $\Gamma_j, x : U \vdash v : V_j$, $\Gamma_j \vdash (x{:}U)V_j : s_j^4$ et $(x{:}U)V_j \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Par inversion encore, il existe $(s_j^1, s_j^2, s_j^3) \in \mathcal{B}$ tel que $\Gamma_j \vdash U : s_j^1$, $\Gamma_j, x : U \vdash V_j : s_j^2$ et $s_j^3 \ \mathbb{C}^*_{\Gamma_j} \ s_j^4$. Par préservation des sortes, $s_j^3 = s_j^4$. Par hypothèse de récurrence, $s_1^1 = s_2^1$ et $s_1^2 = s_2^2$. Donc, par fonctionnalité, $s_1^3 = s_2^3$. Notons $s$ la sorte $s_1^3 = s_2^3 = s_1^4 = s_2^4$. Par correction de la conversion, $\Gamma \vdash s_j : s$. Par inversion et préservation des sortes, $(s_j, s) \in \mathcal{A}$. Donc, par injectivité, $s_1 = s_2$.
- $t = uv$. Par inversion, il existe $V_j$ et $W_j$ tels que $\Gamma_j \vdash u : (x_j{:}V_j)W_j$, $\Gamma_j \vdash v : V_j$ et $W_j\{x_j \mapsto v\} \ \mathbb{C}^*_{\Gamma_j} \ s_j$. Soit $\Gamma'_j = \Gamma_j, x_j : V_j$ et $\theta_j = \{x_j \mapsto v\}$. Par correction des types, il existe $s'_j$ tel que $\Gamma_j \vdash (x_j{:}V_j)W_j : s'_j$. Par hypothèse de récurrence sur $u$, $s'_1 = s'_2$. Posons $s' = s'_1 = s'_2$. Par inversion, il existe $(s_j^1, s_j^2, s_j^3) \in \mathcal{B}$ tel que $\Gamma_j \vdash V_j : s_j^1$, $\Gamma'_j \vdash W_j : s_j^2$ et $s_j^3 \ \mathbb{C}^*_{\Gamma_j} \ s'$. Par préservation des sortes, $s_j^3 = s'$. Par hypothèse de récurrence sur $v$, $s_1^1 = s_2^1$. Donc, par injectivité, $s_1^2 = s_2^2$. Posons $s'' = s_1^2 = s_2^2$. Comme $\theta_j : \Gamma'_j \mapsto \Gamma_j$, par substitution, $\Gamma_j \vdash W_j\theta_j : s''$. Par correction de la conversion, $\Gamma_j \vdash s_j : s''$. Par inversion et préservation des sortes, $(s_j, s'') \in \mathcal{A}$. Donc, par injectivité, $s_1 = s_2$.

**Cas** $i = 1$. Il existe $\Gamma_j$ et $T_j$ tels que $\Gamma_j \vdash t : T_j$ et $\Gamma_j \vdash T_j : s_j$.

- $t = s$. D'après le cas $i = 0$, il existe $s'$ tel que $s' \ \mathbb{C}^*_{\Gamma_j} \ T_j$. Par correction de la conversion, $\Gamma_j \vdash s' : s_j$. Par inversion et préservation des sortes, $(s', s_j) \in \mathcal{A}$. Donc, par fonctionnalité, $s_1 = s_2$.
- $t = f(\vec{t})$. D'après le cas $i = 0$, il existe $s'$ tel que $\Gamma_j \vdash T_j : s'$. Par convertibilité des types et préservation des sortes, $s_1 = s_2 = s'$.
- $t = x \in \mathcal{X}^s$. D'après le cas $i = 0$, $\Gamma_j \vdash T_j : s$. Donc, par convertibilité des types et préservation des sortes, $s_1 = s_2 = s$.
- $t = (x{:}U)V$. D'après le cas $i = 0$, il existe $s$ tel que $s \ \mathbb{C}^*_{\Gamma_j} \ T_j$. Par correction de la conversion, $\Gamma_j \vdash s : s_j$. Par inversion et préservation des sortes, $(s, s_j) \in \mathcal{A}$. Donc, par fonctionnalité, $s_1 = s_2$.
- $t = [x{:}U]v$. D'après le cas $i = 0$, il existe $s$ tel que $\Gamma_j \vdash T_j : s$. Par convertibilité des types et préservation des sortes, $s_1 = s_2 = s$.
- $t = uv$. D'après le cas $i = 0$, il existe $s''$ tel que $\Gamma_j \vdash T_j : s''$. Par convertibilité des types et préservation des sortes, $s_1 = s_2 = s'$. ■

**Lemme 48 (Classification)** Dans un RTS logique et injectif, soit $(s_1, s_2) \in \mathcal{A}$ et

$\mathbb{T}_0^{s_1} \subseteq \mathbb{T}_1^{s_2}$, soit $(s_1, s_2) \notin \mathcal{A}$ et $\mathbb{T}_0^{s_1} \cap \mathbb{T}_1^{s_2} = \emptyset$.

**Preuve.** Si $(s_1, s_2) \in \mathcal{A}$, il est clair que $\mathbb{T}_0^{s_1} \subseteq \mathbb{T}_1^{s_2}$. Montrons alors que $t \in \mathbb{T}_0^{s_1} \cap \mathbb{T}_1^{s_2}$ implique $(s_1, s_2) \in \mathcal{A}$, par récurrence sur $t$. Soient $\Gamma$, $\Gamma'$ et $T$ tels que $\Gamma \vdash t : s_1$, $\Gamma' \vdash t : T$ et $\Gamma' \vdash T : s_2$.

- $t = s$. Par inversion, il existe $s_1'$ et $s_2'$ tels que $(s, s_1') \in \mathcal{A}$, $s_1' \ \mathbb{C}_\Gamma^* \ s_1$, $(s, s_2') \in \mathcal{A}$ et $s_2' \ \mathbb{C}_{\Gamma'}^* \ T$. Par préservation des sortes, $s_1' = s_1$. Par fonctionnalité, $s_1' = s_2'$. Par correction de la conversion, $\Gamma \vdash s_2' : s_2$. Donc, par inversion et préservation des sortes, $(s_2', s_2) \in \mathcal{A}$ et $(s_1, s_2) \in \mathcal{A}$.
- $t = f(\vec{t})$. Soit $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{t}\}$. Par inversion, il existe $s_1'$ et $s_2'$ tels que $\Gamma \vdash U\gamma : s_1'$, $U\gamma \ \mathbb{C}_\Gamma^* \ s_1$, $\Gamma' \vdash U\gamma : s_2'$ et $U\gamma \ \mathbb{C}_{\Gamma'}^* \ T$. Par correction de la conversion, $\Gamma \vdash s_1 : s_1'$ donc, par inversion et préservation des sortes, $(s_1, s_1') \in \mathcal{A}$. Par séparation, $s_1' = s_2'$. Par correction de la conversion, $\Gamma' \vdash T : s_2'$. Donc, par séparation, $s_2' = s_2$ et $(s_1, s_2) \in \mathcal{A}$.
- $t = x \in \mathcal{X}^s$. Par inversion, $\Gamma \vdash x\Gamma : s$, $x\Gamma \ \mathbb{C}_\Gamma^* \ s_1$, $\Gamma' \vdash x\Gamma' : s$ et $x\Gamma' \ \mathbb{C}_{\Gamma'}^* \ T$. Par correction de la conversion, $\Gamma \vdash s_1 : s$. Par inversion et préservation des sortes, $(s_1, s) \in \mathcal{A}$. Par correction de la conversion, $\Gamma' \vdash T : s$. Donc, par séparation, $s = s_2$ et $(s_1, s_2) \in \mathcal{A}$.
- $t = (x : U)V$. Par inversion, il existe $(s_a, s_b, s_c)$ et $(s_a', s_b', s_c') \in \mathcal{B}$ tels que $\Gamma \vdash U : s_a$, $\Gamma, x : U \vdash V : s_b$, $s_c \ \mathbb{C}_\Gamma^* \ s_1$, $\Gamma' \vdash U : s_a'$, $\Gamma', x : U \vdash V : s_b'$ et $s_c' \ \mathbb{C}_{\Gamma'}^* \ T$. Par préservation des sortes, $s_c = s_1$. Par correction de la conversion, $\Gamma' \vdash s_c' : s_2$. Par inversion et préservation des sortes, $(s_c', s_2) \in \mathcal{A}$. Par séparation, $s_a = s_a'$ et $s_b = s_b'$. Donc, par fonctionnalité, $s_c = s_c'$ et $(s_1, s_2) \in \mathcal{A}$.
- $t = [x : U]v$. Par inversion, il existe $V$, $s$, $V'$ et $s'$ tels que $\Gamma, x : U \vdash v : V$, $\Gamma \vdash (x : U)V : s$, $(x : U)V \ \mathbb{C}_\Gamma^* \ s_1$, $\Gamma', x : U \vdash v : V'$, $\Gamma' \vdash (x : U)V' : s'$ et $(x : U)V' \ \mathbb{C}_{\Gamma'}^* \ T$. Par correction de la conversion, $\Gamma \vdash s_1 : s$ et $\Gamma' \vdash T : s'$. Par inversion et préservation des sortes, $(s_1, s) \in \mathcal{A}$. Par séparation $s' = s_2$. Par inversion encore, il existe $(s_a, s_b, s_c)$ et $(s_a', s_b', s_c') \in \mathcal{B}$ tels que $\Gamma \vdash U : s_a$, $\Gamma, x : U \vdash V : s_b$, $s_c \ \mathbb{C}_\Gamma^* \ s$, $\Gamma' \vdash U : s_a'$ et $\Gamma', x : U \vdash V' : s_b'$ et $s_c' \ \mathbb{C}_{\Gamma'}^* \ s'$. Par préservation des sortes, $s_c = s$ et $s_c' = s'$. Par séparation, $s_a = s_a'$ et $s_b = s_b'$. Donc, par fonctionnalité, $s_c = s_c'$ et $(s_1, s_2) \in \mathcal{A}$.
- $t = uv$. Par inversion, il existe $V$, $W$, $s$, $V'$ et $W'$ et $s'$ tels que $\Gamma \vdash u : (x : V)W$, $\Gamma \vdash W\{x \mapsto v\} : s$, $W\{x \mapsto v\} \ \mathbb{C}_\Gamma^* \ s_1$, $\Gamma' \vdash u : (x : V')W'$, $\Gamma' \vdash W'\{x \mapsto v\} : s'$ et $W'\{x \mapsto v\} \ \mathbb{C}_{\Gamma'}^* \ T$. Par correction de la conversion, $\Gamma \vdash s_1 : s$ et $\Gamma' \vdash T : s'$. Par inversion et préservation des sortes, $(s_1, s) \in \mathcal{A}$. Par séparation, $s = s'$ et $s' = s_2$. Donc, $(s_1, s_2) \in \mathcal{A}$. ■

**Remarque 49 (Classes de typage)**

Avec le lemme de correction des types, nous avons vu qu'un terme typable est nécessairement dans l'un des ensembles $\mathbb{T}_i^s$ où $i \in \{0, 1\}$ et $s \in \mathcal{S}$. Avec les lemmes de séparation et de classification précédents, on peut décrire avec précision les liens entre ces ensembles.

Dans un TSM injectif, l'ensemble des axiomes $\mathcal{A}$ est nécessairement une réunion disjointe de "chaînes" maximales, c'est-à-dire d'ensembles $\mathcal{A}'$ tels que :

– si $(s_1, s_2) \in \mathcal{A}'$ et $(s_2, s_3) \in \mathcal{A}$ alors $(s_2, s_3) \in \mathcal{A}'$,
– si $(s_2, s_3) \in \mathcal{A}'$ et $(s_1, s_2) \in \mathcal{A}$ alors $(s_1, s_2) \in \mathcal{A}'$.

Par exemple, dans le cas où $\mathcal{A}$ est fini, une chaîne maximale est de la forme $\{(s_1, s_2), (s_2, s_3), \ldots, (s_n, s_{n+1})\}$ avec $s_1, \ldots, s_n$ distincts deux à deux. Pour une telle chaîne, on obtient les $n$ classes de typage $\mathbb{T}_1^{s_1}$, $\mathbb{T}_1^{s_2}$, $\ldots$, $\mathbb{T}_1^{s_n}$, plus les deux classes $\mathbb{T}_1^{s_{n+1}}$ et $\mathbb{T}_0^{s_{n+1}}$ si $s_{n+1}$ est distinct des autres $s_i$. Finalement, une sorte $s$ qui ne fait partie d'aucun axiome donne lieu à deux classes, $\mathbb{T}_1^s$ et $\mathbb{T}_0^s$.

## 4.4 RTS confluents

Ci-après, nous montrons des résultats de dépendance des types par rapport aux variables ou aux symboles. Le premier résultat, la dépendance par rapport aux variables, est plus connu sous le nom de Lemme de Renforcement. Nous présentons une preuve de ce lemme dans le cas d'un RTS fonctionnel (et confluent) inspirée de celle de H. Geuvers et M.-J. Nederhof [57]. L. S. van Benthem Jutting [114] a montré que tout PTS vérifie cette propriété. Il serait intéressant d'examiner cette preuve pour l'adapter aux RTS.

**Lemme 50 (Dépendance par rapport aux variables)** Dans un RTS confluent et fonctionnel,
si $\Delta, z : V, \Delta' \vdash t : T$ et $z \notin \mathrm{FV}(\Delta', t)$ alors il existe $T'$ tel que $T \to^* T'$ et $\Delta, \Delta' \vdash t : T'$.

**Preuve.** Par récurrence sur $\Delta, z : V, \Delta' \vdash t : T$.

**(ax)** Impossible.

**(symb)** Soit $\Gamma = \Delta, z : V, \Delta'$. Montrons que $U\gamma$ lui-même fait l'affaire. Par hypothèse de récurrence, pour tout $i$, il existe $T_i'$ tel que $T_i\gamma \to^* T_i'$ et $\Delta, \Delta' \vdash t_i : T_i'$. Montrons que $\gamma : \Gamma_f \to \Delta, \Delta'$. Soit $\gamma_i = \{x_1 \mapsto t_1, \ldots, x_i \mapsto t_i\}$ et $\Gamma_i = x_1 : T_1, \ldots, x_i : T_i$. Montrons que $\gamma_i : \Gamma_i \to \Delta, \Delta'$ par récurrence sur $i$. Pour $i = 0$, il n'y a rien à montrer. Supposons alors que $\gamma_i : \Gamma_i \to \Delta, \Delta'$ et montrons que $\gamma_{i+1} : \Gamma_{i+1} \to \Delta, \Delta'$. Comme $\vdash \tau_f : s$, par inversion, pour tout $j$, il existe $s_j$ tel que $\Gamma_{j-1} \vdash T_j : s_j$. Ainsi, par le lemme d'environnement, $\mathrm{FV}(T_j) \subseteq \{x_1, \ldots, x_{j-1}\}$. Donc, pour tout $j \leq i + 1$, $T_j\gamma_{i+1} = T_j\gamma_i$. Ainsi, $\gamma_{i+1} : \Gamma_i \to \Delta, \Delta'$ et il ne nous reste plus qu'à montrer que $\Delta, \Delta' \vdash t_{i+1} : T_{i+1}\gamma_i$. Nous avons $\Delta, \Delta' \vdash t_{i+1} : T_{i+1}'$. Comme $\Gamma_i \vdash T_{i+1} : s_{i+1}$ et $\gamma_i : \Gamma_i \to \Delta, \Delta'$, par substitution, $\Delta, \Delta' \vdash T_{i+1}\gamma_i : s_{i+1}$. Donc, par (conv), $\Delta, \Delta' \vdash t_{i+1} : T_{i+1}\gamma_i$ et $\gamma_{i+1} : \Gamma_{i+1} \to \Delta, \Delta'$. Finalement, $\gamma = \gamma_n : \Gamma_f \to \Delta, \Delta'$. Comme $\Gamma_f \vdash U : s$, par substitution, $\Delta, \Delta' \vdash U\gamma : s$.

**(var)** Soit $\Gamma = \Delta, z : V, \Delta'$. Par hypothèse de récurrence, $\Delta, \Delta' \vdash T : s$. Donc, par (var), $\Delta, \Delta', x : T \vdash x : T$.

**(weak)** Si $z = x$, $T$ lui-même fait l'affaire puisque $\Gamma \vdash t : T$. Sinon, soit $\Gamma = \Delta, z : V, \Delta'$. Par hypothèse de récurrence, il existe $T'$ tel que $T \to^* T'$, $\Delta, \Delta' \vdash t : T'$ et $\Delta, \Delta' \vdash U : s$. Donc, par (weak), $\Delta, \Delta', x : U \vdash t : T'$.

**(prod)** Soit $\Gamma = \Delta, z : V, \Delta'$. Par hypothèse de récurrence, $\Delta, \Delta' \vdash T : s_1$ et $\Delta, \Delta', x : T \vdash U : s_2$. Donc, par (prod), $\Delta, \Delta' \vdash (x : T)U : s_3$.

**(abs)** Soit $\Gamma = \Delta, z{:}V, \Delta'$. Par hypothèse de récurrence, il existe $U'$ tel que $U \to^* U'$ et $\Delta, \Delta', x : T \vdash u : U'$. Montrons que $\Delta, \Delta' \vdash (x : T)U' : s$. Alors, par (abs), $\Delta, \Delta' \vdash [x : T]u : (x : T)U'$. Comme $\Gamma \vdash (x : T)U : s$, par inversion, il existe $(s_1, s_2, s) \in \mathcal{B}$ tel que $\Gamma \vdash T : s_1$ et $\Gamma, x : T \vdash U : s_2$. Comme $z \notin \mathrm{FV}(T)$, par hypothèse de récurrence, $\Delta, \Delta' \vdash T : s_1$. Comme $\Delta, \Delta', x : T \vdash u : U'$, par correction des types, soit $U' = s'$ soit $\Delta, \Delta', x{:}T \vdash U' : s'$. Supposons que $U' = s'$. Comme $\Gamma, x : T \vdash U : s_2$, par préservation du typage, $\Gamma, x : T \vdash U' : s_2$. Donc $(s', s_2) \in \mathcal{A}$ et $\Delta, \Delta', x{:}T \vdash U' : s_2$. Si maintenant $\Delta, \Delta', x{:}T \vdash U' : s'$ alors, par convertibilité des types, $s' = s_2$. Ainsi, dans tous les cas, $\Gamma, x{:}T \vdash U' : s_2$. Donc, par (prod), $\Delta, \Delta' \vdash (x{:}T)U' : s$.

**(app)** Soit $\Gamma = \Delta, z{:}V, \Delta'$. Par hypothèse de récurrence, il existe $U'_1$, $U'_2$ et $V$ tels que $U \to^* U'_1$, $U \to^* U'_2$, $V \to^* V'$, $\Delta, \Delta' \vdash t : (x : U'_1)V'$ et $\Delta, \Delta' \vdash u : U'_2$. Par confluence, il existe $U''$ tel que $U'_1 \to^* U''$ et $U'_2 \to^* U''$. Par préservation du typage, $\Delta, \Delta' \vdash t : (x : U'')V'$ et $\Delta, \Delta' \vdash u : U''$. Donc, par (app), $\Delta, \Delta' \vdash tu : V'\{x \mapsto u\}$.

**(conv)** Par hypothèse de récurrence, il existe $T''$ tel que $T \to^* T''$ et $\Delta, \Delta' \vdash t : T''$. Par confluence, il existe $T'''$ tel que $T'' \to^* T'''$ et $T' \to^* T'''$. Par préservation du typage, $\Delta, \Delta' \vdash t : T'''$. ■

**Corollaire 51** Dans un RTS confluent et fonctionnel, si $\Delta, z : V, \Delta' \vdash t : T$ et $z \notin \mathrm{FV}(\Delta', t, T)$ alors $\Delta, \Delta' \vdash t : T$.

**Preuve.** D'après le lemme, il existe $T'$ tel que $T \to^* T'$ et $\Delta, \Delta' \vdash t : T'$. Par correction des types, soit $T$ est une sorte maximale et $T' = T$, soit $\Delta, z{:}V, \Delta' \vdash T : s$. Alors, d'après le lemme, $\Delta, \Delta' \vdash T : s$. Donc, par (conv), $\Delta, \Delta' \vdash t : T$. ■

**Lemme 52 (Permutation forte)** Si $\Gamma, y : A, z : B, \Gamma' \vdash t : T$ et $y \notin \mathrm{FV}(B)$ alors $\Gamma, z{:}B, y{:}A, \Gamma' \vdash t : T$.

**Preuve.** Soit $\Delta = \Gamma, y : A, z : B, \Gamma'$ et $\Delta' = \Gamma, z : B, y : A, \Gamma'$. D'après le lemme de transitivité, il suffit de montrer que $\Delta'$ est valide et que $\Delta' \vdash \Delta$. Et pour cela, il suffit de montrer que $\Delta'$ est valide. Par le lemme d'environnement, nous avons $\Gamma \vdash A : s$ et $\Gamma, y{:}A \vdash B : s'$. D'après le lemme précédent, $\Gamma \vdash B : s'$. Donc, $\Gamma, z{:}B$ est valide et, par affaiblissement, $\Gamma, z : B, y : A$ aussi. Supposons que $\Gamma' = \vec{x} : \vec{T}$ et posons $\Delta_i = \Gamma, y : A, z : B, x_1 : T_1, \ldots, x_i : T_i$ et $\Delta'_i = \Gamma, z : B, y : A, x_1 : T_1, \ldots, x_i : T_i$. Montrons par récurrence sur $i$ que $\Delta'_i$ est valide. Nous avons montré que $\Delta'_0$ est valide. Supposons que $\Delta'_i$ soit valide. Par le lemme d'environnement, $\Delta_i \vdash T_{i+1} : s_{i+1}$. Comme $\Delta'_i \vdash \Delta_i$, $\Delta'_i \vdash T_{i+1} : s_{i+1}$ et $\Delta'_{i+1}$ est valide. Donc, $\Delta'$ est valide et $\Delta' \vdash t : T$.

■

**Définition 53 (Compatibilité par rapport à un pré-ordre)** Soit $\geq$ un pré--ordre sur $\mathcal{F}$. Étant donné un symbole $g$, nous désignerons par $\vdash_g$ la relation de typage du RTS dont les symboles sont strictement inférieurs à $g$.

– $\to$ est *compatible* avec $\geq$ si, pour tout symbole $g$ et tout terme $t, t'$, dès lors que tous les symboles de $t$ sont strictement inférieurs à $g$ et que $t \to t'$, alors tous les symboles de $t'$ sont strictement inférieurs à $g$.

– $\tau$ est *compatible* avec $\geq$ si, pour tout $g$, tous les symboles de $\tau_g$ sont inférieurs ou égaux à $g$.

**Lemme 54 (Dépendance par rapport aux symboles)** Soit un RTS confluent et fonctionnel et $\geq$ un pré-ordre sur $\mathcal{F}$ tel que $\to$ et $\tau$ soient compatibles avec $\geq$. Si $\Gamma \vdash t : T$ et les symboles de $\Gamma$ et $t$ sont strictement inférieurs à $g$ alors il existe $T'$ tel que $T \to^* T'$ et $\Gamma \vdash_g t : T'$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$.

**(ax)** Immédiat.

**(symb)** Montrons que $\Gamma \vdash_g f(\vec{t}) : U\gamma$. Par hypothèse de récurrence, pour tout $i$, il existe $T'_i$ tel que $T_i\gamma \to^* T'_i$ et $\Gamma \vdash_g t_i : T'_i$. Montrons que $\gamma : \Gamma_f \to \Gamma$ dans $\vdash_g$. Soit $\gamma_i = \{x_1 \mapsto t_1, \ldots, x_i \mapsto t_i\}$ et $\Gamma_i = x_1 : T_1, \ldots, x_i : T_i$. Montrons que $\gamma_i : \Gamma_i \to \Gamma$ par récurrence sur $i$. Pour $i = 0$, il n'y a rien à montrer. Supposons alors que $\gamma_i : \Gamma_i \to \Gamma$ et montrons que $\gamma_{i+1} : \Gamma_{i+1} \to \Gamma$. Comme $\tau$ est compatible avec $\geq$, par hypothèse de récurrence, $\vdash_g \tau_f : s$. Par inversion, pour tout $j$, il existe $s_j$ tel que $\Gamma_{j-1} \vdash_g T_j : s_j$. Ainsi, par le lemme des variables libres, $\mathrm{FV}(T_j) \subseteq \{x_1, \ldots, x_{j-1}\}$. Donc, pour tout $j \leq i+1$, $T_j\gamma_{i+1} = T_j\gamma_i$. Ainsi, $\gamma_{i+1} : \Gamma_i \to \Gamma$ et il ne nous reste plus qu'à montrer que $\Gamma \vdash_g t_{i+1} : T_{i+1}\gamma_i$. Nous avons $\Gamma \vdash_g t_{i+1} : T'_{i+1}$. Comme $\Gamma_i \vdash_g T_{i+1} : s_{i+1}$ et $\gamma_i : \Gamma_i \to \Gamma$, par substitution, $\Gamma \vdash_g T_{i+1}\gamma_i : s_{i+1}$. Donc, par (conv), $\Gamma \vdash_g t_{i+1} : T_{i+1}\gamma_i$ et $\gamma_{i+1} : \Gamma_{i+1} \to \Gamma$. Finalement, $\gamma = \gamma_n : \Gamma_f \to \Gamma$. Comme $\Gamma_f \vdash_g U : s$, par substitution, $\Gamma \vdash_g U\gamma : s$.

**(var)** Par hypothèse de récurrence, $\Gamma \vdash_g T : s$. Donc, par (var), $\Gamma, x : T \vdash_g x : T$.

**(weak)** Par hypothèse de récurrence, il existe $T'$ tel que $T \to^* T'$, $\Gamma \vdash_g t : T'$ et $\Gamma \vdash_g U : s$. Donc, par (weak), $\Gamma, x : U \vdash_g t : T'$.

**(prod)** Par hypothèse de récurrence, $\Gamma \vdash_g T : s_1$ et $\Gamma, x : T \vdash_g U : s_2$. Donc, par (prod), $\Gamma \vdash_g (x : T)U : s_3$.

**(abs)** Par hypothèse de récurrence, il existe $U'$ tel que $U \to^* U'$ et $\Gamma, x : T \vdash_g u : U'$. Montrons que $\Gamma \vdash_g (x : T)U' : s$. Alors, par (abs), $\Gamma \vdash_g [x : T]u : (x : T)U'$. Comme $\Gamma \vdash (x : T)U : s$, par inversion, il existe $(s_1, s_2, s) \in \mathcal{B}$ tel que $\Gamma \vdash T : s_1$ et $\Gamma, x : T \vdash U : s_2$. Par hypothèse de récurrence, $\Gamma \vdash_g T : s_1$. Comme $\Gamma, x : T \vdash_g u : U'$, par correction des types, soit $U' = s'$ soit $\Gamma, x : T \vdash_g U' : s'$. Supposons que $U' = s'$. Comme $\Gamma, x : T \vdash U : s_2$, par préservation du typage, $\Gamma, x : T \vdash U' : s_2$. Donc $(s', s_2) \in \mathcal{A}$ et $\Gamma, x : T \vdash_g U' : s_2$. Si maintenant $\Gamma, x : T \vdash_g U' : s'$ alors, par convertibilité des types, $s' = s_2$. Ainsi, dans tous les cas, $\Gamma, x : T \vdash_g U' : s_2$. Donc, par (prod), $\Gamma \vdash_g (x : T)U' : s$.

**(app)** Par hypothèse de récurrence, il existe $U'_1$, $U'_2$ et $V$ tels que $U \to^* U'_1$, $U \to^* U'_2$, $V \to^* V'$, $\Gamma \vdash_g t : (x : U'_1)V'$ et $\Gamma \vdash_g u : U'_2$. Par confluence, il existe $U''$ tel que $U'_1 \to^* U''$ et $U'_2 \to^* U''$. Par préservation du typage, $\Gamma \vdash_g t : (x : U'')V'$ et $\Gamma \vdash_g u : U''$. Donc, par (app), $\Gamma \vdash_g tu : V'\{x \mapsto u\}$.

**(conv)** Par hypothèse de récurrence, il existe $T''$ tel que $T \to^* T''$ et $\Gamma \vdash_g t : T''$. Par confluence, il existe $T'''$ tel que $T'' \to^* T'''$ et $T' \to^* T'''$. Par préservation du typage, $\Gamma \vdash_g t : T'''$. Comme $\to$ est compatible avec $\geq$, les symboles de $T'''$ sont strictement inférieurs à $g$. ■

# Chapitre 5

# Systèmes de Types Algébriques (ATS)

Maintenant, nous allons étudier le cas des RTS dont la relation de réduction est engendrée par la $\beta$-réduction et des règles de récriture. Avant cela, il convient de bien définir ce qu'on entend par récriture dans un contexte de typage fort et d'ordre supérieur comme c'est le cas dans les systèmes de types.

Au premier ordre, c'est-à-dire dans une algèbre de termes du premier ordre, une règle de récriture est généralement définie comme un couple $l \to r$ de termes tels que $l$ ne soit pas une variable et les variables qui apparaissent dans $r$ apparaissent également dans $l$ (sinon la récriture ne termine pas). Ensuite, on dit qu'un terme $t$ se récrit en $t'$ à la position $p$ s'il existe une substitution $\sigma$ telle que $t|_p = l\sigma$ (on dit que $t|_p$ "filtre" $l$) et $t' = t[r\sigma]_p$ ($t|_p = l\sigma$ est remplacé dans $t$ par $r\sigma$). Nous renvoyons le lecteur vers [43, 3] pour plus de détails sur la récriture du premier ordre.

Ici, nous allons considérer un mécanisme de récriture très similaire en restreignant les membres gauches des règles à appartenir à l'algèbre des termes engendrée par $\mathcal{F}$ et $\mathcal{X}$. Par contre, les membres droits pourront être quelconques. C'est un cas particulier de *Combinatory Reduction System* (CRS) [1] de W. Klop [78] pour lequel il n'est pas nécessaire d'utiliser du *filtrage d'ordre supérieur* à la Klop ou à la Miller [91, 95].

Cependant, nous avons montré dans [20] que le critère de terminaison que nous allons présenter dans le chapitre suivant peut s'adapter, dans le cadre du $\lambda$-calcul simplement typé, à la récriture avec filtrage d'ordre supérieur à la Klop ou à la Miller. Il serait donc intéressant de chercher à définir une notion de récriture avec filtrage d'ordre supérieur en présence de types polymorphes et dépendants, et d'étudier si nos conditions de terminaison peuvent également s'adapter à cette notion de récriture.

**Définition 55 (Termes algébriques)** Un terme est *algébrique* s'il est une variable ou s'il est de la forme $f(\vec{t})$ avec tous les $t_i$ eux-mêmes algébriques. Nous désignerons par $\mathcal{T}(\mathcal{F}, \mathcal{X})$ l'ensemble des termes algébriques bâtis à partir de $\mathcal{F}$ et

1. Pour cela, il suffit de traduire $[x{:}T]u$ par $\Lambda(T, [x]u)$, $(x{:}T)U$ par $\Pi(T, [x]U)$ et $uv$ par $@(u, v)$, où $\Lambda$, $\Pi$ et $@$ sont des symboles d'arité 2 et $[\_]\_$ est l'opérateur d'abstraction des CRS.

$\mathcal{X}$, et par $\mathbb{T}(\mathcal{F}, \mathcal{X})$ l'ensemble des termes algébriques typables.

**Définition 56 (Récriture)** Une *règle de récriture* est une paire $l \to r$ constituée de deux termes $l$ et $r$ tels que $l$ est un terme algébrique différent d'une variable et $\mathrm{FV}(r) \subseteq \mathrm{FV}(l)$. Une règle $l \to r$ est *linéaire-gauche* si aucune variable n'apparaît plus d'une fois dans $l$.

Étant donné un ensemble $\mathcal{R}$ de règles de récriture, la $\mathcal{R}$*-réduction* $\to_{\mathcal{R}}$ est la plus petite relation stable par substitution et contexte contenant $\mathcal{R}$. Un terme de la forme $l\sigma$ avec $l \to r \in \mathcal{R}$ et $\sigma$ une substitution est un $\mathcal{R}$*-radical*.

Étant donné un ensemble de symboles $\mathcal{G}$, nous désignerons par $\mathcal{R}_{\mathcal{G}}$ l'ensemble des règles qui *définissent* un symbole de $\mathcal{G}$, c'est-à-dire l'ensemble des règles dont le symbole de tête du membre gauche est un symbole de $\mathcal{G}$.

Un symbole $f$ est dit *constant* si $\mathcal{R}_{\{f\}} = \emptyset$, sinon il est dit (partiellement) *défini*. Nous désignerons par $\mathcal{CF}$ l'ensemble des symboles constants et par $\mathcal{DF}$ l'ensemble des symboles définis.

**Définition 57 (ATS)** Un *ATS* est un pré-RTS dont la relation de réduction $\to$ est de la forme $\to_{\mathcal{R}} \cup \to_{\beta}$ avec $\mathcal{R}$ un ensemble de règles de récriture.

Maintenant que nous avons introduit notre notion de récriture, nous pouvons nous demander à quelles conditions celle-ci est correcte vis-à-vis de la relation de typage.

Au premier ordre, dans les algèbres sortées, pour que la récriture préserve la sorte des termes, il suffit que, dans toutes les règles, le membre gauche et le membre droit aient la même sorte.

Transposée dans les systèmes de types, cette condition donne : il existe un environnement $\Gamma$ et un type $T$ tel que $\Gamma \vdash l : T$ et $\Gamma \vdash r : T$. Cette condition est celle qui a été retenue dans tous les travaux précédents combinant $\lambda$-calcul typé et récriture.

Cependant, cette condition présente un inconvénient important. En présence de types polymorphes ou dépendants, elle conduit à des règles fortement non linéaires. Cela a deux conséquences importantes. Premièrement, la récriture est fortement ralentie à cause des tests d'égalité nécessaires. Deuxièmement, il est plus difficile de montrer la confluence en présence de règles non linéaires.

Prenons l'exemple de la concaténation de deux listes polymorphes dans le Calcul des Constructions ($\mathcal{S} = \{\star, \Box\}$, $\mathcal{A} = \{(\star, \Box)\}$ et $\mathcal{B} = \{(s_1, s_2, s_3) \in \mathcal{S}^3 \mid s_2 = s_3\}$) :

- $list \in \mathcal{F}_1^{\Box}$ avec $\tau_{list} = \star \to \star$ le constructeur de type des listes polymorphes,
- $nil \in \mathcal{F}_1^{\star}$ avec $\tau_{nil} = (A{:}\star)list(A)$ la liste vide,
- $cons \in \mathcal{F}_3^{\star}$ avec $\tau_{cons} = (A : \star)A \to list(A) \to list(A)$ la fonction ajoutant un élément en tête d'une liste,
- $app \in \mathcal{F}_3^{\star}$ avec $\tau_{app} = (A{:}\star)list(A) \to list(A) \to list(A)$ la fonction de concaténation de deux listes.

Une définition naturelle de *app* est :

- $app(A, nil(A), \ell') \to \ell'$
- $app(A, cons(A, x, \ell), \ell') \to cons(A, x, app(A, \ell, \ell'))$

Cette définition vérifie bien la condition. Il suffit de prendre $\Gamma = A : \star, x : A, \ell : list(A), \ell' : list(A)$ et $T = list(A)$. Mais on peut se demander s'il est bien nécessaire de faire le test d'égalité quand on veut appliquer la seconde règle. En effet, si $app(A, cons(A', x, \ell), \ell')$ est bien typé, alors, par inversion, $cons(A', x, \ell)$ est de type $list(A)$ et, par inversion encore, $list(A')$ est convertible à $list(A)$. Ainsi, autoriser la réduction même si $A'$ est différent de $A$ ne semble pas poser de problème puisque $list(A')$ est convertible à $list(A)$.

En fait, ce qui est important, ce n'est pas que le membre gauche d'une règle soit typable, c'est que si une instance du membre gauche est typable alors l'instance correspondante du membre droit a le même type. Nous exprimons cela en disant qu'il doit exister un environnement $\Gamma$ dans lequel typer le membre droit, et une substitution $\rho$ qui envoie les variables du membres gauches n'appartenant pas à $\Gamma$ sur des termes typables dans $\Gamma$. Ainsi, on peut considérer les règles suivantes :

- $app(A, nil(A'), \ell') \to \ell'$
- $app(A, cons(A', x, \ell), \ell') \to cons(A, x, app(A, \ell, \ell'))$

en prenant $\Gamma = A : \star, x : A, \ell : list(A), \ell' : list(A)$ et $\rho = \{A' \mapsto A\}$. Dans [19], nous donnons 5 conditions, (S1) à (S5), qui doivent être vérifiées par la règle $l \to r$, $\Gamma$ et $\rho$. Supposons que $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{l}\}$. Alors, (S1) est $\mathrm{dom}(\rho) \subseteq \mathrm{FV}(l) \setminus \mathrm{dom}(\Gamma)$ et (S2) est $\Gamma \vdash l\rho : U\gamma\rho$. Bien que ces deux premières conditions soient souvent réalisées, elles ne sont pas nécessaires pour montrer la préservation du typage. C'est pourquoi, dans la définition suivante, elles ne figurent pas. Cependant, nous verrons que (S2) est nécessaire pour montrer la normalisation forte (voir Définition 81).

**Définition 58 (Règle bien typée)** Une règle $l \to r$ est *bien typée* s'il existe un environnement $\Gamma$ et une substitution $\rho$ tels que, si $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{l}\}$ alors :

**(S3)** $\Gamma \vdash r : U\gamma\rho$,
**(S4)** pour tout $\Delta$, $\sigma$ et $T$, si $\Delta \vdash l\sigma : T$ alors $\sigma : \Gamma \to \Delta$,
**(S5)** pour tout $\Delta$, $\sigma$ et $T$, si $\Delta \vdash l\sigma : T$ alors, pour tout $x$, $x\sigma \downarrow x\rho\sigma$.

Par la suite, nous écrirons $(l \to r, \Gamma, \rho) \in \mathcal{R}$ pour signifier que $l \to r$, $\Gamma$ et $\rho$ vérifient les hypothèses précédentes.

Un exemple, utilisant un type dépendant, nous est fourni par la concaténation de deux listes de longueurs données et la fonction *map* qui, à une fonction $f$ et une liste $a_1 \dots a_n$, associe la liste $f(a_1) \dots f(a_n)$ :

- $T \in \mathcal{F}_0^{\square}$ avec $\tau_T = \star$ une constante de type,
- $nat \in \mathcal{F}_0^{\square}$ avec $\tau_{nat} = \star$ le type des entiers naturels,
- $0 \in \mathcal{F}_0^{\star}$ avec $\tau_0 = nat$ la constante zéro,
- $s \in \mathcal{F}_1^{\star}$ avec $\tau_s = nat \to nat$ la fonction successeur,
- $+ \in \mathcal{F}_2^{\star}$ avec $\tau_+ = nat \to nat \to nat$ l'addition sur $nat$,
- $listn \in \mathcal{F}_1^{\square}$ avec $\tau_{listn} = nat \to \star$ le constructeur de types des listes de longueur $n$,

- $niln \in \mathcal{F}_0^\star$ avec $\tau_{niln} = listn(0)$ la liste vide,
- $consn \in \mathcal{F}_3^\star$ avec $\tau_{consn} = T \to (n : nat)listn(n) \to listn(s(n))$ la fonction ajoutant un élément en tête d'une liste,
- $appn \in \mathcal{F}_4^\star$ avec $\tau_{appn} = (n : nat)listn(n) \to (n' : nat)listn(n') \to listn(n + n')$ la fonction de concaténation de deux listes,
- $mapn \in \mathcal{F}_3^\star$ avec $\tau_{mapn} = (T \to T) \to (n : nat)listn(n) \to listn(n)$ la fonction qui, à une fonction $f : T \to T$ et une liste $a_1 \dots a_n$, associe la liste $f(a_1) \dots f(a_n)$,

où $+$, $appn$ et $mapn$ sont définies par :

- $+(0, n') \to n'$
- $+(s(n), n') \to s(n + n')$
- $appn(0, \ell, n', \ell') \to \ell'$
- $appn(p, consn(x, n, \ell), n', \ell') \to consn(x, n + n', appn(n, \ell, n', \ell'))$
- $mapn(f, 0, \ell) \to \ell$
- $mapn(f, p, consn(x, n, \ell)) \to consn(fx, n, mapn(f, n, \ell))$
- $mapn(f, p, appn(n, \ell, n', \ell')) \to appn(n, mapn(f, n, \ell), n', mapn(f, n', \ell'))$

Pour la seconde règle de $appn$, on peut prendre $\Gamma = x : T, n : nat, \ell : listn(n), n' : nat, \ell' : listn(n')$ et $\rho = \{p \mapsto s(n)\}$. Cela nous évite d'avoir à vérifier que $p$ est convertible à $s(n)$.

Pour la troisième règle de $mapn$, on peut prendre $\Gamma = f : T \to T, n : nat, \ell : listn(n), n' : nat, \ell' : listn(n')$ et $\rho = \{p \mapsto n + n'\}$. Cela nous évite d'avoir à vérifier que $p$ est convertible à $n + n'$.

Le lecteur pourra trouver d'autres exemples dans le Sous-chapitre 7.2.

**Lemme 59 (Correction de la récriture)** Si $\mathcal{R}$ est un ensemble de règles bien typées alors $\to_{\mathcal{R}}$ préserve le typage.

**Preuve.** Nous procédons comme pour la correction de $\to_\beta$. Nous ne détaillons que le cas (symb) :

$$\textbf{(symb)}\ \frac{\vdash \tau_f : s \quad \Gamma \text{ valide} \quad \Gamma \vdash t_1 : T_1\gamma \dots \Gamma \vdash t_n : T_n\gamma}{\Gamma \vdash f(\vec{t}) : U\gamma} \qquad \begin{array}{c}(f \in \mathcal{F}_n^s,\\ \tau_f = (\vec{x} : \vec{T})U,\\ \gamma = \{\vec{x} \mapsto \vec{t}\})\end{array}$$

Soit $(l \to r, \Gamma_0, \rho) \in \mathcal{R}$ avec $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$. Supposons que $t = l\sigma$ et montrons que $\Gamma \vdash r\sigma : U\gamma$. Par **(S4)**, $\sigma : \Gamma_0 \to \Gamma$. Par **(S3)**, $\Gamma_0 \vdash r : U\gamma_0\rho$. Donc, par substitution, $\Gamma \vdash r\sigma : U\gamma_0\rho\sigma$. Par **(S5)**, pour tout $x$, $x\rho\sigma$ et $x\sigma$ ont un réduit commun que nous appellerons $t_x$. Donc, en réduisant successivement dans $U\gamma_0\rho\sigma$ chaque $x\rho\sigma$ à $t_x$, et dans $U\gamma_0\sigma$ chaque $x\sigma$ á $t_x$ aussi, on obtient $U\gamma_0\rho\sigma \downarrow U\gamma_0\sigma$. Mais $U\gamma_0\sigma = U\gamma$ et, par inversion, il existe $s'$ tel que $\Gamma \vdash U\gamma : s'$. Donc, par (conv), $\Gamma \vdash r\sigma : U\gamma_0\sigma$. ■

**Théorème 60 (Admissibilité)** Un ATS logique dont toutes les règles de récriture sont bien typées est un RTS, *i.e.* sa relation de réduction préserve le typage.

**Preuve.** Pour $\to_\beta$, cela découle du fait que l'ATS est logique. Pour $\to_{\mathcal{R}}$, cela découle de la correction de la récriture. ■

Comment s'assurer que les conditions (S3), (S4) et (S5) sont bien remplies ? Dans toute leur généralité, elles sont sûrement indécidables. D'une part, on ne sait pas si $\vdash$ et $\downarrow$ sont décidables et, d'autre part, dans (S4) et (S5), nous quantifions arbitrairement sur $\Delta$, $\sigma$ et $T$. Il est donc nécessaire de faire des hypothèses supplémentaires. Nous envisageons successivement les trois conditions ci-après.

Voyons (S3). En pratique, les symboles et les règles qui les définissent sont souvent rajoutés les uns après les autres (ou par paquets, mais l'argument qui suit peut se généraliser). Soit $(\mathcal{F}, \mathcal{R})$ un système dans lequel $\vdash$ est décidable (par exemple, un système fonctionnel, confluent et fortement normalisant), $f \notin \mathcal{F}$ et $\mathcal{R}_f$ un ensemble de règles définissant $f$ et dont tous les symboles appartiennent à $\mathcal{F}' = \mathcal{F} \cup \{f\}$. Alors, dans $(\mathcal{F}', \mathcal{R})$, $\vdash$ est également décidable. On peut donc chercher à vérifier (S3) dans ce système. Cela ne nous paraît pas une grande restriction : il serait surprenant que le typage d'une règle nécessite l'utilisation de la règle elle-même !

Nous considérons maintenant (S4).

**Définition 61 (Type canonique et type dérivé)** Soit $t$ un terme de la forme $l\sigma$ avec $l = f(\vec{l})$ algébrique, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{l}\}$. Nous dirons que $U\gamma\sigma$ est le *type canonique* de $t$.

Soit $p \in \mathrm{Pos}(l)$ avec $p \neq \varepsilon$. Nous définissons le *type de* $t|_p$ *dérivé de* $t$, $\tau(t, p)$, de la manière suivante :

– si $p = i$ alors $\tau(t, p) = T_i\gamma\sigma$,
– si $p = iq$ et $q \neq \varepsilon$ alors $\tau(t, p) = \tau(t_i, q)$.

Le type de $t|_p$ dérivé de $t$ ne dépend en fait que du terme juste au-dessus de $t|_p$ dans $t$.

Le lemme suivant nous montre que le type canonique de $t$ et le type de $t|_p$ dérivé de $t$ sont bien des types pour $t$ et $t|_p$ respectivement.

**Lemme 62** Soit $t$ un terme de la forme $l\sigma$ avec $l = f(\vec{l})$ algébrique et $\Gamma \vdash t : T$, $V$ le type canonique de $t$ et $p \in \mathrm{Pos}(l)$ avec $p \neq \varepsilon$. Dans un TSM quelconque, $\Gamma \vdash t : V$ et $\Gamma \vdash t|_p : \tau(t, p)$.

**Preuve.** De $\Gamma \vdash t : T$, par inversion, on obtient immédiatement $\Gamma \vdash t : V$. Voyons $\Gamma \vdash t|_p : \tau(t, p)$ maintenant. Comme $p \neq \varepsilon$, nous avons $p = qi$, $t|_q$ de la forme $g(\vec{k}\sigma)$ avec $g(\vec{k})$ algébrique et $t|_q$ typable dans $\Gamma$. Supposons que $\tau_g = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{k}\}$. Alors, $\tau(t, p) = T_i\gamma\sigma$ et, par inversion, $\Gamma \vdash t|_p : T_i\gamma\sigma$. ■

**Lemme 63 (S4)** Soit $l \to r$ une règle et $\Gamma$ un environnement. Si, pour tout $x \in \mathrm{dom}(\Gamma)$, il existe $p_x \in \mathrm{Pos}(x, l)$ tel que $\tau(l, p_x) = x\Gamma$, alors (S4) est vérifiée.

**Preuve.** Supposons $\Delta \vdash l\sigma : T$. Comme $l$ est algébrique, par inversion, $\Delta \vdash x\sigma : \tau(l\sigma, p_x) = \tau(l, p_x)\sigma = x\Gamma\sigma$. ■

Pour (S5), par contre, nous n'avons pas de résultat général. Par le Lemme d'inversion, (S5) se ramène à un problème d'unification modulo $\downarrow^*$. La confluence de $\to$ (qui

implique $\downarrow^*=\downarrow$) peut donc être très utile. Malheureusement, il y a peu de résultats généraux sur la confluence de $\to_{\mathcal{R}} \cup \to_\beta$ (voir la discussion après la Définition 91). Par contre, on peut facilement montrer que la confluence locale est préservée.

**Lemme 64 (Confluence locale)** Si $\to_{\mathcal{R}}$ est localement confluente sur $\mathcal{T}(\mathcal{F},\mathcal{X})$ alors $\to=\to_{\mathcal{R}} \cup \to_\beta$ est localement confluente sur $\mathcal{T}$.

**Preuve.** Nous écrirons $t \to^p t'$ pour dire qu'il existe $u$ tel que $t|_p \to u$ et $t' = t[u']_p$ (réduction à la position $p$). Supposons que $t \to^p t_1$ et $t \to^q t_2$ et montrons par récurrence sur $t$ qu'il existe $t'$ tel que $t_1 \to^* t'$ et $t_2 \to^* t'$. Il y a trois cas :

- $p$ et $q$ n'ont pas de préfixe commun. Les réductions en $p$ et $q$ peuvent se faire en parallèle : $t_1 \to^q t'_1$, $t_2 \to^p t'_2$ et $t'_1 = t'_2$.
- $p = ip'$ et $q = iq'$. On peut conclure par hypothèse de récurrence sur $t|_i$.
- $p = \varepsilon$ ou $q = \varepsilon$. Quitte à échanger les rôles de $p$ et $q$, on peut supposer que $p = \varepsilon$. Il y a alors deux cas :
  - $t = [x{:}V]u\ v$ et $t_1 = u\{x \mapsto v\}$. On peut distinguer trois sous-cas :
    - $q = 11q'$ et $V \to^{q'} V'$. Alors $t' = t_1$ convient.
    - $q = 12q'$ et $u \to^{q'} u'$. Alors $t' = u'\{x \mapsto v\}$ convient.
    - $q = 2q'$ et $v \to^{q'} v'$. Alors $t' = u\{x \mapsto v'\}$ convient.
  - $t = l\sigma$, $l \to r \in \mathcal{R}$ et $t_1 = r\sigma$. Il existe un terme algébrique $u$ de taille maximale et une substitution $\theta$ tels que $t = u\theta$ et $x\theta = y\theta$ implique $x = y$ ($u$ et $\theta$ sont uniques au choix des variables près et $u$ a les mêmes non-linéarités que $t$). Comme les membres gauches des règles sont algébriques, $u = l\sigma'$ et $\sigma = \sigma'\theta$. Maintenant, on peut distinguer deux sous-cas :
    - $q \in \mathrm{Pos}(u)$. Comme les membres gauches des règles sont algébriques, nous avons $u \to_{\mathcal{R}} r\sigma'$ et $u \to_{\mathcal{R}} v$. Par confluence locale de $\to_{\mathcal{R}}$ sur $\mathcal{T}(\mathcal{F},\mathcal{X})$, il existe $u'$ tel que $r\sigma' \to^* u'$ et $v \to^* u'$. Ainsi, $t_1 = r\sigma'\theta \to^* u'\theta$ et $t_2 = v\theta \to^* u'\theta$.
    - $q = q_1q'$ et $u|_{q_1} = x$. Soit $q_2, \ldots, q_n$ les positions des autres occurrences de $x$ dans $u$. Si on réduit $t_2$ à chaque position $q_iq'$, on obtient un terme de la forme $l\sigma'\theta'$ où $\theta'$ est la substitution égale à $\theta$ sauf en $x$ où elle est égale au réduit de $x\theta$. Il suffit alors de prendre $t' = r\sigma'\theta'$. ■

# Chapitre 6

# Conditions de Normalisation Forte

Dans ce chapitre, nous allons donner des conditions de normalisation forte pour les ATS basé sur le Calcul des Constructions.

**Définition 65 (CAC)** Un *Calcul des Constructions Algébriques* (CAC) est un ATS $(\mathcal{S}, \mathcal{F}, \mathcal{X}, \mathcal{A}, \mathcal{B}, \tau, \mathcal{R})$ tel que $\mathcal{S} = \{\star, \Box\}$, $\mathcal{A} = \{(\star, \Box)\}$ et $\mathcal{B} = \{(s_1, s_2, s_3) \in \mathcal{S}^3 \mid s_2 = s_3\}$.

Un CAC est un ATS injectif et régulier.

## 6.1 Classification des termes

D'après les lemmes de séparation et de classification, les termes typables sont divisés en trois classes disjointes : $\mathbb{T}_0^\Box$, $\mathbb{T}_1^\Box$ et $\mathbb{T}_1^\star$. Pour les désigner, nous allons introduire des notations plus explicites.

**Définition 66 (Classes de typage)**
- Soit $\mathbb{K} = \mathbb{T}_0^\Box$ la classe des *types de prédicats*.
- Soit $\mathbb{P} = \mathbb{T}_1^\Box$ la classe des *prédicats*.
- Soit $\mathbb{O} = \mathbb{T}_1^\star$ la classe des *objets*.

Qu'un terme appartienne à l'une de ces trois classes peut être facilement décidé en introduisant les classes syntaxiques suivantes :

**Définition 67 (Classes syntaxiques)**
- La classe syntaxique $\mathcal{K}$ des *types de prédicats* :
  - $\star \in \mathcal{K}$,
  - si $x \in \mathcal{X}$, $T \in \mathcal{T}$ et $K \in \mathcal{K}$ alors $(x{:}T)K \in \mathcal{K}$.
- La classe syntaxique $\mathcal{P}$ des *prédicats* :
  - $\mathcal{X}^\Box \subseteq \mathcal{P}$,
  - si $x \in \mathcal{X}$, $T \in \mathcal{T}$ et $P \in \mathcal{P}$ alors $(x{:}T)P \in \mathcal{P}$ et $[x{:}T]P \in \mathcal{P}$,

  - si $P \in \mathcal{P}$ et $t \in \mathcal{T}$ alors $Pt \in \mathcal{P}$,
  - si $F \in \mathcal{F}_n^{\Box}$ et $t_1, \ldots, t_n \in \mathcal{T}$ alors $F(\vec{t}) \in \mathcal{P}$.
- La classe syntaxique $\mathcal{O}$ des *objets* :
  - $\mathcal{X}^\star \subseteq \mathcal{O}$,
  - si $x \in \mathcal{X}$, $T \in \mathcal{T}$ et $u \in \mathcal{O}$ alors $[x{:}T]u \in \mathcal{O}$,
  - si $u \in \mathcal{O}$ et $t \in \mathcal{T}$ alors $ut \in \mathcal{O}$,
  - si $f \in \mathcal{F}_n^\star$ et $t_1, \ldots, t_n \in \mathcal{T}$ alors $f(\vec{t}) \in \mathcal{O}$.

**Lemme 68** Les classes syntaxiques sont deux à deux disjointes et chaque classe de typage est incluse dans la classe syntaxique qui lui correspond : $\mathbb{K} \subseteq \mathcal{K}$, $\mathbb{P} \subseteq \mathcal{P}$ et $\mathbb{O} \subseteq \mathcal{O}$.

**Preuve.** Que les classes syntaxiques soit deux à deux disjointes découle immédiatement de leur définition. Montrons que si $\Gamma \vdash t : T$ alors $t$ appartient à la classe syntaxique correspondant à sa classe de typage, par récurrence sur $\Gamma \vdash t : T$. Nous suivons les notations des règles de typage.

**(ax)** Comme $\mathcal{A} = \{(\star, \Box)\}$, nous avons nécessairement $s_1 = \star$ et $s_2 = \Box$. Or, $\star \in \mathbb{K} \cap \mathcal{K}$.

**(symb)** Par inversion et régularité, $\Gamma \vdash U\gamma : s$. Donc, si $f \in \mathcal{F}^\star$ alors $f(\vec{t}) \in \mathbb{O} \cap \mathcal{O}$, et si $f \in \mathcal{F}^{\Box}$ alors $f(\vec{t}) \in \mathbb{P} \cap \mathcal{P}$.

**(var)** Si $x \in \mathcal{X}^\star$ alors $x \in \mathbb{O} \cap \mathcal{O}$, et si $x \in \mathcal{X}^{\Box}$ alors $x \in \mathbb{P} \cap \mathcal{P}$.

**(weak)** Par hypothèse de récurrence.

**(prod)** Par régularité, $U$ et $(x{:}T)U$ ont le même type. On peut donc conclure par hypothèse de récurrence sur $U$.

**(abs)** Par inversion et régularité, $(x : T)U$ et $U$ ont le même type. On peut donc conclure par hypothèse de récurrence sur $u$.

**(app)** Par inversion et régularité, $V\{x \mapsto u\}$ et $(x{:}U)V$ ont le même type. On peut donc conclure par hypothèse de récurrence sur $t$.

**(conv)** Par correction de la conversion, $T$ et $T'$ ont le même type. On peut donc conclure par hypothèse de récurrence. ■

## 6.2 Types inductifs et constructeurs

Jusqu'à maintenant, nous n'avons fait que très peu d'hypothèses sur les symboles et les règles qui les définissent. Or N. P. Mendler [89] a montré que dans le $\lambda$-calcul simplement typé, si un type et ses constructeurs ne vérifient pas une certaine condition de positivité, alors autoriser des définitions par récursion sur de tels types permet de définir des termes bien typés qui ne sont pas normalisants. Il a également montré que, réciproquement, si tous les types sont positifs alors les définitions par récursion sont fortement normalisantes.

N. P. Mendler dit qu'un type de base $T$ apparaît positivement dans un type $U$ si toutes les occurrences de $T$ dans $U$ sont à gauche d'un nombre pair de $\rightarrow$. Enfin, il dit qu'un type $T$ est positif si $T$ apparaît positivement dans le type de chacun des arguments de ses constructeurs. Ainsi, les types usuels tels que le type des entiers naturels ou le type des listes d'entiers naturels sont des types positifs.

Voyons un exemple de type $T$ non positif. Soit $U$ un type. Supposons que $T$ ait pour constructeur le symbole $c$ de type $(T \to U) \to T$. Alors, $T$ n'est pas positif car $T$ apparaît à une position négative dans $T \to U$. Considérons maintenant la fonction $p$ de type $T \to (T \to U)$ définie par la règle $p(c(x)) \to x$. Si on prend $\omega : T \to U = \lambda x.p(x)x$, alors on vérifie aisément que $\omega c(\omega) : U$ n'est pas normalisant :

$$\omega c(\omega) \ \to_\beta \ p(c(\omega))c(\omega) \ \to_{\mathcal{R}} \ \omega c(\omega) \ \to_\beta \ \ldots$$

Dans le cas où $U = \star$, on peut interpréter cela comme le Théorème de Cantor, à savoir : il n'y a pas de surjection d'un ensemble $T$ dans l'ensemble de ses parties $T \to \star$. Dans cette interprétation, $p$ est l'injection naturelle entre $T$ et $T \to \star$. Dire que $p$ est surjective est équivalent à dire (avec l'axiome du choix) qu'il existe $c$ tel que $p \circ c$ soit l'identité, c'est-à-dire, tel que $p(c(x)) \to x$. Dans [46], G. Dowek montre qu'une telle hypothèse est incohérente. Ici, nous montrons le lien avec la non-normalisation des types non-positifs.

N. P. Mendler donne également une condition, la positivité forte, dans le cas de types dépendants et polymorphes. Une notion similaire mais plus restreinte, la positivité stricte, est utilisée par T. Coquand et C. Paulin pour le Calcul des Constructions Inductives [36]. Enfin, le travail de N. P. Mendler a été étendu par R. Matthes [85, 86].

Nous introduisons ci-après la notion plus générale de *structure inductive admissible*. En particulier, nous ne considérerons pas qu'un constructeur soit constant : il sera possible d'avoir des règles de récriture sur les constructeurs également. Cela nous permettra d'avoir des types quotients tels que le type *int* des entiers relatifs qu'on peut définir de la manière suivante :

- $int \in \mathcal{F}_0^\square$ avec $\tau_{int} = \star$ le type des entiers relatifs,
- $0 \in \mathcal{F}_0^\star$ avec $\tau_0 = int$ la constante zéro,
- $s \in \mathcal{F}_1^\star$ avec $\tau_s = int \to int$ la fonction successeur,
- $p \in \mathcal{F}_1^\star$ avec $\tau_p = int \to int$ la fonction prédécesseur,

où $s$ et $p$ sont définis par :

- $s(p(x)) \to x$
- $p(s(x)) \to x$

**Définition 69 (Constructeurs)** Soit $C$ un symbole de prédicat constant. Un symbole $f$ est *constructeur* de $C$ si $\tau_f$ est de la forme $(\vec{y} : \vec{U})C(\vec{v})$ avec $\alpha_f = |\vec{y}|$. Nous désignerons par $\mathcal{C}o(C)$ l'ensemble des constructeurs de $C$.

Notre notion de constructeur non seulement inclut les constructeurs habituels, constants, mais également tout symbole permettant de produire des termes de type $C$. Par exemple :

- $+ \in \mathcal{F}_2^\star$ avec $\tau_+ = int \to int \to int$ la fonction d'addition sur *int*,
- $\times \in \mathcal{F}_2^\star$ avec $\tau_\times = int \to int \to int$ la fonction de multiplication sur *int*,

ou, dans le cas des listes polymorphes :

– $app \in \mathcal{F}^\star_3$ avec $\tau_{app} = (A{:}\star)list(A) \to list(A) \to list(A)$ la fonction de concaténation.

Remarquons que, si un symbole de prédicat doit avoir des constructeurs, alors son arité ne peut pas être prise n'importe comment :

**Définition 70 (Arité maximale)** Nous dirons qu'un symbole de prédicat $F$ est *d'arité maximale* si $\tau_F = (\vec{x} : \vec{T})\star$ et $\alpha_F = |\vec{x}|$.

**Lemme 71** Soit $C$ un symbole de prédicat constant et $c$ un constructeur de $C$. Dans un CAC logique, si $\vdash \tau_C : \Box$ et $\vdash \tau_c : s$ alors $s = \star$ et $C$ est d'arité maximale.

**Preuve.** Supposons que $\tau_C = (\vec{x} : \vec{V})W$ et $\tau_c = (\vec{y} : \vec{U})C(\vec{v})$. Posons $\gamma = \{\vec{x} \mapsto \vec{v}\}$. Comme $\Box$ est une sorte maximale et $\vdash \tau_C : \Box$, d'après le lemme sur les sortes maximales, $W$ est de la forme $(\vec{x'} : \vec{V'})\star$. Maintenant, de $\vdash \tau_c : s$, par inversion et régularité, on déduit que $\Gamma_c \vdash C(\vec{v}) : s$, $\Gamma_c \vdash C(\vec{v}) : W\gamma$ et $W\gamma \ \mathbb{C}^*_{\Gamma_c} \ s$. Comme $\Gamma_c \vdash W\gamma : \Box$, par correction de la conversion, $\Gamma_c \vdash s : \Box$ et, par inversion, $s = \star$. Donc, d'après le lemme sur les sortes maximales, $|\vec{x'}| = 0$ et $W = \star$. ■

**Définition 72 (Structure inductive)** Une *structure inductive* est donnée par :
- un pré-ordre $\geq_{\mathcal{C}}$ sur $\mathcal{CF}^\Box$ dont la partie stricte, $>_{\mathcal{C}}$, est bien fondée ;
- pour chaque symbole de prédicat constant $C$, un ensemble $\mathrm{Ind}(C) \subseteq \{1, \ldots, \alpha_C\}$ dit des *positions inductives* de $C$ tel que si $\tau_C = (\vec{x} : \vec{T})\star$ et $i \in \mathrm{Ind}(C)$ alors $x_i \in \mathcal{X}^\Box$ ;
- pour chaque constructeur $c$, un ensemble $\mathrm{Acc}(c) \subseteq \{1, \ldots, \alpha_c\}$ dit des *positions accessibles* de $c$.

Les positions accessibles permettent de désigner les arguments des constructeurs qu'on veut utiliser dans les membres droits des règles de récriture. Les positions inductives permettent de désigner les paramètres en lesquels un constructeur de type doit être monotone.

**Définition 73 (Positions positives et négatives)** Soit $T \in \mathcal{T} \setminus \mathcal{O}$. L'ensemble des *positions positives* de $T$, $\mathrm{Pos}^+(T)$, et l'ensemble des *positions négatives* de $T$, $\mathrm{Pos}^-(T)$, sont simultanément définis par récurrence sur la structure de $T$ de la manière suivante :
– $\mathrm{Pos}^+(s) = \mathrm{Pos}^+(F(\vec{t})) = \mathrm{Pos}^+(X) = \varepsilon$,
– $\mathrm{Pos}^-(s) = \mathrm{Pos}^-(F(\vec{t})) = \mathrm{Pos}^-(X) = \emptyset$,
– $\mathrm{Pos}^\delta((x{:}V)W) = 1.\mathrm{Pos}^{-\delta}(V) \cup 2.\mathrm{Pos}^\delta(W)$,
– $\mathrm{Pos}^\delta([x{:}V]W) = 1.\mathrm{Pos}(V) \cup 2.\mathrm{Pos}^\delta(W)$,
– $\mathrm{Pos}^\delta(Vu) = 1.\mathrm{Pos}^\delta(V) \cup 2.\mathrm{Pos}(u)$,
– $\mathrm{Pos}^\delta(VU) = 1.\mathrm{Pos}^\delta(V)$,
– $\mathrm{Pos}^+(C(\vec{t})) = \{\varepsilon\} \cup \bigcup \{i.\mathrm{Pos}^+(t_i) \mid i \in \mathrm{Ind}(C)\}$,
– $\mathrm{Pos}^-(C(\vec{t})) = \bigcup \{i.\mathrm{Pos}^-(t_i) \mid i \in \mathrm{Ind}(C)\}$,

où $\delta \in \{-, +\}$, $-+ = -$, $-- = +$ (règle habituelle des signes). L'ensemble des *positions neutres* de $T$ est $\mathrm{Pos}^0(T) = \mathrm{Pos}^+(T) \cap \mathrm{Pos}^-(T)$. L'ensemble des *positions non-neutres* de $T$ est $\mathrm{Pos}^{\neq 0}(T) = (\mathrm{Pos}^+(T) \cup \mathrm{Pos}^-(T)) \setminus \mathrm{Pos}^0(T)$.

Les positions positives et négatives ne forment pas deux ensembles disjoints. Leur intersection forme les positions neutres. Par exemple, toutes les positions de $u$ dans $Vu$ ou toutes les positions de $V$ dans $[x:V]W$ sont neutres. Nous verrons dans le Sous-chapitre 8.3 que ces sous-termes ne prennent pas part á l'interprétation d'un type.

Dans [19], nous donnons 6 conditions, (I1) à (I6), pour définir ce qu'est une structure inductive admissible. Mais il est en fait possible d'éliminer la condition (I1) en modifiant légèrement la condition (I2). C'est pourquoi, dans la définition suivante, (I1) ne figure pas et (I2) est placé après (I6).

**Définition 74 (Structures inductives admissibles)** Une structure inductive est *admissible* si, pour tout prédicat constant $C$, pour tout constructeur $c$ de type $(\vec{y}:\vec{U})C(\vec{v})$ et pour tout $j \in \mathrm{Acc}(c)$ :

**(I3)** $\forall D \in \mathcal{CF}^{\Box}, D =_{\mathcal{C}} C \Rightarrow \mathrm{Pos}(D,U_j) \subseteq \mathrm{Pos}^{+}(U_j)$ (les symboles équivalents à $C$ doivent être à des positions positives),

**(I4)** $\forall D \in \mathcal{CF}^{\Box}, D >_{\mathcal{C}} C \Rightarrow \mathrm{Pos}(D,U_j) \subseteq \mathrm{Pos}^{0}(U_j)$ (les symboles plus grands que $C$ doivent être à des positions neutres),

**(I5)** $\forall F \in \mathcal{DF}^{\Box}, \mathrm{Pos}(F,U_j) \subseteq \mathrm{Pos}^{0}(U_j)$ (les symboles de prédicat définis doivent être à des positions neutres),

**(I6)** $\forall Y \in \mathrm{FV}^{\Box}(U_j), \exists\, \iota_Y \leq \alpha_C, v_{\iota_Y} = Y$ (toute variable de prédicat de $U_j$ doit être paramètre de $C$),

**(I2)** $\forall Y \in \mathrm{FV}^{\Box}(U_j), \iota_Y \in \mathrm{Ind}(C) \Rightarrow \mathrm{Pos}(Y,U_j) \subseteq \mathrm{Pos}^{+}(U_j)$ (toute variable de prédicat de $U_j$ qui est paramètre inductif de $C$ doit être à une position positive).

Par exemple, $\mathrm{Ind}(list) = \{1\}$, $\mathrm{Acc}(nil) = \{1\}$ et $\mathrm{Acc}(cons) = \{1,2,3\}$ forme une structure inductive admissible. Soit maintenant :

– $tree \in \mathcal{F}_0^{\Box}$ avec $\tau_{tree} = \star$ le type des arbres à branchement fini et

– $node \in \mathcal{F}_1^{\star}$ avec $\tau_{node} = list(tree) \to tree$ son constructeur.

Comme $1 \in \mathrm{Ind}(list)$, $\mathrm{Ind}(tree) = \emptyset$ et $\mathrm{Acc}(node) = \{1\}$ forme une structure inductive admissible.

Autoriser des symboles plus grands ou définis n'a pas d'importance du moment qu'ils n'apparaissent qu'à des positions neutres car les sous-termes neutres ne sont pas pris en compte dans l'interprétation d'un type.

La condition (I6) signifie que les arguments-prédicats d'un constructeur doivent être des paramètres du type dont il est le constructeur. Une condition similaire apparaît dans les travaux de M. Stefanova [108] ("safeness") et D. Walukiewicz [117] ("$\star$-dependancy"). Par contre, dans le Calcul des Constructions Inductives (CIC) [99], il n'y a pas de telle restriction. Cependant, en examinant les règles de typage des schémas d'élimination, il apparaît qu'aucune fonction intéressante ne peut être définie sur un type ne vérifiant pas cette condition.

Prenons comme exemple le type des listes hétérogènes non vide (nous suivons la syntaxe CIC) :

– $listh = Ind(X : \star)\{C_1|C_2\}$ où $C_1 = (A{:}\star)(x{:}A)X$ et $C_2 = (A{:}\star)(x{:}A)X \to X$,
– $endh = Constr(1, listh)$,
– $consh = Constr(2, listh)$.

La règle de typage du schéma d'élimination non dépendant ($\text{Nodep}_{\star,\star}$) est :

$$\frac{\Gamma \vdash \ell : listh \quad \Gamma \vdash Q : \star \quad \Gamma \vdash f_1 : C_1\{listh, Q\} \quad \Gamma \vdash f_2 : C_2\{listh, Q\}}{\Gamma \vdash Elim(\ell, Q)\{f_1|f_2\} : Q}$$

où $C_1\{listh, Q\} = (A : \star)(x : A)Q$ et $C_2\{listh, Q\} = (A : \star)(x : A)listh \to Q \to Q$. Ainsi, $Q$, $f_1$ et $f_2$ doivent être typables dans $\Gamma$. Le résultat de $f_1$ ou $f_2$ ne peut donc pas dépendre de $A$ ou de $x$. Cela signifie qu'il est possible de calculer la longueur d'une telle liste par exemple, mais qu'on ne peut pas en extraire un élément. Or, la longueur d'une telle liste est une information qui peut sûrement être obtenue sans utiliser une telle structure de données.

Parmi l'ensemble des symboles de prédicat appartenant à une structure inductive, on peut distinguer différentes catégories.

**Définition 75 (Prédicats primitifs, basiques et strictement positifs)**
Un symbole de prédicat constant $C$ est :
– *primitif* si, pour tout $D =_{\mathcal{C}} C$, pour tout constructeur $d$ de type $\tau_d = (\vec{y} : \vec{U})D(\vec{w})$ et pour tout $j \in \text{Acc}(d)$, $U_j$ est soit de la forme $E(\vec{t})$ avec $E <_{\mathcal{C}} D$ et $E$ primitif, soit de la forme $E(\vec{t})$ avec $E =_{\mathcal{C}} D$.
– *basique* si, pour tout $D =_{\mathcal{C}} C$, pour tout constructeur $d$ de type $\tau_d = (\vec{y} : \vec{U})D(\vec{w})$ et pour tout $j \in \text{Acc}(d)$, si $E =_{\mathcal{C}} D$ apparaît dans $U_j$ alors $U_j$ est de la forme $E(\vec{t})$.
– *strictement positif* si, pour tout $D =_{\mathcal{C}} C$, pour tout constructeur $d$ de type $\tau_d = (\vec{y} : \vec{U})D(\vec{w})$ et pour tout $j \in \text{Acc}(d)$, si $E =_{\mathcal{C}} D$ apparaît dans $U_j$ alors $U_j$ est de la forme $(\vec{z} : \vec{V})E(\vec{t})$ et aucune occurrence de $D' =_{\mathcal{C}} D$ n'apparaît dans les $V_i$.

Il est aisé de voir qu'un prédicat primitif est basique et qu'un prédicat basique est strictement positif. Le type $listn$ des listes de longueur $n$ est primitif. Le type $list$ des listes polymorphes est basique mais non-primitif.

Les prédicats strictement positifs sont les prédicats qui sont autorisés dans le Calcul des Constructions Inductives. Comme exemple, on peut citer le type des arbres bien fondés ou ordinaux de Brouwer [1] :
– $ord \in \mathcal{F}_0^{\square}$ avec $\tau_{ord} = \star$ le type des ordinaux de Brouwer,
– $0 \in \mathcal{F}_0^{\star}$ avec $\tau_0 = ord$ l'ordinal zéro,
– $s \in \mathcal{F}_1^{\star}$ avec $\tau_s = ord \to ord$ l'ordinal successeur,
– $lim \in \mathcal{F}_1^{\star}$ avec $\tau_{lim} = (nat \to ord) \to ord$ l'ordinal limite.

1. Un terme de type $ord$ de la forme $lim(f)$ ne correspond pas forcément à un ordinal limite. Toutefois, si on choisit bien les fonctions $f$, on peut représenter un certain segment initial des ordinaux dénombrables.

Un autre exemple nous est donné par l'algèbre de processus qui utilise un opérateur de choix $\Sigma$ dépendant de la donnée qui lui est fournie [107] :

- $data \in \mathcal{F}_0^{\square}$ avec $\tau_{data} = \star$ le type des données,
- $proc \in \mathcal{F}_0^{\square}$ avec $\tau_{proc} = \star$ le type des processus,
- $\circ \in \mathcal{F}_2^{\star}$ avec $\tau_{\circ} = proc \to proc \to proc$ la mise en séquence de deux processus,
- $+ \in \mathcal{F}_2^{\star}$ avec $\tau_{+} = proc \to proc \to proc$ la mise en parallèle de deux processus,
- $\delta \in \mathcal{F}_0^{\star}$ avec $\tau_{\delta} = proc$ le processus bloquant (*deadlock* en anglais),
- $\Sigma \in \mathcal{F}_1^{\star}$ avec $\tau_{\Sigma} = (data \to proc) \to proc$ le choix d'un processus selon une donnée.

Un dernier exemple est celui des formules du calcul des prédicats du premier ordre :

- $term \in \mathcal{F}_0^{\square}$ avec $\tau_{term} = \star$ le type des termes,
- $form \in \mathcal{F}_0^{\square}$ avec $\tau_{form} = \star$ le type des formules,
- $\vee \in \mathcal{F}_2^{\star}$ avec $\tau_{\vee} = form \to form \to form$ le "ou" de deux formules,
- $\neg \in \mathcal{F}_1^{\star}$ avec $\tau_{\neg} = form \to form$ le "non" d'une formule,
- $\forall \in \mathcal{F}_1^{\star}$ avec $\tau_{\forall} = (term \to form) \to form$ la quantification universelle.

Pour l'instant, nous n'interdisons pas d'avoir des prédicats positifs non stricts. Toutefois, dans le sous-chapitre suivant, nous verrons que, dans leur état actuel, les conditions sur les règles de récriture ne permettent pas de définir des fonctions sur des prédicats positifs non stricts.

Les types positifs non stricts peuvent pourtant être utiles comme nous le montre cette exemple d'affichage en largeur des labels d'un arbre binaire à l'aide de continuations [87] :

- $tree \in \mathcal{F}_0^{\square}$ avec $\tau_{tree} = \star$ le type des arbres binaires,
- $L \in \mathcal{F}_1^{\star}$ avec $\tau_L = nat \to tree$ le constructeur "feuille",
- $N \in \mathcal{F}_3^{\star}$ avec $\tau_N = nat \to tree \to tree \to tree$ le constructeur "noeud",
- $list \in \mathcal{F}_0^{\square}$ avec $\tau_{list} = \star$ le type des listes d'entiers,
- $nil \in \mathcal{F}_0^{\star}$ avec $\tau_{nil} = list$ la liste vide,
- $cons \in \mathcal{F}_2^{\star}$ avec $\tau_{cons} = nat \to list \to list$ l'ajout d'un élément en tête,
- $cont \in \mathcal{F}_0^{\square}$ avec $\tau_{cont} = \star$ le type des continuations,
- $D \in \mathcal{F}_0^{\star}$ avec $\tau_D = cont$,
- $C \in \mathcal{F}_1^{\star}$ avec $\tau_C = ((cont \to list) \to list) \to cont$ les constructeurs de continuations,
- $@ \in \mathcal{F}_2^{\star}$ avec $\tau_{@} = cont \to (cont \to list) \to list$ l'application sur les continuations définie par :
- $@(D, g) \to g\ D$
- $@(C(f), g) \to f\ g$
- $ex \in \mathcal{F}_1^{\star}$ avec $\tau_{ex} = cont \to list$ l'itérateur sur les continuations défini par :

– $ex(D) \to nil$
– $ex(C(f)) \to f\ [k{:}cont]ex(k)$

– $br \in \mathcal{F}_2^\star$ avec $\tau_{br} = tree \to cont \to cont$,
– $br\_fst \in \mathcal{F}_1^\star$ avec $\tau_{br\_fst} = tree \to list$ la fonction d'affichage en largeur des labels définie par :
– $br\_fst(t) \to ex(br(t, D))$
– $br(L(x), k) \to C([g{:}cont \to list]cons(x, @(k, g)))$
– $br(N(x, s, t), k) \to C([g{:}cont \to list]cons(x, @(k, [m{:}cont]g\ br(s, br(t, m)))))$

Il est connu qu'une telle définition est fortement normalisante car elle peut être codée dans le $\lambda$-calcul polymorphe [85, 86]. Cependant, il n'est pas clair comment définir une condition syntaxique, un schéma, qui permette d'assurer la normalisation d'une telle définition. En effet, dans le membre droit de la seconde règle de $ex$, $ex$ n'est explicitement appliqué à aucun argument plus petit que $f$. Mais il ne pourra être appliqué qu'à des sous-termes de réduits de $f$. Malheureusement, pas tous les sous-termes d'un terme calculable ne sont *a priori* calculables (voir le Paragraphe 6.3.2).

## 6.3 Schéma Général

### 6.3.1 Récriture d'ordre supérieur

Nous allons maintenant nous intéresser aux règles de récriture. Quelles conditions peuvent nous permettre d'assurer la normalisation forte de $\to{=}\to_{\mathcal{R}} \cup \to_\beta$ ? Depuis les travaux de V. Breazu-Tannen et J. Gallier [24] et M. Okada [97] sur le $\lambda$-calcul simplement typé et le $\lambda$-calcul polymorphe, puis les travaux de F. Barbanera [4] sur le Calcul des Constructions et de D. Dougherty [44] sur le $\lambda$-calcul pur, il est bien connu que l'ajout de récriture du premier ordre au niveau objet à un $\lambda$-calcul typé préserve la normalisation forte. Cela vient du fait que la récriture du premier ordre ne peut pas créer de nouveaux $\beta$-radicaux. Nous montrerons que ce résultat, sous certaines conditions, s'étend également à la récriture au niveau des prédicats.

Cependant, il y a de nombreuses fonctions très utiles dont la définition par récriture ne rentre pas dans ce contexte de la récriture de premier ordre, soit par ce que certains de ses arguments ne sont pas de type primitif (la fonction *app* de concaténation de deux listes polymorphes), soit par ce que leur définition exploite les possibilités offertes par l'ordre supérieur. C'est le cas par exemple de la fonction *map* qui, à une fonction $f$ (argument d'ordre supérieur) et une liste $a_1 \dots a_n$ d'éléments, associe la liste $f(a_1) \dots f(a_n)$ :

– $map \in \mathcal{F}_4^\star$ avec $\tau_{map} = (A{:}\star)(B{:}\star)(A \to B) \to list(A) \to list(B)$,
– $map(A, B, f, nil(A')) \to nil(B)$
– $map(A, B, f, cons(A', x, \ell)) \to cons(B, fx, map(A, B, f, \ell))$
– $map(A, B, f, app(A', \ell, \ell')) \to app(B, map(A, f, \ell), map(A, f, \ell'))$

C'est également le cas des récurseurs ou principes de récurrence qu'on peut associer aux prédicats inductifs (le récurseur étant une version non dépendante du principe de récurrence). Voyons par exemple le cas du récurseur sur *nat* qui peut nous permettre de définir des fonctions par récurrence (mais, bien sûr, on préférera donner une définition directement à l'aide de règles de récriture !) :

- $natrec \in \mathcal{F}_4^\star$ avec $\tau_{natrec} = (A{:}\star)A \to (nat \to A \to A) \to nat \to A$,
- $natrec(A, x, f, 0) \to x$
- $natrec(A, x, f, s(n)) \to f\ n\ natrec(A, x, f, n)$
- $plus \in \mathcal{F}_0^\star$ avec $\tau_{plus} = nat \to nat \to nat$ l'addition sur les entiers défini à l'aide du récurseur :
- $plus \to [p{:}nat][q{:}nat]natrec(nat, p, [q'{:}nat][r{:}nat]s(r), q)$

et du principe de récurrence sur *nat*, qui nous permet de représenter des preuves par récurrence sur les entiers :

- $natind \in \mathcal{F}_4^\star$ avec $\tau_{natind} = (P : nat \to \star)P0 \to ((n : nat)Pn \to Ps(n)) \to (n : nat)Pn$,
- $natind(P, h_0, h_s, 0) \to h_0$
- $natrec(P, h_0, h_s, s(n)) \to h_s\, n\ natind(P, h_0, h_s, n)$

Les méthodes utilisées dans les travaux précédemment cités ne peuvent pas être appliquées à notre calcul car, d'une part, contrairement à la récriture du premier ordre, la récriture d'ordre supérieur peut créer des $\beta$-radicaux, et d'autre part, la récriture est incluse dans la règle de conversion (conv) ce qui permet à davantage de termes d'être typés.

Mais il existe d'autres méthodes, valables dans le $\lambda$-calcul simplement typé seulement ou dans des systèmes de types plus riches, pour montrer la terminaison de ce type de définitions :

- Le *Schéma Général*, initialement formulé par J.-P. Jouannaud et M. Okada [72] pour le $\lambda$-calcul polymorphe et étendu au Calcul des Constructions par F. Barbanera, M. Fernández et H. Geuvers [7], est essentiellement une extension du schéma de récursion primitive : dans le membre droit d'une règle $f(\vec{l}) \to r$, les appels à $f$ doivent se faire sur des sous-termes strictes de $\vec{l}$. Il permet de traiter des définitions au niveau objet de symboles simplement typés définis sur des types primitifs. Il a ensuite été reformulé et étendu aux types simples strictement positifs par J.-P. Jouannaud, M. Okada et moi-même pour le $\lambda$-calcul simplement typé [22] et le Calcul des Constructions [21].

- Le *Higher-Order Recursive Path Ordering* (HORPO) de J.-P. Jouannaud et A. Rubio [74] [2] est une extension aux termes du $\lambda$-calcul simplement typé de l'ordre RPO [100, 42] utilisé au premier ordre. Il a été récemment étendu par D. Walukiewicz [117] au Calcul des Constructions avec des symboles de type polymorphe

2. Signalons également les travaux précédents de C. Loria-Saenz et J. Steinbach [79], O. Lysne et J. Piris [82] et J.-P. Jouannaud et A. Rubio [75].

et dépendant, mais toujours au niveau objet et sur des types basiques. Le Schéma Général peut être vu comme une version non-récursive de HORPO.

- Il est également possible de chercher une interprétation pour chacun des symboles de telle sorte que l'interprétation d'un terme décroisse strictement quand on lui applique une règle. Cette méthode, introduite par J. van de Pol pour le $\lambda$-calcul simplement typé [115], étend à l'ordre supérieur la méthode des interprétations connue au premier ordre. C'est donc une méthode très puissante mais difficile à utiliser en pratique, d'une part parce que les interprétations sont bien sûr elles-mêmes d'ordre supérieur, et d'autre part parce qu'elle est très peu modulaire : rajouter de nouvelles règles ou de nouveaux symboles nécessite de trouver une nouvelle interprétation et de redémontrer sa stricte monotonie.

Pour traiter la récriture au niveau des prédicats, avec des symboles de type polymorphe et dépendant et des types strictement positifs, nous avons choisi d'étendre la méthode du Schéma Général. C'est cette extension que nous allons présenter dans le reste de ce sous-chapitre.

Cependant, le Schéma Général ne représente qu'une partie des conditions de normalisation forte, celle concernant la récriture d'ordre supérieur. Comme dans les travaux précédents [72, 7], une autre partie des conditions concerne la récriture du premier ordre que nous étendons ici au niveau des types.

### 6.3.2 Définition du schéma

À vrai dire, les conditions que nous allons présenter ci-après ne se comprennent pleinement qu'à la lumière de la méthode employée pour montrer la normalisation forte, à savoir la méthode de Tait et Girard des candidats de réductibilité [111, 62]. Grosso modo, cette méthode consiste à associer un ensemble de termes fortement normalisables, les termes *calculables*, à chaque type et à montrer que tout terme bien typé appartient à l'ensemble associé à son type.

L'idée du Schéma Général est alors de définir, à partir d'un membre gauche de règle $f(\vec{l})$, un ensemble de termes dont on peut prouver la calculabilité si les $l_i$ sont eux-mêmes calculables. Nous appelons cet ensemble la *clôture calculable* de $f(\vec{l})$. Pour montrer la terminaison d'une définition, il suffit alors de vérifier que, pour chaque règle, le membre droit appartient à la clôture calculable du membre gauche.

Pour construire la clôture calculable, dans un premier temps, nous définissons, parmi les sous-termes des $l_i$, un sous-ensemble dont les éléments sont calculables si les $l_i$ sont calculables. Nous appellerons ces sous-termes les sous-termes *accessibles*. Nous ne considérons qu'un sous-ensemble car pas tous les sous-termes d'un terme calculable sont *a priori* calculables. Alors, nous construirons la clôture calculable en appliquant à ces sous-termes accessibles des opérations préservant la calculabilité.

Pour pouvoir définir des fonctions intéressantes, parmi ces opérations, on doit trouver celle consistant à appliquer des arguments à $f$ (récursivité). Se pose alors le problème d'avoir un ordre bien fondé dans lequel les arguments décroissent strictement. La relation sous-terme strict $\triangleright$ (en fait restreinte aux sous-termes accessibles pour préserver la calculabilité) est suffisante pour traiter toutes les définitions sur

des types basiques. Dans l'exemple de *map*, $\ell$ et $\ell'$ sont des sous-termes stricts accessibles de $app(A', \ell, \ell')$. Par contre, sur les types strictement positifs, cela n'est pas suffisant comme le montre la définition suivante d'une "addition" sur les ordinaux de Brouwer :

- $+ \in \mathcal{F}_2^\star$ avec $\tau_+ = ord \to ord \to ord$,
- $+(x, 0) \to x$
- $+(x, s(y)) \to s(+(x, y))$
- $+(x, lim(f)) \to lim([n{:}nat] + (x, fn))$

Un autre exemple nous est donné par les règles de simplification de l'algèbre de processus *proc* [107] :

- $+(p, p) \to p$
- $+(p, \delta) \to p$
- ...
- $\circ(\Sigma(f), p) \to \Sigma([d{:}data] \circ (fd, p))$

C'est pourquoi, dans la suite, nous utiliserons deux ordres distincts. Le premier, $>_1$, servira pour les arguments de type basique. Le second, $>_2$, servira pour les arguments de type strictement positif.

Enfin, pour apporter plus de souplesse dans la comparaison des arguments des fonctions, à chaque symbole, nous associerons un *statut* qui décrit comment comparer deux séquences d'arguments à l'aide d'une combinaison simple de comparaisons lexicographiques et multi-ensembles [73].

**Définition 76 (Relations d'accessibilité)** Soit $c$ un constructeur de type $(\vec{y} : \vec{U})C(\vec{v})$, $\vec{u}$ des arguments de $c$, $\gamma = \{\vec{y} \mapsto \vec{u}\}$ et $j \in \mathrm{Acc}(c)$ une position accessible de $c$. Alors :

- $u_j : U$ est *faiblement accessible modulo* $\rho$ dans $c(\vec{u}) : T$, $c(\vec{u}) : T \ \rhd_1^\rho \ u_j : U$, si $T\rho = C(\vec{v})\gamma\rho$ et $U\rho = U_j\gamma\rho$.
- $u_j : U$ est *fortement accessible modulo* $\rho$ dans $c(\vec{u}) : T$, $c(\vec{u}) : T \ \rhd_2^\rho \ u_j : U$, si $T\rho = C(\vec{v})\gamma\rho$, $U\rho = U_j\gamma\rho$ et $U_j$ est de la forme $(\vec{x} : \vec{T})D(\vec{w})$.

Ces relations d'accessibilité vont nous servir à définir les ordres $>_1$ et $>_2$. Nous avons $\rhd_2^\rho \subseteq \rhd_1^\rho$. Pour des raisons techniques, nous prenons en compte non seulement les termes mais également leur type. Cela est dû au fait que nous ne sommes en mesure de montrer que deux types convertibles ont la même interprétation que si ces deux types sont calculables. Cela induit certaines restrictions sur les types des symboles.

En effet, l'accessibilité impose l'égalité (modulo l'application de $\rho$) entre les types canoniques et les types dérivés (voir Définition 61). Plus précisément, pour que $t : T \rhd_1^\rho u : U$, il faut que $T$ soit égal (modulo $\rho$) au type canonique de $t$ et $U$ au type de $u$ dérivé de $t$. Si de plus $u : U \rhd_1^\rho v : V$, on voit alors que $U$ doit aussi être égal (modulo $\rho$) au type canonique de $u$.

**Définition 77 (Précédence)** Une *précédence* est un pré-ordre $\geq_{\mathcal{F}}$ sur $\mathcal{F}$ dont la partie stricte $>_{\mathcal{F}}$ est bien fondée. Nous noterons par $=_{\mathcal{F}}$ sa relation d'équivalence associée.

**Définition 78 (Statuts)** Soit $(x_i)_{i\geq 1}$ une famille de variables.

- **Statut.** Un *statut* est un terme linéaire de la forme $lex(m_1, \ldots, m_k)$ avec $k \geq 1$ et chaque $m_i$ de la forme $mul(x_{k_1}, \ldots, x_{k_p})$ avec $p \geq 1$. L'*arité* d'un statut *stat* est le plus grand indice $i$ tel que $x_i$ apparaîsse dans *stat*.
- **Assignement de statuts.** Un *assignement de statuts* est une application *stat* qui, à chaque symbole $f$ d'arité $n > 0$ et de type $(\vec{x} : \vec{T})U$, associe un statut $stat_f = lex(\vec{m})$ d'arité inférieure ou égale à $n$ tel que :
  - si $x_i \in \mathrm{FV}(stat_f)$ alors $T_i$ est de la forme $C_f^i(\vec{u})$ avec $C_f^i$ un symbole de prédicat constant,
  - si $m_i = mul(x_{k_1}, \ldots, x_{k_p})$ alors $C_f^{k_1} =_{\mathcal{C}} \ldots =_{\mathcal{C}} C_f^{k_p}$.
- **Positions strictement positives.** Soit $f$ un symbole de statut $lex(\vec{m})$. Nous noterons par $SP(f)$ l'ensemble des $i$ tels que si $m_i = mul(x_{k_1}, \ldots, x_{k_p})$ alors $C_f^{k_1}$ (et donc aussi $C_f^{k_2}, \ldots, C_f^{k_p}$) est strictement positif.
- **Compatibilité d'un assignement.** Un assignement *stat* est *compatible* avec une précédence $\geq_{\mathcal{F}}$ si :
  - $f =_{\mathcal{F}} g$ implique $stat_f = stat_g$ et, pour tout $i$, $C_f^i =_{\mathcal{F}} C_g^i$.
- **Ordre engendré par un statut.** Soit $>$ un ordre sur les termes et $stat = lex(\vec{m})$ un statut d'arité $n$. L'*extension* par *stat* de $>$ aux séquences de termes de longueur au moins $n$ est l'ordre $>_{stat}$ défini par :
  - $\vec{u} >_{stat} \vec{v}$ si $\vec{m}\{\vec{x} \mapsto \vec{u}\}\ (>^m)_{\mathrm{lex}}\ \vec{m}\{\vec{x} \mapsto \vec{v}\}$,
  - $mul(\vec{u}) >^m mul(\vec{v})$ si $\{\vec{u}\} >_{\mathrm{mul}} \{\vec{v}\}$.

Par exemple, avec $stat = lex(mul(x_2), mul(x_1, x_3))$, $\vec{u} >_{stat} \vec{v}$ si $(\{u_2\}, \{u_1, u_3\})$ $(>_{\mathrm{mul}})_{\mathrm{lex}}$ $(\{v_2\}, \{v_1, v_3\})$. Une propriété importante de l'ordre $>_{stat}$ est qu'il est bien fondé dès lors que $>$ est bien fondé.

Nous allons maintenant définir les ordres $>_1$ et $>_2$.

**Définition 79 (Ordre sur les arguments d'un symbole)** Soit $(l \to r, \Gamma_0, \rho)$ une règle telle que $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$, $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$ et $stat_f = lex(\vec{m})$. Nous dirons que :

- $t : T >_1 u : U$ si $t : T\ (\triangleright_1^\rho)^+\ u : U$.
- $t : T >_2 u : U$ si :
  - $t$ est de la forme $c(\vec{t})$ avec $c$ un constructeur de type $(\vec{x} : \vec{T})C(\vec{v})$,
  - $u$ est de la forme $x\vec{u}$ avec $x \in \mathrm{dom}(\Gamma_0)$, $x\Gamma_0$ de la forme $(\vec{y} : \vec{U})D(\vec{w})$ et $D =_{\mathcal{C}} C$,
  - $t : T\ (\triangleright_2^\rho)^+\ x : V$ avec $V\rho = x\Gamma_0$.

Maintenant, nous définissons l'ordre $>$ sur les arguments de $f$ (en fait les paires argument-type $u : U$). C'est une adaptation de $>_{stat_f}$ où l'ordre utilisé est soit $>_1$ soit $>_2$ selon que l'argument est à une position strictement positive ou non. Supposons que $stat_f = lex(m_1, \ldots, m_k)$. Alors :

- $\vec{u} > \vec{v}$ si $\vec{m}\{\vec{x} \mapsto \vec{u}\}\ (>^1, \ldots, >^k)_{\mathrm{lex}}\ \vec{m}\{\vec{x} \mapsto \vec{v}\}$,

- $mul(\vec{u}) >^i mul(\vec{v})$ si $\{\vec{u}\}\ (>_{\phi(i)})_{\text{mul}}\ \{\vec{v}\}$ avec $\phi(i) = 2$ si $i \in SP(f)$ et $\phi(i) = 1$ sinon.

Par exemple, on vérifie aisément que, dans la troisième règle de l'addition sur les ordinaux, $lim(f) : ord >_2 fn : ord$. Pour cette règle, nous pouvons prendre $\Gamma_0 = x : ord, f : nat \to ord$ et l'identité pour $\rho$. Alors, $f \in \text{dom}(\Gamma_0)$, $\tau(lim(f), 1)\rho = f\Gamma_0 = nat \to ord$ et $lim(f) : ord \rhd_2^\rho f : nat \to ord$.

**Définition 80 (Clôture calculable)** Soit $R = (l \to r, \Gamma_0, \rho)$ une règle telle que $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$. La *clôture calculable* de $R$ relativement à une précédence $\geq_{\mathcal{F}}$ et un assignement de statuts *stat* compatible avec $\geq_{\mathcal{F}}$ est la plus petite relation ternaire $\vdash_{\text{c}} \subseteq \mathcal{E} \times \mathcal{T} \times \mathcal{T}$ définie par les règles d'inférence de la Figure 6.1. Nous désignerons par $\vdash_{\text{c}}^<$ la restriction de $\vdash_{\text{c}}$ aux règles différentes de (symb$^=$).

On vérifie aisément que si $\Gamma \vdash_{\text{c}} t : T$ alors $\Gamma = \Gamma_0, \Gamma'$. Et aussi que $\vdash_{\text{c}} \subseteq \vdash_s$ et $\vdash_{\text{c}}^< \subseteq \vdash_f$.

Il est important de noter que la clôture calculable peut facilement être étendue en rajoutant de nouvelles règles d'inférence. Pour préserver la normalisation forte, il suffit alors de compléter la preuve du Théorème 146 où il est montré que les règles de la clôture calculable préservent bien la calculabilité.

**Définition 81 (Règle bien formée)** Soit $R = (l \to r, \Gamma_0, \rho)$ une règle telle que $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$. La règle $R$ est *bien formée* si :
– $\Gamma_0 \vdash l\rho : U\gamma_0\rho$,
– pour tout $x \in \text{dom}(\Gamma_0)$, il existe $i$ tel que $l_i : T_i\gamma_0\ (\rhd_1^\rho)^*\ x : x\Gamma_0$,
– $\text{dom}(\rho) \cap \text{dom}(\Gamma_0) = \emptyset$.

FIGURE 6.1 – Clôture calculable de $(f(\vec{l}) \to r, \Gamma_0, \rho)$

$$\text{(ax)} \quad \frac{}{\Gamma_0 \vdash_{\mathrm{c}} \star : \Box}$$

$$\text{(symb}^{<}\text{)} \quad \frac{\Gamma_0 \vdash_{\mathrm{c}} \tau_g : s \quad \Gamma \text{ valide pour } \vdash_{\mathrm{c}} \quad \Gamma \vdash_{\mathrm{c}} u_1 : U_1\gamma \quad \ldots \quad \Gamma \vdash_{\mathrm{c}} u_n : U_n\gamma}{\Gamma \vdash_{\mathrm{c}} g(\vec{u}) : V\gamma} \quad \begin{array}{c}(g \in \mathcal{F}_n^s,\ g <_{\mathcal{F}} f,\\ \tau_g = (\vec{y} : \vec{U})V,\\ \gamma = \{\vec{y} \mapsto \vec{u}\},\ \vdash \tau_g : s)\end{array}$$

$$\text{(symb}^{=}\text{)} \quad \frac{\Gamma_0 \vdash_{\mathrm{c}} \tau_g : s \quad \Gamma \vdash_{\mathrm{c}} u_1 : U_1\gamma \quad \ldots \quad \Gamma \vdash_{\mathrm{c}} u_n : U_n\gamma}{\Gamma \vdash_{\mathrm{c}} g(\vec{u}) : V\gamma} \quad \begin{array}{c}(g \in \mathcal{F}_n^s,\ g =_{\mathcal{F}} f,\\ \tau_g = (\vec{y} : \vec{U})V,\\ \gamma = \{\vec{y} \mapsto \vec{u}\},\ \vdash \tau_g : s,\\ \vec{l} : \vec{T}\gamma_0 > \vec{u} : \vec{U}\gamma)\end{array}$$

$$\text{(acc)} \quad \frac{\Gamma_0 \vdash_{\mathrm{c}} x\Gamma_0 : s}{\Gamma_0 \vdash_{\mathrm{c}} x : x\Gamma_0} \quad (x \in \mathrm{dom}^s(\Gamma_0))$$

$$\text{(var)} \quad \frac{\Gamma \vdash_{\mathrm{c}} T : s}{\Gamma, x{:}T \vdash_{\mathrm{c}} x : T} \quad \begin{array}{c}(x \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma)\\ \cup \mathrm{FV}(l))\end{array}$$

$$\text{(weak)} \quad \frac{\Gamma \vdash_{\mathrm{c}} t : T \quad \Gamma \vdash_{\mathrm{c}} U : s}{\Gamma, x{:}U \vdash_{\mathrm{c}} t : T} \quad \begin{array}{c}(x \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma)\\ \cup \mathrm{FV}(l))\end{array}$$

$$\text{(prod)} \quad \frac{\Gamma, x{:}T \vdash_{\mathrm{c}} U : s}{\Gamma \vdash_{\mathrm{c}} (x{:}T)U : s}$$

$$\text{(abs)} \quad \frac{\Gamma, x{:}T \vdash_{\mathrm{c}} u : U \quad \Gamma \vdash_{\mathrm{c}} (x{:}T)U : s}{\Gamma \vdash_{\mathrm{c}} [x{:}T]u : (x{:}T)U}$$

$$\text{(app)} \quad \frac{\Gamma \vdash_{\mathrm{c}} t : (x{:}U)V \quad \Gamma \vdash_{\mathrm{c}} u : U}{\Gamma \vdash_{\mathrm{c}} tu : V\{x \mapsto u\}}$$

$$\text{(conv)} \quad \frac{\Gamma \vdash_{\mathrm{c}} t : T \quad \Gamma \vdash_{\mathrm{c}} T : s \quad \Gamma \vdash_{\mathrm{c}} T' : s}{\Gamma \vdash_{\mathrm{c}} t : T'} \quad (T \downarrow T')$$

Par exemple, considérons la règle :

$$appn(p, consn(x, n, \ell), n', \ell') \to consn(x, n + n', appn(n, \ell, n', \ell'))$$

avec $\Gamma_0 = x : T, n : nat, \ell : listn(n), n' : nat, \ell' : listn(n')$ et $\rho = \{p \mapsto s(n)\}$ (voir Chapitre 5). Nous avons $\Gamma_0 \vdash l\rho : listn(p + n')\rho$. Pour $x$, nous avons $consn(x, n, \ell) : listn(p) \rhd_1^\rho x : T$. On vérifie aisément que les conditions sont également vérifiées pour les autres variables.

**Définition 82 (Système récursif)** Soit $R = (l \to r, \Gamma_0, \rho)$ une règle telle que $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$. La règle $R$ satisfait le *Schéma Général* relativement à une précédence $\geq_\mathcal{F}$ et un assignement de statuts *stat* compatible avec $\geq_\mathcal{F}$ si $R$ est bien formée et si $\Gamma_0 \vdash_c r : U\gamma_0\rho$.

Un ensemble de règles $\mathcal{R}$ est *récursif* s'il existe une précédence $\geq_\mathcal{F}$ et un assignement de statuts *stat* compatible avec $\geq_\mathcal{F}$ pour lequel toute règle de $\mathcal{R}$ satisfait le Schéma Général relativement à $\geq_\mathcal{F}$ et *stat*.

Au total, une règle $(l \to r, \Gamma_0, \rho)$ avec $l = f(\vec{l})$, $\tau_f = (\vec{x} : \vec{T})U$ et $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$, est bien typée et bien formée si elle vérifie les conditions suivantes :

- $\mathrm{dom}(\rho) \cap \mathrm{dom}(\Gamma_0) = \emptyset$,
- pour tout $x \in \mathrm{dom}(\Gamma_0)$, il existe $i$ tel que $l_i : T_i\gamma_0\ (\rhd_1^\rho)^*\ x : x\Gamma_0$,
- $\Gamma_0 \vdash l\rho : U\gamma_0\rho$,
- $\Gamma_0 \vdash_c r : U\gamma_0\rho$,
- pour tout $\Delta$, $\sigma$ et $T$, si $\Delta \vdash l\sigma : T$ alors $\sigma : \Gamma \to \Delta$,
- pour tout $\Delta$, $\sigma$ et $T$, si $\Delta \vdash l\sigma : T$ alors, pour tout $x$, $x\sigma \downarrow x\rho\sigma$.

**Remarque 83 (Décidabilité)**
A priori, du fait de la règle (conv) et de la condition $\vdash \tau_g : s$ pour les règles (symb$^<$) et (symb$^=$), la relation $\vdash_c$ peut ne pas être décidable. Cependant, si on suppose $\vdash \tau_g : s$ et on restreint la règle (conv) à un fragment confluent et fortement normalisant de $\to$, $\vdash_c$ devient décidable (avec un algorithme semblable à celui pour $\vdash$). En pratique, les symboles et les règles qui les définissent sont souvent rajoutés les uns après les autres (ou par paquets, mais l'argument qui suit peut se généraliser).

Soit $(\mathcal{F}, \mathcal{R})$ un système confluent et fortement normalisant, $f \notin \mathcal{F}$ et $\mathcal{R}_f$ un ensemble de règles définissant $f$ et dont tous les symboles appartiennent à $\mathcal{F}' = \mathcal{F} \cup \{f\}$. Alors, $(\mathcal{F}', \mathcal{R})$ est également confluent et fortement normalisant. On peut donc chercher à vérifier si les règles de $\mathcal{R}_f$ vérifie le Schéma Général avec la règle (conv) restreinte au cas où $T \downarrow_{\beta R} T'$. Cela ne nous paraît pas une grande restriction : il serait surprenant que le typage d'une règle nécessite l'utilisation de la règle elle-même !

Avant de détailler un exemple, nous allons montrer quelques propriétés simples qui nous faciliterons les preuves d'appartenance à la clôture calculable. En effet, pour montrer $\Gamma_0 \vdash_c r : U\gamma_0\rho$, sachant que par (S3) nous avons $\Gamma_0 \vdash r : U\gamma_0\rho$, on peut se demander dans quelle mesure une dérivation de $\Gamma_0 \vdash r : U\gamma_0\rho$ peut être transformée en une dérivation de $\Gamma_0 \vdash_c r : U\gamma_0\rho$. L'idéal serait qu'il suffise que les symboles de $r$ soient inférieurs ou égaux à $f$ et que, dans le cas d'un appel récursif, les

arguments soient strictement plus petits. Nous montrons ci-après quelques résultats qui permettent de traiter les termes ne contenant que des symboles strictement inférieurs à $f$.

**Définition 84** Une règle $l \to r$ est *compatible* avec $\geq_{\mathcal{F}}$ si tous les symboles de $r$ sont inférieurs ou égaux à ceux de $l$. Un ensemble de règle $\mathcal{R}$ est *compatible* avec $\geq_{\mathcal{F}}$ si chacune des règles de $\mathcal{R}$ est compatible avec $\geq_{\mathcal{F}}$.

Il est facile de vérifier que $\to$ est compatible avec $\geq_{\mathcal{F}}$ (Définition 53) si et seulement si $\mathcal{R}$ est compatible avec $\geq_{\mathcal{F}}$.

**Lemme 85** Supposons $\mathcal{R}$ et $\tau$ compatibles avec $\geq_{\mathcal{F}}$. Quitte à renommer les variables de $\Gamma$, si $\Gamma \vdash_f t : T$ alors $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} t : T$.

**Preuve.** Par récurrence sur $\Gamma \vdash_f t : T$. ■

**Lemme 86 (Substitution pour $\vdash_{\mathrm{c}}$)** Si $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} t : T$ et $\theta : \Gamma_0, \Gamma \to \Gamma_0, \Gamma'$ dans $\vdash_{\mathrm{c}}$ (resp. dans $\vdash_{\mathrm{c}}^{<}$) alors $\Gamma_0, \Gamma' \vdash_{\mathrm{c}} t\theta : T\theta$ (resp. $\Gamma_0, \Gamma' \vdash_{\mathrm{c}}^{<} t\theta : T\theta$).

**Preuve.** Par récurrence sur $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} t : T$. On procède comme pour le Lemme 24.

■

**Lemme 87** Si $\mathcal{R}$ et $\tau$ sont compatibles avec $\geq_{\mathcal{F}}$, $\to$ est confluente et les symboles de $\Gamma_0$ sont inférieurs strictement à $f$ alors, pour tout $x \in \mathrm{dom}^s(\Gamma_0)$, $\Gamma_0 \vdash_{\mathrm{c}}^{<} x\Gamma_0 : s$.

**Preuve.** Supposons que $\Gamma_0 = \vec{y} : \vec{U}$ et que $y_i$ soit de sorte $s_i$. Montrons par récurrence sur $i$ que, pour tout $j \leq i$, $\Gamma_0 \vdash_{\mathrm{c}}^{<} U_j : s_j$. Si $i = 0$, c'est immédiat. Supposons alors $i > 0$. Par hypothèse de récurrence, pour tout $j < i$, $\Gamma_0 \vdash_{\mathrm{c}}^{<} U_j : s_j$, et il nous faut montrer $\Gamma_0 \vdash_{\mathrm{c}}^{<} U_i : s_i$.

Posons $\Gamma = y_1 : U_1, \ldots, y_{i-1} : U_{i-1}$. Soit $\vec{z}$ une séquence de $i-1$ nouvelles variables distinctes de $\vec{y}$, $\theta = \{\vec{y} \mapsto \vec{z}\}$, $\theta' = \{\vec{z} \mapsto \vec{y}\}$ et $\Gamma' = \vec{z} : \vec{U}\theta$. Par **(S3)**, $\Gamma_0$ est valide. Donc, par le lemme d'environnement, $\Gamma \vdash U_i : s_i$. Comme les symboles de $\Gamma$ et $U_i$ sont strictement inférieurs à $f$ et $\to$ est confluente, par le Lemme 54, $\Gamma \vdash_f U_i : s_i$. Par remplacement, $\Gamma' \vdash_f U_i\theta : s_i$. Alors, par le Lemme 85, $\Gamma_0, \Gamma' \vdash_{\mathrm{c}}^{<} U_i\theta : s_i$.

Montrons maintenant que $\theta' : \Gamma_0, \Gamma' \to \Gamma_0$ dans $\vdash_{\mathrm{c}}^{<}$. Pour cela, il suffit de montrer que, pour tout $j < i$, $\Gamma_0 \vdash_{\mathrm{c}}^{<} z_j\theta' : U_j\theta\theta'$, c'est-à-dire, $\Gamma_0 \vdash_{\mathrm{c}}^{<} y_j : U_j$. Or, par hypothèse de récurrence, pour tout $j < i$, $\Gamma_0 \vdash_{\mathrm{c}}^{<} U_j : s_j$. Donc, par (acc), $\Gamma_0 \vdash_{\mathrm{c}}^{<} y_j : U_j$. Ainsi, $\theta' : \Gamma' \to \Gamma_0$ dans $\vdash_{\mathrm{c}}^{<}$ et, par le Lemme 86, $\Gamma_0 \vdash_{\mathrm{c}}^{<} U_i : s_i$. ■

**Lemme 88** Si $\mathcal{R}$ et $\tau$ sont compatibles avec $\geq_{\mathcal{F}}$, $\to$ est confluente et $\Gamma_0, \Gamma \vdash_f t : T$ alors $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} t : T$.

**Preuve.** Par récurrence sur la taille de $\Gamma$. Supposons que $\Gamma_0, \Gamma = \vec{y} : \vec{U}$. Soit $\vec{z}$ une séquence de $|\vec{y}|$ nouvelles variables distinctes de $\vec{y}$, $\theta = \{\vec{y} \mapsto \vec{z}\}$, $\theta' = \{\vec{z} \mapsto \vec{y}\}$ et $\Delta = \vec{z} : \vec{U}\theta$. Par remplacement, $\Delta \vdash_f t\theta : T\theta$. Par le Lemme 85, $\Gamma_0, \Delta \vdash_{\mathrm{c}}^{<} t\theta : T\theta$. Montrons maintenant que $\theta' : \Gamma_0, \Delta \to \Gamma_0, \Gamma$ dans $\vdash_{\mathrm{c}}^{<}$ afin de pouvoir appliquer le Lemme 86.

Pour cela, il faut montrer que, pour tout $x \in \mathrm{dom}(\Gamma_0, \Delta)$, $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} x\theta' : x(\Gamma_0, \Delta)$. Si $x \in \mathrm{dom}(\Gamma_0)$, il faut montrer que $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} x : x\Gamma_0$, et si $x = z_i \in \mathrm{dom}(\Delta)$, il faut

montrer que $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} y_i : U_i$. Autrement dit, il suffit de montrer que $\Gamma_0, \Gamma$ est valide dans $\vdash_{\mathrm{c}}^{<}$. Si $\Gamma$ est vide, c'est immédiat car, d'après le Lemme 87, $\Gamma_0$ est valide dans $\vdash_{\mathrm{c}}^{<}$. Supposons maintenant que $\Gamma = \Gamma', y : U$. Alors, $\Delta = \Delta', z : U\theta$. Par le lemme d'environnement, $\Gamma_0, \Gamma' \vdash_f U : s$. Par hypothèse de récurrence, $\Gamma_0, \Gamma' \vdash_{\mathrm{c}}^{<} U : s$. Donc, par (var), $\Gamma_0, \Gamma \vdash_{\mathrm{c}}^{<} y : U$ et $\Gamma_0, \Gamma$ est valide dans $\vdash_{\mathrm{c}}^{<}$. ■

Un cas particulier utile du lemme est :

**Corollaire 89** Si $\mathcal{R}$ et $\tau$ sont compatibles avec $\geq_{\mathcal{F}}$ et $\rightarrow$ est confluente alors, pour tout $g \leq_{\mathcal{F}} f$, si $\vdash \tau_g : s$ alors $\Gamma_0 \vdash_{\mathrm{c}}^{<} \tau_g : s$.

**Preuve.** Comme $\tau$ est compatible avec $\geq_{\mathcal{F}}$ et $\rightarrow$ est confluente, par le Lemme 54, $\vdash_f \tau_g : s$. Donc, par le Lemme 88, $\Gamma_0 \vdash_{\mathrm{c}}^{<} \tau_g : s$. ■

Maintenant, nous sommes en mesure de détailler un exemple. Considérons la règle :

$$appn(p, consn(x, n, \ell), n', \ell') \rightarrow consn(x, n + n', appn(n, \ell, n', \ell'))$$

avec $\Gamma_0 = x : T, n : nat, \ell : listn(n), n' : nat, \ell' : listn(n')$ et $\rho = \{p \mapsto s(n)\}$ (voir Chapitre 5). Prenons $stat_{appn} = lex(mul(x_2))$ ; $appn >_{\mathcal{F}} consn, +$ ; $consn >_{\mathcal{F}} T$ et $+ >_{\mathcal{F}} s, 0 >_{\mathcal{F}} nat$. Nous avons déjà vu que cette règle est bien formée. Montrons que $\Gamma_0 \vdash_{\mathrm{c}} r : listn(s(n))$. $\mathcal{R}$ et $\tau$ sont compatibles avec $\geq_{\mathcal{F}}$.

Pour appliquer (symb$^{<}$), nous devons montrer $\vdash \tau_{consn} : \star$, $\Gamma_0 \vdash_{\mathrm{c}} \tau_{consn} : \star$, $\Gamma_0 \vdash_{\mathrm{c}} x : T$, $\Gamma_0 \vdash_{\mathrm{c}} n + n' : nat$ et $\Gamma_0 \vdash_{\mathrm{c}} appn(n, \ell, n', \ell') : listn(n + n')$. Il est aisé de vérifier que $\vdash \tau_{consn} : \star$. Alors, par le Corollaire 89, on en déduit que $\Gamma_0 \vdash_{\mathrm{c}} \tau_{consn} : \star$. $\Gamma_0 \vdash_{\mathrm{c}} x : T$ et $\Gamma_0 \vdash_{\mathrm{c}} n + n' : nat$ découlent du Lemme 88. Reste à montrer $\Gamma_0 \vdash_{\mathrm{c}} appn(n, \ell, n', \ell') : listn(n + n')$.

Pour appliquer (symb$^{=}$), nous devons montrer $\vdash \tau_{appn} : \star$, $\Gamma_0 \vdash_{\mathrm{c}} \tau_{appn} : \star$, $\Gamma_0 \vdash_{\mathrm{c}} n : nat$, $\Gamma_0 \vdash_{\mathrm{c}} \ell : listn(n)$, $\Gamma_0 \vdash_{\mathrm{c}} n' : nat$, $\Gamma_0 \vdash_{\mathrm{c}} \ell' : listn(n')$ et $consn(x, n, \ell) : listn(s(n)) >_1 \ell : listn(n)$. Il est aisé de vérifier que $\vdash \tau_{appn} : \star$. Alors, par le Corollaire 89, on en déduit que $\Gamma_0 \vdash_{\mathrm{c}} \tau_{appn} : \star$. Nous avons déjà montré que $consn(x, n, \ell) : listn(p) >_1 \ell : listn(n)$. Les autres assertions découlent du Lemme 88.

## 6.4 Conditions de normalisation forte

**Définition 90 (Systèmes de récriture)** Soit $\mathcal{G} \subseteq \mathcal{F}$. $(\mathcal{G}, \mathcal{R}_{\mathcal{G}})$ est un *système de récriture* :

- *du premier ordre* si :
  - $\mathcal{G}$ est constitué de symboles de prédicat d'arité maximale ou de constructeurs de prédicats primitifs,
  - toutes les règles de $\mathcal{R}_{\mathcal{G}}$ ont un membre droit algébrique ;
- *non-dupliquant* si, pour toute règle de $\mathcal{R}_{\mathcal{G}}$, aucune variable n'a plus d'occurrences libres dans le membre droit que dans le membre gauche ;
- *primitif* si toutes les règles de $\mathcal{R}_{\mathcal{G}}$ ont un membre droit de la forme $[\vec{x} : \vec{T}]g(\vec{u})\vec{v}$ avec $g$ soit un symbole de $\mathcal{G}$ soit un symbole de prédicat primitif ;

- *simple* s'il n'y a pas de paires critiques entre $\mathcal{R}_\mathcal{G}$ et $\mathcal{R}$ :
  - pas de filtrage sur des symboles définis,
  - pas d'ambiguïté dans l'application des règles ;
- *petit* si, pour toute règle $g(\vec{l}) \to r \in \mathcal{R}_\mathcal{G}$ et tout $X \in \mathrm{FV}^\square(r)$, il existe $\kappa_X$ tel que $l_{\kappa_X} = X$ ;
- *positif* si, pour tout symbole $g \in \mathcal{G}$ et toute règle $l \to r \in \mathcal{R}_\mathcal{G}$, $\mathrm{Pos}(g, r) \subseteq \mathrm{Pos}^+(r)$ ;
- *sûr* si, pour toute règle $(g(\vec{l}) \to r, \Gamma, \rho) \in \mathcal{R}_\mathcal{G}$ avec $\tau_g = (\vec{x} : \vec{T})U$ et $\gamma = \{\vec{x} \mapsto \vec{l}\}$ :
  - pour tout $X \in \mathrm{FV}^\square(\vec{T}U)$, $X\gamma\rho \in \mathrm{dom}^\square(\Gamma)$,
  - pour tout $X, X' \in \mathrm{FV}^\square(\vec{T}U)$, $X\gamma\rho = X'\gamma\rho \Rightarrow X = X'$.

**Définition 91 (Conditions de normalisation forte)**

**(A0)** Toutes les règles sont bien typées.

**(A1)** La relation $\to = \to_\mathcal{R} \cup \to_\beta$ est confluente sur $\mathcal{T}$.

**(A2)** Il existe une structure inductive admissible.

**(A3)** Il existe une précédence $\succeq$ sur $\mathcal{DF}^\square$ avec laquelle $\mathcal{R}_{\mathcal{DF}^\square}$ est compatible [3] et dont chaque classe d'équivalence modulo $\simeq$, la relation d'équivalence associée à $\succeq$, forme un système qui est soit :
- **(p)** primitif,
- **(q)** positif, petit et simple,
- **(r)** récursif, petit et simple.

**(A4)** Il existe une partition $\mathcal{F}_1 \uplus \mathcal{F}_\omega$ de $\mathcal{DF}$ (les symboles *de premier ordre* et les symboles *d'ordre supérieur*) telle que : [4]
- **(a)** $(\mathcal{F}_\omega, \mathcal{R}_\omega)$ est récursif,
- **(b)** $(\mathcal{F}_\omega, \mathcal{R}_\omega)$ est sûr,
- **(c)** aucun symbole de $\mathcal{F}_\omega$ n'apparaît dans les règles de $\mathcal{R}_1$,
- **(d)** $(\mathcal{F}_1, \mathcal{R}_1)$ est du premier ordre,
- **(e)** si $\mathcal{R}_\omega \neq \emptyset$ alors $(\mathcal{F}_1, \mathcal{R}_1)$ est non-dupliquant,
- **(f)** $\to_{\mathcal{R}_1}$ est fortement normalisante sur $\mathbb{T}(\mathcal{F}_1, \mathcal{X})$.

La condition (A1) assure, entre autres choses, que le CAC est logique. Avec (A0), cela implique qu'il est admissible, c'est-à-dire, que la réduction préserve le typage (Théorème 60). On peut se demander dans quelle mesure la confluence est nécessaire pour montrer la compatibilité avec le produit et donc, la correction de la $\beta$-réduction. H. Geuvers [56] parvient à montrer cette propriété pour $\mathcal{C} = \leftrightarrow^*_{\beta\eta}$ alors que $\to_{\beta\eta}$ n'est pas confluente sur les termes mal typés. M. Fernández [51] parvient également à montrer cette propriété pour $\mathcal{C} = \to^*_{\beta\mathcal{R}} \cup {}^*_{\beta R}\!\leftarrow$ avec $\to_\mathcal{R}$ une relation de récriture au niveau objet, sans faire d'hypothèse sur la confluence de $\to_{\beta\mathcal{R}}$. Mais ce dernier résultat utilise de manière essentielle le fait que la récriture est cantonnée aux objets. Il n'est pas clair comment ce résultat peut être étendu à la récriture sur les types.

On retiendra donc que les hypothèses (A1) et (A3) ne sont utiles qu'au cas où il y a de la récriture au niveau des types.

3. Voir Définition 84.
4. Nous dénoterons $\mathcal{R}_{\mathcal{F}_1}$ par $\mathcal{R}_1$ et $\mathcal{R}_{\mathcal{F}_\omega}$ par $\mathcal{R}_\omega$.

La condition (A1) peut paraître difficile à remplir car, bien souvent, on montre la confluence à partir de la normalisation forte et de la confluence (locale) des paires critiques (résultat de D. Knuth et P. Bendix [17] pour le premier ordre, étendu à l'ordre supérieur par T. Nipkow [95]).

On sait que $\rightarrow_\beta$ est confluente et qu'il n'y a pas de paires critiques entre $\mathcal{R}$ et la règle de $\beta$-réduction car les membres gauches de $\mathcal{R}$ sont algébriques.

F. Müller [92] a montré dans ce cas que, si $\rightarrow_\mathcal{R}$ est confluente et toutes les règles de $\mathcal{R}$ sont linéaires-gauches, alors $\rightarrow_\mathcal{R} \cup \rightarrow_\beta$ est confluente. Ainsi, la possibilité que nous avons introduite dans les ATS de pouvoir linéariser certaines règles (substitutions $\rho$) tout en assurant la préservation du typage se révèle très utile (voir Chapitre 5).

Dans ce cas-là, et en supposant que $\rightarrow_{\mathcal{R}_1}$ soit fortement normalisante comme c'est demandé en (f), comment peut-on montrer la confluence de $\rightarrow$ ? Dans le cas où $\rightarrow_{\mathcal{R}_1}$ est fortement normalisante et non-dupliquante si $\mathcal{R}_\omega \neq \emptyset$, on peut montrer que $\rightarrow_{\mathcal{R}_1} \cup \rightarrow_{\mathcal{R}_\omega}$ est aussi fortement normalisante (Théorème 144). Ainsi, il suffit d'examiner les paires critiques de $\mathcal{R}$.

Dans (A4), dans le cas où $\mathcal{R}_\omega \neq \emptyset$, on demande à ce que les règles de $\mathcal{R}_1$ ne dupliquent pas de variables. En effet, au premier ordre déjà, la normalisation forte n'est pas une propriété modulaire [113], même avec des systèmes confluents [50]. Par contre, elle est modulaire pour les systèmes non-dupliquants disjoints [104]. Ici, $\mathcal{R}_1$ et $\mathcal{R}_\omega$ ne sont pas disjoints mais définis hiérarchiquement : aucun symbole de $\mathcal{F}_\omega$ n'apparaît dans les règles de $\mathcal{R}_1$. Dans [40], N. Dershowitz rassemble de nombreux résultats sur la modularité de la normalisation forte pour les systèmes du premier ordre, notamment dans le cas de systèmes hiérarchiquement définis. Il serait intéressant d'étudier la modularité de la normalisation dans le cas de la récriture d'ordre supérieur et, en particulier, d'autres conditions que la non-duplication qui nous empêche par exemple de considérer l'exemple suivant :

$$\begin{array}{rcl} 0/y & \rightarrow & 0 \\ s(x)/y & \rightarrow & s((x-y)/y) \\ \\ 0-y & \rightarrow & 0 \\ s(x)-0 & \rightarrow & s(x) \\ s(x)-s(y) & \rightarrow & x-y \end{array}$$

qui est un système du premier ordre dupliquant et ne satisfaisant pas le Schéma Général ; il ne peut donc être placé ni dans $\mathcal{R}_1$ ni dans $\mathcal{R}_\omega$. E. Giménez [59] parvient à traiter cet exemple en utilisant le fait que le résultat de $x-y$ est plus petit que $s(x)$.

Dans (A3), la condition de petitesse pour les systèmes récursifs et positifs est équivalente dans le Calcul des Constructions Inductive à la restriction de l'élimination forte aux types inductifs "petits", c'est-à-dire aux types dont les constructeurs n'ont d'autres arguments-prédicats que ceux du type. Par exemple, le type $list$ des listes polymorphes est "petit" car, dans $(A:\star)A \rightarrow list(A) \rightarrow list(A)$, le type de

son constructeur *cons*, $A$ est un argument de *list*. Par contre, un type $T$ ayant un constructeur $c$ de type $\star \to T$ n'est pas petit. Ainsi, on ne peut pas définir de fonction $f$ de type $T \to \star$ avec la règle $f(c(A)) \to A$. Une telle règle n'est pas petite et ne forme pas un système primitif non plus. D'une certaine manière, les systèmes primitifs peuvent toujours être vus comme des systèmes petits car ils ne contiennent pas de projections et les types primitifs n'ont pas d'arguments-prédicats.

Enfin, dans (A4), la condition de sûreté pour les symboles d'ordre supérieur signifie qu'on ne peut pas faire de filtrage ou tester l'égalité des arguments-prédicats qui sont nécessaires au typage d'autres arguments. Dans son extension de HORPO [74] au Calcul des Constructions, D. Walukiewicz [117] demande une condition similaire. Elle donne quelques exemples (pathologiques) de non normalisation dûs à la violation de cette condition comme $J(A, A, a) \to a$ avec $J : (A : \star)(B : \star)B \to A$ ou $J(A, A, a, b) \to b$ avec $J : (A : \star)(B : \star)A \to B \to A$. Par contre, la règle $map(A, A, [x : A]x, \ell) \to \ell$ qui paraît bien inoffensive ne satisfait pas non plus cette condition de sûreté.

Nous pouvons maintenant énoncer notre principal résultat :

**THÉORÈME :** Un CAC satisfaisant les conditions de la Définition 91 est admissible (*i.e.* la réduction préserve le typage) et fortement normalisant.

La preuve de ce théorème essentiel fait l'objet du Chapitre 8. Il généralise les résultats de M. Fernández [51] et de J.-P. Jouannaud, M. Okada et moi-même [21]. Au Chapitre 7, nous donnons plusieurs exemples importants de CAC vérifiant ces conditions de normalisation forte : un sous-système avec élimination forte du Calcul des Constructions Inductives (CIC) et la Déduction Naturelle Modulo (NDM) de nombreuses théories équationnelles.

Cependant, ces conditions ne permettent pas d'accepter la procédure de décision pour les tautologies propositionnelles classiques de la Figure 1.3. Voyons pourquoi :

- Nous n'avons pas considérer de récriture modulo associativité et commutativité.
- Du fait que le système n'est pas linéaire-gauche, nous ne savons pas comment montrer sa confluence quand il est combiné avec $\beta$.
- Le système n'est pas primitif puisqu'il y a des projections ($P \operatorname{xor} \bot \to P$). Il est récursif (et positif aussi) et petit. Malheureusement, il n'est pas simple.

La récriture modulo AC ne nous paraıt pas une extension très difficile à conduire. Par contre, la confluence et la simplicité sont des problèmes qui nous paraissent non triviaux mais que nous espérons résoudre après cette thèse. Dans le Chapitre 9, nous donnons encore d'autres directions de recherche mais ces trois-là sont assurément les plus importantes.

De la normalisation forte, rappelons-le, on peut déduire la décidabilité de la relation de typage, qui est la propriété sur laquelle repose les systèmes d'aide à la démonstration comme Coq [52] ou LEGO [81].

**Théorème 92 (Décidabilité de $\vdash$)** Soit $\Gamma$ un environnement valide et $T$ un terme égal à $\Box$ ou typable dans $\Gamma$. Dans un CAC satisfaisant les conditions de la Définition 91, vérifier qu'un terme $t$ est de type $T$ dans $\Gamma$ est décidable.

**Preuve.** $\Gamma$ étant valide, il est possible de dire si $t$ est typable ou non, et si $t$ est typable, il est possible d'inférer un type $T'$ pour $t$. Comme les types sont convertibles, il suffit ensuite de vérifier que $T$ et $T'$ ont la même forme normale. Nous renvoyons le lecteur vers [32, 11] pour plus de détails. ■

**Remarque 93 (Cohérence logique)**

Dans le Calcul des Constructions pur (CC), il est assez aisé de vérifier que, dans l'environnement vide, il ne peut y avoir de preuve de $\bot = (P{:}\star)P$ en forme normale [10]. Donc, pour CC, la normalisation forte implique la cohérence logique.

Malheureusement, dans un CAC, la situation n'est pas aussi simple. D'un point de vue logique, avoir des symboles est équivalent à travailler dans un environnement non vide. Il se pourrait donc que les symboles et les règles qui les définissent permettent de former une preuve de $\bot$ en forme normale. Dans [106], J. Seldin montre la cohérence des environnements "fortement cohérents" [5] de manière syntaxique. Cependant, pour montrer la cohérence d'environnements plus riches, il peut être nécessaire d'utiliser des outils sémantiques.

5. Un environnement $\Gamma$ est fortement cohérent si, pour tout $x \in \mathrm{dom}(\Gamma)$, soit $x\Gamma$ est un type de prédicat, soit $x\Gamma$ est $\beta$-équivalent à un terme de la forme $y\vec{t}$.

# Chapitre 7

# Exemples de CAC

## 7.1 Calcul des Constructions Inductives

Nous allons voir que nos conditions de normalisation forte nous permettent de montrer la normalisation d'un sous-système important du Calcul des Constructions Inductives (CIC) de B. Werner [118] qui sert de base au système de spécification et de démonstration Coq [52]. Mais, comme CIC est exprimé dans un formalisme différent du nôtre, cela va nous demander un travail assez important, certes routinier, pour traduire CIC et montrer que cette traduction préserve le typage et réfléchit la normalisation forte.

Afin de pouvoir typer les schémas d'élimination forte de manière polymorphe, ce qu'il n'est pas possible de faire dans le Calcul des Constructions, B. Werner se place dans un système de types avec les sortes $\mathcal{S} = \{\star, \Box, \triangle\}$, les axiomes $\mathcal{A} = \{(\star, \Box), (\Box, \triangle)\}$ et les règles $\mathcal{B} = \{(s_1, s_2, s_3) \in \mathcal{S}^3 \mid s_1 \in \{\star, \Box\}, s_2 = s_3\}$ (en fait, il dénote $\star$ par Set, $\Box$ par Type et $\triangle$ par Extern).

Ensuite, il ajoute des termes pour représenter les types inductifs, leurs constructeurs et les définitions par récursion sur ces types :

- **types inductifs** : Un type inductif est dénoté par $I = Ind(X:A)\{\vec{C}\}$ où les $C_i$ sont les types des constructeurs de $I$. Par exemple, $Nat = Ind(X:\star)\{X, X \to X\}$ représente tout type isomorphe aux entiers naturels. Le terme $A$ doit être de la forme $(\vec{x} : \vec{A})\star$ et les $C_i$ de la forme $(\vec{z} : \vec{B})X\vec{m}$ avec $X \notin \mathrm{FV}(\vec{m})$. De plus, les types inductifs doivent être strictement positifs. Dans CIC, cela signifie que, si $C_i = (\vec{z} : \vec{B})X\vec{m}$ alors, pour tout $j$, soit $X$ n'apparaît pas dans $B_j$, soit $B_j$ est de la forme $(\vec{y}:\vec{D})X\vec{q}$ et $X$ n'apparaît ni dans $\vec{D}$ ni dans $\vec{q}$.
- **constructeurs** : Le $i$-ème constructeur d'un type inductif $I$ est dénoté par $Constr(i, I)$. Par exemple, $Constr(1, Nat)$ représente zéro et $Constr(2, Nat)$ représente la fonction successeur.
- **définitions par récursion** : Une définition par récursion sur un type inductif $I$ est dénotée par $Elim(I, Q, \vec{a}, c)$ où $Q$ est le type du résultat, $\vec{a}$ les arguments de $I$ et $c$ un terme de type $I\vec{a}$. L'élimination forte (dans le cas où $Q$ est un type de prédicat) est restreinte aux types inductifs dits "petits", dont les constructeurs n'ont pas d'autres arguments-prédicats que ceux du types. Plus précisément, un

type inductif $I = Ind(X:A)\{\vec{C}\}$ est *petit* si tous les types de ses constructeurs sont petits et un type de constructeur $C = (\vec{z}:\vec{B})X\vec{m}$ est *petit* si $\{\vec{z}\} \cap \mathcal{X}^{\Box} = \emptyset$ (cela signifie que les arguments-prédicats doivent faire partie de l'environnement de typage $\Gamma$ ; ils ne peuvent pas faire partie de $\vec{C}$).

Pour formuler la relation de réduction associée à *Elim* (appelée *ι-réduction* et notée $\rightarrow_\iota$) et les règles de typage de ces constructions (Figure 7.1), il est nécessaire d'introduire quelques définitions préalables.

Étant donné un type de constructeur $C$, on définit $\Delta\{I, X, C, Q, c\}$ de la manière suivante :

- $\Delta\{I, X, X\vec{m}, Q, c\} = Q\vec{m}c$
- $\Delta\{I, X, (z:B)D, Q, c\} = (z:B)\Delta\{I, X, D, Q, cz\}$ si $X \notin \mathrm{FV}(B)$
- $\Delta\{I, X, (z:B)D, Q, c\} = (z:B\{X \mapsto I\})((\vec{y}:\vec{D})Q\,\vec{q}\,(z\vec{y})) \rightarrow \Delta\{I, X, D, Q, cz\}$ si $B = (\vec{y}:\vec{D})X\vec{q}$

La *ι-réduction* est définie par la règle :

$$Elim(I, Q, \vec{x}, Constr(i, I')\,\vec{z})\{\vec{f}\} \;\rightarrow_\iota\; \Delta[I, X, C_i, f_i, FunElim(I, Q, \vec{f})]\,\vec{z}$$

où $I = Ind(X:A)\{\vec{C}\}$, $FunElim(I, Q, \vec{f}) = [\vec{x}:\vec{A}][y:I\vec{x}]Elim(I, Q, \vec{x}, y)\{\vec{f}\}$ et $\Delta[I, X, C, f, F]$ est défini de la manière suivante :

- $\Delta[I, X, X\vec{m}, f, F] = f$
- $\Delta[I, X, (z:B)D, f, F] = [z:B]\Delta[I, X, D, fz, F]$ si $X \notin \mathrm{FV}(B)$
- $\Delta[I, X, (z:B)D, f, F] = [z:B\{X \mapsto I\}]\Delta[I, X, D, fz\,[\vec{y}:\vec{D}](F\vec{q}\,(z\vec{y})), F]$ si $B = (\vec{y}:\vec{D})X\vec{q}$

Enfin, dans la règle de conversion (conv), en plus de la $\beta$-réduction et de la $\iota$-réduction, B. Werner considère la $\eta$-réduction : $[x:T]ux \rightarrow_\eta u$ si $x \notin \mathrm{FV}(u)$. Du fait que $\rightarrow_{\beta\eta}$ n'est pas confluente sur les termes mal typés [1], considérer la $\eta$-réduction créé de nombreuses difficultés [56]. Pour ce qui nous concerne, comme la condition (A1) ne peut pas être vérifiée en présence de $\eta$-réduction, nous ne pouvons pas considérer la $\eta$-réduction. Trouver une condition plus faible que (A1) qui soit vérifiée en présence de $\eta$-réduction et permette de montrer la correction de la réduction et la normalisation forte est un problème que nous avons provisoirement laissé ouvert.

La $\iota$-réduction telle qu'elle est définie dans CIC introduit de nombreux $\beta$-radicaux, et les appels récursifs à *Elim* sont faits sur des variables liées qui seront instanciées par des sous-termes stricts. Ainsi, d'une part, cela n'est pas très efficace car ces instanciations pourraient être faites immédiatement après la $\iota$-réduction, et d'autre part, la forme actuelle du Schéma Général ne permet pas de traiter des appels récursifs sur des variables liées, même si celles-ci ne sont instanciables que par des sous-termes stricts.

C'est pourquoi nous n'allons pas montrer la normalisation forte de la relation $\rightarrow_{\beta\iota}$ mais de la relation $\rightarrow_{\beta\iota'}$ où $\rightarrow_{\iota'}$ est la relation de réduction dont un pas correspond à une $\iota$-réduction suivie d'une série de $\beta$-réductions pour éliminer les $\beta$-radicaux introduits par la $\iota$-réduction. C'est d'ailleurs cette relation de réduction qui est implémentée dans le système Coq [52].

1. Remarque dûe à R. Nederpelt [93] : $[x:A]x\,{}_\beta\!\leftarrow [x:A]([y:B]y\ x) \rightarrow_\eta [y:B]y$.

FIGURE 7.1 – Règles de typage de CIC

$$\text{(Ind)}\quad \frac{A = (\vec{x} : \vec{A}) \star \quad \Gamma \vdash A : \Box \quad \forall i,\, \Gamma, X : A \vdash C_i : \star}{\Gamma \vdash Ind(X\!:\!A)\{\vec{C}\} : A}$$

$$\text{(Constr)}\quad \frac{I = Ind(X\!:\!A)\{\vec{C}\} \quad \Gamma \vdash I : T}{\Gamma \vdash Constr(i, I) : C_i\{X \mapsto I\}}$$

$$\text{($\star$-Elim)}\quad \frac{\begin{array}{c} A = (\vec{x} : \vec{A}) \star \quad I = Ind(X\!:\!A)\{\vec{C}\} \\ \Gamma \vdash Q : (\vec{x} : \vec{A}) I\vec{x} \to \star \\ T_i = \Delta\{I, X, C_i, Q, Constr(i, I)\} \\ \gamma = \{\vec{x} \mapsto \vec{a}\} \quad \forall j,\, \Gamma \vdash a_j : A_j\gamma \quad \Gamma \vdash c : I\vec{a} \quad \forall i,\, \Gamma \vdash f_i : T_i \end{array}}{\Gamma \vdash Elim(I, Q, \vec{a}, c)\{\vec{f}\} : Q\vec{a}c}$$

$$\text{($\Box$-Elim)}\quad \frac{\begin{array}{c} A = (\vec{x} : \vec{A}) \star \quad I = Ind(X\!:\!A)\{\vec{C}\} \text{ est petit} \\ \Gamma \vdash Q : (\vec{x} : \vec{A}) I\vec{x} \to \Box \\ T_i = \Delta\{I, X, C_i, Q, Constr(i, I)\} \\ \gamma = \{\vec{x} \mapsto \vec{a}\} \quad \forall j,\, \Gamma \vdash a_j : A_j\gamma \quad \Gamma \vdash c : I\vec{a} \quad \forall i,\, \Gamma \vdash f_i : T_i\} \end{array}}{\Gamma \vdash Elim(I, Q, \vec{a}, c)\{\vec{f}\} : Q\vec{a}c}$$

$$\text{(Conv)}\quad \frac{\Gamma \vdash t : T \quad T \leftrightarrow^*_{\beta\eta\iota} T' \quad \Gamma \vdash T' : s}{\Gamma \vdash t : T'}$$

**Définition 94 ($\iota'$-réduction)** La *$\iota'$-réduction* est la relation de réduction définie par la règle :

$$Elim(I, Q, \vec{x}, Constr(i, I')\,\vec{z})\{\vec{f}\} \;\to_{\iota'}\; \Delta'[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$$

où $I = Ind(X\!:\!A)\{\vec{C}\}$ et $\Delta'[I, X, C, f, Q, \vec{f}, \vec{z}]$ est défini de la manière suivante :
- $\Delta'[I, X, X\vec{m}, f, Q, \vec{f}, \emptyset] = f$
- $\Delta'[I, X, (z\!:\!B)D, f, Q, \vec{f}, z\vec{z}] = \Delta'[I, X, D, fz, Q, \vec{z}]$ si $X \notin \mathrm{FV}(B)$
- $\Delta'[I, X, (z\!:\!B)D, f, Q, \vec{f}, z\vec{z}] = \Delta'[I, X, D, fz\,[\vec{y}\!:\!\vec{D}]Elim(I, Q, \vec{q}, z\vec{y}), Q, \vec{z}]$ si $B = (\vec{y}\!:\!\vec{D})X\vec{q}$

Nous pensons que la normalisation forte de $\to_{\beta\iota'}$ implique celle de $\to_{\beta\iota}$. Mais, comme ce problème ne nous paraît pas très simple et n'est pas directement lié à notre travail, nous en remettons la résolution à plus tard.

**Conjecture 95** Si $\to_{\beta\iota'}$ est fortement normalisante alors $\to_{\beta\iota}$ est fortement normalisante.

**Définition 96 (Type inductif admissible)** Un type inductif $I = Ind(X\!:\!A)\{\vec{C}\}$ est *admissible* s'il vérifie les conditions (I5), (I6) (adaptées à la syntaxe CIC, une

élimination forte étant considérée comme un symbole de prédicat défini) et la condition de *sûreté* suivante. Si $A = (\vec{x} : \vec{A})\star$ et $C_i = (\vec{z} : \vec{B})X\vec{m}$ alors :

– $\forall x_i \in \mathcal{X}^{\square}$, $m_i \in \mathcal{X}^{\square}$,
– $\forall x_i, x_j \in \mathcal{X}^{\square}$, $m_i = m_j \Rightarrow x_i = x_j$.

**Définition 97 (CIC$^-$)** Le sous-système que nous allons considérer, CIC$^-$, s'obtient en appliquant les restrictions suivantes :

- On exclut toute utilisation de la sorte $\triangle$ afin de rester dans le Calcul des Constructions.
- Dans la règle (Ind), au lieu de demander que $I = Ind(X : A)\{\vec{C}\}$ soit typable dans un environnement $\Gamma$ quelconque, on demande que $I$ soit typable dans l'environnement vide car, dans CAC, les types des symboles doivent être typables dans l'environnement vide. De plus, on se restreint aux types admissibles en forme normale.

  La restriction à $\Gamma = \emptyset$ n'est pas une réelle restriction car tout type $I = Ind(X : A)\{\vec{C}\}$ typable dans un environnement $\Gamma = \vec{y} : \vec{U}$ peut être remplacé par un type $I' = Ind(X' : A')\{\vec{C'}\}$ équivalent : il suffit de prendre $A' = (\vec{y} : \vec{U})A$, $C'_i = (\vec{y} : \vec{U})C_i\{X \mapsto X'\vec{y}\}$ et de remplacer $I$ par $I'\vec{y}$ et $Constr(i, I)$ par $Constr(i, I')\vec{y}$.

  Cependant, on est amené à modifier la définition de type de constructeur *petit* de la manière suivante : un type de constructeur $C$ d'un type $I = Ind(X : A)\{\vec{C}\}$ avec $A = (\vec{x} : \vec{A})\star$ est *petit* s'il est de la forme $(\vec{x} : \vec{A})(\vec{z} : \vec{B})X\vec{m}$ avec $\{\vec{z}\} \cap \mathcal{X}^{\square} = \emptyset$.
- Dans la règle ($\star$-Elim), au lieu de demander que $Q$ soit typable dans un environnement $\Gamma$ quelconque, on demande que $Q$ soit typable dans l'environnement vide. De plus, on demande explicitement que $I$ et $T_i = \Delta\{I, X, C_i, Q, Constr(i, I)\}$ soient typables.
- Dans la règle ($\square$-Elim), au lieu de demander $\vdash Q : (\vec{x} : \vec{A})I\vec{x} \to \square$, ce qui n'est pas possible dans le Calcul des Constructions, on demande que $Q$ soit de la forme $[\vec{x} : \vec{A}][y : I\vec{x}]K$ avec $\vec{x} : \vec{A}, y : I\vec{x} \vdash K : \square$, et que $f_i$ soit de type $T_i = \Delta'\{I, X, C_i, \vec{x}y, K, Constr(i, I)\}$ où $\Delta'\{I, X, C, \vec{x}y, K, c\}$ est défini de la manière suivante :
  – $\Delta'\{I, X, X\vec{m}, \vec{x}y, K, c\} = K\{\vec{x} \mapsto \vec{m}, y \mapsto c\}$,
  – si $B = (\vec{y} : \vec{D})X\vec{q}$ alors $\Delta'\{I, X, (z : B)D, \vec{x}y, K, c\} = (z : B\{X \mapsto I\})((\vec{y} : \vec{D})K\{\vec{x} \mapsto \vec{q}, y \mapsto z\vec{y}\}) \to \Delta'\{I, X, D, \vec{x}y, K, cz\}$.

  De plus, on demande que $Q$ soit en forme normale, que $T_i$ soit typable et que les types inductifs qui apparaissent dans $Q$ soient des sous-termes de $I$. Enfin, comme conclusion, on prend $\Gamma \vdash Elim(I, Q, \vec{a}, c)\{\vec{f}\} : K\{\vec{x} \mapsto \vec{a}, y \mapsto c\}$ au lieu de $\Gamma \vdash Elim(I, Q, \vec{a}, c)\{\vec{f}\} : Q\vec{a}c$.

  Demander que $Q$ soit de la forme $[\vec{x} : \vec{A}][y : I\vec{x}]K$ n'est pas une très grande restriction car, comme l'a montré B. Werner (Corollaire 2.9 page 57 de [118]), si $\Gamma \vdash Q : \square$ alors il existe $Q'$ de la forme $(\vec{y} : \vec{U})\star$ tel que $Q \to^*_\beta Q'$. Ainsi, si $\vdash Q : (\vec{x} : \vec{A})I\vec{x} \to \square$ alors $\vec{x} : \vec{A}, y : I\vec{x} \vdash Q\vec{x}y : \square$. Donc, il existe $Q'$ de la forme $(\vec{y} : \vec{U})\star$ tel que $Q\vec{x}y \to^*_\beta Q'$. Alors, $[\vec{x} : \vec{A}][y : I\vec{x}]Q\vec{x}y \to^*_\beta [\vec{x} : \vec{A}][y : I\vec{x}]Q'$ et $[\vec{x} : \vec{A}][y : I\vec{x}]Q\vec{x}y \to^*_\eta Q$. Donc, par confluence, il existe $Q''$ de la forme $[\vec{x} : \vec{A}][y : I\vec{x}](\vec{y} : \vec{U'})\star$ tel que $Q \to^*_{\beta\eta} Q''$.

Par contre, demander que les inductifs de $Q$ soient des sous-termes de $I$ est une restriction plus sévère dûe uniquement au fait qu'on se restreint au Calcul des Constructions et qu'on ne peut donc pas typer le schéma d'élimination forte de manière polymorphe (c'est pourquoi B. Werner se place dans un PTS plus général).

- Dans la règle (conv), au lieu de demander $T \leftrightarrow^*_{\beta\eta\iota} T'$, on demande $T \leftrightarrow^*_{\beta\iota'} T'$ qui est équivalent à $T \downarrow_{\beta\iota'} T'$ puisque $\rightarrow_{\beta\iota'}$ est confluente (CRS orthogonal).

Nous désignerons par $\rightarrow_{\beta\iota'}$ la relation de réduction de CIC$^-$, par $\mathcal{NF}$ l'ensemble des termes de CIC$^-$ qui admettent une forme normale pour $\rightarrow_{\beta\iota'}$ (unique par confluence), par $t\downarrow$ la forme normale de $t$, et par $\vdash$ la relation de typage de CIC$^-$ (Figure 7.2).

FIGURE 7.2 – Règles de typage de CIC$^-$

$$\text{(Ind)}\quad \frac{A = (\vec{x} : \vec{A})\star \quad \vdash A : \Box \quad \forall i,\ X : A \vdash C_i : \star \quad I = Ind(X{:}A)\{\vec{C}\} \in \mathcal{NF} \text{ est admissible}}{\vdash I : A}$$

$$\text{(Constr)}\quad \frac{I = Ind(X{:}A)\{\vec{C}\} \quad \Gamma \vdash I : T}{\Gamma \vdash Constr(i, I) : C_i\{X \mapsto I\}}$$

$$\text{($\star$-Elim)}\quad \frac{\begin{array}{c} A = (\vec{x} : \vec{A})\star \quad I = Ind(X{:}A)\{\vec{C}\} \quad \Gamma \vdash I : T \\ \vdash Q : (\vec{x} : \vec{A})I\vec{x} \rightarrow \star \\ T_i = \Delta\{I, X, C_i, Q, Constr(i, I)\} \quad \vdash T_i : \star \\ \gamma = \{\vec{x} \mapsto \vec{a}\} \quad \forall j, \Gamma \vdash a_j : A_j\gamma \quad \Gamma \vdash c : I\vec{a} \quad \forall i, \Gamma \vdash f_i : T_i \end{array}}{\Gamma \vdash Elim(I, Q, \vec{a}, c)\{\vec{f}\} : Q\vec{a}c}$$

$$\text{($\Box$-Elim)}\quad \frac{\begin{array}{c} A = (\vec{x} : \vec{A})\star \quad I = Ind(X{:}A)\{\vec{C}\} \\ Q = [\vec{x}{:}\vec{A}][y{:}I\vec{x}]K \in \mathcal{NF} \quad \vec{x} : \vec{A}, y : I\vec{x} \vdash K : \Box \\ \text{les inductifs de } Q \text{ sont des sous-termes de } I \\ T_i = \Delta'\{I, X, C_i, \vec{x}y, K, Constr(i, I)\} \quad \vdash T_i : \Box \\ \gamma = \{\vec{x} \mapsto \vec{a}\} \quad \forall j, \Gamma \vdash a_j : A_j\gamma \quad \Gamma \vdash c : I\vec{a} \quad \forall i, \Gamma \vdash f_i : T_i \end{array}}{\Gamma \vdash Elim(I, Q, \vec{a}, c)\{\vec{f}\} : K\{\vec{x} \mapsto \vec{a}, y \mapsto c\}}$$

$$\text{(Conv)}\quad \frac{\Gamma \vdash t : T \quad T \leftrightarrow^*_{\beta\iota'} T' \quad \Gamma \vdash T' : s}{\Gamma \vdash t : T'}$$

**Théorème 98** Il existe un CAC $\Upsilon$ (de relation de typage $\vdash_\Upsilon$ et de relation de réduction $\rightarrow$) vérifiant les conditions de normalisation forte de la Définition 91, et une fonction $\langle\_\rangle$ qui, à un terme de CIC$^-$, associe un terme de $\Upsilon$ telle que :

– si $\Gamma \vdash t : T$ alors $\langle\Gamma\rangle \vdash_\Upsilon \langle t\rangle : \langle T\rangle$,

– si de plus $t \rightarrow_{\beta\iota'} t'$ alors $\langle t\rangle \rightarrow^+ \langle t'\rangle$.

Ainsi, $\rightarrow_{\beta\iota'}$ est fortement normalisante dans CIC$^-$.

**Définition 99 (Traduction)** Nous définissons $\langle t\rangle$ sur les termes bien typés, par récurrence sur $\Gamma \vdash t : T$ :

- Soit $I = Ind(X:A)\{\vec{C}\}$ avec $A = (\vec{x} : \vec{A})\star$. Nous prenons $\langle I\rangle = [\vec{x}:\langle\vec{A}\rangle]Ind_I(\vec{x})$ où $Ind_I$ est un symbole de type $\langle A\rangle$.
- Par hypothèse, $C_i = (\vec{z} : \vec{B})X\vec{m}$. Nous prenons $\langle Constr(i,I)\rangle = [\vec{z} : \langle\vec{B}\rangle\theta']\ Constr_i^I(\vec{z})$ où $\theta' = \{X \mapsto \langle I\rangle\}$, $Constr_i^I$ est un symbole de type $(\vec{z}:\vec{B'})Ind_I(\langle\vec{m}\rangle)$, $B'_j = \langle B_j\rangle$ si $X$ n'apparaît pas dans $B_j$, et $B'_j = (\vec{y}:\langle\vec{D}\rangle)Ind_I(\langle\vec{q}\rangle)$ si $B_j = (\vec{y}:\vec{D})X\vec{q}$.
- Soit $Q$ un terme qui n'est pas de la forme $[\vec{x} : \vec{A}][y : I\vec{x}]K$ avec $K = (\vec{y} : \vec{U})\star$. Nous prenons $\langle Elim(I,Q,\vec{a},c)\{\vec{f}\}\rangle = WElim_I(\langle Q\rangle,\langle\vec{a}\rangle,\langle c\rangle,\langle\vec{f}\rangle)$ où $WElim_I$ est un symbole de type $(Q : (\vec{x} : \langle\vec{A}\rangle)\ \langle I\rangle\vec{x} \to \star)(\vec{x} : \langle\vec{A}\rangle)(y : \langle I\rangle\vec{x})(\vec{f} : \langle\vec{T}\rangle)\langle Q\rangle\vec{x}y$ et $T_i = \Delta\{I, X, C_i, Q, Constr(i,I)\}$.
- Soit $Q$ un terme de la forme $[\vec{x} : \vec{A}][y : I\vec{x}]K$ avec $K = (\vec{y} : \vec{U})\star$. Nous prenons $\langle Elim(I,Q,\vec{a},c)\{\vec{f}\}\rangle = SElim_I^Q(\langle\vec{a}\rangle,\langle c\rangle,\langle\vec{f}\rangle)$ où $SElim_I^Q$ est un symbole de type $(\vec{x}:\langle\vec{A}\rangle)(y:\langle I\rangle\vec{x})(\vec{f}:\langle\vec{T}\rangle)\langle K\rangle$, $T_i = \Delta'\{I, X, C_i, \vec{x}y, K, Constr(i,I)\}$.
- Les autres termes sont définis récursivement : $\langle uv\rangle = \langle u\rangle\langle v\rangle, \ldots$

Soit $\Upsilon$ le CAC dont les symboles sont ceux décrits précédemment et dont les règles sont :

$$\begin{array}{rcl} WElim_I(Q,\vec{x},Constr_i^I(\vec{z}),\vec{f}) & \to & \Delta'_W[I,X,C_i,f_i,Q,\vec{f},\vec{z}] \\ SElim_I^Q(\vec{x},Constr_i^I(\vec{z}),\vec{f}) & \to & \Delta'_S[I,X,C_i,f_i,Q,\vec{f},\vec{z}] \end{array}$$

où $\Delta'_W[I,X,C,f,Q,\vec{f},\vec{z}]$ et $\Delta'_S[I,X,C,f,Q,\vec{f},\vec{z}]$ sont définis de la manière suivante :

- $\Delta'_W[I,X,X\vec{m},f,Q,\vec{f},\vec{z}] = \Delta'_S[I,X,X\vec{m},f,Q,\vec{f},\vec{z}] = f$,
- $\Delta'_S[I,X,(z:B)D,f,Q,\vec{f},z\vec{z}] = \Delta'_S[I,X,D,f\,z,Q,\vec{f},\vec{z}]$
  et $\Delta'_W[I,X,(z:B)D,f,Q,\vec{f},z\vec{z}] = \Delta'_W[I,X,D,f\,z,Q,\vec{f},\vec{z}]$ si $X \notin \mathrm{FV}(B)$
- $\Delta'_S[I,X,(z:B)D,f,Q,\vec{f},z\vec{z}] = \Delta'_S[I,X,D,f\,z\,[\vec{y}:\vec{D}]SElim_I^Q(\vec{f},\vec{q},z\vec{y}),Q,\vec{f},\vec{z}]$
  et $\Delta'_W[I,X,(z:B)D,f,Q,\vec{f},z\vec{z}] =$
  $\Delta'_W[I,X,D,f\,z\,[\vec{y}:\vec{D}]WElim_I(Q,\vec{f},\vec{q},z\vec{y}),Q,\vec{f},\vec{z}]$ si $B = (\vec{y}:\vec{D})X\vec{q}$

Comme $\to_{\beta\iota'}$ est confluente, $\Upsilon$ est un ATS logique. Donc, la $\beta$-réduction préserve le typage. Cela va nous être utile pour montrer que la traduction préserve le typage :

**Lemme 100** Si $\Gamma \vdash t : T$ alors $\langle\Gamma\rangle \vdash_\Upsilon \langle t\rangle : \langle T\rangle$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$.

**(Ind)** Nous devons montrer $\vdash_\Upsilon \langle I\rangle : \langle A\rangle$. Nous avons $\langle I\rangle = [\vec{x} : \langle\vec{A}\rangle]Ind_I(\vec{x})$ avec $Ind_I$ de type $\langle A\rangle = (\vec{x}:\langle\vec{A}\rangle)\star$. Comme $\vdash A : \Box$, par hypothèse de récurrence, nous avons $\vdash_\Upsilon \langle A\rangle : \Box$, c'est-à-dire, $\vdash_\Upsilon \tau_{Ind_I} : \Box$. Par inversion, on obtient $\vec{x} : \langle\vec{A}\rangle \vdash_\Upsilon \star : \Box$. Donc, $\Gamma = \vec{x} : \langle\vec{A}\rangle$ est valide et, par le lemme d'environnement et (weak), pour tout $i$, $\Gamma \vdash_\Upsilon x_i : \langle A_i\rangle$. Ainsi, par (symb), $\Gamma \vdash_\Upsilon Ind_I(\vec{x}) : \star$ et, par (abs), $\Gamma \vdash_\Upsilon \langle I\rangle : \langle A\rangle$.

**(Constr)** Nous devons montrer $\langle\Gamma\rangle \vdash_\Upsilon \langle Constr(i,I)\rangle : \langle C_i\theta\rangle$ où $\theta = \{X \mapsto I\}$. Nous avons $C_i = (\vec{z} : \vec{B})\ X\vec{m}$, $\langle Constr(i,I)\rangle = [\vec{z}:\langle\vec{B}\rangle\theta']Constr_i^I(\vec{z})$, $\theta' = \{X \mapsto \langle I\rangle\}$,

$Constr_i^I$ de type $(\vec{z} : \vec{B}')Ind_I(\langle\vec{m}\rangle)$, $B'_j = \langle B_j\rangle$ si $X$ n'apparaît pas dans $B_j$, $B'_j = (\vec{y}:\langle\vec{D}\rangle)Ind_I(\langle\vec{q}\rangle)$ si $B_j = (\vec{y}:\vec{D})X\vec{q}$, $\langle C_i\theta\rangle = \langle C_i\rangle\theta' = (\vec{z}:\langle\vec{B}\rangle\theta')\langle I\rangle\langle\vec{m}\rangle$ et $\langle I\rangle = (\vec{x}:\langle\vec{A}\rangle)Ind_I(\vec{x})$.

Ainsi, $\langle I\rangle\langle\vec{m}\rangle \to_\beta^* Ind_I(\langle\vec{m}\rangle)$. De plus, si $X$ n'apparaît pas dans $B_j$ alors $B'_j = \langle B_j\rangle = \langle B_j\rangle\theta'$. Si $B_j = (\vec{y}:\vec{D})X\vec{q}$ alors $B'_j = (\vec{y}:\langle\vec{D}\rangle)Ind_I(\langle\vec{q}\rangle)$ et $\langle B_j\rangle\theta' = (\vec{y}:\langle\vec{D}\rangle)\langle I\rangle\langle\vec{q}\rangle$. Comme, $\langle I\rangle\langle\vec{q}\rangle \to_\beta^* Ind_I(\langle\vec{q}\rangle)$, pour tout $j$, $\langle B_j\rangle\theta' \to_\beta^* B'_j$.

Comme $\Gamma \vdash I : T$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle I\rangle : \langle T\rangle$ et $\langle\Gamma\rangle$ est valide. Par ailleurs, par inversion, $\vdash I : A$ et $X : A \vdash C_i : \star$. Par inversion encore, $X : A, \vec{z} : \vec{B} \vdash X\vec{m} : \star$. Par hypothèse de récurrence, $\vdash_\Upsilon \langle I\rangle : \langle A\rangle$ et $X : \langle A\rangle, \vec{z} : \langle\vec{B}\rangle \vdash_\Upsilon X\langle\vec{m}\rangle : \star$. Par substitution, $\vec{z} : \langle\vec{B}\rangle\theta' \vdash_\Upsilon \langle I\rangle\langle\vec{m}\rangle : \star$. Donc, $\Delta = \vec{z}:\langle\vec{B}\rangle\theta'$ est valide. Comme $\langle\vec{B}\rangle\theta' \to_\beta^* \vec{B}'$, par préservation du typage sur les environnements, il en va de même de $\Delta' = \vec{z} : \vec{B}'$. Donc, $\vec{z} : \vec{B}' \vdash_\Upsilon \langle I\rangle\langle\vec{m}\rangle : \star$ et, par (prod), $\vdash_\Upsilon (\vec{z}:\vec{B}')Ind_I(\langle\vec{m}\rangle) : \star$, c'est-à-dire, $\vdash_\Upsilon \tau_{Constr_i^I} : \star$.

Par le lemme d'environnement et (conv), pour tout $j$, $\Delta \vdash_\Upsilon z_j : B'_j$. Donc, par (symb), $\Delta \vdash_\Upsilon Constr_i^I(\vec{z}) : Ind_I(\langle\vec{m}\rangle)$ et, par (abs), $\vdash_\Upsilon \langle Constr(i,I)\rangle : (\vec{z}:\langle\vec{B}\rangle\theta')Ind_I(\langle\vec{m}\rangle)$. Finalement, par (conv) et (weak), $\langle\Gamma\rangle \vdash_\Upsilon \langle Constr(i,I)\rangle : \langle C_i\theta\rangle$.

**($\star$-Elim)** Nous devons montrer $\langle\Gamma\rangle \vdash_\Upsilon \langle Elim(I,Q,\vec{a},c)\{\vec{f}\}\rangle : \langle Q\vec{a}c\rangle$. Nous avons $\langle Elim(I,Q,\vec{a},c)\{\vec{f}\}\rangle = WElim_I(\langle Q\rangle,\langle\vec{a}\rangle,\langle c\rangle,\langle\vec{f}\rangle)$, $WElim_I$ de type $(Q:\langle B\rangle)(\vec{x}:\langle\vec{A}\rangle)(y:I\vec{x})(\vec{f}:\langle\vec{T}\rangle)\langle Q\rangle\vec{x}y$, $B = (\vec{x}:\vec{A})I\vec{x} \to \star$, $T_i = \Delta\{I, X, C_i, Q, Constr(i,I)\}$ et $\langle Q\vec{a}c\rangle = \langle Q\rangle\langle\vec{a}\rangle\langle c\rangle$. Afin de pouvoir appliquer (symb), montrons (1) $\langle\Gamma\rangle \vdash_\Upsilon \langle Q\rangle : \langle B\rangle$, (2) $\langle\Gamma\rangle \vdash_\Upsilon \langle\vec{a}\rangle : \langle\vec{A}\rangle\gamma'$, (3) $\langle\Gamma\rangle \vdash_\Upsilon \langle c\rangle : \langle I\rangle\vec{x}\gamma'$, (4) $\langle\Gamma\rangle \vdash_\Upsilon \langle\vec{f}\rangle : \langle\vec{T}\rangle\gamma'$ et (5) $\vdash_\Upsilon \tau_{WElim_I} : \star$, où $\gamma' = \{Q \mapsto \langle Q\rangle, \vec{x} \mapsto \langle\vec{a}\rangle, y \mapsto \langle c\rangle, \vec{f} \mapsto \langle\vec{f}\rangle\}$. D'abord, remarquons que, comme $\Gamma \vdash c : I\vec{a}$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle c\rangle : \langle I\vec{a}\rangle$ et $\langle\Gamma\rangle$ est valide.

(1) Comme $\vdash Q : B$, par hypothèse de récurrence, $\vdash_\Upsilon \langle Q\rangle : \langle B\rangle$. Donc, par affaiblissement, $\langle\Gamma\rangle \vdash_\Upsilon \langle Q\rangle : \langle B\rangle$.

(2) Comme $\Gamma \vdash a_j : A_j\gamma$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle a_j\rangle : \langle A_j\gamma\rangle$. Or, $\langle A_j\gamma\rangle = \langle A_j\rangle\gamma'$ car $\mathrm{FV}(A_j) \subseteq \{\vec{x}\}$.

(3) Comme $\Gamma \vdash c : I\vec{a}$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle c\rangle : \langle I\vec{a}\rangle$. Or, $\langle I\vec{a}\rangle = \langle I\rangle\langle\vec{a}\rangle = \langle I\rangle\vec{x}\gamma'$.

(4) Comme $\Gamma \vdash f_i : T_i$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle f_i\rangle : \langle T_i\rangle$. Or, $\langle T_i\rangle\gamma' = \langle T_i\rangle$ car $T_i$ est clos ($\vdash T_i : \star$).

(5) Soit $\Delta = Q : \langle B\rangle, \vec{x} : \langle\vec{A}\rangle, (y : \langle I\rangle\vec{x})$ et $\Delta' = \Delta, \vec{f} : \langle\vec{T}\rangle$. Montrons que $\Delta'$ est valide. En effet, si c'est le cas, alors $\Delta' \vdash_\Upsilon \langle Q\rangle\vec{x}y : \star$ et, par (prod), $\vdash_\Upsilon \tau_{WElim_I} : \star$. Nous avons $\vdash_\Upsilon \langle Q\rangle : \langle B\rangle$. Donc, $\vdash_\Upsilon \langle B\rangle : \Box$. Par (var), $Q : \langle B\rangle \vdash_\Upsilon Q : \langle B\rangle$. Par inversion, $Q : \langle B\rangle, \vec{x} : \langle\vec{A}\rangle, y : \langle I\rangle\vec{x} \vdash_\Upsilon \star : \Box$ et $\Delta$ est valide. Soit $\Delta_i = \Delta, f_1:\langle T_1\rangle, \ldots, f_i:\langle T_i\rangle$. Montrons par récurrence sur $i$ que $\Delta_i$ est valide. Si $i = 0$, c'est immédiat. Montrons alors que si $\Delta_i$ est valide alors $\Delta_{i+1}$ l'est aussi. Comme $\vdash_\Upsilon \langle T_{i+1}\rangle : \star$, par affaiblissement, $\Delta_i \vdash_\Upsilon \langle T_{i+1}\rangle : \star$. Donc, par (var), $\Delta_{i+1} \vdash_\Upsilon f_{i+1} : T_{i+1}$ et $\Delta_{i+1}$ est valide.

**($\Box$-Elim)** Nous devons montrer $\langle\Gamma\rangle \vdash_\Upsilon \langle Elim(I,Q,\vec{a},c)\{\vec{f}\}\rangle : \langle K\rangle$. Nous avons $\langle Elim(I,Q,\vec{a},c)\{\vec{f}\}\rangle = SElim_I^Q(\langle\vec{a}\rangle,\langle c\rangle,\langle\vec{f}\rangle)$, $SElim_I^Q$ de type $(\vec{x}:\langle\vec{A}\rangle)(y:I\vec{x})(\vec{f}:\langle\vec{T}\rangle)\langle K\rangle$ et $T_i = \Delta'\{I, X, C_i, Q, Constr(i,I)\}$. Afin de pouvoir appliquer (symb),

montrons (1) $\langle\Gamma\rangle \vdash_\Upsilon \langle\vec{a}\rangle : \langle\vec{A}\rangle\gamma'$, (2) $\langle\Gamma\rangle \vdash_\Upsilon \langle c\rangle : \langle I\rangle\vec{x}\gamma'$, (3) $\langle\Gamma\rangle \vdash_\Upsilon \langle\vec{f}\rangle : \langle\vec{T}\rangle\gamma'$ et (4) $\vdash_\Upsilon \tau_{SElim_I} : \Box$, où $\gamma' = \{\vec{x} \mapsto \langle\vec{a}\rangle, y \mapsto \langle c\rangle, \vec{f} \mapsto \langle\vec{f}\rangle\}$. D'abord, remarquons que, comme $\Gamma \vdash c : I\vec{a}$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle c\rangle : \langle I\vec{a}\rangle$ et $\langle\Gamma\rangle$ est valide.

(1) Comme $\Gamma \vdash a_j : A_j\gamma$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle a_j\rangle : \langle A_j\gamma\rangle$. Or, $\langle A_j\gamma\rangle = \langle A_j\rangle\gamma'$ car $\mathrm{FV}(A_j) \subseteq \{\vec{x}\}$.

(2) Comme $\Gamma \vdash c : I\vec{a}$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle c\rangle : \langle I\vec{a}\rangle$. Or, $\langle I\vec{a}\rangle = \langle I\rangle\langle\vec{a}\rangle = \langle I\rangle\vec{x}\gamma'$.

(3) Comme $\Gamma \vdash f_i : T_i$, par hypothèse de récurrence, $\langle\Gamma\rangle \vdash_\Upsilon \langle f_i\rangle : \langle T_i\rangle$. Or, $\langle T_i\rangle\gamma' = \langle T_i\rangle$ car $T_i$ est clos ($\vdash T_i : \Box$).

(4) Soit $\Delta = \vec{x} : \langle\vec{A}\rangle, y : \langle I\rangle\vec{x}$ et $\Delta' = \Delta, \vec{f} : \langle\vec{T}\rangle$. Nous avons $\Delta \vdash_\Upsilon \langle K\rangle : \Box$. Montrons que $\Delta'$ est valide. En effet, si c'est le cas, alors, par affaiblissement, $\Delta' \vdash_\Upsilon \langle K\rangle : \Box$ et, par (prod), $\vdash_\Upsilon \tau_{SElim_I} : \star$. Soit $\Delta_i = \Delta, f_1 : \langle T_1\rangle, \ldots, f_i : \langle T_i\rangle$. Montrons par récurrence sur $i$ que $\Delta_i$ est valide. Si $i = 0$, c'est immédiat. Montrons alors que si $\Delta_i$ est valide alors $\Delta_{i+1}$ l'est aussi. Comme $\vdash_\Upsilon \langle T_{i+1}\rangle : \Box$, par affaiblissement, $\Delta_i \vdash_\Upsilon \langle T_{i+1}\rangle : \Box$. Donc, par (var), $\Delta_{i+1} \vdash_\Upsilon f_{i+1} : T_{i+1}$ et $\Delta_{i+1}$ est valide.

Les autres cas se traitent sans difficulté. ■

**Lemme 101** Les règles de $\Upsilon$ sont bien typées.

**Preuve.** Nous devons montrer que chacune des règles vérifie les conditions (S3) à (S5). Voyons le cas de $WElim_I(Q, \vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_W[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$. Le cas de $SElim_I^Q(\vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_S[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$ se traite de manière similaire. Soit $B = (\vec{x} : \vec{A})I\vec{x} \to \star$. Nous avons $\tau_{WElim_I} = (Q : \langle B\rangle)(\vec{x} : \langle\vec{A}\rangle)(y : \langle I\rangle\vec{x})(\vec{f} : \langle\vec{T}\rangle)Q\vec{x}y$, $T_i = \Delta\{I, X, C_i, Q, Constr(i, I)\}$, $C_i = (\vec{z} : \vec{B})X\vec{m}$, $B_j = (\vec{y^j} : \vec{D^j})X\vec{q^j}$ si $X \in \mathrm{FV}(B_j)$, $\tau_{Constr_i^I} = (\vec{z} : \vec{B'})Ind_I(\langle\vec{m}\rangle)$, $B'_j = \langle B_j\rangle$ si $X \notin \mathrm{FV}(B_j)$, $B'_j = (\vec{y^j} : \langle\vec{D^j}\rangle)Ind_I(\langle\vec{q^j}\rangle)$ sinon, et $\tau_{Ind_I} = (\vec{x} : \langle\vec{A}\rangle)\star$. Soit $l = WElim_I(Q, \vec{x}, c, \vec{f})$, $r = \Delta'_W[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$, $c = Constr_i^I(\vec{z})$ et $\gamma = \{y \mapsto c\}$. Nous prenons $\Gamma = Q : \langle B\rangle, \vec{z} : \vec{B'}, \vec{f} : \langle\vec{T}\rangle$ et $\rho = \{\vec{x} \mapsto \langle\vec{m}\rangle\}$.

**(S2)** Nous devons montrer $\Gamma \vdash_\Upsilon r : Q\langle\vec{m}\rangle c$. Nous avons $r = \Delta'_W[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$ et $T_i = \Delta\{I, X, C_i, Q, Constr(i, I)\}$. Cela ne présente pas de difficultés.

**(S3)** Nous devons montrer que si $\Delta \vdash_\Upsilon l\sigma : T$ alors $\sigma : \Gamma \to \Delta$. Nous avons $\Delta \vdash_\Upsilon WElim_I(Q\sigma, \vec{x}\sigma, Constr_i^I(\vec{z}\sigma), \vec{f}\sigma) : T$. Alors, par inversion, on déduit $\Delta \vdash_\Upsilon Q\sigma : \langle B\rangle\sigma$, $\Delta \vdash_\Upsilon \vec{z}\sigma : \vec{B'}\sigma$ et $\Delta \vdash_\Upsilon \vec{f}\sigma : \langle\vec{T}\rangle\sigma$, c'est-à-dire, $\sigma : \Gamma \to \Delta$.

**(S4)** Nous devons montrer que si $\Delta \vdash_\Upsilon l\sigma : T$ alors, pour tout $x$, $x\rho\sigma \downarrow x\sigma$. Par inversion, nous avons $\Delta \vdash_\Upsilon c\sigma : \langle I\rangle\vec{x}\sigma$ et $Ind_I(\langle\vec{m}\rangle\sigma)\ \mathcal{C}^*_\Delta\ \langle I\rangle\vec{x}\sigma$. Par ailleurs, $\langle I\rangle\vec{x}\sigma \to^*_\beta Ind_I(\vec{x}\sigma)$. Donc, $Ind_I(\langle\vec{m}\rangle\sigma)\ \mathcal{C}^*\ Ind_I(\vec{x}\sigma)$ et, par confluence, $Ind_I(\langle\vec{m}\rangle\sigma) \downarrow Ind_I(\vec{x}\sigma)$. Comme $Ind_I$ est constant et $\langle\vec{m}\rangle\sigma = \vec{x}\rho\sigma$, on obtient $\vec{x}\sigma \downarrow \vec{x}\rho\sigma$. ■

**Lemme 102** Les règles de $\Upsilon$ sont bien formées.

**Preuve.** Voyons le cas de $WElim_I(Q, \vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_W[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$. Le cas de $SElim_I^Q(\vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_S[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$ se traite de manière-

re similaire. Soit $B = (\vec{x} : \vec{A})I\vec{x} \to \star$. Nous avons $\Gamma = Q : \langle B\rangle, \vec{z} : \vec{B'}, \vec{f} : \langle\vec{T}\rangle$ et $\rho = \{\vec{x} \mapsto \langle\vec{m}\rangle\}$. Nous devons montrer que chaque variable $x \in \mathrm{dom}(\Gamma)$ est faiblement accessible dans un des arguments de $WElim_I$, que $x\Gamma$ est égal à $T\rho$ où $T$ est le type de $x$ dérivé de $l$ et que $\Gamma \vdash_\Upsilon l\rho : (Q\vec{x}y)\gamma\rho$.

L'accessibilité est immédiate pour $Q$ et $\vec{f}$. Les $z_j$ sont faiblement accessibles puisque toutes les positions d'un constructeur sont accessibles (voir la définition de $\mathrm{Acc}(Constr_i^I)$). Le type de $z_j$ dérivé de $l$ est $B'_j$ qui ne dépend pas de $\vec{x}$. Donc, $B'_j\rho = B'_j = z_j\Gamma$.

Voyons $\Gamma \vdash_\Upsilon l\rho : (Q\vec{x}y)\gamma\rho$ maintenant. Nous avons $l\rho = WElim_I(Q, \langle\vec{m}\rangle,$ $Constr_i^I(\vec{z}), \vec{f})$ et $(Q\vec{x}y)\gamma\rho = Q\langle\vec{m}\rangle c$. Pour appliquer (symb), nous devons montrer (1) $\vdash_\Upsilon \tau_{WElim_I} : \star$, (2) $\Gamma \vdash_\Upsilon Q : \langle B\rangle$, (3) $\Gamma \vdash_\Upsilon \langle\vec{m}\rangle : \langle\vec{A}\rangle\rho$, (4) $\Gamma \vdash_\Upsilon c : \langle I\rangle\langle\vec{m}\rangle$ et (5) $\Gamma \vdash_\Upsilon \vec{f} : \langle\vec{T}\rangle$.

Montrons alors que $\Gamma$ est valide. Tout d'abord, $WElim_I$ n'est défini que s'il existe un terme bien typé de la forme $Elim(I, Q', \vec{a}, c')\{\vec{f}\}$. Et, dans ce cas, nous avons $\vdash Q' : B$ et $\vdash \vec{T} : \star$. Donc, $\vdash B : \Box$ et $\vdash_\Upsilon \langle B\rangle : \Box$. Ainsi, $Q : \langle B\rangle$ est valide. Par ailleurs, si $WElim_I$ est bien défini alors $Ind_I$ est également bien défini, et donc $Constr_i^I$ aussi. Or, nous avons montré dans le lemme précédent que, dans ce cas, $\vdash_\Upsilon \tau_{Constr_i^I} : \star$. Par affaiblissement, $Q : \langle B\rangle \vdash_\Upsilon \tau_{Constr_i^I} : \star$. Par inversion, $Q : \langle B\rangle, \vec{z} : \vec{B'} \vdash_\Upsilon Ind_I(\langle\vec{m}\rangle) : \star$ et $Q : \langle B\rangle, \vec{z} : \vec{B'}$ est valide. Enfin, comme $\vdash \vec{T} : \star$, $\vdash_\Upsilon \langle\vec{T}\rangle : \star$ et, par affaiblissement, $Q : \langle B\rangle, \vec{z} : \vec{B'} \vdash_\Upsilon \langle\vec{T}\rangle : \star$. Donc $\Gamma$ est valide.

(1) Déjà montré dans le lemme précédent.

(2) Par le lemme d'environnement.

(3) De $\vec{z} : \vec{B'} \vdash_\Upsilon Ind_I(\langle\vec{m}\rangle) : \star$, par inversion, nous déduisons $\vec{z} : \vec{B'} \vdash_\Upsilon \langle\vec{m}\rangle : \langle\vec{A}\rangle\rho$. Donc, par affaiblissement, $\Gamma \vdash_\Upsilon \langle\vec{m}\rangle : \langle\vec{A}\rangle\rho$.

(4) Comme $\Gamma \vdash_\Upsilon \vec{z} : \vec{B'}$, par (symb), $\Gamma \vdash_\Upsilon c : Ind_I(\langle\vec{m}\rangle)$. Or, $\langle I\rangle\langle\vec{m}\rangle \to_\beta^* Ind_I(\langle\vec{m}\rangle)$. D'après (3), $\Gamma \vdash_\Upsilon \langle\vec{m}\rangle : \langle\vec{A}\rangle\rho$. Donc, par (app), $\Gamma \vdash_\Upsilon \langle I\rangle\langle\vec{m}\rangle : \star$. Donc, par (conv), $\Gamma \vdash_\Upsilon c : \langle I\rangle\langle\vec{m}\rangle$.

(5) Par le lemme d'environnement. ■

**Lemme 103** $\Upsilon$ vérifie les conditions de normalisation forte de la Définition 91.

**Preuve.**

**(A0)** Déjà montré.

**(A1)** Nous avons déjà vu que $\to$ est confluente.

**(A2)** Pour la structure inductive, nous prenons :

- $Ind_I >_\mathcal{C} Ind_J$ si $J$ est un sous-terme strict de $I$ est un pré-ordre bien fondé,
- $\mathrm{Ind}(Ind_I) = \emptyset$,
- $\mathrm{Acc}(Constr_i^I) = \{1, \ldots, n\}$ où $n$ est l'arité de $Constr_i^I$.

Montrons que cette structure est admissible. Soit $C$ un prédicat constant. Alors $C = Ind_I$ avec $I = Ind(X : A)\{\vec{C}\}$ et $A = (\vec{x} : \vec{A})\star$, et $C$ est de type $(\vec{x} : \langle\vec{A}\rangle)\star$. Soit $c = Constr_i^I$ un des constructeurs de $C$. Par hypothèse, $C_i = (\vec{z} : \vec{B})X\vec{m}$ et $B_j = (\vec{y^j} : \vec{D^j})X\vec{q^j}$ si $X \in \mathrm{FV}(B_j)$. Donc $c$ est de type $(\vec{z} : \vec{B'})$ avec $B'_j = \langle B_j\rangle$ si $X \notin \mathrm{FV}(B_j)$, et $B'_j = (\vec{y^j} : \langle\vec{D^j}\rangle)C(\langle\vec{q^j}\rangle)$ avec $X \notin \mathrm{FV}(\vec{D^j}\vec{q^j})$ sinon. Soit $\vec{v} = \vec{m}$, $j \in \mathrm{Acc}(c)$ et $U_j = B'_j$.

**(I3)** $\forall D \in \mathcal{CF}^{\square}, D =_{\mathcal{C}} C \Rightarrow \mathrm{Pos}(D, U_j) \subseteq \mathrm{Pos}^{+}(U_j)$. Nécessairement, $D = C$. Soit $X \notin \mathrm{FV}(B_j)$ et $\mathrm{Pos}(C, U_j) = \emptyset$, soit $B_j = (\vec{y} : \vec{D})X\vec{q}$, $X \notin \mathrm{FV}(\vec{D}\vec{q})$. Dans tous les cas, $\mathrm{Pos}(C, U_j) \subseteq \mathrm{Pos}^{+}(U_j)$.

**(I4)** $\forall D \in \mathcal{CF}^{\square}, D >_{\mathcal{C}} C \Rightarrow \mathrm{Pos}(D, U_j) \subseteq \mathrm{Pos}^{0}(U_j)$. Si $D = Ind_J >_{\mathcal{C}} C = Ind_I$ alors $I$ est un sous-terme strict de $J$. Donc, $J$ ne peut pas apparaître dans $I$ et $\mathrm{Pos}(D, U_j) = \emptyset$.

**(I5)** $\forall F \in \mathcal{DF}^{\square}, \mathrm{Pos}(F, U_j) \subseteq \mathrm{Pos}^{0}(U_j)$. Par hypothèse sur les types de CIC$^{-}$.

**(I6)** $\forall Y \in \mathrm{FV}^{\square}(U_j), \exists\, \iota_Y \leq \alpha_C, v_{\iota_Y} = Y$. Par hypothèse sur les types de CIC$^{-}$.

**(I2)** $\forall Y \in \mathrm{FV}^{\square}(U_j), \iota_Y \in \mathrm{Ind}(C) \Rightarrow \mathrm{Pos}(Y, U_j) \subseteq \mathrm{Pos}^{+}(U_j)$. Car $\mathrm{Ind}(C) = \emptyset$.

**(A3)** Pour $\succeq$, on prend l'égalité. Montrons que les règles qui définissent $SElim_I^Q$ forment un système :

- **récursif** : On le montre ci-après pour tous les symboles.
- **petit** : Nous avons $SElim_I^Q(\vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_S[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$. Nous désignerons par $\vec{l}$ les arguments de $SElim_I^Q$ et par $r$ le membre droit de la règle. Nous devons montrer que, pour tout $X \in \mathrm{FV}^{\square}(r)$, il existe un unique $\kappa_X$ tel que $l_{\kappa_X} = X$. Nous avons $\mathrm{FV}^{\square}(r) = \{\vec{f}\} \cup (\{\vec{z}\} \cap \mathcal{X}^{\square})$. Pour les $f_j$, c'est immédiat. Pour les $z_j \in \mathcal{X}^{\square}$, cela découle de la restriction de l'élimination forte aux types inductifs petits : $\vec{z} = \vec{x}\vec{z'}$ avec $\{\vec{z'}\} \cap \mathcal{X}^{\square} = \emptyset$.
- **simple** :
  - **(B1)** Les symboles qui apparaissent dans les arguments de $WElim_I$ ou $SElim_I^Q$ sont constants.
  - **(B2)** Au plus une règle peut s'appliquer au sommet d'un terme de la forme $WElim_I(Q, \vec{a}, c, \vec{f})$ ou $SElim_I^Q(\vec{a}, c, \vec{f})$.

**(A4)** Nous avons $\mathcal{F}_1 = \emptyset$. Nous n'avons donc qu'à vérifier les conditions (a) et (b) :

**(a)** $(\mathcal{F}, \mathcal{R})$ **est récursif :** Voyons le cas de $WElim_I(Q, \vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_W[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$. Nous désignerons respectivement par $l$ et par $r$ le membre gauche et le membre droit de cette règle. Le cas de $SElim_I^Q(\vec{x}, Constr_i^I(\vec{z}), \vec{f}) \to \Delta'_S[I, X, C_i, f_i, Q, \vec{f}, \vec{z}]$ se traite de manière similaire.

Pour la précédence $\geq_{\mathcal{F}}$, nous prenons $WElim_I >_{\mathcal{F}} WElim_J$, $WElim_I >_{\mathcal{F}} SElim_J^Q$, $SElim_I >_{\mathcal{F}} WElim_J$ et $SElim_I >_{\mathcal{F}} SElim_J^Q$ si $J$ est un sous-terme strict de $I$, et tous les symboles définis supérieurs aux symboles constants.

Pour le statut, nous prenons $stat_{WElim_I} = lex(mul(x_k))$ où $k$ est la position de l'argument de type $\langle I \rangle \vec{x}$. De même pour $SElim_I$. Il est donc immédiat que cet assignement est compatible avec $\geq_{\mathcal{F}}$.

Nous avons déjà montrer que les règles étaient bien formées. Montrons maintenant que $r$ appartient à la clôture calculable de $l$, c'est-à-dire, $\Gamma \vdash_{\mathrm{c}} r : Q\langle \vec{m} \rangle c$ où $c = Constr_i^I(\vec{z})$. Tout d'abord, remarquons que $\mathcal{R}$ et $\tau$ sont compatibles avec $\geq_{\mathcal{F}}$. C'est clair pour $\mathcal{R}$. Pour $\tau$, cela est dû à notre restriction sur $SElim_I^Q$ : les inductifs de $Q$ sont des sous-termes de $I$. Ainsi, par les Lemmes 89 et 87, nous avons $\Gamma \vdash_{\mathrm{c}} x\Gamma : s$ pour tout $x \in \mathrm{dom}^s(\Gamma)$, et $\Gamma \vdash_{\mathrm{c}} \tau_g : s$ pour tout $g \leq_{\mathcal{F}} WElim_I$. À partir de là, on vérifie aisément que $\Gamma \vdash_{\mathrm{c}} r : Q\langle \vec{m} \rangle c$.

**(b)** $(\mathcal{F},\mathcal{R})$ **est sûr :** Soit $\vec{T}U$ la séquence $\langle Q\rangle, \langle\vec{A}\rangle, \langle I\rangle\vec{x}, \langle\vec{T}\rangle, Q\vec{x}y$. Nous devons montrer :

– $\forall X \in \mathrm{FV}^{\Box}(\vec{T}U)$, $X\gamma\rho \in \mathrm{dom}^{\Box}(\Gamma)$,
– $\forall X, X' \in \mathrm{FV}^{\Box}(\vec{T}U)$, $X\gamma\rho = X'\gamma\rho \Rightarrow X = X'$.

Nous avons $\mathrm{FV}^{\Box}(\vec{T}U) = \{Q\} \cup \{\vec{x}\} \cap \mathcal{X}^{\Box}$, $Q\gamma\rho = Q \in \mathrm{dom}^{\Box}(\Gamma)$ et $x_i\gamma\rho = \langle m_i\rangle$. Donc les propriétés précédentes sont vérifiées du fait de l'hypothèse de sûreté que nous faisons sur les types de CIC$^-$. ■

Il ne nous reste plus maintenant qu'à montrer que la traduction réfléchit la normalisation forte :

**Lemme 104** Si $\Gamma \vdash t : T$ et $t \to_{\beta\iota'} t'$ alors $\langle t\rangle \to^+ \langle t'\rangle$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$.

**(Ind)** Comme $I \in \mathcal{NF}$, aucune réduction n'est possible.

**(Constr)** Comme $\Gamma \vdash I : T$, par inversion, $I \in \mathcal{NF}$ et aucune réduction n'est possible.

**($\star$-Elim)** Nous avons $\langle t\rangle = \mathit{WElim}_I(\langle Q\rangle, \langle\vec{a}\rangle, \langle c\rangle, \langle\vec{f}\rangle)$. Comme $\Gamma \vdash I : T$, par inversion, $I \in \mathcal{NF}$ et aucune réduction n'est possible dans $I$. Si $Q \to_{\beta\iota'} Q'$ alors, comme $\vdash Q : (\vec{x} : \vec{A})I\vec{x} \to \star$, par hypothèse de récurrence, $\langle Q\rangle \to^+ \langle Q'\rangle$ et $\langle t\rangle \to^+ \langle t'\rangle$. Si $\vec{a} \to_{\beta\iota'} \vec{a}'$ alors, comme $\Gamma \vdash \vec{a} : \vec{A}\gamma$, par hypothèse de récurrence, $\langle\vec{a}\rangle \to^+ \langle\vec{a}'\rangle$ et $\langle t\rangle \to^+ \langle t'\rangle$. Enfin, si $c \to_{\beta\iota'} c'$ alors, comme $\Gamma \vdash c : I\vec{a}$, par hypothèse de récurrence, $\langle c\rangle \to^+ \langle c'\rangle$ et $\langle t\rangle \to^+ \langle t'\rangle$.

**($\Box$-Elim)** Nous avons $\langle t\rangle = \mathit{SElim}_I^Q(\langle\vec{a}\rangle, \langle c\rangle, \langle\vec{f}\rangle)$. Comme $\Gamma \vdash I : T$, par inversion, $I \in \mathcal{NF}$ et aucune réduction n'est possible dans $I$. De même, comme $Q \in \mathcal{NF}$, aucune réduction n'est possible dans $Q$. Si $\vec{a} \to_{\beta\iota'} \vec{a}'$ alors, comme $\Gamma \vdash \vec{a} : \vec{A}\gamma$, par hypothèse de récurrence, $\langle\vec{a}\rangle \to^+ \langle\vec{a}'\rangle$ et $\langle t\rangle \to^+ \langle t'\rangle$. Enfin, si $c \to_{\beta\iota'} c'$ alors, comme $\Gamma \vdash c : I\vec{a}$, par hypothèse de récurrence, $\langle c\rangle \to^+ \langle c'\rangle$ et $\langle t\rangle \to^+ \langle t'\rangle$.

Les autres cas se traitent sans difficultés. ■

## 7.2 Calcul des Constructions Inductives + Récriture

Nous venons de voir qu'une grande partie du Calcul des Constructions Inductives est formalisable dans un CAC. Nous allons voir que nous pouvons rajouter à ce CAC des règles de récriture qui ne sont pas formalisables dans CIC. Prenons les symboles $\mathit{nat} : \star$, $0 : \mathit{nat}$, $s : \mathit{nat} \to \mathit{nat}$, $+, \times : \mathit{nat} \to \mathit{nat} \to \mathit{nat}$, $\mathit{list} : \star \to \star$, $\mathit{nil} : (A:\star)\mathit{list}(A)$, $\mathit{cons} : (A:\star)A \to \mathit{list}(A) \to \mathit{list}(A)$, $\mathit{app} : (A:\star)\mathit{list}(A)) \to \mathit{list}(A) \to \mathit{list}(A)$, $\mathit{len} : (A:\star)\mathit{list}(A) \to \mathit{nat}$ la longueur d'une liste, $\mathit{in} : (A:\star)A \to \mathit{list}(A) \to \star$ le prédicat d'appartenance d'un élément à une liste, $\mathit{incl} : (A:\star)\mathit{list}(A) \to \mathit{list}(A) \to \star$ le prédicat d'appartenance des éléments d'une liste à un autre liste, $\mathit{sub} : (A:\star)\mathit{list}(A) \to \mathit{list}(A) \to \star$ le prédicat "être une sous-liste de", $\mathit{eq} : (A:\star)A \to A \to \star$ l'égalité polymorphe, $\top : \star$ la proposition toujours vraie, $\bot : \star$ la proposition toujours fausse, $\neg : \star \to \star$, $\vee, \wedge : \star \to \star \to \star$, et les règles suivantes :

$$
\begin{array}{rcl@{\qquad}rcl}
x+0 & \to & x & x\times 0 & \to & 0\\
0+x & \to & x & 0\times x & \to & 0\\
x+s(y) & \to & s(x+y) & x\times s(y) & \to & (x\times y)+x\\
s(x)+y & \to & s(x+y) & s(0)\times x & \to & x\\
(x+y)+z & \to & x+(y+z) & x\times s(0) & \to & x
\end{array}
$$

$$
\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl}
\neg\top & \to & \bot & P\vee\top & \to & \top & P\wedge\top & \to & P\\
\neg\bot & \to & \top & P\vee\bot & \to & P & P\wedge\bot & \to & \bot
\end{array}
$$

$$
\begin{array}{rcl}
eq(A,0,0) & \to & \top\\
eq(A,0,s(x)) & \to & \bot\\
eq(A,s(x),0) & \to & \bot\\
eq(A,s(x),s(y)) & \to & eq(nat,x,y)
\end{array}
$$

$$
\begin{array}{rcl}
app(A,nil(A'),\ell) & \to & \ell\\
app(A,cons(A',x,\ell),\ell') & \to & cons(A,x,app(A,\ell,\ell'))\\
app(A,app(A',\ell,\ell'),\ell'') & \to & app(A,\ell,app(A,\ell',\ell''))
\end{array}
$$

$$
\begin{array}{rcl}
len(A,nil(A')) & \to & 0\\
len(A,cons(A',x,\ell)) & \to & s(len(A,\ell))\\
len(A,app(A',\ell,\ell')) & \to & len(A,\ell)+len(A,\ell')
\end{array}
$$

$$
\begin{array}{rcl}
in(A,x,nil(A')) & \to & \bot\\
in(A,x,cons(A',y,l)) & \to & eq(A,x,y)\vee in(A,x,l)
\end{array}
$$

$$
\begin{array}{rcl}
sub(A,nil(A'),l) & \to & \top\\
sub(A,cons(A',x,l),nil(A'')) & \to & \bot\\
sub(A,cons(A',x,l),cons(A'',x',l')) & \to & (eq(A,x,x')\wedge sub(A,l,l'))\\
 & & \vee sub(A,cons(A,x,l),l')
\end{array}
$$

$$
\begin{array}{rcl}
incl(A,nil(A'),l) & \to & \top\\
incl(A,cons(A',x,l),l') & \to & in(A,x,l')\wedge incl(A,l,l')
\end{array}
$$

$$
\begin{array}{rcl}
eq(L,nil(A),nil(A')) & \to & \top\\
eq(L,nil(A),cons(A',x,l)) & \to & \bot\\
eq(L,cons(A',x,l),nil(A)) & \to & \bot\\
eq(L,cons(A,x,l),cons(A',x',l')) & \to & eq(A,x,x')\wedge eq(list(A),l,l')
\end{array}
$$

Le système de récriture est récursif, simple, petit, sûr et confluent (cela peut être prouvé automatiquement par CiME [30]). Comme les règles sont linéaires-gauches, la combinaison avec $\to_\beta$ est confluente. Donc, les conditions de normalisation forte sont vérifiées.

On notera en particulier la dernière règle où $\Gamma = A:\star, x:A, x':A, \ell:list(A), \ell':list(A)$ et $\rho = \{A'\mapsto A, L\mapsto list(A)\}$. Elle est bien formée : par exemple, $cons(A',x',\ell') : L \rhd_1^\rho x' : A'$. Et elle vérifie le Schéma Général : $\{cons(A,x,\ell):L,\ cons(A',x',\ell'):L\}\ (\rhd_1^\rho)_{\rm mul}\ \{x:A,x':A'\},\{\ell:list(A),\ell':list(A)\}$.

Cependant, il manque de nombreuses autres règles importantes pour avoir une procédure de décision pour les tautologies propositionnelles classiques (Figure 1.3) ou d'autres règles de simplification de l'égalité (Figure 1.4). Pour cela, il faudrait avoir de la récriture modulo commutativité et associativité et affaiblir certaines conditions dans la définition du Schéma Général.

## 7.3 Déduction Naturelle Modulo

La Déduction Naturelle Modulo (NDM) pour la logique du premier ordre [47] peut être présentée comme une extension de la Déduction Naturelle avec la règle d'inférence suivante :

$$\frac{\Gamma \vdash P}{\Gamma \vdash Q} \quad \text{if } P \equiv Q$$

où $\equiv$ est une relation d'équivalence sur les propositions stable par substitution et contexte. C'est une extension très puissante de la logique du premier ordre puisque la logique d'ordre supérieur et la théorie des ensembles skolémisée peuvent toutes deux être décrites comme des théories modulo (en utilisant des substitutions explicites [1]).

Dans [48], G. Dowek et B. Werner étudient la normalisation forte de l'élimination des coupures dans le cas où $\equiv$ est engendré par un système de récriture du premier ordre confluent et faiblement normalisant. En particulier, ils prouvent la normalisation dans deux cas généraux : quand le système est positif et quand il est sans quantificateur. Dans [49], ils fournissent un exemple de système confluent et faiblement normalisant pour lequel l'élimination des coupures n'est pas normalisante. Le problème vient du fait que la règle d'élimination du quantificateur $\forall$ introduit une substitution :

$$\frac{\Gamma \vdash \forall x.P(x)}{\Gamma \vdash P(t)}$$

Ainsi, quand un symbole de prédicat est défini par une règle dont le membre droit contient des quantificateurs, sa combinaison avec la $\beta$-réduction peut ne pas préserver la normalisation. Un critère de normalisation pour la récriture d'ordre supérieur comme celui que nous fournissons est donc nécessaire.

Maintenant, puisque NDM est un CAC (les connecteurs logiques peuvent être vus comme des symboles de prédicat constants), on peut comparer les conditions générales données dans [48] avec les nôtres.

**(A1)** Dans [48], seulement $\rightarrow_{\mathcal{R}}$ doit être confluent. Nous ne savons pas si cela implique toujours la confluence de $\rightarrow_{\mathcal{R}} \cup \rightarrow_{\beta}$. C'est vrai si $\mathcal{R}$ est linéaire-gauche car, alors, nous avons une union de CRS linéaires-gauches et confluents n'ayant aucune paire critique l'un par rapport à l'autre (résultat général dû à V. van Oostrom [116] et montré dans le cas particulier de $\rightarrow_{\beta}$ par F. Müller [92]). Mais nous ne connaissons pas de travaux qui montrent que, en présence de types dépendants et de récriture sur les types, $\rightarrow_{\mathcal{R}} \cup \rightarrow_{\beta}$ est confluent même si $\mathcal{R}$ n'est pas linéaire-gauche (V. Breazu-Tannen et J. Gallier ont montré dans [26] la préservation de la confluence pour le $\lambda$-calcul polymorphe d'ordre supérieur avec récriture du premier ordre au niveau objet).

**(A2)** Les types de NDM sont primitifs et forment une structure inductive admissible si on les prend tous équivalents.

**(A3)** Dans [48], la normalisation forte de l'élimination des coupures est prouvée dans deux cas généraux : quand $(\mathcal{DF}^{\Box}, \mathcal{R}_{\mathcal{DF}^{\Box}})$ est sans quantificateur et quand il est positif. Les systèmes sans quantificateur sont primitifs. Donc, dans ce cas, (A3) est satisfait. Dans le cas positif par contre, nous demandons en plus que les arguments des membres gauches des règles soient faits de symboles constants et qu'au plus une règle puisse s'appliquer au sommet d'un terme (conditions de "simplicité"). D'un autre côté, nous fournissons un nouveau cas : $(\mathcal{DF}^{\Box}, \mathcal{R}_{\mathcal{DF}^{\Box}})$ peut être récursif et simple (il est forcément petit).

**(A4)** Les règles sans quantificateur sont de premier ordre et ceux avec quantificateurs ne le sont pas. Dans [48], ces deux types de règles sont traitées de la même manière. Mais le contre-exemple donné dans [49] nous montre qu'elles ne le devraient pas. Dans nos conditions, nous demandons à ce que les symboles définis par des règles avec quantificateurs satisfassent le Schéma Général.

**Théorème 105** Un système NDM satisfaisant les conditions (A1), (A3) et (A4) est fortement normalisant.

# Chapitre 8

# Correction des conditions de normalisation forte

Notre preuve de normalisation forte est basée sur l'extension au Calcul des Constructions par T. Coquand et J. Gallier [33] de la méthode de Tait et Girard des candidats de réductibilités [62]. L'idée est d'interpréter chaque type $T$ par un ensemble $[\![T]\!]$ de termes fortement normalisants et de prouver que tout terme de type $T$ appartient à $[\![T]\!]$. Nous renvoyons le lecteur vers le Chapitre 3 de la thèse de B. Werner [118] pour une introduction sur les candidats, et vers l'article [54] de J. Gallier pour une présentation plus détaillée. Une différence importante entre les candidats de T. Coquand et J. Gallier et les candidats de B. Werner pour le Calcul des Constructions Inductives est que les premiers sont constitués de termes bien typés alors que les seconds sont constitués de termes purs.

## 8.1 Espace des termes interprétés

Afin de disposer de l'environnement dans lequel un terme est typable, nous raisonnerons plutôt sur des fermetures (paires environnement-terme) que sur des termes seuls.

**Définition 106 (Fermeture)** Une *fermeture* est un couple $\Gamma \vdash t$ constitué d'un environnement $\Gamma \in \mathcal{E}$ et d'un terme $t \in \mathcal{T}$. Une fermeture $\Gamma \vdash t$ est *typable* s'il existe un terme $T \in \mathcal{T}$ tel que $\Gamma \vdash t : T$. On notera par $\overline{\mathbb{T}}$ l'ensemble des fermetures typables.

L'ensemble des *fermetures de type* $\Gamma \vdash T$ est $\overline{\mathbb{T}}_{\Gamma\vdash T} = \{\Gamma' \vdash t \in \overline{\mathbb{T}} \mid \Gamma' \supseteq \Gamma \text{ et } \Gamma' \vdash t : T\}$. L'ensemble de fermetures de type $\Gamma \vdash T$ dont les termes sont fortement normalisables sera noté $\overline{\mathbb{SN}}_{\Gamma\vdash T}$. La *restriction* d'un ensemble $S \subseteq \overline{\mathbb{T}}_{\Gamma\vdash T}$ à un environnement $\Gamma' \supseteq \Gamma$ est $S|_{\Gamma'} = S \cap \overline{\mathbb{T}}_{\Gamma'\vdash T} = \{\Gamma''\vdash t \in S \mid \Gamma'' \supseteq \Gamma'\}$.

On vérifie aisément les propriétés élémentaires suivantes :

**Lemme 107**
(a) si $\Gamma'\vdash t \in \overline{\mathbb{T}}_{\Gamma\vdash T}$ et $\Gamma' \subseteq \Gamma'' \in \mathbb{E}$ alors $\Gamma''\vdash t \in \overline{\mathbb{T}}_{\Gamma\vdash T}$.
(b) Si $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $\overline{\mathbb{T}}_{\Gamma'\vdash T} \subseteq \overline{\mathbb{T}}_{\Gamma\vdash T}$ et $\overline{\mathbb{T}}_{\Gamma\vdash T}|_{\Gamma'} = \overline{\mathbb{T}}_{\Gamma'\vdash T}$.

(c) Si $T\ \mathbb{C}_\Gamma\ T'$ alors $\overline{\mathbb{T}}_{\Gamma\vdash T} = \overline{\mathbb{T}}_{\Gamma\vdash T'}$.

Nous devons définir une interprétation pour tous les termes qui peuvent être le type d'un autre terme, c'est-à-dire pour tous les termes $T$ pour lesquels il existe $\Gamma$ et $t$ tels que $\Gamma \vdash t : T$. D'après le lemme de correction des types, il existe $s$ tel que $T = s$ ou $\Gamma \vdash T : s$. Ainsi, il nous faut une interprétation pour les termes des ensembles suivants :

- $\overline{\mathbb{B}} = \{\Gamma\vdash T \in \mathcal{E}\times\mathcal{T} \mid \Gamma \in \mathbb{E} \text{ et } T = \Box\}$,
- $\overline{\mathbb{TY}}^\star = \overline{\mathbb{P}}^0 = \{\Gamma\vdash T \in \mathcal{E}\times\mathcal{T} \mid \Gamma \vdash T : \star\}$,
- $\overline{\mathbb{TY}}^\Box = \overline{\mathbb{K}} = \{\Gamma\vdash K \in \mathcal{E}\times\mathcal{T} \mid \Gamma \vdash K : \Box\}$.

Un terme $T$ tel que $\Gamma\vdash T \in \overline{\mathbb{P}}^0$ peut être obtenu par application d'un terme $U$ à un terme $v$. Par inversion, $U$ doit avoir un type de la forme $(x{:}V)W$. Par correction des types, il existe $s$ tel que $\Gamma \vdash (x{:}V)W : s$. Comme $T$ appartient à la même classe que $U$, $T \in \mathbb{P}^0 \subseteq \mathbb{P} = \mathbb{T}_1^\Box$ et $U \in \mathbb{T}_1^s$, par classification, on obtient $s = \Box$. Donc, d'après le lemme sur les sortes maximales, $\Gamma\vdash U$ ne peut pas appartenir à $\overline{\mathbb{P}}^0$. Il nous faut donc aussi donner une interprétation aux termes des ensembles suivants :

- $\overline{\mathbb{P}}^\star = \{\Gamma\vdash T \in \mathcal{E}\times\mathcal{T} \mid \exists\, x, U, K,\ \Gamma \vdash T : (x{:}U)K\ \wedge\ \Gamma\vdash U : \star\ \wedge\ \Gamma\vdash K : \Box\}$,
- $\overline{\mathbb{P}}^\Box = \{\Gamma\vdash T \in \mathcal{E}\times\mathcal{T} \mid \exists\, X, K, L,\ \Gamma \vdash T : (X{:}K)L\ \wedge\ \Gamma\vdash K : \Box\ \wedge\ \Gamma\vdash L : \Box\}$,
- $\overline{\mathbb{P}} = \overline{\mathbb{P}}^0 \cup \overline{\mathbb{P}}^\star \cup \overline{\mathbb{P}}^\Box$,
- $\overline{\mathbb{TY}} = \overline{\mathbb{B}} \cup \overline{\mathbb{K}} \cup \overline{\mathbb{P}}$.

Afin de justifier notre définition de $\overline{\mathbb{P}}$ et de s'assurer que les termes à interpréter sont bien tous dans $\overline{\mathbb{TY}}$, il suffit de voir, d'après le lemme sur les sortes maximales, que la projection de $\overline{\mathbb{P}}$ sur $\mathcal{T}$, c'est-à-dire $\{T \in \mathcal{T} \mid \exists\Gamma \in \mathcal{E}, \Gamma\vdash T \in \overline{\mathbb{P}}\}$, est égale à $\mathbb{P}$.

**Lemme 108** Les ensembles $\overline{\mathbb{P}}^0$, $\overline{\mathbb{P}}^\star$, $\overline{\mathbb{P}}^\Box$, $\overline{\mathbb{K}}$ et $\overline{\mathbb{B}}$ sont disjoints deux à deux.

**Preuve.** Nous venons de voir que $\overline{\mathbb{P}}^0$ est disjoint de $\overline{\mathbb{P}}^\star$ et $\overline{\mathbb{P}}^\Box$. Puisque $\Box$ n'est pas typable, $\overline{\mathbb{B}}$ est disjoint de tous les autres. $\overline{\mathbb{P}}$ et $\overline{\mathbb{K}}$ sont disjoints puisque leurs projections $\mathbb{P}$ et $\mathbb{K}$ sont disjointes. Il nous reste donc à vérifier que $\overline{\mathbb{P}}^\star$ et $\overline{\mathbb{P}}^\Box$ sont bien disjoints. Supposons qu'il existe $\Gamma\vdash T \in \overline{\mathbb{P}}^\star \cap \overline{\mathbb{P}}^\Box$. Alors il existe $x$, $U$, $K$, $X$, $K'$ et $L$ tels que $\Gamma \vdash T : (x{:}U)K$, $\Gamma \vdash U : \star$, $\Gamma \vdash K : \Box$, $\Gamma \vdash T : (X{:}K')L$, $\Gamma \vdash K' : \Box$ et $\Gamma \vdash L : \Box$. Par convertibilité des types, $(x{:}U)K\ \mathbb{C}_\Gamma^*\ (X{:}K')L$. Par compatibilité avec le produit, $U\ \mathbb{C}_\Gamma^*\ K'$. Par correction de la conversion, $\star = \Box$, ce qui n'est pas possible. ■

Nous allons maintenant introduire une mesure sur $\overline{\mathbb{TY}}$ qui nous permettra de faire des définitions récursives bien fondées.

**Définition 109** $\mu(\Gamma\vdash T) = \begin{cases} 0 & \text{si } T = \Box \text{ or } \Gamma \vdash T : \Box \\ \nu(K) & \text{si } \Gamma \vdash T : K \text{ et } \Gamma \vdash K : \Box \end{cases}$

où $\nu$ est défini sur les types de prédicats de la manière suivante :

- $\nu(\star) = 0$
- $\nu((x{:}U)K) = 1 + \nu(K)$
- $\nu((X{:}K)L) = 1 + \max(\nu(K), \nu(L))$

On doit vérifier que cette définition ne dépend pas de $K$. Comme tous les types de $T$ sont convertibles les uns aux autres, il suffit de vérifier que $\nu$ est invariante par conversion :

**Lemme 110** Si $K\ \mathbb{C}^*_\Gamma\ K'$ alors $\nu(K) = \nu(K')$.

**Preuve.** Par récurrence sur la taille de $K$ et $K'$. D'après le lemme sur les sortes maximales, $K$ est de la forme $(\vec{x} : \vec{T})\star$, $K'$ est de la forme $(\vec{x} : \vec{T'})\star$ et $|\vec{x}| = |\vec{x'}|$. Soit $n = |\vec{x}| = |\vec{x'}|$. Par compatibilité avec le produit et $\alpha$-équivalence, on peut supposer que $\vec{x'} = \vec{x}$. Si $n = 0$ alors $K = K'$ et $\nu(K) = \nu(K')$. Supposons alors $n > 0$. Soit $L = (x_2 : T_2) \dots (x_n : T_n)\star$ et $L' = (x_2 : T'_2) \dots (x_n : T'_n)\star$. Par compatibilité avec le produit, $T_1\ \mathbb{C}^*_\Gamma\ T'_1$ et $L\ \mathbb{C}^*_{\Gamma, x_1:T_1}\ L'$. Par correction de la conversion, $T_1$ et $T'_1$ sont typables par la même sorte $s$. Par inversion et régularité, $\Gamma, x_1 : T_1 \vdash L : \Box$ et $\Gamma, x_1 : T_1 \vdash L' : \Box$. Ainsi, par hypothèse de récurrence, $\nu(L) = \nu(L')$ et, si $s = \Box$, $\nu(T_1) = \nu(T'_1)$. Donc, $\nu(K) = \nu(K')$. ■

**Lemme 111** Si $\Gamma \vdash T \in \overline{\mathbb{TY}}$ et $\theta : \Gamma \to \Delta$ alors $\Delta \vdash T\theta \in \overline{\mathbb{TY}}$ et $\mu(\Delta \vdash T\theta) = \mu(\Gamma \vdash T)$.

**Preuve.** Tout d'abord, il est facile de montrer par récurrence sur la structure des types de prédicats que, si $K$ est un type de prédicats et $\theta$ une substitution, alors $K\theta$ est un type de prédicats et $\nu(K\theta) = \nu(K)$. Montrons maintenant le lemme par cas sur $T$.

- $T = \Box$. $\Delta \vdash T\theta = \Delta \vdash \Box \in \overline{\mathbb{TY}}$ et $\mu(\Delta \vdash T\theta) = 0 = \mu(\Gamma \vdash T)$.
- $\Gamma \vdash T : \Box$. Par substitution, $\Delta \vdash T\theta : \Box$, $\Delta \vdash T\theta \in \overline{\mathbb{TY}}$ et $\mu(\Delta \vdash T\theta) = 0 = \mu(\Gamma \vdash T)$.
- $\Gamma \vdash T : K$ et $\Gamma \vdash K : \Box$. Par substitution, $\Delta \vdash T\theta : K\theta$ et $\Delta \vdash K\theta : \Box$. Donc $\Delta \vdash T\theta \in \overline{\mathbb{TY}}$. Maintenant, $\mu(\Delta \vdash T\theta) = \nu(K\theta)$ et $\mu(\Gamma \vdash T) = \nu(K)$. Or, $\nu(K\theta) = \nu(K)$. ■

## 8.2 Candidats de réductibilité

On notera par :

- $\mathcal{SN}$ l'ensemble des termes fortement normalisables pour $\to$,
- $\mathcal{WN}$ l'ensemble des termes faiblement normalisables pour $\to$,
- $\mathcal{CR}$ l'ensemble des termes $t$ tels que deux réductions partant de $t$ confluent toujours.

**Définition 112 (Termes neutres)** Un terme est *neutre* s'il n'est ni une abstraction ni de la forme $c(\vec{t})$ où $c$ est un constructeur.

**Définition 113 (Candidats de réductibilité)** Pour chaque $\Gamma \vdash T \in \overline{\mathbb{TY}}$, nous allons définir par récurrence sur $\mu(\Gamma \vdash T)$ :

- l'ensemble $\mathcal{R}_{\Gamma \vdash T}$ des *candidats de réductibilité de type* $\Gamma \vdash T$,

– la *restriction* $R|_{\Gamma'}$ d'un candidat $R \in \mathcal{R}_{\Gamma\vdash T}$ à un environnement $\Gamma' \supseteq \Gamma$,
– la relation $\leq_{\Gamma\vdash T}$ sur $\mathcal{R}_{\Gamma\vdash T}$,
– l'élément $\top_{\Gamma\vdash T}$ de $\mathcal{R}_{\Gamma\vdash T}$,
– la fonction $\bigwedge_{\Gamma\vdash T}$ de l'ensemble des parties de $\mathcal{R}_{\Gamma\vdash T}$ dans $\mathcal{R}_{\Gamma\vdash T}$.

- $T = \Box$.
  – $\mathcal{R}_{\Gamma\vdash\Box} = \{\overline{\mathbb{SN}}_{\Gamma\vdash\Box}\}$.
  – $R|_{\Gamma'} = R \cap \overline{\mathbb{T}}_{\Gamma'\vdash\Box}$.
  – $R_1 \leq_{\Gamma\vdash\Box} R_2$ si $R_1 \subseteq R_2$.
  – $\top_{\Gamma\vdash\Box} = \overline{\mathbb{SN}}_{\Gamma\vdash\Box}$.
  – $\bigwedge_{\Gamma\vdash\Box}(\Re) = \top_{\Gamma\vdash\Box}$.
- $\Gamma \vdash T : s$.
  – $\mathcal{R}_{\Gamma\vdash T}$ est l'ensemble de tous les sous-ensembles $R$ de $\overline{\mathbb{T}}_{\Gamma\vdash T}$ tels que :
    **(R1)** $R \subseteq \mathcal{SN}$ (normalisation forte) ;
    **(R2)** si $\Gamma'\vdash t \in R$ et $t \to t'$ alors $\Gamma'\vdash t' \in R$ (stabilité par réduction) ;
    **(R3)** si $\Gamma'\vdash t \in \overline{\mathbb{T}}_{\Gamma\vdash T}$, $t$ est neutre et, pour tout $t'$ tel que $t \to t'$, $\Gamma'\vdash t' \in R$, alors $\Gamma'\vdash t \in R$ (stabilité par expansion pour les termes neutres) ;
    **(R4)** si $\Gamma'\vdash t \in R$ et $\Gamma' \subseteq \Gamma'' \in \mathbb{E}$ alors $\Gamma''\vdash t \in R$ (stabilité par affaiblissement).
  – $R|_{\Gamma'} = R \cap \overline{\mathbb{T}}_{\Gamma'\vdash T}$.
  – $R_1 \leq_{\Gamma\vdash T} R_2$ si $R1 \subseteq R_2$.
  – $\top_{\Gamma\vdash T} = \overline{\mathbb{SN}}_{\Gamma\vdash T}$.
  – $\bigwedge_{\Gamma\vdash T}(\Re) = \bigcap \Re$ si $\Re \neq \emptyset$, $\top_{\Gamma\vdash T}$ sinon.
- $\Gamma \vdash T : (x{:}U)K$.
  – $\mathcal{R}_{\Gamma\vdash T}$ est l'ensemble de toutes les fonctions $R$ qui, à $\Gamma'\vdash u \in \overline{\mathbb{T}}_{\Gamma\vdash U}$, associent un élément de $\mathcal{R}_{\Gamma'\vdash Tu}$ et vérifient :
    **(P1)** si $u \to u'$ alors $R(\Gamma'\vdash u) = R(\Gamma'\vdash u')$ (stabilité par réduction),
    **(P2)** si $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $R(\Gamma\vdash u)|_{\Gamma'} = R(\Gamma'\vdash u)$ (compatibilité avec l'affaiblissement).
  – $R|_{\Gamma'} = R|_{\overline{\mathbb{T}}_{\Gamma'\vdash U}}$.
  – $R_1 \leq_{\Gamma\vdash T} R_2$ si, pour tout $\Gamma'\vdash u \in \overline{\mathbb{T}}_{\Gamma\vdash U}$, $R_1(\Gamma'\vdash u) \leq_{\Gamma'\vdash Tu} R_2(\Gamma'\vdash u)$.
  – $\top_{\Gamma\vdash T}(\Gamma'\vdash u) = \top_{\Gamma'\vdash Tu}$.
  – $\bigwedge_{\Gamma\vdash T}(\Re)(\Gamma'\vdash u) = \bigwedge_{\Gamma'\vdash Tu}(\{R(\Gamma'\vdash u) \mid R \in \Re\})$.
- $\Gamma \vdash T : (X : K)L$. Notons $\Sigma_{\Gamma\vdash K}$ l'ensemble des couples $(\Gamma' \vdash U, S)$ tels que $\Gamma'\vdash U \in \overline{\mathbb{T}}_{\Gamma\vdash K}$ et $S \in \mathcal{R}_{\Gamma'\vdash U}$.
  – $\mathcal{R}_{\Gamma\vdash T}$ est l'ensemble de toutes les fonctions $R$ qui, à un couple $(\Gamma' \vdash U, S) \in \Sigma_{\Gamma\vdash K}$, associent un élément de $\mathcal{R}_{\Gamma'\vdash TU}$ et vérifient :
    **(P1)** si $U \to U'$ alors $R(\Gamma'\vdash U, S) = R(\Gamma'\vdash U', S)$ (stabilité par réduction),
    **(P2)** si $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $R(\Gamma\vdash U, S)|_{\Gamma'} = R(\Gamma'\vdash U, S|_{\Gamma'})$ (compatibilité avec l'affaiblissement).
  – $R|_{\Gamma'} = R|_{\Sigma_{\Gamma'\vdash K}}$.
  – $R_1 \leq_{\Gamma\vdash T} R_2$ si, pour tout $(\Gamma'\vdash U, S) \in \Sigma_{\Gamma\vdash K}$, $R_1(\Gamma'\vdash U, S) \leq_{\Gamma'\vdash TU} R_2(\Gamma'\vdash U, S)$.
  – $\top_{\Gamma\vdash T}(\Gamma'\vdash U, S) = \top_{\Gamma'\vdash TU}$.
  – $\bigwedge_{\Gamma\vdash T}(\Re)(\Gamma'\vdash U, S) = \bigwedge_{\Gamma'\vdash TU}(\{R(\Gamma'\vdash U, S) \mid R \in \Re\})$.

Le lemme suivant assure que tous ces objets sont bien définis.

**Lemme 114 (Propriétés des candidats)**
(a) $\mathcal{R}_{\Gamma\vdash T}$, $\leq_{\Gamma\vdash T}$, $\top_{\Gamma\vdash T}$ et $\bigwedge_{\Gamma\vdash T}$ sont bien définis.
(b) Si $T \to T'$ alors $\mathcal{R}_{\Gamma\vdash T} = \mathcal{R}_{\Gamma\vdash T'}$.
(c) Si $R \in \mathcal{R}_{\Gamma\vdash T}$ et $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $R|_{\Gamma'} \in \mathcal{R}_{\Gamma'\vdash T}$.
(d) $\top_{\Gamma\vdash T} \in \mathcal{R}_{\Gamma\vdash T}$.
(e) Si $T \to T'$ alors $\top_{\Gamma\vdash T} = \top_{\Gamma\vdash T'}$.
(f) Si $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $\top_{\Gamma\vdash T}|_{\Gamma'} = \top_{\Gamma'\vdash T}$.
(g) Si $\Re \subseteq \mathcal{R}_{\Gamma\vdash T}$ alors $\bigwedge_{\Gamma\vdash T}(\Re) \in \mathcal{R}_{\Gamma\vdash T}$.
(h) Si $T \to T'$ alors $\bigwedge_{\Gamma\vdash T} = \bigwedge_{\Gamma\vdash T'}$.
(i) Si $\Gamma \subseteq \Gamma' \in \mathbb{E}$ alors $\bigwedge_{\Gamma\vdash T}(\Re)|_{\Gamma'} = \bigwedge_{\Gamma'\vdash T}(\{R|_{\Gamma'} \mid R \in \Re\})$.

**Preuve.** Par récurrence sur $\mu(\Gamma\vdash T)$.

- $T = \Box$.
  (a) Immédiat.
  (b) $\Box$ n'est pas réductible.
  (c) Nécessairement, $R = \overline{\mathbb{SN}}_{\Gamma\vdash\Box}$. Donc $R|_{\Gamma'} = \overline{\mathbb{SN}}_{\Gamma\vdash\Box} \cap \overline{\mathbb{T}}_{\Gamma'\vdash\Box} = \overline{\mathbb{SN}}_{\Gamma'\vdash\Box} \in \mathcal{R}_{\Gamma'\vdash\Box}$.
  (d) Immédiat.
  (e) $\Box$ n'est pas réductible.
  (f) $\top_{\Gamma\vdash\Box}|_{\Gamma'} = \overline{\mathbb{SN}}_{\Gamma\vdash\Box} \cap \overline{\mathbb{T}}_{\Gamma'\vdash\Box} = \overline{\mathbb{SN}}_{\Gamma'\vdash\Box} = \top_{\Gamma'\vdash\Box}$.
  (g) $\bigwedge_{\Gamma\vdash\Box}(\Re) = \top_{\Gamma\vdash\Box}$.
  (h) $\Box$ n'est pas réductible.
  (i) $\bigwedge_{\Gamma\vdash\Box}(\Re)|_{\Gamma'} = \top_{\Gamma\vdash\Box}|_{\Gamma'} = \top_{\Gamma'\vdash\Box} = \bigwedge_{\Gamma'\vdash\Box}(\{R|_{\Gamma'} \mid R \in \Re\})$.
- $\Gamma \vdash T : s$.
  (a) Immédiat.
  (b) Par préservation du typage, $\overline{\mathbb{T}}_{\Gamma\vdash T} = \overline{\mathbb{T}}_{\Gamma\vdash T'}$.
  (c) Par affaiblissement, $R|_{\Gamma'} \subseteq \overline{\mathbb{T}}_{\Gamma'\vdash T}$. Montrons maintenant que $R|_{\Gamma'}$ vérifie (R1) à (R4). Pour (R1), (R2) et (R4), c'est immédiat. Pour (R3), soit $\Gamma''\vdash t \in \overline{\mathbb{T}}_{\Gamma'\vdash T}$ tel que $t$ soit neutre et, pour tout $t'$ tel que $t \to t'$, $\Gamma''\vdash t' \in R|_{\Gamma'}$. Comme $R|_{\Gamma'} \subseteq R$, $\Gamma''\vdash t \in R$. Mais $\Gamma''\vdash t \in \overline{\mathbb{T}}_{\Gamma'\vdash T}$. Donc $\Gamma''\vdash t \in R|_{\Gamma'}$.
  (d) Par définition, $\top_{\Gamma\vdash T} \subseteq \overline{\mathbb{T}}_{\Gamma\vdash T}$ et il est facile de vérifier que $\top_{\Gamma\vdash T}$ vérifie (R1) à (R4).
  (e) Par préservation du typage.
  (f) Immédiat.
  (g) Comme chaque élément de $\Re$ est inclus dans $\overline{\mathbb{T}}_{\Gamma\vdash T}$ et vérifie (R1) à (R4), il est facile de vérifier qu'il en est de même pour $\bigcap\Re$.
  (h) Immédiat.
  (i) $(\bigcap\Re)|_{\Gamma'} = (\bigcap\Re) \cap \overline{\mathbb{T}}_{\Gamma'\vdash T} = \bigcap\{R \cap \overline{\mathbb{T}}_{\Gamma'\vdash T} \mid R \in \Re\} = \bigcap\{R|_{\Gamma'} \mid R \in \Re\}$.
- $\Gamma \vdash T : (x{:}U)K$.
  (a) Il nous suffit de vérifier que $\mu(\Gamma'\vdash Tu) < \mu(\Gamma\vdash T)$ et que les définitions ne dépendent pas du choix d'un type pour $T$.
  Par affaiblissement, $\Gamma' \vdash T : (x:U)K$ et $\Gamma' \vdash (x:U)K : \Box$. Par (app),

$\Gamma' \vdash Tu : K\{x \mapsto u\}$. Par inversion et régularité, $\Gamma', x : U \vdash K : \Box$. Par substitution, $\Gamma' \vdash K\{x \mapsto u\} : \Box$. Donc $\Gamma' \vdash Tu \in \overline{\mathbb{TY}}$ et $\mu(\Gamma' \vdash Tu) = \nu(K\{x \mapsto u\})$. Par invariance par substitution, $\nu(K\{x \mapsto u\}) = \nu(K)$. Donc $\mu(\Gamma \vdash T) = \nu((x{:}U)K) = 1 + \nu(K) > \mu(\Gamma \vdash Tu)$.

Montrons maintenant que les définitions ne dépendent pas du choix d'un type pour $T$. Supposons que $\Gamma \vdash T : (x'{:}U')K'$. Par convertibilité des types et compatibilité avec le produit, $U \ \mathbb{C}^*_\Gamma \ U'$. Donc $\overline{\mathbb{T}}_{\Gamma\vdash U} = \overline{\mathbb{T}}_{\Gamma\vdash U'}$ et $\mathcal{R}_{\Gamma\vdash T}$, $\leq_{\Gamma\vdash T}$, $\top_{\Gamma\vdash T}$ et $\bigwedge_{\Gamma\vdash T}$ restent inchangés si on remplace $U$ par $U'$.

(b) Par hypothèse de récurrence, $\mathcal{R}_{\Gamma'\vdash Tu} = \mathcal{R}_{\Gamma'\vdash T'u}$.

(c) Immédiat.

(d) Vérifions que $\top_{\Gamma\vdash T}$ vérifie (P1) et (P2).

(P1) Par hypothèse de récurrence (e), $\top_{\Gamma'\vdash Tu} = \top_{\Gamma'\vdash Tu'}$.

(P2) Par hypothèse de récurrence (f), $\top_{\Gamma'\vdash Tu}|_{\Gamma''} = \top_{\Gamma''\vdash Tu}$.

(e) Par préservation du typage, $\top_{\Gamma\vdash T}$ et $\top_{\Gamma\vdash T'}$ ont même domaine. De plus, ils sont égaux puisque, par hypothèse de récurrence, $\top_{\Gamma'\vdash Tu}$ vérifie (e).

(f) $\top_{\Gamma\vdash T}|_{\Gamma'}$ et $\top_{\Gamma'\vdash T}$ ont même domaine et sont égaux.

(g) Soit $\Re' = \{R(\Gamma' \vdash u) \mid R \in \Re\}$. Par définition, si $R \in \mathcal{R}_{\Gamma\vdash T}$ et $\Gamma' \vdash u \in \overline{\mathbb{T}}_{\Gamma\vdash U}$ alors $R(\Gamma' \vdash u) \in \mathcal{R}_{\Gamma'\vdash Tu}$. Par hypothèse de récurrence, $\bigwedge_{\Gamma'\vdash Tu}$ vérifie (g). Donc, $\bigwedge_{\Gamma'\vdash Tu}(\Re') \in \mathcal{R}_{\Gamma'\vdash Tu}$. Reste à voir si $\bigwedge_{\Gamma\vdash T}$ vérifie (P1) et (P2).

(P1) Soit $\Re'' = \{R(\Gamma' \vdash u') \mid R \in \Re\}$. Comme chaque $R \in \Re$ vérifie (P1), $R(\Gamma'\vdash u') = R(\Gamma'\vdash u)$. De plus, par hypothèse de récurrence, $\bigwedge_{\Gamma'\vdash Tu}$ vérifie (h). Donc $\bigwedge_{\Gamma'\vdash Tu}(\Re') = \bigwedge_{\Gamma'\vdash Tu'}(\Re'')$.

(P2) Soit $\Re_1 = \{R(\Gamma \vdash u) \mid R \in \Re\}$ et $\Re_2 = \{R(\Gamma \vdash u)|_{\Gamma'} \mid R \in \Re\}$. Comme chaque $R \in \Re$ vérifie (P2), $R(\Gamma \vdash u)|_{\Gamma'} = R(\Gamma' \vdash u)$. Par hypothèse de récurrence, $\bigwedge_{\Gamma\vdash Tu}$ vérifie (i). Donc $\bigwedge_{\Gamma\vdash Tu}(\Re_1)|_{\Gamma'} = \bigwedge_{\Gamma'\vdash Tu}(\Re_2)$.

(h) D'après (a), $\mathcal{R}_{\Gamma\vdash T} = \mathcal{R}_{\Gamma\vdash T'}$. Donc, $\bigwedge_{\Gamma\vdash T}$ et $\bigwedge_{\Gamma\vdash T'}$ ont le même domaine. Soit $\Re \subseteq \mathcal{R}_{\Gamma\vdash T}$. Alors, $\bigwedge_{\Gamma\vdash T}(\Re)$ et $\bigwedge_{\Gamma\vdash T'}(\Re)$ ont le même domaine et sont égales car, par hypothèse de récurrence, $\bigwedge_{\Gamma'\vdash Tu}$ vérifie (h).

(i) $\bigwedge_{\Gamma\vdash T}(\Re)|_{\Gamma'}$ et $\bigwedge_{\Gamma'\vdash T}(\{R|_{\Gamma'} \mid R \in \Re\})$ ont le même domaine et sont égales car, si $\Gamma'' \vdash u \in \overline{\mathbb{T}}_{\Gamma'\vdash U}$ alors $R(\Gamma'' \vdash u) = R|_{\Gamma'}(\Gamma'' \vdash u)$.

• $\Gamma \vdash T : (X{:}K)L$. La preuve est similaire au cas précédent. ■

**Lemme 115** Soit $\overline{\mathbb{X}}_{\Gamma\vdash T} = \{\Gamma' \vdash t \in \overline{\mathbb{T}}_{\Gamma\vdash T} \mid t = x\vec{t},\ x \in \mathcal{X},\ \vec{t} \in \mathcal{SN}\}$. Si $\Gamma \vdash T : s$ alors $\overline{\mathbb{X}}_{\Gamma\vdash T} \neq \emptyset$ et, pour tout $R \in \mathcal{R}_{\Gamma\vdash T}$, $\overline{\mathbb{X}}_{\Gamma\vdash T} \subseteq R$.

**Preuve.** $\overline{\mathbb{X}}_{\Gamma\vdash T} \subseteq \overline{\mathbb{T}}_{\Gamma\vdash T}$. Comme $\mathcal{X}^s$ est infini et $\mathrm{dom}(\Gamma)$ est fini, il existe $x \in \mathcal{X}^s \setminus \mathrm{dom}(\Gamma)$. Donc, par (var), $\Gamma, x{:}T \vdash x : T$. Ainsi, $\Gamma, x{:}T \vdash x \in \overline{\mathbb{T}}_{\Gamma\vdash T}$ et $\overline{\mathbb{X}}_{\Gamma\vdash T} \neq \emptyset$. Maintenant, soient $R \in \mathcal{R}_{\Gamma\vdash T}$, $\Gamma' \in \mathcal{E}$, $x \in \mathcal{X}$ et $\vec{t} \in \mathcal{SN}$ tels que $\Gamma' \vdash x\vec{t} \in \overline{\mathbb{T}}_{\Gamma\vdash T}$. Montrons que $\Gamma' \vdash x\vec{t} \in R$ par récurrence sur $\vec{t}$ avec $\to_{\mathrm{lex}}$ comme ordre bien fondé. Comme $x\vec{t}$ est neutre, par **(R3)**, il suffit de montrer que tout réduit immédiat de $x\vec{t}$ appartient à $R$. Mais c'est justement l'hypothèse de récurrence. ■

**Lemme 116 (Complétude du treillis des candidats)** Pour tout $\Gamma \vdash T \in \overline{\mathbb{TY}}$, $(\mathcal{R}_{\Gamma\vdash T}, \leq_{\Gamma\vdash T})$ est un inf-demi-treillis complet. De plus, comme il admet $\top_{\Gamma\vdash T}$ comme

plus grand élément, c'est un treillis complet. La borne inférieure d'une partie de $\mathcal{R}_{\Gamma\vdash T}$ est donnée par $\bigwedge_{\Gamma\vdash T}$.

**Preuve.** Il est facile de vérifier par récurrence sur $\mu(\Gamma\vdash T)$ que $\leq_{\Gamma\vdash T}$ est une relation d'ordre (*i.e.* réflexive, transitive et anti-symétrique) et que $\top_{\Gamma\vdash T}$ est le plus grand élément de $\mathcal{R}_{\Gamma\vdash T}$. ■

## 8.3 Schéma d'interprétation des types

Nous définissons l'interprétation d'un type $\Gamma\vdash T$ relativement à une substitution $\theta:\Gamma\to\Delta$ par récurrence sur la structure de $T$. Pour cela, il nous faut donner une interprétation aux variables et symboles de prédicat qui apparaissent dans $T$. Nous allons donc commencer par définir un *schéma d'interprétation* $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi}$ qui utilise un *assignement de candidats* $\xi$ pour les variables de prédicat et une interprétation $I$ pour les symboles de prédicat. Dans le Sous-chapitre 8.4, nous définissons l'interprétation des symboles de prédicat constants et, dans le Sous-chapitre 8.6, nous définissons l'interprétation des symboles de prédicat définis.

**Définition 117 (Assignement de candidats)** Un *assignement de candidats* est une fonction $\xi$ de $\mathcal{X}^\Box$ dans $\bigcup\{\mathcal{R}_{\Delta\vdash T}\mid\Delta\vdash T\in\overline{\mathbb{TY}}\}$ de domaine fini. Étant donné une substitution $\theta:\Gamma\to\Delta$, on dit que $\xi$ est *compatible avec* $(\theta,\Gamma,\Delta)$ si, pour tout $X\in\mathrm{dom}^\Box(\Gamma)$, $X\xi\in\mathcal{R}_{\Delta\vdash X\theta}$. La *restriction* de $\xi$ à un environnement $\Gamma'$ est l'assignement $\xi|_{\Gamma'}$ défini par $X(\xi|_{\Gamma'})=(X\xi)|_{\Gamma'}$.

À toute substitution $\theta:\Gamma\to\Delta$, on peut associer l'*assignement canonique* $\xi_\theta$ tel que, pour tout $X\in\mathrm{dom}^\Box(\Gamma)$, $X\xi_\theta=\top_{\Delta\vdash x\theta}$. D'après le Lemme 114 (d), $\xi_\theta$ est compatible avec $(\theta,\Gamma,\Delta)$.

**Lemme 118** Soit $\theta:\Gamma\to\Delta$ une substitution et $\xi$ un assignement de candidats compatible avec $(\theta,\Gamma,\Delta)$.

(a) Si $\theta\to\theta'$ alors $\theta':\Gamma\to\Delta$ et $\xi$ est compatible avec $(\theta',\Gamma,\Delta)$.

(b) Si $\Delta\subseteq\Delta'\in\mathbb{E}$ alors $\theta:\Gamma\to\Delta'$ et $\xi|_{\Delta'}$ est compatible avec $(\theta,\Gamma,\Delta')$.

**Preuve.**

(a) D'après le Lemme 35, nous savons que $\theta':\Gamma\to\Delta$. Soit $X\in\mathrm{dom}^\Box(\Gamma)$. Puisque $\xi$ est compatible avec $(\theta,\Gamma,\Delta)$, $X\xi\in\mathcal{R}_{\Delta\vdash X\theta}$. D'après le Lemme 114 (b), $\mathcal{R}_{\Delta\vdash X\theta'}=\mathcal{R}_{\Delta\vdash X\theta}$. Donc $X\xi\in\mathcal{R}_{\Delta\vdash X\theta'}$.

(b) Par affaiblissement, $\theta:\Gamma\to\Delta'$. Soit $X\in\mathrm{dom}^\Box(\Gamma)$. Par définition, $X(\xi|_{\Delta'})=(X\xi)|_{\Delta'}$. Puisque $\xi$ est compatible avec $(\theta,\Gamma,\Delta)$, $X\xi\in\mathcal{R}_{\Delta\vdash X\theta}$. Donc, d'après le Lemme 114 (c), $(X\xi)|_{\Delta'}\in\mathcal{R}_{\Delta'\vdash X\theta}$. ■

Soit $F$ un symbole de prédicat de type $(\vec{x}:\vec{T})\star$. Par la suite, on se permettra de considérer $F$, qui n'est pas un terme en général, comme représentant sa forme $\eta$-longue $[\vec{x}:\vec{T}]F(\vec{x})$.

**Définition 119 (Interprétation d'un symbole de prédicat)** Une *interprétation* pour un symbole de prédicat $F$ est une fonction $I$ qui à un environnement $\Delta$ associe un élément de $\mathcal{R}_{\Delta\vdash F}$ tel que :

**(P3)** si $\Delta \subseteq \Delta' \in \mathbb{E}$ alors $I_\Delta|_{\Delta'} = I_{\Delta'}$ (compatibilité avec l'affaiblissement).
Une *interprétation* pour un ensemble $\mathcal{G}$ de symboles de prédicat est une fonction qui, à un symbole $G \in \mathcal{G}$, associe une interprétation pour $G$.

**Définition 120 (Schéma d'interprétation)** L'*interprétation* de $\Gamma \vdash T \in \overline{\mathbb{TY}}$ par rapport à un environnement $\Delta \in \mathbb{E}$, une substitution $\theta : \Gamma \to \Delta$, un assignement de candidats $\xi$ compatible avec $(\theta, \Gamma, \Delta)$ et une interprétation $I_F$ pour chaque $F \in \mathcal{F}^\square$, est un élément de $\mathcal{R}_{\Delta \vdash T\theta}$ défini par récurrence sur $T$ :

- $[\![\Gamma \vdash s]\!]^I_{\Delta,\theta,\xi} = \overline{\mathbb{SN}}_{\Delta \vdash s}$,
- $[\![\Gamma \vdash F(\vec{t})]\!]^I_{\Delta,\theta,\xi} = I_{\Delta \vdash F}(\vec{a})$ où, si $\tau_F = (\vec{x} : \vec{T})U$ :
  - $a_i = \Delta \vdash t_i\theta$ si $x_i \in \mathcal{X}^\star$,
  - $a_i = (\Delta \vdash t_i\theta, [\![\Gamma \vdash t_i]\!]^I_{\Delta,\theta,\xi})$ si $x_i \in \mathcal{X}^\square$,
- $[\![\Gamma \vdash X]\!]^I_{\Delta,\theta,\xi} = X\xi$,
- $[\![\Gamma \vdash (x:U)V]\!]^I_{\Delta,\theta,\xi} = \{\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta \vdash (x:U\theta)V\theta} \mid \forall \Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}},\ \Delta'' \vdash tu \in [\![\Gamma, x:U \vdash V]\!]^I_{\Delta'',\theta\cup\{x\mapsto u\},\xi|_{\Delta''}}\}$,
- $[\![\Gamma \vdash (X : K)V]\!]^I_{\Delta,\theta,\xi} = \{\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta \vdash (X:K\theta)V\theta} \mid \forall \Delta'' \vdash U \in [\![\Gamma \vdash K]\!]^I_{\Delta',\theta,\xi|_{\Delta'}},\ \Delta'' \vdash tU \in \bigcap\{[\![\Gamma, X:K \vdash V]\!]^I_{\Delta'',\theta\cup\{X\mapsto U\},\xi|_{\Delta''}\cup\{X\mapsto S\}} \mid S \in \mathcal{R}_{\Delta'' \vdash U}\}\}$,
- $[\![\Gamma \vdash [x:U]V]\!]^I_{\Delta,\theta,\xi}(\Delta' \vdash u) = [\![\Gamma, x:U \vdash V]\!]^I_{\Delta',\theta\cup\{x\mapsto u\},\xi|_{\Delta'}}$,
- $[\![\Gamma \vdash [X:K]V]\!]^I_{\Delta,\theta,\xi}(\Delta' \vdash U, S) = [\![\Gamma, X:K \vdash V]\!]^I_{\Delta',\theta\cup\{X\mapsto U\},\xi|_{\Delta'}\cup\{X\mapsto S\}}$,
- $[\![\Gamma \vdash Vu]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Delta \vdash u\theta)$,
- $[\![\Gamma \vdash VU]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Delta \vdash U\theta, [\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi})$.

Dans le cas où $\Gamma \vdash T : s$, nous dirons qu'une fermeture appartenant à $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ est *calculable*. Enfin, nous dirons que $(\theta, \Gamma, \Delta)$ est *valide* par rapport à $\xi$ si, pour tout $x \in \mathrm{dom}(\Gamma)$, $\Delta \vdash x\theta \in [\![\Gamma \vdash x\Gamma]\!]^I_{\Delta,\theta,\xi}$.

D'après le Lemme 115, l'identité est valide par rapport à tout assignement compatible avec elle.

**Lemme 121 (Correction du schéma d'interprétation)**
(a) $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ est bien défini.
(b) $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \in \mathcal{R}_{\Delta \vdash T\theta}$.
(c) Si $\theta \to \theta'$ alors $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi}$.
(d) Si $\Delta \subseteq \Delta' \in \mathbb{E}$ alors $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'} = [\![\Gamma \vdash T]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$.

**Preuve.** Notons tout d'abord que, pour (c), $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi}$ existe car, d'après le Lemme 118 (a), $\theta' : \Gamma \to \Delta$ et $\xi$ est compatible avec $(\theta', \Gamma, \Delta)$. Pour (d), $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'}$ existe car, d'après (b), $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \in \mathcal{R}_{\Delta \vdash T\theta}$, et $[\![\Gamma \vdash T]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ existe car, d'après le Lemme 118 (b), $\theta : \Gamma \to \Delta'$ et $\xi|_{\Delta'}$ est compatible avec $(\theta, \Gamma, \Delta')$.

- $T = s$.
  (a) Immédiat.
  (b) D'après le Lemme 114 (d).
  (c) Car $[\![\Gamma \vdash s]\!]^I_{\Delta,\theta,\xi}$ ne dépend pas de $\theta$.

(d) $[\![\Gamma\vdash s]\!]^I_{\Delta,\theta,\xi}|_{\Delta'} = \overline{\mathbb{SN}}_{\Delta\vdash s} \cap \overline{\mathbb{T}}_{\Delta'\vdash s} = \overline{\mathbb{SN}}_{\Delta'\vdash s} = [\![\Gamma\vdash s]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$.

- $T = F(\vec{t})$.
  (a) $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} = I_{\Delta\vdash F}(\vec{a})$ où $a_i = \Delta\vdash t_i\theta$ si $x_i \in \mathcal{X}^\star$ et $a_i = (\Delta\vdash t_i\theta, [\![\Gamma\vdash t_i]\!]^I_{\Delta,\theta,\xi})$ si $x_i \in \mathcal{X}^\square$. Par hypothèse de récurrence (a) et (b), $[\![\Gamma\vdash t_i]\!]^I_{\Delta,\theta,\xi}$ est bien défini et appartient à $\mathcal{R}_{\Delta\vdash t_i\theta}$. Donc $\vec{a}$ est dans le domaine de $I_{\Delta\vdash F}$ et $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi}$ est bien défini.
  (b) Par définition de $I_{\Delta\vdash F}$.
  (c) Par **(P1)** .
  (d) Par **(P2)** .
- $T = X$.
  (a) Immédiat.
  (b) Car $\xi$ est compatible avec $(\theta, \Gamma, \Delta)$.
  (c) Car $[\![\Gamma\vdash X]\!]^I_{\Delta,\theta,\xi}$ ne dépend pas de $\theta$.
  (d) Par définition de $\xi|_{\Delta'}$.
- $T = (x{:}U)V$. Soit $\Gamma' = \Gamma, x{:}U$.
  (a) Supposons que $\Delta \subseteq \Delta' \in \mathbb{E}$. D'après le Lemme 118 (b), $\theta : \Gamma \to \Delta'$ et $\xi|_{\Delta'}$ est compatible avec $(\theta, \Gamma, \Delta')$. Ainsi, par hypothèse de récurrence (a) et (b), $[\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ est bien défini et appartient à $\mathcal{R}_{\Delta'\vdash U\theta}$.

  Par correction des types, il existe $s$ tel que $\Gamma \vdash T : s$. Par inversion, $\Gamma' \vdash V : s$. D'après le lemme d'environnement, $\Gamma \vdash U : \star$. Donc, par substitution, $\Delta' \vdash U\theta : \star$ et $[\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}} \subseteq \overline{\mathbb{T}}_{\Delta'\vdash U\theta}$.

  Maintenant, soit $\Delta'' \vdash u \in [\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $\sigma = \theta \cup \{x \mapsto u\}$. Puisque $\theta : \Gamma \to \Delta$, $\Delta'' \vdash u : U\theta$ et $x \notin \mathrm{FV}(\Gamma')$, nous avons $\sigma : \Gamma' \to \Delta''$. D'après le Lemme 118 (b), $\xi|_{\Delta''}$ est compatible avec $(\theta, \Gamma, \Delta'')$. Et puisque $\mathrm{dom}^\square(\sigma) = \mathrm{dom}^\square(\theta)$ et $\sigma$ et $\theta$ sont égales sur ce domaine, $\xi|_{\Delta''}$ est compatible avec $(\sigma, \Gamma', \Delta'')$. Donc, par hypothèse de récurrence (a) et (b), $[\![\Gamma'\vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ est bien défini et appartient à $\mathcal{R}_{\Delta''\vdash V\sigma}$.

  Enfin, par substitution, $\Delta'' \vdash V\sigma : s$. Donc, $[\![\Gamma'\vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}} \subseteq \overline{\mathbb{T}}_{\Delta''\vdash V\sigma}$ et $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi}$ est bien défini.
  (b) Par substitution, $\Delta \vdash T\theta : s$. Donc, on doit prouver que $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi}$ est inclus dans $\overline{\mathbb{T}}_{\Delta\vdash T\theta}$ (immédiat) et satisfait (R1) à (R4). Nous avons vu en (a) que, si $\Delta \subseteq \Delta' \in \mathbb{E}$, $\Delta'' \vdash u \in [\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $\sigma = \theta \cup \{x \mapsto u\}$, alors $[\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $[\![\Gamma'\vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ sont respectivement inclus dans $\overline{\mathbb{T}}_{\Delta'\vdash U\theta}$ et $\overline{\mathbb{T}}_{\Delta''\vdash V\sigma}$ et satisfont (R1) à (R4).

  **(R1)** D'après le Lemme 115, il existe $y \in \mathcal{X}^\star \setminus \mathrm{dom}(\Delta')$ tel que $\Delta', y{:}U\theta \vdash y \in [\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$. Posons $\Delta'' = \Delta', y{:}U\theta$ et $\sigma = \theta \cup \{x \mapsto y\}$. Alors, par définition, $\Delta''\vdash ty \in [\![\Gamma'\vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ et, puisque $[\![\Gamma'\vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ satisfait **(R1)**, $ty \in \mathcal{SN}$ et $t \in \mathcal{SN}$.

  **(R2)** Soit $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $\sigma = \theta \cup \{x \mapsto u\}$. Par définition, $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ et, puisque $tu \to t'u$ et $[\![\Gamma' \vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ satisfait **(R2)**, $t'u \in [\![\Gamma'\vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ et $t' \in [\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi}$.

**(R3)** C'est dans ce cas que nous utilisons de manière essentielle la notion d'arité d'un symbole et la distinction syntaxique entre application du $\lambda$-calcul et application d'un symbole évoquée dans la Remarque 10. Soit $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $\sigma = \theta \cup \{x \mapsto u\}$. Par définition, $\Delta'' \vdash tu \in \overline{\mathbb{T}}_{\Delta'' \vdash V\sigma}$ et $tu$ est neutre. Puisque $[\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ satisfait **(R1)**, $u \in \mathcal{SN}$.

Prouvons que tous les réduits immédiats $v'$ de $tu$ appartiennent à $[\![\Gamma' \vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$, par récurrence sur $u$ avec $\to$ comme ordre bien fondé. Comme $t$ n'est pas une abstraction, $v'$ est soit de la forme $t'u$ avec $t'$ un réduit immédiat de $t$, soit de la forme $tu'$ avec $u'$ un réduit immédiat de $u$. Dans le premier cas, $\Delta'' \vdash t'u \in \overline{\mathbb{T}}_{\Delta'' \vdash V\sigma}$ puisque, par hypothèse, $\Delta'' \vdash t' \in [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ et $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$. Dans le second cas, $\Delta'' \vdash tu' \in \overline{\mathbb{T}}_{\Delta'' \vdash V\sigma}$ par hypothèse de récurrence.

Puisque $[\![\Gamma' \vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ satisfait **(R3)**, $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ et $\Delta'' \vdash t \in [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$.

**(R4)** Supposons que $\Delta' \subseteq \Delta'' \in \mathbb{E}$, $\Delta''' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta'',\theta,\xi|_{\Delta''}}$ et $\sigma = \theta \cup \{x \mapsto u\}$. Par hypothèse de récurrence (d), $[\![\Gamma \vdash U]\!]^I_{\Delta'',\theta,\xi|_{\Delta''}} = [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}|_{\Delta''} = [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}} \cap \overline{\mathbb{T}}_{\Delta'' \vdash U\theta}$. Ainsi, $\Delta''' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$, $\Delta''' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta''',\sigma,\xi|_{\Delta'''}}$ et $\Delta'' \vdash t \in [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$.

(c) Montrons que $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \subseteq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi}$. L'inclusion inverse se traite de manière similaire. Puisque $\Delta \vdash T\sigma : s$, par conversion, $\overline{\mathbb{T}}_{\Delta \vdash T\theta'} = \overline{\mathbb{T}}_{\Delta \vdash T\theta}$ et $\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta \vdash T\theta'}$. Maintenant, soit $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta',\xi|_{\Delta'}}$, $\sigma = \theta \cup \{x \mapsto u\}$ et $\sigma' = \sigma \cup \{x \mapsto u\}$. Par hypothèse de récurrence (c), $[\![\Gamma \vdash U]\!]^I_{\Delta',\theta',\xi|_{\Delta'}} = [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$. Donc, puisque $[\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ satisfait **(R3)**, $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta,\sigma,\xi}$. Et puisque $\sigma \to \sigma'$, par hypothèse de récurrence (c), $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta,\sigma',\xi}$ et $\Delta' \vdash t \in [\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi}$.

(d) Montrons que $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'} \subseteq [\![\Gamma \vdash T]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$. L'inclusion inverse se traite de manière similaire. Par définition, $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \cap \overline{\mathbb{T}}_{\Delta' \vdash T\theta}$. Soit $\Delta'' \vdash t \in [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'}$, $\Delta''' \vdash u \in [\![\Gamma \vdash T]\!]^I_{\Delta'',\theta,\xi|_{\Delta''}}$ et $\sigma = \theta \cup \{x \mapsto u\}$. Par définition de $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$, $\Delta''' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta''',\sigma,\xi|_{\Delta'''}}$. Donc, puisque $\Delta'' \vdash t \in \overline{\mathbb{T}}_{\Delta' \vdash T\theta}$, $\Delta'' \vdash t \in [\![\Gamma \vdash T]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$.

- $T = (X\!:\!K)V$. Similaire au cas précédent.
- $T = [x\!:\!U]V$. Soit $\Gamma' = \Gamma, x\!:\!U$.

(a) Soit $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$ et $\sigma = \theta \cup \{x \mapsto u\}$. Puisque $\Delta \subseteq \Delta' \in \mathbb{E}$, par le Lemme 118 (b), $\theta : \Gamma \to \Delta'$ et $\xi|_{\Delta'}$ est compatible avec $(\theta, \Gamma, \Delta')$. De plus, puisque $\Delta' \vdash u : U\theta$ et $x \notin \mathrm{FV}(\Gamma)$, $\sigma : \Gamma' \to \Delta'$. Par ailleurs, $\xi|_{\Delta'}$ est compatible avec $(\sigma, \Gamma', \Delta')$ puisque $\mathrm{dom}^{\square}(\sigma) = \mathrm{dom}^{\square}(\theta)$. Ainsi, par hypothèse de récurrence (a), $[\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma,\xi|_{\Delta'}}$ est bien défini et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ est bien défini.

(b) $\mathcal{R}_{\Delta \vdash T\theta}$ est l'ensemble des fonctions qui à $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$ associent un élément de $\mathcal{R}_{\Delta' \vdash (T\theta\ u)}$ et vérifient (P1) et (P2). Par hypothèse de récurrence (b), $[\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma,\xi|_{\Delta'}} \in \mathcal{R}_{\Delta' \vdash V\sigma}$. Comme $(T\theta\ u) \to_\beta V\sigma$, par le Lemme 114 (b), $\mathcal{R}_{\Delta' \vdash V\sigma} = \mathcal{R}_{\Delta' \vdash (T\theta\ u)}$.

**(P1)** Supposons que $u \to u'$. $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}(\Delta' \vdash u') = [\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma',\xi|_{\Delta'}}$ où $\sigma' = \theta \cup \{x \mapsto u'\}$. Comme $\sigma \to \sigma'$, par hypothèse de récurrence (c), $[\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma',\xi|_{\Delta'}} = [\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma,\xi|_{\Delta'}}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \in \mathcal{R}_{\Delta \vdash T\theta}$.

**(P2)** Supposons que $\Delta \subseteq \Delta' \in \mathbb{E}$. $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}(\Delta \vdash u)|_{\Delta'} = [\![\Gamma' \vdash V]\!]^I_{\Delta,\sigma,\xi}|_{\Delta'}$. Par hypothèse de récurrence (d), $[\![\Gamma' \vdash V]\!]^I_{\Delta,\sigma,\xi}|_{\Delta'} = [\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma,\xi'|_{\Delta'}} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}(\Delta' \vdash u)$.

(c) $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi}(\Delta' \vdash u) = [\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma',\xi|_{\Delta'}}$ où $\sigma' = \sigma \cup \{x \mapsto u\}$. $\sigma \to \sigma'$ donc, par hypothèse de récurrence (c), $[\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma',\xi|_{\Delta'}} = [\![\Gamma' \vdash V]\!]^I_{\Delta',\sigma,\xi|_{\Delta'}}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$.

(d) $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\overline{\mathbb{T}}_{\Delta' \vdash U\theta}}$ est la fonction qui, à $\Delta'' \vdash u \in \overline{\mathbb{T}}_{\Delta' \vdash U\theta}$, associe $[\![\Gamma' \vdash V]\!]^I_{\Delta'',\sigma,\xi|_{\Delta''}}$ où $\sigma = \theta \cup \{x \mapsto u\}$. C'est $[\![\Gamma \vdash T]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$.

- $T = [X\!:\!K]V$. Similaire au cas précédent.
- $T = Vu$.

(a) Par hypothèse de récurrence (a), $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}$ est bien défini et appartient à $\mathcal{R}_{\Delta \vdash V\theta}$. Puisque $T \in \overline{\mathbb{T}\mathbb{Y}}$ et $T \neq \Box$, $T$ est typable dans $\Gamma$. Par inversion, il existe $U$ et $K$ tels que $\Gamma \vdash V : (x : U)K$ et $\Gamma \vdash u : U$. Par substitution, $\Delta \vdash V\theta : (x : U\theta)K\theta$ et $\Delta \vdash u\theta : U\theta$. Donc, $\mathcal{R}_{\Delta \vdash V\theta}$ est l'ensemble des fonctions qui à $\Delta' \vdash u' \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$ associent un élément de $\mathcal{R}_{\Delta' \vdash V\theta u'}$. Ainsi, comme $\Delta \vdash u\theta \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$, $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ est bien défini.

(b) $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \in \mathcal{R}_{\Delta \vdash (V\theta u\theta)} = \mathcal{R}_{\Delta \vdash T\theta}$.

(c) Par hypothèse de récurrence (c), $[\![\Gamma \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}$. Comme $u\theta \to^* u\sigma$ et $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}$ vérifie **(P1)**, $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta',\xi}$.

(d) $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}|_{\Delta'} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Delta \vdash u\theta)|_{\Delta'}$. Par **(P2)**, $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Delta \vdash u\theta)|_{\Delta'} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Delta' \vdash u\theta) = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}|_{\Delta'}(\Delta' \vdash u\theta)$. Par hypothèse de récurrence (d), $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}|_{\Delta'}(\Delta' \vdash u\theta) = [\![\Gamma \vdash V]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}(\Delta' \vdash u\theta) = [\![\Gamma \vdash T]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$.

- $T = (V\ U)$. Similaire au cas précédent. ■

**Lemme 122** Soient deux interprétations $I$ et $I'$ égales sur les symboles de prédicat apparaissant dans $T$, et soient deux assignements de candidats $\xi$ et $\xi'$ égaux sur les variables libres de $T$. Alors, $[\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi'} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$.

**Preuve.** Par récurrence sur $T$. ■

**Lemme 123 (Substitution de candidats)** Soient $\Gamma_0$, $\Gamma_1$ et $\Gamma_2$, trois environnements valides, soit $\Gamma_0 \vdash T \in \overline{\mathbb{T}\mathbb{Y}}$, soient $\theta_1 : \Gamma_0 \to \Gamma_1$ et $\theta_2 : \Gamma_1 \to \Gamma_2$, deux substitutions, et soit $\xi_2$, un assignement de candidats compatible avec $(\theta_2, \Gamma_1, \Gamma_2)$. Alors, l'assignement de candidats $\xi_{12}$ défini par $X\xi_{12} = [\![\Gamma_1 \vdash X\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2}$ est compatible avec $(\theta_1\theta_2, \Gamma_0, \Gamma_2)$ et $[\![\Gamma_1 \vdash T\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2} = [\![\Gamma_0 \vdash T]\!]^I_{\Gamma_2,\theta_1\theta_2,\xi_{12}}$.

**Preuve.** D'après le Lemme 121 (b), $X\xi_{12} \in \mathcal{R}_{\Gamma_2 \vdash X\theta_1\theta_2}$. Donc $\xi_{12}$ est compatible avec $(\theta_1\theta_2, \Gamma_0, \Gamma_2)$. Soit $R = [\![\Gamma_0 \vdash T]\!]^I_{\Gamma_2,\theta_1\theta_2,\xi_{12}}$ et $R' = [\![\Gamma_1 \vdash T\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2}$. Montrons que $R = R'$ par récurrence sur $T$. D'après le Lemme 121 (b), $R$ et $R'$ appartiennent tous deux à $\mathcal{R}_{\Gamma_2 \vdash T\theta_1\theta_2}$.

- $T = s$. $R' = [\![\Gamma_1 \vdash s]\!]^I_{\Gamma_2,\theta_2,\xi_2} = \overline{\mathbb{SN}}_{\Gamma_2 \vdash s} = R$.
- $T = F(\vec{t})$. Supposons que $\tau_F = (\vec{x} : \vec{T})\star$. Alors, $R = I_{\Gamma_2 \vdash F}(\vec{a})$ où $a_i = \Gamma_2 \vdash t_i\theta_1\theta_2$ si $x_i \in \mathcal{X}^\star$, et $a_i = (\Gamma_2 \vdash t_i\theta_1\theta_2, [\![\Gamma_0 \vdash t_i]\!]^I_{\Gamma_2,\theta_1\theta_2,\xi_{12}})$ si $x_i \in \mathcal{X}^\square$. De même, $R' = I_{\Gamma_2 \vdash F}(\vec{a}')$ où $a'_i = \Gamma_2 \vdash t_i\theta_1\theta_2$ si $x_i \in \mathcal{X}^\star$, et $a'_i = (\Gamma_2 \vdash t_i\theta_1\theta_2, [\![\Gamma_1 \vdash t_i\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2})$ si $x_i \in \mathcal{X}^\square$. Par hypothèse de récurrence, pour tout $x_i \in \mathcal{X}^\square$, $[\![\Gamma_0 \vdash t_i]\!]^I_{\Gamma_2,\theta_1\theta_2,\xi_{12}} = [\![\Gamma_1 \vdash t_i\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2}$. Donc, $\vec{a} = \vec{a}'$ et $R = R'$.
- $T = X$. $R = X\xi_{12} = [\![\Gamma_1 \vdash X\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2} = R'$.
- $T = (x : U)V$. Soit $\Gamma'_0 = \Gamma_0, x : U$, $\Gamma'_1 = \Gamma_1, x : U\theta_1$ et $\Gamma'_2 \vdash t \in R$. Comme $R \in \mathcal{R}_{\Gamma_2 \vdash T\theta_1\theta_2}$, $\Gamma'_2 \vdash t \in \overline{\mathbb{T}}_{\Gamma_2 \vdash T\theta_1\theta_2}$. Soit $\Gamma''_2 \vdash u \in [\![\Gamma_1 \vdash U\theta_1]\!]^I_{\Gamma'_2,\theta_2,\xi_2|_{\Gamma'_2}}$. D'après le Lemme 121 (d), pour tout $X \in \mathrm{dom}^\square(\Gamma_0)$, $X\xi_{12}|_{\Gamma'_2} = [\![\Gamma_1 \vdash X\theta_1]\!]^I_{\Gamma'_2,\theta_2,\xi_2|_{\Gamma'_2}}$. Par hypothèse de récurrence, $[\![\Gamma_1 \vdash U\theta_1]\!]^I_{\Gamma'_2,\theta_2,\xi_2|_{\Gamma'_2}} = [\![\Gamma_0 \vdash U]\!]^I_{\Gamma'_2,\theta_1\theta_2,\xi_{12}|_{\Gamma'_2}}$. Par définition de $\mathcal{R}_{\Gamma_2 \vdash T\theta_1\theta_2}$, $\Gamma''_2 \vdash tu \in [\![\Gamma'_0 \vdash V]\!]^I_{\Gamma''_2,\theta_1\theta_2\cup\{x\mapsto u\},\xi_{12}|_{\Gamma''_2}}$. Par ailleurs, $X\xi_{12}|_{\Gamma''_2} = [\![\Gamma_1 \vdash X\theta_1]\!]^I_{\Gamma''_2,\theta_2,\xi_2|_{\Gamma''_2}}$, $\theta_1 : \Gamma'_0 \to \Gamma'_1$, $\theta_2 \cup \{x \mapsto u\} : \Gamma'_1 \to \Gamma''_2$ et $\theta_1\theta_2 \cup \{x \mapsto u\} = \theta_1(\theta_2 \cup \{x \mapsto u\})$. Donc, par hypothèse de récurrence, $[\![\Gamma'_0 \vdash V]\!]^I_{\Gamma''_2,\theta_1\theta_2\cup\{x\mapsto u\},\xi_{12}|_{\Gamma''_2}} = [\![\Gamma'_1 \vdash V\theta_1]\!]^I_{\Gamma''_2,\theta_2\cup\{x\mapsto u\},\xi_2|_{\Gamma''_2}}$ et $\Gamma'_2 \vdash t \in R'$. Ainsi, $R \subseteq R'$. De même, on montrerait que $R' \subseteq R$.
- $T = (X : K)V$. Similaire au cas précédent.
- $T = [x : U]V$. Soit $\Gamma'_0 = \Gamma_0, x : U$, $\Gamma'_1 = \Gamma_1, x : U\theta_1$ et $\Gamma'_2 \vdash u \in \overline{\mathbb{T}}_{\Gamma_2 \vdash U\theta_1\theta_2}$. Par définition, $R(\Gamma'_2 \vdash u) = [\![\Gamma'_0 \vdash V]\!]^I_{\Gamma'_2,\theta_1\theta_2\cup\{x\mapsto u\},\xi_{12}|_{\Gamma'_2}}$. Par hypothèse de récurrence, $[\![\Gamma'_0 \vdash V]\!]^I_{\Gamma'_2,\theta_1\theta_2\cup\{x\mapsto u\},\xi_{12}|_{\Gamma'_2}} = [\![\Gamma'_1 \vdash V\theta_1]\!]^I_{\Gamma'_2,\theta_2\cup\{x\mapsto u\},\xi_2|_{\Gamma'_2}}$. Donc, $R(\Gamma'_2 \vdash u) = R'(\Gamma'_2 \vdash u)$ et $R = R'$.
- $T = [X : K]V$. Similaire au cas précédent.
- $T = Vu$. $R = [\![\Gamma_0 \vdash V]\!]^I_{\Gamma_2,\theta_1\theta_2,\xi_{12}}(\Gamma_2 \vdash u\theta_1\theta_2)$. Par hypothèse de récurrence, $[\![\Gamma_0 \vdash V]\!]^I_{\Gamma_2,\theta_1\theta_2,\xi_{12}} = [\![\Gamma_1 \vdash V\theta_1]\!]^I_{\Gamma_2,\theta_2,\xi_2}$. Donc, $R = R'$.
- $T = VU$. Similaire au cas précédent. ■

## 8.4 Interprétation des symboles de prédicat constants

Dans ce sous-chapitre, nous définissons l'interprétation $I$ pour les symboles de prédicat constants par récurrence sur $>_\mathcal{C}$. Soit $C$ un symbole de prédicat constant et supposons que nous ayons déjà défini une interprétation $K$ pour tous les symboles plus petits que $C$.

Comme N. P. Mendler [89] ou B. Werner [118], nous allons définir cette interprétation comme le point fixe d'une fonction monotone sur un treillis complet, la monotonie étant assurée par les conditions de positivité des prédicats inductifs (Définition 74). La principale différence avec ces travaux est que nous avons une notion de constructeur beaucoup plus générale puisqu'elle inclut toute fonction dont le type de sortie est un symbole de prédicat constant, afin de pouvoir définir des fonctions ou des prédicats par filtrage non seulement sur des constructeurs au sens

habituel (constants) mais aussi sur des symboles définis.

Nous désignerons par :

– $[C]$ l'ensemble des symboles de prédicats constants équivalents à $C$,
– $\mathcal{I}$ l'ensemble des interprétations pour $[C]$,
– $\leq$ la relation sur $\mathcal{I}$ définie par $I \leq I'$ si, pour tout $D \in [C]$ et $\Delta \in \mathbb{E}$, $I_{\Delta\vdash D} \leq_{\Delta\vdash D} I'_{\Delta\vdash D}$.

Pour alléger les notations, étant donnée $I \in \mathcal{I}$, nous désignerons $[\![\Gamma\vdash T]\!]^{K\cup I}_{\Delta,\theta,\xi}$ par $[\![\Gamma\vdash T]\!]^{I}_{\Delta,\theta,\xi}$.

Soit $D$ un symbole de prédicat constant équivalent à $C$. Supposons que $D$ soit d'arité $n$ et de type $(\vec{x} : \vec{T})\star$. Étant donné un environnement $\Delta$, par définition, $\mathcal{R}_{\Delta\vdash D}$ est l'ensemble des fonctions qui, à $a_1 \in A_1, \ldots, a_n \in A_n$, associe un élément de $\mathcal{R}_{\Delta_n\vdash D(\vec{t})}$ où :

– $a_i = \Delta_i\vdash t_i$ et $A_i = \overline{\mathbb{T}}_{\Delta_{i-1}\vdash T_i\theta}$ si $x_i \in \mathcal{X}^\star$,
– $a_i = (\Delta_i\vdash t_i, S_i)$ et $A_i = \Sigma_{\Delta_{i-1}\vdash T_i\theta}$ si $x_i \in \mathcal{X}^\Box$,
– $\Delta_0 = \Delta$ et $\theta = \{\vec{x} \mapsto \vec{t}\}$.

**Définition 124 (Interprétation monotone)** Soit $I \in \mathcal{I}$, $x_i \in \mathcal{X}^\Box$, $\Delta \in \mathbb{E}$ et $\vec{a}, \vec{a}'$ deux séquences d'arguments pour $I$ telles que $a_i = (\Delta_i\vdash t_i, S_i)$, $a'_i = (\Delta_i\vdash t_i, S'_i)$ et, pour tout $j \neq i$, $a_j = a'_j$. Alors $I$ est *monotone en son $i$-ème argument* si $S_i \leq S'_i$ implique $I_{\Delta\vdash D}(\vec{a}) \leq I_{\Delta\vdash D}(\vec{a}')$. Nous désignerons par $\mathcal{I}^m$ l'ensemble des interprétations qui sont monotones en chacun de leurs arguments inductifs $i \in \mathrm{Ind}(D)$.

**Lemme 125** $(\mathcal{I}^m, \leq)$ est un treillis complet.

**Preuve.** Tout d'abord, $\leq$ est bien une relation d'ordre puisque, pour tout $D =_{\mathcal{C}} C$ et $\Delta \in \mathbb{E}$, $\leq_{\Delta\vdash D}$ est une relation d'ordre.

Montrons que $\mathcal{I}^m$ admet comme plus grand élément la fonction $I^\top$ défini par $I^\top_{\Delta\vdash D} = \top_{\Delta\vdash D}$. La fonction $I^\top_D$ est bien une interprétation car, d'après le Lemme 114 (d) et (f), $I^\top_{\Delta\vdash D} \in \mathcal{R}_{\Delta\vdash D}$ et si $\Delta \subseteq \Delta' \in \mathbb{E}$ alors $I^\top_{\Delta\vdash D}\Delta = \top_{\Delta\vdash D}|_{\Delta'} = \top_{\Delta'\vdash D} = I^\top_{\Delta'\vdash D}$. Par ailleurs, $I^\top$ est le plus grand élément de $\mathcal{I}$ puisque $\top_{\Delta\vdash D}$ est le plus grand élément de $\mathcal{R}_{\Delta\vdash D}$.

Montrons maintenant que $I^\top$ est monotone en ses arguments inductifs. Soit $i \in \mathrm{Ind}(D)$ et $\vec{a}, \vec{a}'$ deux séquences d'arguments pour $I^\top_{\Delta\vdash D}$ telles que $a_i = (\Delta_i\vdash t_i, S_i)$, $a'_i = (\Delta_i\vdash t_i, S'_i)$, $S_i \leq S'_i$ et, pour tout $j \neq i$, $a_j = a'_j$. Alors, $I^\top_{\Delta\vdash D}(\vec{a}) = \top_{\Delta_n\vdash D(\vec{t})} = I^\top_{\Delta\vdash D}(\vec{a}')$.

Montrons maintenant que toute partie de $\mathcal{I}^m$ admet une borne inférieure. Soit $\Im \subseteq \mathcal{I}^m$ et $I^\wedge$ la fonction définie par $I^\wedge_{\Delta\vdash D} = \bigwedge_{\Delta\vdash D}(\Re_{\Delta\vdash D})$ où $\Re_{\Delta\vdash D} = \{I_{\Delta\vdash D} \mid I \in \Im\}$. La fonction $I^\wedge$ est bien une interprétation car, d'après le Lemme 114 (g) et (i), $I^\wedge_{\Delta\vdash D} \in \mathcal{R}_{\Delta\vdash D}$ et si $\Delta \subseteq \Delta' \in \mathbb{E}$ alors $I^\wedge_{\Delta\vdash D}|_{\Delta'} = I^\wedge_{\Delta'\vdash D}$.

Montrons que $I^\wedge$ est monotone en ses arguments inductifs. Soit $i \in \mathrm{Ind}(D)$ et $\vec{a}, \vec{a}'$ deux séquences d'arguments de $I^\wedge_{\Delta\vdash D}$ vérifiant les conditions pour montrer la monotonie. Alors, $I^\wedge_{\Delta\vdash D}(\vec{a}) = \bigcap\{I_{\Delta\vdash D}(\vec{a}) \mid I \in \Im\}$ et $I^\wedge_{\Delta\vdash D}(\vec{a}') = \bigcap\{I_{\Delta\vdash D}(\vec{a}') \mid I \in$

$\Im\}$. Comme chaque $I$ est monotone en ses arguments inductifs, $I_{\Delta\vdash D}(\vec{a}) \leq I_{\Delta\vdash D}(\vec{a}')$. Donc, $I^{\wedge}_{\Delta\vdash D}(\vec{a}) \leq I^{\wedge}_{\Delta\vdash D}(\vec{a}')$.

Il ne nous reste plus qu'à montrer que $I^{\wedge}$ est la borne inférieure de $\Im$. Tout d'abord, pour tout $I \in \Im$, $I^{\wedge} \leq I$ puisque $I^{\wedge}_{\Delta\vdash D}$ est la borne inférieure de $\Re_{\Delta\vdash D}$. Supposons maintenant qu'il existe $I' \in \mathcal{I}^m$ tel que, pour tout $I \in \Im$, $I' \leq I$. Alors, puisque $I^{\wedge}_{\Delta\vdash D}$ est la borne inférieure de $\Re_{\Delta\vdash D}$, $I' \leq I^{\wedge}$. ■

**Définition 126 (Interprétation des symboles de prédicat constants)** Soit $\varphi$ la fonction qui, à $I \in \mathcal{I}^m$, associe l'interprétation $\varphi^I$ telle que $\varphi^I_{\Delta\vdash D}(\vec{a})$ soit l'ensemble des $\Delta' \vdash u \in \overline{\mathbb{SN}}_{\Delta_n\vdash D(\vec{t})}$ tels que si $u$ se réduit en un terme de la forme $d(\vec{u})$ avec $d$ un constructeur de type $(\vec{y} : \vec{U})D(\vec{v})$ alors, pour tout $j \in \mathrm{Acc}(d)$, $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ où $\theta = \{\vec{y} \mapsto \vec{u}\}$ et, pour tout $Y \in \mathrm{FV}^{\Box}(U_j)$, $Y\xi = S_{\iota_Y}$. Nous montrons ci-après que $\varphi$ est monotone. Nous pouvons donc prendre $I_{\Delta\vdash D} = \mathrm{lfp}(\varphi)_{\Delta\vdash D}$ où $\mathrm{lfp}(\varphi)$ est le plus petit point fixe de $\varphi$.

Cette définition a pour but essentiel d'assurer la correction de la relation d'accessibilité (Lemme 134) : si $d(\vec{u})$ est calculable alors chaque $u_j$ accessible ($j \in \mathrm{Acc}(d)$) est calculable. C'est ce qui nous permettra d'assurer la calculabilité des variables d'un membre gauche de règle si les arguments du membre gauche sont calculables, et donc la calculabilité des membres droits qui appartiennent à la clôture calculable de la règle.

**Lemme 127** $\varphi^I$ est bien défini et est bien une interprétation.

**Preuve.** Montrons tout d'abord que $\varphi^I$ est bien défini. Tout d'abord, $\xi$ existe du fait de l'hypothèse **(I6)**. Soit $\Gamma_d = \vec{y} : \vec{U}$. Il faut s'assurer que $\theta : \Gamma_d \to \Delta'$ et que $\xi|_{\Delta'}$ est compatible avec $(\theta, \Gamma_d, \Delta')$. Par préservation du typage, $\Delta' \vdash d(\vec{u}) : D(\vec{t})$. Par inversion, $\Delta' \vdash d(\vec{u}) : D(\vec{v}\theta)$, $D(\vec{v}\theta)\ \mathbb{C}^*_{\Delta'}\ D(\vec{t})$ et, pour tout $j$, $\Delta' \vdash u_j : U_j\theta$. Donc, $\theta : \Gamma_d \to \Delta'$. Soit $Y \in \mathrm{FV}^{\Box}(U_j)$. Nous avons $Y\xi = S_{\iota_Y} \in \mathcal{R}_{\Delta_{\iota_Y}\vdash t_{\iota_Y}}$. Par le Lemme 114 (c), $Y\xi|_{\Delta'} \in \mathcal{R}_{\Delta'\vdash t_{\iota_Y}}$. Par **(A1)**, comme $D$ est constant, pour tout $i$, $v_i\theta \downarrow t_i$. Ainsi, par le Lemme 114 (b), $\mathcal{R}_{\Delta'\vdash t_{\iota_Y}} = \mathcal{R}_{\Delta'\vdash v_{\iota_Y}\theta}$. Par **(I6)**, $v_{\iota_Y} = Y$. Donc $Y\xi \in \mathcal{R}_{\Delta'\vdash Y\theta}$.

Enfin, il faut s'assurer que les interprétations nécessaires à la définition de $[\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ sont toutes bien définies. L'interprétation des symboles de prédicat constants plus petits que $D$ est $K$. L'interprétation des symboles de prédicat constants équivalents à $D$ est $I$. Par **(I4)** et **(I5)**, les symboles de prédicat constants plus grands que $D$ ou les symboles de prédicat définis ne peuvent apparaître qu'à des positions neutres. Or il est facile de vérifier que les termes qui se trouvent à de telles positions ne sont pas interprétés.

Maintenant, montrons que $\varphi^I_{\Delta\vdash D} \in \mathcal{R}_{\Delta\vdash D}$. Pour cela, on doit montrer que $R = \varphi^I_{\Delta\vdash D}(\vec{a})$ est inclus dans $\overline{\mathbb{T}}_{\Delta_n\vdash D(\vec{t})}$ (immédiat) et satisfait les propriétés (R1) à (R4) :

**(R1)** Par définition.

**(R2)** Soit $\Delta' \vdash u \in R$ et $u'$ tel que $u \to u'$. Par définition, $\Delta' \vdash u \in \overline{\mathbb{SN}}_{\Delta_n\vdash D(\vec{t})}$. Donc, $\Delta' \vdash u' \in \overline{\mathbb{SN}}_{\Delta_n\vdash D(\vec{t})}$. Supposons que $u' \to^* d(\vec{u})$ avec $\tau_d = (\vec{y} : \vec{U})D(\vec{v})$. Alors, $u \to^* d(\vec{u})$. Donc, pour tout $j \in \mathrm{Acc}(d)$, $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $\Delta' \vdash u' \in R$.

**(R3)** Soit $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta_n \vdash D(\vec{t})}$ tel que $u$ soit neutre et, pour tout $u'$ tel que $u \to u'$, $\Delta' \vdash u' \in R$. Alors, $\Delta' \vdash u \in \overline{\mathbb{SN}}_{\Delta_n \vdash D(\vec{t})}$. Supposons maintenant que $u \to^* d(\vec{u})$ avec $\tau_d = (\vec{y} : \vec{U})D(\vec{v})$. Puisque $u$ est neutre, il existe $u'$ tel que $u \to u'$ et $u' \to^* d(\vec{u})$. Donc, pour tout $j \in \mathrm{Acc}(d)$, $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $\Delta' \vdash u \in R$.

**(R4)** Soit $\Delta' \vdash u \in R$ et supposons que $\Delta' \subseteq \Delta'' \in \mathbb{E}$. Alors, $\Delta'' \vdash u \in \overline{\mathbb{SN}}_{\Delta_n \vdash D(\vec{t})}$. Supposons maintenant que $u \to^* d(\vec{u})$ avec $\tau_d = (\vec{y} : \vec{U})D(\vec{v})$. Alors, pour tout $j \in \mathrm{Acc}(d)$, $\Delta' \vdash u_j \in R_j$ où $R_j = [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$. D'après le Lemme 121 (b), $R_j$ appartient à $\mathcal{R}_{\Delta_n \vdash U_j\theta}$ et donc satisfait **(R4)**. Donc $\Delta'' \vdash u_j \in R_j$ et $\Delta'' \vdash u \in R$.

Enfin, il ne nous reste plus qu'à montrer les propriétés (P1) à (P3). Pour (P1), la stabilité par réduction, c'est immédiat car, si $\vec{t} \to \vec{t'}$ alors $\overline{\mathbb{SN}}_{\Delta_n \vdash D(\vec{t})} = \overline{\mathbb{SN}}_{\Delta_n \vdash D(\vec{t'})}$. Pour (P2) et (P3), il est facile de voir que les fonctions ont le même domaine et sont égales. ■

**Lemme 128** $\varphi^I$ est monotone en ses arguments inductifs.

**Preuve.** Soit $i \in \mathrm{Ind}(D)$. Nous devons montrer que $S_i \leq S'_i$ implique $\varphi^I_{\Delta \vdash D}(\vec{a}) \subseteq \varphi^I_{\Delta \vdash D}(\vec{a'})$. Soit alors $\Delta' \vdash u \in \varphi^I_{\Delta \vdash D}(\vec{a})$ et montrons que $\Delta' \vdash u \in \varphi^I_{\Delta \vdash D}(\vec{a'})$. Tout d'abord, $\Delta' \vdash u \in \overline{\mathbb{SN}}_{\Delta_n \vdash D(\vec{t})}$. Supposons maintenant que $u$ se réduise sur un terme de la forme $d(\vec{u})$ avec $d$ un constructeur de type $(\vec{y} : \vec{U})D(\vec{v})$. Soit $j \in \mathrm{Acc}(d)$. Nous devons montrer que $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}}$ où $\theta = \{\vec{y} \mapsto \vec{u}\}$ et, pour tout $Y \in \mathrm{FV}^\Box(U_j)$, $Y\xi' = S'_{\iota_Y}$.

Nous avons $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ où, pour tout $Y \in \mathrm{FV}^\Box(U_j)$, $Y\xi = S_{\iota_Y}$. Si, pour tout $Y \in \mathrm{FV}^\Box(U_j)$, $\iota_Y \neq i$, alors $\xi = \xi'$ et $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}}$. Supposons maintenant qu'il existe $Y \in \mathrm{FV}^\Box(U_j)$ tel que $\iota_Y = i$. Alors, $Y\xi = S_i \leq Y\xi' = S'_i$. Par **(I2)**, $\mathrm{Pos}(Y, U_j) \subseteq \mathrm{Pos}^+(U_j)$. Enfin, $U_j$ vérifie **(I3)**, **(I4)** et **(I5)**.

Montrons alors par récurrence sur $T$ que, pour tout $\Gamma \vdash T \in \overline{\mathbb{TY}}$, $\Delta \in \mathbb{E}$, $\theta : \Gamma \to \Delta$, $\xi, \xi'$ compatibles avec $(\theta, \Gamma, \Delta)$ tels que $Y\xi \leq Y\xi'$ et, pour tout $X \neq Y$, $X\xi = X\xi'$ :

– si $\mathrm{Pos}(Y, T) \subseteq \mathrm{Pos}^+(T)$ et $T$ vérifie (I3), (I4) et (I5) alors $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$,

– si $\mathrm{Pos}(Y, T) \subseteq \mathrm{Pos}^-(T)$ et $T$ vérifie (I3$^-$), (I4) et (I5) alors $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \geq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$,

où (I3$^-$) est la propriété $\forall D \in \mathcal{CF}^\Box, D =_{\mathcal{C}} C \Rightarrow \mathrm{Pos}(D, T) \subseteq \mathrm{Pos}^-(T)$. Nous détaillons le premier cas ; le second se traite de manière similaire.

- $T = s$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = \overline{\mathbb{SN}}_{\Delta \vdash s} = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$.
- $T = E(\vec{t})$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = I_{\Delta \vdash E}(\vec{a})$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'} = I_{\Delta \vdash E}(\vec{a}')$ avec $a_i = a'_i = \Delta \vdash t_i\theta$ si $x_i \in \mathcal{X}^\star$, et $a_i = (\Delta \vdash t_i\theta, S_i)$, $a'_i = (\Delta \vdash t_i\theta, S'_i)$, $S_i = [\![\Gamma \vdash t_i]\!]^I_{\Delta,\theta,\xi}$ et $S'_i = [\![\Gamma \vdash t_i]\!]^I_{\Delta,\theta,\xi'}$ si $x_i \in \mathcal{X}^\Box$. Comme $T$ vérifie (I3), (I4) et (I5), nous avons $E \in \mathcal{CF}^\Box$ et $E \leq_{\mathcal{C}} D$. Donc, $I_{\Delta \vdash E}$ est monotone en ses arguments inductifs. Soit $i \leq \alpha_E$. Montrons que $t_i$ vérifie (I3), (I4) et (I5) :

**(I3)** Soit $D' =_{\mathcal{C}} D$. Nous devons montrer que $\mathrm{Pos}(D', t_i) \subseteq \mathrm{Pos}^+(t_i)$. Si $\mathrm{Pos}(D', t_i) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D', t_i)$ alors $i.p \in \mathrm{Pos}(D', T)$. Comme $\mathrm{Pos}(D', T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = \{\varepsilon\} \cup \bigcup \{i.\mathrm{Pos}^+(t_i) \mid i \in \mathrm{Ind}(E)\}$, nous avons $i \in \mathrm{Ind}(E)$ et $p \in \mathrm{Pos}^+(t_i)$.

**(I4)** et **(I5)** Soit $D' >_{\mathcal{C}} D$ ou $D' \in \mathcal{DF}^{\square}$. Nous devons montrer que $\mathrm{Pos}(D', t_i) \subseteq \mathrm{Pos}^0(t_i)$. Si $\mathrm{Pos}(D', t_i) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D', t_i)$ alors $i.p \in \mathrm{Pos}(D', T)$. Comme $\mathrm{Pos}(D', T) \subseteq \mathrm{Pos}^0(T)$ et $\mathrm{Pos}^0(T) = \{\varepsilon\} \cup \bigcup \{i.\mathrm{Pos}^0(t_i) \mid i \in \mathrm{Ind}(E)\}$, nous avons $i \in \mathrm{Ind}(E)$ et $p \in \mathrm{Pos}^0(t_i)$.

Voyons maintenant les relations entre $S_i$ et $S'_i$. Si $\mathrm{Pos}(Y, t_i) = \emptyset$ alors, par hypothèse de récurrence, $S_i = S'_i$. S'il existe $p \in \mathrm{Pos}(Y, t_i)$ alors $i.p \in \mathrm{Pos}(Y, T)$. Comme $\mathrm{Pos}(Y, T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = \{\varepsilon\} \cup \bigcup \{i.\mathrm{Pos}^+(t_i) \mid i \in \mathrm{Ind}(E)\}$, nous avons nécessairement $i \in \mathrm{Ind}(E)$ et $p \in \mathrm{Pos}^+(t_i)$. Donc, par hypothèse de récurrence, $S_i \leq S'_i$. Finalement, comme $I_{\Delta \vdash E}$ est monotone en ses arguments inductifs, on conclut que $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$.

- $T = Y$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = Y\xi \leq Y\xi' = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$. Or, $\mathrm{Pos}(Y, Y) = \varepsilon \subseteq \mathrm{Pos}^+(Y)$.
- $T = X \neq Y$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = X\xi = X\xi' = [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$.
- $T = (x : U)V$. Soit $\Gamma' = \Gamma, x : U$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = \{\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta \vdash (x:U\theta)V\theta} \mid \forall \Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}},\ \Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi|_{\Delta''}}\}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'} = \{\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta \vdash (x:U\theta)V\theta} \mid \forall \Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}},\ \Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi'|_{\Delta''}}\}$ où $\theta' = \theta \cup \{x \mapsto u\}$. Nous devons montrer que $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \subseteq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$. Soit $\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta \vdash (x:U\theta)V\theta}$ et $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}}$. Comme $\mathrm{Pos}(Y, T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}^-(U) \cup 2.\mathrm{Pos}^+(V)$, nous avons $\mathrm{Pos}(Y, U) \subseteq \mathrm{Pos}^-(U)$ et $\mathrm{Pos}(Y, V) \subseteq \mathrm{Pos}^+(V)$. Montrons que $U$ vérifie (I3$^-$), (I4) et (I5) :

  **(I3$^-$)** Soit $D' =_{\mathcal{C}} D$. Nous devons montrer que $\mathrm{Pos}(D', U) \subseteq \mathrm{Pos}^-(U)$. Si $\mathrm{Pos}(D', U) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D', U)$ alors $1.p \in \mathrm{Pos}(D', T)$. Comme $\mathrm{Pos}(D', T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}^-(U) \cup 2.\mathrm{Pos}^+(V)$, nous avons $p \in \mathrm{Pos}^-(U)$.

  **(I4)** et **(I5)** Soit $D' =_{\mathcal{C}} D$ ou $D' \in \mathcal{DF}^{\square}$. Nous devons montrer que $\mathrm{Pos}(D', U) \subseteq \mathrm{Pos}^0(U)$. Si $\mathrm{Pos}(D', U) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D', U)$ alors $1.p \in \mathrm{Pos}(D', T)$. Comme $\mathrm{Pos}(D', T) \subseteq \mathrm{Pos}^0(T)$ et $\mathrm{Pos}^0(T) = 1.\mathrm{Pos}^0(U) \cup 2.\mathrm{Pos}^0(V)$, nous avons $p \in \mathrm{Pos}^0(U)$.

  De même, $V$ vérifie (I3), (I4) et (I5). Donc, par hypothèse de récurrence, $[\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}} \subseteq [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $[\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi|_{\Delta''}} \subseteq [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi'|_{\Delta''}}$. Ainsi, $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$, $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi|_{\Delta''}}$ et $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi'|_{\Delta''}}$. Donc, $\Delta' \vdash t \in [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \subseteq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$.
- $T = (X : K)V$. Similaire au cas précédent.
- $T = [x : U]V$. $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ est la fonction qui à $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$ associe $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}}$ où $\Gamma' = \Gamma, x : U$ et $\theta' = \theta \cup \{x \mapsto u\}$. $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$ est la fonction qui à $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$ associe $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi'|_{\Delta'}}$ où $\Gamma' = \Gamma, x : U$ et $\theta' = \theta \cup \{x \mapsto u\}$. Nous devons montrer que, pour tout $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta \vdash U\theta}$, $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}} \leq [\![\Gamma' \vdash$

$V]\!]^I_{\Delta',\theta',\xi'|_{\Delta'}}$. Comme $\mathrm{Pos}(Y,T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}(U) \cup 2.\mathrm{Pos}^+(V)$, nous avons $\mathrm{Pos}(Y,V) \subseteq \mathrm{Pos}^+(V)$. Montrons que $V$ vérifie (I3), (I4) et (I5).

**(I3)** Soit $D' =_{\mathcal{C}} D$. Nous devons montrer que $\mathrm{Pos}(D',V) \subseteq \mathrm{Pos}^+(V)$. Si $\mathrm{Pos}(D',V) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D',V)$ alors $2.p \in \mathrm{Pos}(D',T)$. Comme $\mathrm{Pos}(D',T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}(U) \cup 2.\mathrm{Pos}^+(V)$, nous avons $p \in \mathrm{Pos}^+(V)$.

**(I4)** et **(I5)** Soit $D' =_{\mathcal{C}} D$ ou $D' \in \mathcal{DF}^{\Box}$. Nous devons montrer que $\mathrm{Pos}(D',V) \subseteq \mathrm{Pos}^0(V)$. Si $\mathrm{Pos}(D',V) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D',V)$ alors $2.p \in \mathrm{Pos}(D',T)$. Comme $\mathrm{Pos}(D',T) \subseteq \mathrm{Pos}^0(T)$ et $\mathrm{Pos}^0(T) = 1.\mathrm{Pos}(U) \cup 2.\mathrm{Pos}^0(V)$, nous avons $p \in \mathrm{Pos}^0(V)$.

Donc, par hypothèse de récurrence, $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}} \leq [\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi'|_{\Delta'}}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$.

- $T = [X\!:\!K]V$. Similaire au cas précédent.
- $T = Vu$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Gamma \vdash u)$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi'}(\Gamma \vdash u)$. Comme $\mathrm{Pos}(Y,T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}^+(V) \cup 2.\mathrm{Pos}(u)$, nous avons $\mathrm{Pos}(Y,V) \subseteq \mathrm{Pos}^+(V)$. Montrons que $V$ vérifie (I3), (I4) et (I5).

  **(I3)** Soit $D' =_{\mathcal{C}} D$. Nous devons montrer que $\mathrm{Pos}(D',V) \subseteq \mathrm{Pos}^+(V)$. Si $\mathrm{Pos}(D',V) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D',V)$ alors $1.p \in \mathrm{Pos}(D',T)$. Comme $\mathrm{Pos}(D',T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}^+(V) \cup 2.\mathrm{Pos}(u)$, nous avons $p \in \mathrm{Pos}^+(V)$.

  **(I4)** et **(I5)** Soit $D' =_{\mathcal{C}} D$ ou $D' \in \mathcal{DF}^{\Box}$. Nous devons montrer que $\mathrm{Pos}(D',V) \subseteq \mathrm{Pos}^0(V)$. Si $\mathrm{Pos}(D',V) = \emptyset$, c'est immédiat. S'il existe $p \in \mathrm{Pos}(D',V)$ alors $1.p \in \mathrm{Pos}(D',T)$. Comme $\mathrm{Pos}(D',T) \subseteq \mathrm{Pos}^0(T)$ et $\mathrm{Pos}^0(T) = 1.\mathrm{Pos}^0(V) \cup 2.\mathrm{Pos}(u)$, nous avons $p \in \mathrm{Pos}^0(V)$.

  Donc, par hypothèse de récurrence, $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi'}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$.
- $T = VU$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Gamma \vdash U, [\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi})$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi'}(\Gamma \vdash U, [\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi'})$. Comme $\mathrm{Pos}(Y,T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}^+(V)$, nous avons $\mathrm{Pos}(Y,V) \subseteq \mathrm{Pos}^+(V)$ et $\mathrm{Pos}(Y,U) = \emptyset$. Nous avons vu dans le cas précédent que $V$ vérifie (I3), (I4) et (I5). Montrons que $U$ vérifie aussi (I3), (I3$^-$), (I4) et (I5).

  **(I3)** et **(I3$^-$)** Soit $D' =_{\mathcal{C}} D$. Nous devons montrer que $\mathrm{Pos}(D',U) \subseteq \mathrm{Pos}^+(U) \cup \mathrm{Pos}^-(U)$. S'il existe $p \in \mathrm{Pos}(D',U)$ alors $2.p \in \mathrm{Pos}(D',T)$. Comme $\mathrm{Pos}(D',T) \subseteq \mathrm{Pos}^+(T)$ et $\mathrm{Pos}^+(T) = 1.\mathrm{Pos}^+(V)$, ce n'est pas possible. Donc, $\mathrm{Pos}(D',U) = \emptyset \subseteq \mathrm{Pos}^+(U) \cup \mathrm{Pos}^-(U)$.

  **(I4)** et **(I5)** Soit $D' =_{\mathcal{C}} D$ ou $D' \in \mathcal{DF}^{\Box}$. Nous devons montrer que $\mathrm{Pos}(D',V) \subseteq \mathrm{Pos}^0(V)$. S'il existe $p \in \mathrm{Pos}(D',U)$ alors $2.p \in \mathrm{Pos}(D',T)$. Comme $\mathrm{Pos}(D',T) \subseteq \mathrm{Pos}^0(T)$ et $\mathrm{Pos}^0(T) = 1.\mathrm{Pos}^0(V)$, ce n'est pas possible. Donc, $\mathrm{Pos}(D',U) = \emptyset \subseteq \mathrm{Pos}^0(U)$.

  Donc, par hypothèse de récurrence, $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi'}$, $[\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi'}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'}$. ■

**Lemme 129** $\varphi$ est monotone.

**Preuve.** Soient $I, I' \in \mathcal{I}^m$ tel que $I \leq I'$. Nous devons montrer que, pour tout $D =_{\mathcal{F}} C$, $\Delta \in \mathbb{E}$ et $\vec{a}$, $\varphi^I_{\Delta\vdash D}(\vec{a}) \leq \varphi^{I'}_{\Delta\vdash D}(\vec{a})$. Soit alors $\Delta' \vdash u \in \varphi^I_{\Delta\vdash D}(\vec{a})$ et montrons que $\Delta' \vdash u \in \varphi^{I'}_{\Delta\vdash D}(\vec{a})$. Tout d'abord, $\Delta' \vdash u \in \overline{\mathbb{SN}}_{\Delta_n\vdash D(\vec{t})}$. Supposons maintenant que $u$ se réduise sur un terme de la forme $d(\vec{u})$ avec $d$ un constructeur de type $(\vec{y} : \vec{U})D(\vec{v})$. Soit $j \in \mathrm{Acc}(d)$. Nous devons montrer que $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^{I'}_{\Delta',\theta,\xi|_{\Delta'}}$ où $\theta = \{\vec{y} \mapsto \vec{u}\}$ et, pour tout $Y \in \mathrm{FV}^{\square}(U_j)$, $Y\xi = S_{\iota_Y}$.

Nous avons $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $U_j$ vérifie **(I3)**, **(I4)** et **(I5)**. Montrons alors par récurrence sur $T$ que, pour tout $\Gamma\vdash T \in \overline{\mathbb{TY}}$, $\Delta \in \mathbb{E}$, $\theta : \Gamma \to \Delta$, $\xi$ compatible avec $(\theta, \Gamma, \Delta)$ :

– si $T$ vérifie (I3), (I4) et (I5) alors $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma\vdash T]\!]^{I'}_{\Delta,\theta,\xi}$,

– si $T$ vérifie (I3$^-$), (I4) et (I5) alors $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} \geq [\![\Gamma\vdash T]\!]^{I'}_{\Delta,\theta,\xi}$,

où (I3$^-$) est la propriété $\forall D \in \mathcal{CF}^{\square}, D =_{\mathcal{C}} C \Rightarrow \mathrm{Pos}(D, T) \subseteq \mathrm{Pos}^-(T)$. Nous détaillons le premier cas ; le second se traite de manière similaire.

- $T = s$. Nous avons $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} = \overline{\mathbb{SN}}_{\Delta\vdash s} = [\![\Gamma\vdash T]\!]^{I'}_{\Delta,\theta,\xi}$.
- $T = E(\vec{t})$. Nous avons $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} = I_{\Delta\vdash E}(\vec{a})$ et $[\![\Gamma\vdash T]\!]^{I'}_{\Delta,\theta,\xi} = I'_{\Delta\vdash E}(\vec{a}')$ avec $a_i = a'_i = \Delta\vdash t_i\theta$ si $x_i \in \mathcal{X}^\star$, et $a_i = (\Delta\vdash t_i\theta, S_i)$, $a'_i = (\Delta\vdash t_i\theta, S'_i)$, $S_i = [\![\Gamma\vdash t_i]\!]^I_{\Delta,\theta,\xi}$ et $S'_i = [\![\Gamma\vdash t_i]\!]^{I'}_{\Delta,\theta,\xi}$ si $x_i \in \mathcal{X}^{\square}$. Comme $T$ vérifie (I3), (I4) et (I5), nous avons $E \in \mathcal{CF}^{\square}$ et $E \leq_{\mathcal{C}} D$. Donc, $I_{\Delta\vdash E}$ et $I'_{\Delta\vdash E}$ sont monotones en leurs arguments inductifs. Nous avons vu dans le lemme précédent que $t_i$ vérifie (I3), (I4) et (I5). Donc, par hypothèse de récurrence, $S_i \leq S'_i$. Finalement, comme $I_{\Delta\vdash E}$ est monotone en ses arguments inductifs et $I \leq I'$, on conclut que $I_{\Delta\vdash E}(\vec{a}) \leq I_{\Delta\vdash E}(\vec{a}') \leq I'_{\Delta\vdash E}(\vec{a}')$ et donc que $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi'}$.
- $T = X$. Nous avons $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} = Y\xi = [\![\Gamma\vdash T]\!]^{I'}_{\Delta,\theta,\xi}$.
- $T = (x : U)V$. Soit $\Gamma' = \Gamma, x : U$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = \{\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta\vdash(x:U\theta)V\theta} \mid \forall \Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}, \Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi|_{\Delta''}}\}$ et $[\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi} = \{\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta\vdash(x:U\theta)V\theta} \mid \forall \Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^{I'}_{\Delta',\theta,\xi|_{\Delta'}}, \Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^{I'}_{\Delta'',\theta',\xi|_{\Delta''}}\}$ où $\theta' = \theta \cup \{x \mapsto u\}$. Nous devons montrer que $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} \subseteq [\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi}$. Soit $\Delta' \vdash t \in \overline{\mathbb{T}}_{\Delta\vdash(x:U\theta)V\theta}$ et $\Delta'' \vdash u \in [\![\Gamma \vdash U]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}}$. Nous avons vu dans le lemme précédent que $U$ vérifie (I3$^-$), (I4) et (I5) et que $V$ vérifie (I3), (I4) et (I5). Donc, par hypothèse de récurrence, $[\![\Gamma\vdash U]\!]^{I'}_{\Delta',\theta,\xi|_{\Delta'}} \subseteq [\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ et $[\![\Gamma'\vdash V]\!]^I_{\Delta'',\theta',\xi|_{\Delta''}} \subseteq [\![\Gamma'\vdash V]\!]^{I'}_{\Delta'',\theta',\xi|_{\Delta''}}$. Ainsi, $\Delta''\vdash u \in [\![\Gamma\vdash U]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$, $\Delta''\vdash tu \in [\![\Gamma' \vdash V]\!]^I_{\Delta'',\theta',\xi|_{\Delta''}}$ et $\Delta'' \vdash tu \in [\![\Gamma' \vdash V]\!]^{I'}_{\Delta'',\theta',\xi|_{\Delta''}}$. Donc, $\Delta' \vdash t \in [\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi}$ et $[\![\Gamma\vdash T]\!]^I_{\Delta,\theta,\xi} \subseteq [\![\Gamma\vdash T]\!]^{I'}_{\Delta,\theta,\xi}$.
- $T = (X : K)V$. Similaire au cas précédent.
- $T = [x : U]V$. $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi}$ est la fonction qui à $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta\vdash U\theta}$ associe $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}}$ où $\Gamma' = \Gamma, x : U$ et $\theta' = \theta \cup \{x \mapsto u\}$. $[\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi}$ est la fonction qui à $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta\vdash U\theta}$ associe $[\![\Gamma' \vdash V]\!]^{I'}_{\Delta',\theta',\xi|_{\Delta'}}$ où $\Gamma' = \Gamma, x : U$ et $\theta' = \theta \cup \{x \mapsto u\}$. Nous devons montrer que, pour tout $\Delta' \vdash u \in \overline{\mathbb{T}}_{\Delta\vdash U\theta}$, $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}} \leq [\![\Gamma' \vdash V]\!]^{I'}_{\Delta',\theta',\xi|_{\Delta'}}$. Nous avons vu dans le lemme précédent que $V$ vérifie (I3), (I4) et (I5). Donc, par hypothèse de récurrence, $[\![\Gamma' \vdash V]\!]^I_{\Delta',\theta',\xi|_{\Delta'}} \leq [\![\Gamma' \vdash V]\!]^{I'}_{\Delta',\theta',\xi|_{\Delta'}}$ et

$[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi}$.

- $T = [X\!:\!K]V$. Similaire au cas précédent.
- $T = Vu$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Gamma \vdash u)$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi'}(\Gamma \vdash u)$. Nous avons vu dans le lemme précédent que $V$ vérifie (I3), (I4) et (I5). Donc, par hypothèse de récurrence, $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash V]\!]^{I'}_{\Delta,\theta,\xi}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi}$.
- $T = VU$. Nous avons $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi}(\Gamma \vdash U, [\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi})$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi'} = [\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi'}(\Gamma \vdash U, [\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi'})$. Nous avons vu dans le lemme précédent que $U$ vérifie (I3), (I3$^-$), (I4) et (I5), et que $V$ vérifie (I3), (I4) et (I5). Donc, par hypothèse de récurrence, $[\![\Gamma \vdash V]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash V]\!]^{i'}_{\Delta,\theta,\xi}$, $[\![\Gamma \vdash U]\!]^I_{\Delta,\theta,\xi} = [\![\Gamma \vdash U]\!]^{I'}_{\Delta,\theta,\xi}$ et $[\![\Gamma \vdash T]\!]^I_{\Delta,\theta,\xi} \leq [\![\Gamma \vdash T]\!]^{I'}_{\Delta,\theta,\xi}$. ■

Comme $(\mathcal{I}^m, \leq)$ est un treillis complet, $\varphi$ admet un plus petit point fixe $I$ qui est une interprétation pour tous les symboles de prédicat constants équivalents à $C$. Par récurrence sur $>_{\mathcal{C}}$ on obtient ainsi une interprétation $I$ pour tous les symboles de prédicat constants.

Notons que, dans le cas d'un symbole de prédicat constant primitif, l'interprétation est simplement l'ensemble des termes fortement normalisables de ce type.

**Lemme 130 (Interprétation des prédicats constants primitifs)** Si $C$ est un symbole de prédicat constant primitif alors $I_{\Delta \vdash C} = \top_{\Delta \vdash C}$.

**Preuve.** Comme $I_{\Delta \vdash C} \leq \top_{\Delta \vdash C}$, il nous suffit de montrer que, pour tout $u \in \mathcal{SN}$, $C$ primitif de type $(\vec{x} : \vec{T})\star$, $\vec{a}$ arguments de $I_{\Delta \vdash C}$, $a_i = \Delta_i \vdash t_i$ si $x_i \in \mathcal{X}^\star$, $a_i = (\Delta_i \vdash t_i, S_i)$ si $x_i \in \mathcal{X}^\square$, et $\Delta' \supseteq \Delta_n$, si $\Delta' \vdash u : C(\vec{t})$ alors $\Delta' \vdash u \in I_{\Delta \vdash C}(\vec{a})$, par récurrence sur $u$ avec $\rightarrow \cup \rhd$ comme ordre bien fondé. Supposons que $u \rightarrow^* c(\vec{u})$ avec $c$ un constructeur de type $(\vec{y} : \vec{U})C(\vec{v})$. Si $u \rightarrow^+ c(\vec{u})$, nous pouvons conclure par hypothèse de récurrence. Supposons alors que $u = c(\vec{u})$. Dans ce cas, nous devons montrer que, pour tout $j \in \mathrm{Acc}(c)$, $\Delta' \vdash u_j \in [\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}}$ où $\theta = \{\vec{y} \mapsto \vec{u}\}$ et, pour tout $Y \in \mathrm{FV}^\square(U_j)$, $Y\xi = S_{\iota_Y}$. Par définition des symboles de prédicat primitifs, pour tout $j \in \mathrm{Acc}(c)$, $U_j$ est de la forme $D(\vec{w})$ avec $D$ un symbole de prédicat primitif. Supposons que $\tau_D = (\vec{z} : \vec{V})\star$. Soit $a'_i = \Delta' \vdash w_i\theta$ si $z_i \in \mathcal{X}^\star$, et $a'_i = (\Delta' \vdash w_i\theta, [\![\vec{y} : \vec{U} \vdash w_i]\!]^I_{\Delta',\theta,\xi'|_{\Delta'}})$ si $z_i \in \mathcal{X}^\square$. Comme $u_j \in \mathcal{SN}$ et $\Delta' \vdash u_j : D(\vec{w}\theta)$, par hypothèse de récurrence, $\Delta' \vdash u_j \in I_{\Delta \vdash D}(\vec{a}')$. Or, par **(P3)**, $I_{\Delta \vdash D}(\vec{a}') = I_{\Delta' \vdash D}(\vec{a}')$ et $[\![\vec{y} : \vec{U} \vdash U_j]\!]^I_{\Delta',\theta,\xi|_{\Delta'}} = I_{\Delta' \vdash D}(\vec{a}')$. Donc, $\Delta' \vdash u \in I_{\Delta \vdash C}(\vec{a})$. ■

## 8.5 Ordre de réductibilité

Dans ce sous-chapitre, on suppose donnée une interprétation $J$ pour les symboles de prédicat définis et nous désignons $[\![\Gamma \vdash T]\!]^{I \cup J}_{\Delta,\theta,\xi}$ par $[\![\Gamma \vdash T]\!]_{\Delta,\theta,\xi}$.

Le point fixe de la fonction $\varphi$ définie dans le sous-chapitre précédent peut être atteint par itération transfinie à partir du plus petit élément de $\mathcal{I}^m$. Notons par $\varphi^{\mathfrak{a}}$ l'interprétation obtenue après $\mathfrak{a}$ itérations.

**Définition 131 (Ordre d'une fermeture)** L'*ordre* de $\Delta' \vdash t \in I_{\Delta\vdash C}(\vec{a})$, $o(\Delta' \vdash t)$, est le plus petit ordinal $\mathfrak{a}$ tel que $\Delta' \vdash t \in \varphi^{\mathfrak{a}}_{\Delta\vdash C}(\vec{a})$.

Cette notion d'ordre va nous permettre de définir un ordre bien fondé dans lequel les définitions sur des prédicats strictement positifs décroissent strictement. En effet, dans ce cas-là, l'ordre sous-terme ne suffit pas. Dans l'exemple de l'addition sur les ordinaux de Brouwer (voir page 69), nous avons la règle :

$$+(x, lim(f)) \;\to\; lim([n\!:\!nat] + (x, fn))$$

Nous avons un appel récursif avec $fn$ comme argument, qui n'est pas un sous-terme de $lim(f)$. Cependant, du fait de la définition de l'interprétation des symboles de prédicat constant et de la définition de l'interprétation du produit, nous pouvons dire que, si $lim(f)$ est calculable alors $f$ est calculable, et donc que, pour tout $n$ calculable, $fn$ est calculable. Autrement dit, l'ordre de $lim(f)$ est plus grand que celui de $fn$ : $o(lim(f)) > o(fn)$.

**Définition 132 (Ordre de réductibilité)** On suppose donnés une précédence $\geq_{\mathcal{F}}$ sur $\mathcal{F}$ et un assignement de statuts *stat* compatible avec $\geq_{\mathcal{F}}$. Soit $f$ un symbole d'arité non nulle et de statut $stat_f = lex(m_1, \ldots, m_k)$. Notons $\Theta_f$ l'ensemble des quadruplets $(g, \Delta, \theta, \xi)$ tels que, si $\tau_g = (\vec{x} : \vec{T})U$ et $\Gamma_g = \vec{x} : \vec{T}$, alors $g =_{\mathcal{F}} f$, $\Delta \in \mathbb{E}$, $\theta : \Gamma_g \to \Delta$, $\xi$ est compatible avec $(\theta, \Gamma_g, \Delta)$, $(\theta, \Gamma_g, \Delta)$ est valide par rapport à $\xi$. Nous munissons $\Theta_f$ de l'ordre $\sqsupset_f$ défini par :

- $(g, \Delta, \theta, \xi) \sqsupset_f (g', \Delta', \theta', \xi')$ si $\vec{m}\theta \; (\sqsupset_f^1, \ldots, \sqsupset_f^k)_{\text{lex}} \; \vec{m}\theta'$,
- $mul(\vec{u}) \sqsupset_f^i mul(\vec{u}')$ si $\{\vec{u}\} \; (\sqsupset^i)_{\text{mul}} \; \{\vec{u}'\}$ avec :
  - $\sqsupset^i = \gg$ si $i \in SP(f)$ où $u \gg u'$ si $o(\Delta \vdash u) > o(\Delta' \vdash u')$,
  - $\sqsupset^i = \to \cup \rhd$ sinon.

Nous munissons $\Theta = \bigcup\{\Theta_f \mid f \in \mathcal{F}\}$ de l'*ordre de réductibilité* $\sqsupset$ défini par $(g, \Delta, \theta, \xi) \sqsupset (g', \Delta', \theta', \xi')$ si $g >_{\mathcal{F}} g'$ ou, $g =_{\mathcal{F}} g'$ et $(g, \Delta, \theta, \xi) \sqsupset_f (g', \Delta', \theta', \xi')$.

**Lemme 133** L'ordre de réductibilité est bien fondé et compatible avec la réduction : si $\theta \to \theta'$ alors $(g, \Delta, \theta, \xi) \sqsupseteq (g, \Delta, \theta', \xi)$.

**Preuve.** L'ordre de réductibilité est bien fondé car les ordinaux sont bien fondés et les extensions lexicographiques et multi-ensembles préservent la bonne fondation. Il est compatible avec la réduction par définition de l'interprétation des symboles de prédicat constants. ■

Nous vérifions ci-après, d'une part, que la relation d'accessibilité est correcte : un sous-terme accessible dans un terme calculable est calculable, et d'autre part, que l'ordre sur les arguments est également correct : si $l_i >_2 u_j$ et $l_i$ est calculable alors $u_j$ est calculable et admet un ordre plus petit que celui de $l_i$.

**Lemme 134 (Correction de l'accessibilité)** Si $t : T \rhd_1^\rho u : U$, $\Gamma \vdash t\rho : T\rho$, $\sigma : \Gamma \to \Delta$ et $\Delta \vdash t\sigma \in [\![\Gamma \vdash T\rho]\!]_{\Delta,\sigma,\xi}$ alors $\Gamma \vdash u\rho : U\rho$ et $\Delta \vdash u\sigma \in [\![\Gamma \vdash U\rho]\!]_{\Delta,\sigma,\xi}$.

**Preuve.** Par définition de $\rhd_1^\rho$, nous avons $t$ de la forme $c(\vec{u})$ avec $c$ un constructeur de type $(\vec{y} : \vec{U})C(\vec{v})$, $T\rho = C(\vec{v})\gamma\rho$ où $\gamma = \{\vec{y} \mapsto \vec{u}\}$, $u = u_j$ avec $j \in \mathrm{Acc}(c)$ et $U\rho = U_j\gamma\rho$. Supposons que $\tau_C = (\vec{x} : \vec{T})\star$. Alors, $[\![\Gamma \vdash C(\vec{v})\gamma\rho]\!]_{\Delta,\sigma,\xi} = I_{\Delta\vdash C}(\vec{a})$ avec $a_i = \Delta \vdash v_i\gamma\rho\sigma$ si $x_i \in \mathcal{X}^\star$, et $a_i = (\Delta \vdash v_i\gamma\rho\sigma, S_i)$ où $S_i = [\![\Gamma \vdash v_i\gamma\rho]\!]_{\Delta,\sigma,\xi}$ si $x_i \in \mathcal{X}^\square$. Par définition de $I_{\Delta\vdash C}$, $\Delta \vdash u_j\sigma \in [\![\Gamma_c \vdash U_j]\!]_{\Delta,\gamma\rho\sigma,\xi'}$ avec, pour tout $Y \in \mathrm{FV}^\square(U_j)$, $Y\xi' = S_{\iota_Y} = [\![\Gamma \vdash v_{\iota_Y}\gamma\rho]\!]_{\Delta,\sigma,\xi}$. Par **(I6)**, $v_{\iota_Y} = Y$ et $Y\xi' = [\![\Gamma \vdash Y\gamma\rho]\!]_{\Delta,\sigma,\xi}$. De $\Gamma \vdash t\rho : T\rho$, par inversion, on déduit que, pour tout $i$, $\Gamma \vdash u_i\rho : U_i\gamma\rho$. Donc, $\gamma\rho : \Gamma_c \to \Gamma$. Ainsi, par substitution de candidats, $[\![\Gamma_c \vdash U_j]\!]_{\Delta,\gamma\rho\sigma,\xi'} = [\![\Gamma \vdash U_j\gamma\rho]\!]_{\Delta,\sigma,\xi}$ et $\Delta \vdash u_j\sigma \in [\![\Gamma \vdash U\rho]\!]_{\Delta,\sigma,\xi}$. ■

**Lemme 135 (Correction de l'ordre sur les arguments)** Supposons que $t : T >_2 u : U$, c'est-à-dire que $t$ soit de la forme $c(\vec{t})$ avec $c$ un constructeur de type $(\vec{x} : \vec{T})C(\vec{v})$, $u$ soit de la forme $x\vec{u}$ avec $x \in \mathrm{dom}(\Gamma_0)$, $x\Gamma_0$ de la forme $(\vec{y} : \vec{U})D(\vec{w})$ et $D =_{\mathcal{C}} C$, et que $t : T\ (\rhd_2^\rho)^+\ x : V$ avec $V\rho = x\Gamma_0$.

Soit $\theta = \{\vec{y} \mapsto \vec{u}\}$. Si $\Gamma \vdash t\rho : T\rho$, $\sigma : \Gamma \to \Delta$, $\Delta \vdash t\sigma \in [\![\Gamma \vdash T\rho]\!]_{\Delta,\sigma,\xi}$ et, pour tout $i$, $\Delta \vdash u_i\sigma \in [\![\Gamma \vdash U_i\theta]\!]_{\Delta,\sigma,\xi}$, alors $\Delta \vdash x\sigma\vec{u}\sigma \in [\![\Gamma \vdash D(\vec{w})\theta]\!]_{\Delta,\sigma,\xi}$ et $o(\Delta \vdash t\sigma) > o(\Delta \vdash x\sigma\vec{u}\sigma)$.

**Preuve.** Soit $p$ le chemin suivi de $t$ à $x$ dans $t : T\ (\rhd_2^\rho)^+\ x : V$. Montrons que si $p = j_1 \dots j_n j_{n+1}$ alors $c(\vec{t}) : C(\vec{v})\gamma = c_0(\vec{t}^0) : C_0(\vec{v}^0)\gamma_0 \rhd_2^\rho c_1(\vec{t}^1) : C_1(\vec{v}^1)\gamma_1 \rhd_2^\rho \dots \rhd_2^\rho c_n(\vec{t}^n) : C_n(\vec{v}^n)\gamma_n \rhd_2^\rho x : V$ avec, si $\tau_{c_i} = (\vec{x}^i : \vec{T}^i)C(\vec{v}^i)$ et $\gamma_i = \{\vec{x}^i \mapsto \vec{t}^i\}$ :

– pour tout $i < n$, $c_{i+1}(\vec{t}^{i+1}) = t^i_{j_{i+1}}$, $T^i_{j_{i+1}} = C_{i+1}(\vec{w}^{i+1})$ et $\vec{w}^{i+1}\gamma_i\rho = \vec{v}^{i+1}\gamma_{i+1}\rho$,

– $T^n_{j_{n+1}} = (\vec{y} : \vec{U})D(\vec{w}')$ et $\vec{w}'\gamma_n = \vec{w}$.

Procédons par récurrence sur $n$. Si $n = 0$, c'est immédiat. Supposons alors que $c(\vec{t}) : C(\vec{v})\gamma \rhd_2^\rho t_{j_1} : T_{j_1}\gamma_0\ (\rhd_2^\rho)^+\ x : V$. Comme $t_{j_1} : T_{j_1}\gamma_0\ (\rhd_2^\rho)^+\ x : V$, $t_{j_1} = c_1(\vec{t}^1)$ avec $\tau_{c_1} = (\vec{x}^1 : \vec{T}^1)C_1(\vec{v}^1)$ et $T_{j_1}\gamma_0\rho = C_1(\vec{v}^1)\gamma_1\rho$ où $\gamma_1 = \{\vec{x}^1 \mapsto \vec{t}^1\}$. Par ailleurs, par définition de $\rhd_2^\rho$, $T_{j_1}$ est de la forme $(\vec{z} : \vec{V})C'_1(\vec{w}^1)$. Donc, $|\vec{z}| = 0$, $C'_1 = C_1$ et $\vec{w}^1\gamma_0\rho = \vec{v}^1\gamma_1\rho$. Ainsi, on peut conclure par récurrence sur $t_{j_1} : T_{j_1}\gamma\ (\rhd_2^\rho)^+\ x : V$.

Comme nous sommes dans une structure inductive admissible **(A2)**, on en déduit que $C \geq_{\mathcal{F}} C_1 \geq_{\mathcal{F}} \dots \geq_{\mathcal{F}} C_n \geq D$. Et comme $D =_{\mathcal{F}} C$ et $>_{\mathcal{F}}$ est bien fondé, nous avons $C =_{\mathcal{F}} C_1 =_{\mathcal{F}} \dots =_{\mathcal{F}} D$.

Posons $w_{n+1} = t|_p\vec{u}$, $W_{n+1} = D(\vec{w})\theta$ et, pour tout $i \leq n$, $w_i = t|_{j_1\dots j_i}$ et $W_i = C_i(\vec{v}^i)\gamma_i$. Montrons par récurrence sur $n$ que, pour tout $i \leq n+1$, $\Delta \vdash w_i\sigma \in [\![\Gamma \vdash W_i]\!]_{\Delta,\sigma,\xi}$ et que, pour tout $i \leq n$, $o(\Delta \vdash w_i\sigma) > o(\Delta \vdash w_{i+1}\sigma)$.

- Cas $n > 0$. $c(\vec{t}) : C(\vec{v})\gamma \rhd_2^\rho t_{j_1} : T_{j_1}\gamma\ (\rhd_2^\rho)^+\ x : V$. Par hypothèse, nous avons $\Gamma \vdash c(\vec{t})\rho : C(\vec{v})\gamma\rho$ et $\Delta \vdash c(\vec{t})\sigma \in [\![\Gamma \vdash C(\vec{v})\gamma\rho]\!]_{\Delta,\sigma,\xi}$. Par correction de l'accessibilité, $\Gamma \vdash t_{j_1}\rho : T_{j_1}\gamma\rho$ et $\Delta \vdash t_{j_1}\sigma \in [\![\Gamma \vdash T_{j_1}\gamma\rho]\!]_{\Delta,\sigma,\xi}$. Comme $T_{j_1}\gamma\rho = C_1(\vec{v}^1)\gamma_1\rho$ et $C_1 =_{\mathcal{F}} C$, $o(\Delta \vdash t\sigma) > o(\Delta \vdash t_{j_1}\sigma)$. Ainsi, on peut conclure par hypothèse de récurrence sur $t_{j_1} : T_{j_1}\gamma$.
- Cas $n = 0$. $c(\vec{t}) : C(\vec{v})\gamma \rhd_2^\rho x : V$. Comme dans le cas $n > 0$, nous avons $\Delta \vdash x\sigma \in [\![\Gamma \vdash V\rho]\!]_{\Delta,\sigma,\xi}$. Nous avons $V\rho = (\vec{y} : \vec{U})D(\vec{w})$. Soit $m = |\vec{u}|$ et $\theta_k = \{y_1 \mapsto u_1, \dots, y_k \mapsto u_k\}$. Montrons par récurrence sur $k \leq m$ que $\Delta \vdash x\sigma\, u_1\sigma \dots u_k\sigma \in [\![\Gamma \vdash (y_{k+1} : U_{k+1}\theta_k) \dots (y_m : U_m\theta_k)D(\vec{w})\theta_k]\!]_{\Delta,\sigma,\xi}$ et donc que $o(\Delta \vdash t\sigma) > o(\Delta \vdash t'\sigma)$.

  Si $k = 0$, c'est immédiat. Supposons donc $k > 0$. Par hypothèse de récurrence, $\Delta \vdash x\sigma\, u_1\sigma \dots\ u_{k-1}\sigma \in [\![\Gamma \vdash (y_k : U_k\theta_{k-1}) \dots (y_m : U_m\theta_{k-1})D(\vec{w})\theta_{k-1}]\!]_{\Delta,\sigma,\xi}$. Comme

$\Delta \vdash u_k\sigma \in [\![\Gamma \vdash U_k\theta_{k-1}]\!]_{\Delta,\sigma,\xi}$, nous avons $\Delta \vdash x\sigma\, u_1\sigma \ldots u_k\sigma \in [\![\Gamma, y_k : U_k\theta_{k-1} \vdash (y_{k+1}:U_{k+1}\theta_{k-1}) \ldots (y_m:U_m\theta_{k-1})D(\vec{w})\theta_{k-1}]\!]_{\Delta,\sigma',\xi'}$ où $\sigma' = \sigma \cup \{y_k \mapsto u_k\sigma\}$, $\xi' = \xi$ si $y_k \in \mathcal{X}^\star$ et $\xi' = \xi \cup \{y_k \mapsto [\![\Gamma \vdash u_k]\!]_{\Delta,\sigma,\xi}\}$ si $y_k \in \mathcal{X}^\Box$. Ainsi, par substitution de candidats, $[\![\Gamma, y_k : U_k\theta_{k-1} \vdash (y_{k+1} : U_{k+1}\theta_{k-1}) \ldots (y_m : U_m\theta_{k-1})D(\vec{w})\theta_{k-1}]\!]_{\Delta,\sigma',\xi'} = [\![\Gamma \vdash (y_{k+1}:U_{k+1}\theta_k) \ldots (y_m:U_m\theta_k)D(\vec{w})\theta_k]\!]_{\Delta,\sigma,\xi}$. ■

## 8.6 Interprétation des symboles de prédicat définis

Dans ce sous-chapitre, nous définissons l'interprétation $J$ pour les symboles de prédicat définis par récurrence sur $\succ$.

Soit un symbole de prédicat défini $F$ et supposons que nous ayons déjà défini une interprétation $K$ pour tous les symboles inférieurs à $F$. Soit $[F]$ l'ensemble des symboles équivalents à $F$. Nous définissons l'interprétation des symboles de $[F]$ selon que $[F]$ est primitif, positif ou récursif.

Pour alléger les notations, nous désignerons $[\![\Gamma \vdash T]\!]^{I\cup K\cup J}_{\Delta,\theta,\xi}$ par $[\![\Gamma \vdash T]\!]^{J}_{\Delta,\theta,\xi}$. Et pour les arguments des interprétations, nous utiliserons les notations données au début du Sous-chapitre 8.4.

### 8.6.1 Systèmes primitifs

**Définition 136** Pour chaque $G \in [F]$, nous prenons $J_{\Delta\vdash G} = \top_{\Delta\vdash G}$.

### 8.6.2 Systèmes positifs, petits et simples

Soit $\mathcal{J}$ l'ensemble des interprétations pour $[F]$ et $\leq$ la relation sur $\mathcal{J}$ telle que $J \leq J'$ si, pour tout $G \in [F]$ et $\Delta \in \mathbb{E}$, $J_{\Delta\vdash G} \leq_{\Delta\vdash G} J'_{\Delta\vdash G}$. Comme $(\mathcal{R}_{\Delta\vdash G}, \leq_{\Delta\vdash G})$ est un treillis complet, il est facile de voir que $(\mathcal{J}, \leq)$ est également un treillis complet.

**Définition 137** Soit $\psi$ la fonction qui, à $J \in \mathcal{J}$, associe l'interprétation $\psi^J$ telle que :

$$\psi^J_{\Delta\vdash G}(\vec{a}) = \begin{cases} [\![\Gamma \vdash r]\!]^J_{\Delta_n,\sigma,\xi} & \text{si } \vec{t} \in \mathcal{WN} \cap \mathcal{CR},\ \vec{t}{\downarrow} = \vec{l}\sigma \text{ et } (G(\vec{l}) \to r, \Gamma, \rho) \in \mathcal{R} \\ \top_{\Delta_n\vdash G(\vec{t})} & \text{sinon} \end{cases}$$

où, pour chaque $X \in \mathrm{FV}^\Box(r)$, $X\xi = S_{\kappa_X}|_{\Delta_n}$. Ci-après, nous montrons que $\psi$ est monotone. Donc, nous pouvons prendre $J_{\Delta\vdash G} = \mathrm{lfp}(\psi)_{\Delta\vdash G}$.

**Lemme 138** $\psi^J$ est bien défini et est bien une interprétation.

**Preuve.** Par simplicité, au plus une seule régle peut s'appliquer au sommet de $G(\vec{t}\downarrow)$. L'existence de $\kappa_X$ est l'hypothèse de petitesse. Maintenant, par **(S4)**, si $\vec{t} = \vec{l}\sigma$ alors $\sigma : \Gamma \to \Delta_n$. Par ailleurs, $\xi$ est compatible avec $(\sigma, \Gamma, \Delta_n)$ puisque, pour chaque $X \in \mathrm{FV}^\Box(r)$, $X\xi = S_{\kappa_X}|_{\Delta_n} \in \mathcal{R}_{\Delta_n\vdash X\sigma}$ car $S_{\kappa_X} \in \mathcal{R}_{\Delta_{\kappa_X}\vdash t_{\kappa_X}}$ et, par petitesse, $t_{\kappa_X} = l_{\kappa_X}\sigma = X\sigma$.

Montrons maintenant que $\psi^J$ est une interprétation pour $[F]$. D'après le Lemme 121 (b), $[\![\Gamma \vdash r]\!]^J_{\Delta_n,\sigma,\xi} \in \mathcal{R}_{\Delta_n\vdash r\sigma} = \mathcal{R}_{\Delta_n\vdash F(\vec{t})}$. Par ailleurs, $\top_{\Delta_n\vdash F(\vec{t})} \in \mathcal{R}_{\Delta_n\vdash F(\vec{t})}$. Il ne nous reste plus qu'à vérifier les propriétés (P1) à (P3).

**(P1)** Supposons que $\vec{t} \to \vec{t'}$. Par **(A1)**, $\to$ est confluente, donc $\{\vec{t}\} \subseteq \mathcal{WN}$ si et seulement si $\{\vec{t'}\} \subseteq \mathcal{WN}$, et si $\{\vec{t}\} \subseteq \mathcal{WN}$ alors $\vec{t}\downarrow = \vec{t'}\downarrow$. Donc, $\psi^J_{\Delta\vdash G}(\vec{a}) = \psi^J_{\Delta\vdash G}(\vec{a'})$.

**(P2)** Supposons que $\Delta_n \subseteq \Delta' \in \mathbb{E}$. D'après le Lemme 121 (d), $[\![\Gamma\vdash r]\!]^J_{\Delta_n,\sigma,\xi}|_{\Delta'} = [\![\Gamma\vdash r]\!]^I_{\Delta',\sigma,\xi|_{\Delta'}}$. Par ailleurs, $\top_{\Delta_n\vdash F(\vec{t})}|_{\Delta'} = \top_{\Delta'\vdash F(\vec{t})}$. Donc, $\psi^J_{\Delta\vdash G}(\vec{a})|_{\Delta'} = \psi^J_{\Delta'\vdash G}(\vec{a})$. Maintenant, supposons que $\Delta_k \subseteq \Delta'_k \in \mathbb{E}$. Prenons $a'_k = \Delta'_k \vdash t_k$ si $x_k \in \mathcal{X}^\star$, et $a'_k = (\Delta'_k \vdash t_k, S_k|_{\Delta'_k})$ si $x_k \in \mathcal{X}^\square$. Alors, $\psi^J_{\Delta\vdash G}(a_1,\ldots,a_{k-1},a_k)|_{\Delta'_k}$ et $\psi^J_{\Delta\vdash G}(a_1,\ldots,a_{k-1},a'_k)$ sont égales puisqu'elles ont le même domaine et sont égales sur ce domaine.

**(P3)** $\psi_{\Delta\vdash G}|_{\Delta'}$ et $\psi_{\Delta'\vdash G}$ sont égales puisqu'elles ont le même domaine et sont égales sur ce domaine. ■

**Lemme 139** $\psi$ est monotone.

**Preuve.** Comme pour le Lemme 129. ■

### 8.6.3 Systèmes récursifs, petits et simples

Soit $G \in [F]$, $\Delta \in \mathbb{E}$. À une séquence d'arguments $\vec{a}$ pour $J_{\Delta\vdash G}$, nous associons la substitution $\theta = \{\vec{x} \mapsto \vec{t}\} : \Gamma_G \to \Delta_n$ et l'assignement $\xi = \{\vec{x} \mapsto \vec{S}|_{\Delta_n}\}$ compatible avec $(\theta, \Gamma_G, \Delta_n)$. Soit $\mathcal{D}$ l'ensemble des séquences $\vec{a}$ telles que $(\theta, \Gamma_G, \Delta_n)$ soit valide par rapport à $\xi$. Nous munissons $\mathcal{D}$ de l'ordre bien fondé suivant : $\vec{a} \sqsupset \vec{a'}$ si $(G, \Delta_n, \theta, \xi) \sqsupset (G, \Delta'_n, \theta', \xi')$.

**Définition 140** Nous définissons alors $J_{\Delta\vdash G}(\vec{a})$ par récurrence sur $\sqsupset$ :

$$J_{\Delta\vdash G}(\vec{a}) = \begin{cases} [\![\Gamma\vdash r]\!]^J_{\Delta_n,\sigma,\xi} & \text{si } \begin{array}{c}\{\vec{t}\} \subseteq \mathcal{WN}\cap\mathcal{CR},\ \vec{t}\downarrow = \vec{l}\sigma,\ \vec{a}\downarrow \in \mathcal{D} \\ \text{et } (G(\vec{l}) \to r, \Gamma, \rho) \in \mathcal{R}\end{array} \\ \top_{\Delta_n\vdash G(\vec{t})} & \text{sinon} \end{cases}$$

où, pour chaque $X \in \mathrm{FV}^\square(r)$, $X\xi = S_{\kappa_X}|_{\Delta_n}$.

**Lemme 141** $J$ est bien défini et est bien une interprétation.

**Preuve.** Par simplicité, au plus une seule régle peut s'appliquer au sommet de $G(\vec{t}\downarrow)$. L'existence de $\kappa_X$ est l'hypothèse de petitesse. Maintenant, par **(S4)**, si $\vec{t} = \vec{l}\sigma$ alors $\sigma : \Gamma \to \Delta_n$. Par ailleurs, $\xi$ est compatible avec $(\sigma, \Gamma, \Delta_n)$ puisque, pour chaque $X \in \mathrm{FV}^\square(r)$, $X\xi = S_{\kappa_X}|_{\Delta_n} \in \mathcal{R}_{\Delta_n\vdash X\sigma}$ car $S_{\kappa_X} \in \mathcal{R}_{\Delta_{\kappa_X}\vdash t_{\kappa_X}}$ et, par petitesse, $t_{\kappa_X} = l_{\kappa_X}\sigma = X\sigma$.

La bonne fondation de la définition de $J$ découle du Lemme 147 et du Théorème 146. Dans le Lemme 147 de réductibilité des symboles d'ordre supérieur, nous montrons qu'à partir d'une séquence $\vec{a} \in \mathcal{D}$ il est possible d'appliquer le Théorème 146 de correction de la clôture calculable. Et dans ce théorème, nous montrons que, dans un appel récursif $G'(\vec{a'})$ (cas (symb$^=$)), nous avons $\vec{a} \sqsupset \vec{a'}$. Comme $\sqsupset$ est compatible avec la réduction, $\vec{a} \sqsupset \vec{a'}\downarrow$.

Enfin, pour s'assurer que $J$ est bien une interprétation, on procède de la même manière que dans le cas d'un système simple et positif. ■

## 8.7 Correction des conditions de normalisation forte

**Définition 142 (Cap et aliens)** Soit $\zeta$ une injection entre les classes de termes modulo $\leftrightarrow^*$ et $\mathcal{X}$. Le *cap* d'un terme $t$ par rapport à un ensemble $\mathcal{G}$ de symboles est le plus grand terme $cap_{\mathcal{G}}(t) = t[x_1]_{p_1} \dots [x_n]_{p_n}$ tel que, pour tout $i$ :

- $t|_{p_i}$ n'est pas de la forme $f(\vec{t})$ avec $f \in \mathcal{G}$,
- $x_i = \zeta(t|_{p_i})$.

Les $t|_{p_i}$ sont les *aliens* de $t$. Nous noterons par $aliens_{\mathcal{G}}(t)$ leur multi-ensemble.

**Lemme 143 (Pré-réductibilité des symboles de premier ordre)** Soit $f \in \mathcal{F}_1$ et $\vec{t}$ des termes tels que $f(\vec{t})$ soit typable. Si les $t_i$ sont fortement normalisables alors $f(\vec{t})$ est fortement normalisable.

**Preuve.** Montrons que tout réduit immédiat $t'$ de $t = f(\vec{t})$ est fortement normalisable. Ci-après, par *cap*, on désigne le cap par rapport à $\mathcal{F}_1$.

**Cas $\mathcal{R}_\omega \neq \emptyset$.** Par récurrence sur $(aliens(t), cap(t))_{\text{lex}}$ avec $((\rightarrow \cup \rhd)_{\text{mul}}, \rightarrow_{\mathcal{R}_1})_{\text{lex}}$ comme ordre bien fondé (les aliens sont fortement normalisables et, par **(f)**, $\rightarrow_{\mathcal{R}_1}$ est fortement normalisant sur $\mathbb{T}(\mathcal{F}_1, \mathcal{X})$).

Si la réduction a lieu dans $cap(t)$ alors c'est une $\mathcal{R}_1$-réduction. Par **(c)**, aucun symbole de $\mathcal{F}_\omega$ n'apparaît dans les règles de $\mathcal{R}_1$. Donc, $cap(t) \rightarrow_{\mathcal{R}_1} cap(t')$. Par **(d)**, les membres droits des règles de $\mathcal{R}_1$ sont algébriques et, par **(e)**, les règles de $\mathcal{R}_1$ sont non-dupliquantes. Donc, $aliens(t) \rhd_{\text{mul}} aliens(t')$ et nous pouvons conclure par hypothèse de récurrence.

Si la réduction a lieu dans un alien alors $aliens(t)$ $(\rightarrow \cup \rhd)_{\text{mul}}$ $aliens(t')$ et nous pouvons conclure par hypothèse de récurrence.

**Cas $\mathcal{R}_\omega = \emptyset$.** Puisque les $t_i$ sont fortement normalisables et qu'aucune $\beta$-réduction ne peut avoir lieu au sommet de $t$, $t$ admet une forme $\beta$-normale. Notons $\beta cap(t)$ le cap de sa forme $\beta$-normale. Montrons que tout réduit immédiat $t'$ de $t$ est fortement normalisable, par récurrence sur $(\beta cap(t), aliens(t))_{\text{lex}}$ avec $(\rightarrow_{\mathcal{R}_1}, (\rightarrow \cup \rhd)_{\text{mul}})_{\text{lex}}$ comme ordre bien fondé (de même, les aliens sont fortement normalisables et, par **(f)**, $\rightarrow_{\mathcal{R}_1}$ est fortement normalisant sur $\mathbb{T}(\mathcal{F}_1, \mathcal{X})$).

Si la réduction a lieu dans $cap(t)$ alors c'est une $\mathcal{R}_1$-réduction. Par **(d)**, les membres droits des règles de $\mathcal{R}_1$ sont algébriques. Donc, $t'$ admet une forme $\beta$-normale et $\beta cap(t) \rightarrow_{\mathcal{R}_1} \beta cap(t')$. On peut donc conclure par hypothèse de récurrence.

Si la réduction est une $\beta$-réduction dans un alien alors $\beta cap(t) = \beta cap(t')$ et $aliens(t)$ $(\rightarrow \cup \rhd)_{\text{mul}}$ $aliens(t')$. Nous pouvons donc conclure par hypothèse de récurrence.

Reste le cas où la réduction est une $\mathcal{R}_1$-réduction ayant lieu dans un alien $u$. Alors, $aliens(t) \rightarrow_{\text{mul}} aliens(t')$ et $\beta cap(t) \rightarrow^*_{\mathcal{R}_1} \beta cap(t')$. Nous pouvons donc conclure par hypothèse de récurrence. Pour voir que $\beta cap(t) \rightarrow^*_{\mathcal{R}_1} \beta cap(t')$, il suffit de remarquer le fait suivant : si on $\beta$-normalise $u$, tous les résidus du $\mathcal{R}_1$-radical restent réductibles car, par **(c)**, aucun symbole de $\mathcal{F}_\omega$ n'apparaît dans $\mathcal{R}_1$. ■

**Théorème 144 (Normalisation forte de $\rightarrow_{\mathcal{R}}$)** La relation $\rightarrow_{\mathcal{R}_1} \cup \rightarrow_{\mathcal{R}_\omega}$ est fortement normalisante sur les termes typables.

**Preuve.** Par récurrence sur la structure des termes. Le seul cas un peu ennuyeux est celui d'un symbole. Dans le cas d'un symbole de premier ordre, on utilise le lemme de pré-réductibilité des symboles de premier ordre. Dans le cas d'un symbole d'ordre supérieur, il faut montrer que si $f \in \mathcal{F}_\omega$, $\vec{t} \in \mathcal{SN}$ et $f(\vec{t}) \in \mathbb{T}$ alors $t = f(\vec{t}) \in \mathcal{SN}$ où $\mathcal{SN}$ désigne ici la normalisation forte par rapport à $\rightarrow_{\mathcal{R}} = \rightarrow_{\mathcal{R}_1} \cup \rightarrow_{\mathcal{R}_\omega}$.

Pour cela, montrons que tout réduit immédiat $t'$ de $t$ est fortement normalisable par récurrence sur $(f, \varpi(\vec{t}), \vec{t}, \vec{t})$ avec $(>_{\mathcal{F}}, (>_{\mathbb{N}})_{stat_f}, (\rhd \cup \rightarrow_{\mathcal{R}})_{stat_f}, (\rightarrow_{\mathcal{R}})_{\text{lex}})_{\text{lex}}$ comme ordre bien fondé où $\varpi(t) = 0$ si $t$ n'est pas de la forme $g(\vec{u})$ et $\varpi(t) = 1$ sinon. Supposons que $t' = f(\vec{t'})$ avec $t_i \rightarrow_{\mathcal{R}} t'_i$ et, pour tout $j \neq i$, $t_j = t'_j$. Alors $\vec{t}\ (\rightarrow_{\mathcal{R}})_{\text{lex}}\ \vec{t'}$ et $\varpi(t_i) \geq \varpi(t'_i)$ car si $t_i$ n'est pas de la forme $g(\vec{u})$ alors $t'_i$ ne l'est pas non plus.

Supposons maintenant qu'il existe $f(\vec{l}) \rightarrow r \in \mathcal{R}_\omega$ telle que $\vec{t} = \vec{l}\sigma$ et $t' = r\sigma$. Par **(a)**, $r$ appartient à la clôture calculable de $l$. Il est alors aisé de montrer que $r\sigma$ est fortement normalisable en raisonnant par récurrence sur la structure de $r$. À nouveau, le cas seul cas un peu ennuyeux est celui d'un symbole. Mais, soit le symbole est plus petit que $f$, soit il est équivalent à $f$ et ses arguments $\vec{u}$ sont plus petits que $\vec{l}$. Si $l_i >_1 u_j$ alors $l_i \rhd u_j$ et $\text{FV}(u_j) \subseteq \text{FV}(l_i)$. Donc $l_i\sigma \rhd u_j\sigma$ et $\varpi(l_i\sigma) = 1 \geq \varpi(u_j\sigma)$. Si maintenant $l_i >_2 u_j$ alors $u_j$ est de la forme $x\vec{v}$ et $\varpi(l_i\sigma) = 1 > \varpi(u_j\sigma) = 0$. ■

**Lemme 145 (Réductibilité des symboles de premier ordre)** Soit $f \in \mathcal{F}_1$, $\tau_f = (\vec{x} : \vec{T})U$, $\Delta \in \mathbb{E}$, $\theta : \Gamma_f \rightarrow \Delta$ et $\xi$ compatible avec $(\theta, \Gamma_f, \Delta)$. Si $(\theta, \Gamma_f, \Delta)$ est valide par rapport à $\xi$ alors $\Delta \vdash f(\vec{x}\theta) \in [\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi}$.

**Preuve.** Posons $t_i = x_i\theta$ et $t = f(\vec{t})$. Si $f$ n'est pas un constructeur alors $t$ est neutre et il suffit de prouver que, pour tout réduit immédiat $t'$ de $t$, $\Delta \vdash t' \in [\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi}$.

Si $f$ est un constructeur alors $U = C(\vec{v})$ et $[\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi} = I_{\Delta\vdash C}(\vec{a})$ où $a_i = \Delta \vdash v_i\theta$ si $x_i \in \mathcal{X}^\star$ et $a_i = (\Delta \vdash v_i\theta, S'_i)$ avec $S'_i = [\![\Gamma_f \vdash v_i]\!]_{\Delta,\theta,\xi}$ si $x_i \in \mathcal{X}^\square$. Soit $j \in \text{Acc}(f)$. Puisque $\theta$ est valide par rapport à $\xi$, $\Delta \vdash t_j \in [\![\Gamma_f \vdash T_j]\!]_{\Delta,\theta,\xi}$. Donc, dans ce cas également, il suffit de prouver que, pour tout réduit immédiat $t'$ de $t$, $\Delta \vdash t' \in [\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi}$.

Par ailleurs, pour les symboles de premier ordre, $U = \star$ ou $U = C(\vec{v})$ avec $C$ un symbole de prédicat primitif. Donc $[\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi} = \overline{\mathbb{SN}}_{\Delta\vdash U\theta}$ et il suffit de prouver que tout réduit immédiat $t'$ de $t$ est fortement normalisable. C'est le lemme de pré-réductibilité précédent. ■

**Théorème 146 (Correction de la clôture calculable)** Soit $f \in \mathcal{F}$, $\tau_f = (\vec{x} : \vec{T})$ $U$, $R = (f(\vec{l}) \rightarrow r, \Gamma_0, \rho) \in \mathcal{R}$, $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$, $\Delta \in \mathbb{E}$, $\sigma : \Gamma_0 \rightarrow \Delta$ et $\xi$ compatible avec $(\sigma, \Gamma_0, \Delta)$ tels que :

- $(\sigma, \Gamma_0, \Delta)$ est valide par rapport à $\xi$ ;
- pour tout $i$, $\Delta \vdash l_i\sigma \in [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta,\sigma,\xi}$ ;
- pour tout $g \leq_{\mathcal{F}} f$, si $\tau_g = (\vec{y} : \vec{U})V$, $\Delta' \in \mathbb{E}$, $\theta' : \Gamma \rightarrow \Delta$, $\xi'$ est compatible avec $(\theta', \Gamma_g, \Delta')$ et $(\theta', \Gamma_g, \Delta')$ est valide par rapport à $\xi'$, alors $\Delta' \vdash g(\vec{y})\theta' \in [\![\Gamma_g \vdash V]\!]_{\Delta',\theta',\xi|_{\Delta'}}$ dès lors que $(f, \Delta, \gamma_0\sigma, \xi) \sqsupset (g, \Delta', \theta', \xi')$.

Si $\Gamma_0, \Gamma \vdash_{\mathrm{c}} t : T$, $\Delta \subseteq \Delta' \in \mathbb{E}$, $\sigma' : \Gamma \to \Delta'$, $\xi'$ est compatible avec $(\sigma', \Gamma, \Delta')$ et $(\sigma', \Gamma, \Delta')$ est valide par rapport à $\xi'$, alors $\Delta' \vdash t\sigma\sigma' \in [\![\Gamma_0, \Gamma \vdash T]\!]_{\Delta', \sigma\sigma', \xi''}$ où $\xi'' = \xi|_{\Delta'} \cup \xi'$.

**Preuve.** Par récurrence sur $\Gamma' \vdash_{\mathrm{c}} t : T$ ($\Gamma' = \Gamma_0, \Gamma$), nous montrons :

(a) $\Delta' \vdash t\sigma\sigma' \in [\![\Gamma' \vdash T]\!]_{\Delta', \sigma\sigma', \xi''}$,
(b) si $t \notin \mathcal{O}$ et $t \to t'$ alors $[\![\Gamma' \vdash t]\!]_{\Delta', \sigma\sigma', \xi''} = [\![\Gamma' \vdash t']\!]_{\Delta', \sigma\sigma', \xi''}$.

**(ax)** $\Gamma_0 \vdash_{\mathrm{c}} \star : \Box$

(a) $\Delta' \vdash \star \in [\![\Gamma_0 \vdash \Box]\!]_{\Delta', \sigma, \xi|_{\Delta'}} = \overline{\mathbb{SN}}_{\Delta' \vdash \Box}$.
(b) $\star$ est irréductible.

**(symb$^<$)** $\dfrac{\Gamma_0 \vdash_{\mathrm{c}} \tau_g : s \quad \Gamma' \vdash_{\mathrm{c}} u_1 : U_1\gamma \ldots \Gamma' \vdash_{\mathrm{c}} u_n : U_n\gamma}{\Gamma' \vdash_{\mathrm{c}} g(\vec{u}) : V\gamma}$

(a) Par hypothèse de récurrence, $\Delta' \vdash u_i\sigma\sigma' \in [\![\Gamma' \vdash U_i\gamma]\!]_{\Delta', \sigma\sigma', \xi''}$. Par substitution de candidats, il existe $\xi'''$ tel que $[\![\Gamma' \vdash U_i\gamma]\!]_{\Delta, \theta, \xi} = [\![\Gamma_g \vdash U_i]\!]_{\Delta', \gamma\sigma\sigma', \xi'''}$ et $[\![\Gamma' \vdash V\gamma]\!]_{\Delta, \theta, \xi} = [\![\Gamma_g \vdash V]\!]_{\Delta', \gamma\sigma\sigma', \xi'''}$. Ainsi, $(\gamma\sigma\sigma', \Gamma_g, \Delta')$ est valide par rapport à $\xi'''$ et, par hypothèse sur $g$, $\Delta' \vdash g(\vec{y})\gamma\sigma\sigma' \in [\![\Gamma_g \vdash V]\!]_{\Delta', \gamma\sigma\sigma', \xi'''}$.

(b) Nous avons $[\![\Gamma' \vdash g(\vec{u})]\!]_{\Delta', \sigma\sigma', \xi''} = I_{\Delta' \vdash g}(\vec{a})$ avec $a_i = \Delta' \vdash u_i\sigma\sigma'$ si $y_i \in \mathcal{X}^\star$, et $a_i = (\Delta' \vdash u_i\sigma\sigma', S_i)$ et $S_i = [\![\Gamma' \vdash u_i]\!]_{\Delta', \sigma\sigma', \xi''}$ si $y_i \in \mathcal{X}^\Box$. Il y a deux cas :

– $u_i \to u_i'$. On conclut par **(P1)**.
– $\vec{u} = \vec{l}'\sigma''$ et $(g(\vec{l}') \to r', \Gamma'', \rho') \in \mathcal{R}$. Soit $\sigma''' = \sigma''\sigma\sigma'$. Par **(A3)**, il y a deux sous-cas :

• $g$ **appartient à un système primitif.** Alors $I_{\Delta' \vdash g} = \top_{\Delta' \vdash g}$ et $[\![\Gamma' \vdash g(\vec{u})]\!]_{\Delta', \sigma\sigma', \xi''} = \top_{\Delta \vdash g(\vec{l}'\sigma''')} = \top_{\Delta \vdash r'\sigma'''}$. Par substitution de candidats, il existe $\xi'''$ tel que $[\![\Gamma' \vdash r'\sigma'']\!]_{\Delta', \sigma\sigma', \xi''} = [\![\Gamma'' \vdash r']\!]_{\Delta', \sigma''', \xi'''}$. Par ailleurs, $r'$ est de la forme $[\vec{x} : \vec{T}]g'(\vec{u})\vec{v}$ avec $g' \simeq g$ ou $g'$ un symbole de prédicat constant primitif. Si $g' \simeq g$ alors $I_{\Delta' \vdash g'} = \top_{\Delta' \vdash g'}$, et si $g'$ est un symbole de prédicat constant primitif alors, d'après le Lemme 130, $I_{\Delta' \vdash g'} = \top_{\Delta' \vdash g'}$. Donc, $[\![\Gamma'' \vdash r']\!]_{\Delta', \sigma''', \xi'''} = \top_{\Delta' \vdash r'\sigma'''}$ et $[\![\Gamma' \vdash g(\vec{l}'\sigma'')]\!]_{\Delta', \sigma\sigma', \xi''} = [\![\Gamma' \vdash r'\sigma'']\!]_{\Delta', \sigma\sigma', \xi''}$.

• $g$ **appartient à un système positif ou récursif, petit et simple.** Comme $\Delta' \vdash u_i\sigma\sigma' \in [\![\Gamma' \vdash U_i\gamma]\!]_{\Delta', \sigma\sigma', \xi''} \subseteq \mathcal{SN}$, par **(A1)**, $u_i\sigma\sigma'$ admet une forme normale. Par simplicité, les symboles de $\vec{l}'$ sont constants. Donc $u_i\sigma\sigma'$ est de la forme $\vec{l}'\sigma'''$ avec $\sigma''\sigma\sigma' \to^* \sigma'''$. Par simplicité encore, au plus une règle peut s'appliquer au sommet d'un terme. Donc $I_{\Delta' \vdash g}(\vec{a}) = [\![\Gamma'' \vdash r']\!]_{\Delta', \sigma''', \xi'''}$ avec, pour tout $X \in \mathrm{FV}^\Box(r')$, $X\xi''' = S_{\kappa_X} = [\![\Gamma' \vdash l_{\kappa_X}\sigma'']\!]_{\Delta', \sigma\sigma', \xi''}$. Par petitesse, $l_{\kappa_X} = X$. Donc $X\xi''' = [\![\Gamma' \vdash X\sigma'']\!]_{\Delta', \sigma\sigma', \xi''}$ et, par substitution de candidats, $[\![\Gamma'' \vdash r']\!]_{\Delta', \sigma''', \xi'''} = [\![\Gamma' \vdash r'\sigma'']\!]_{\Delta', \sigma\sigma', \xi''}$.

**(symb$^=$)** $\dfrac{\Gamma_0 \vdash_{\mathrm{c}} \tau_g : s \quad \Gamma' \vdash_{\mathrm{c}} u_1 : U_1\gamma \ldots \Gamma' \vdash_{\mathrm{c}} u_n : U_n\gamma}{\Gamma' \vdash_{\mathrm{c}} g(\vec{u}) : V\gamma} \quad (\vec{l} : \vec{T}\gamma_0 > \vec{u} : \vec{U}\gamma)$

(a) Comme dans le cas précédent, nous avons $(\gamma\sigma\sigma', \Gamma_g, \Delta')$ valide par rapport à $\xi'''$. Montrons que $(f, \Delta, \gamma_0\sigma, \xi) \sqsupset_f (g, \Delta', \gamma\sigma\sigma', \xi''')$. Pour cela, il suffit de montrer que, si $l_i : T_i\gamma_0 >_1 u_j : U_j\gamma$ alors $l_i\sigma \rhd u_j\sigma\sigma'$, et si $l_i : T_i\gamma_0 >_2 u_j : U_j\gamma$ alors $o(\Delta \vdash l_i\sigma) > o(\Delta' \vdash u_j\sigma\sigma')$.

Supposons que $l_i : T_i\gamma_0 >_1 u_j : U_j\gamma$. Alors, $l_i \rhd u_j$ et $\mathrm{FV}(u_j) \subseteq \mathrm{FV}(l_i)$. Donc, $l_i\sigma\sigma' = l_i\sigma \rhd u_j\sigma\sigma'$.

Supposons maintenant que $l_i : T_i\gamma_0 >_2 u_j : U_j\gamma$. Par définition de $>_2$, nous avons $u_j = x\vec{v}$, $x \in \mathrm{dom}(\Gamma_0)$ et $x\Gamma_0 = (\vec{z} : \vec{V})W$. Soit $\theta = \{\vec{z} \mapsto \vec{v}\}$. Par correction de l'ordre sur les arguments, il suffit de vérifier que :

(1) $\Gamma' \vdash l_i\rho : T_i\gamma_0\rho$,

(2) $\Delta' \vdash l_i\sigma\sigma' \in [\![\Gamma' \vdash T_i\gamma_0\rho]\!]_{\Delta',\sigma\sigma',\xi''}$,

(3) pour tout $k$, $\Delta' \vdash v_k\sigma\sigma' \in [\![\Gamma' \vdash V_k\theta]\!]_{\Delta',\sigma\sigma',\xi''}$.

En effet, de cela on déduit que $o(\Delta' \vdash l_i\sigma\sigma') > o(\Delta' \vdash u_j\sigma\sigma')$. Or $l_i\sigma = l_i\sigma\sigma'$ et $o(\Delta \vdash l_i\sigma) \geq o(\Delta' \vdash l_i\sigma)$.

(1) Par définition d'une règle bien formée, $\Gamma_0 \vdash f(\vec{l})\rho : U\gamma_0\rho$. Donc, par inversion, $\Gamma_0 \vdash l_i\rho : T_i\gamma_0\rho$. Comme $\Gamma_0 \subseteq \Gamma' \in \mathbb{E}$, par affaiblissement, $\Gamma' \vdash l_i\rho : T_i\gamma_0\rho$.

(2) Par hypothèse, $\Delta \vdash l_i\sigma \in [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta,\sigma,\xi}$. Or $l_i\sigma\sigma' = l_i\sigma$ et $[\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta',\sigma,\xi|_{\Delta'}} = [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta,\sigma,\xi}|_{\Delta'}$.

(3) Par hypothèse, on a $\Gamma' \vdash_{\mathrm{c}} x\vec{v} : U_j\gamma$. Soit $q = |\vec{v}|$. Par inversion, il existe $\vec{V'}$ et $\vec{W'}$ tels que $\Gamma' \vdash_{\mathrm{c}} xv_1 \dots v_{q-1} : (x_q : V'_q)W'_q$, $\Gamma' \vdash_{\mathrm{c}} v_q : V'_q$ et $W'_q\{x_q \mapsto v_q\}$ $\mathbb{C}^*_{\Gamma'}$ $U_j\gamma$, et pour tout $k < q-1$, $\Gamma' \vdash_{\mathrm{c}} xv_1 \dots v_k : (x_{k+1} : V'_{k+1})W'_{k+1}$, $\Gamma' \vdash_{\mathrm{c}} v_{k+1} : V'_{k+1}$ et $W'_{k+1}\{x_{k+1} \mapsto v_{k+1}\}$ $\mathbb{C}^*_{\Gamma'}$ $(x_{k+2} : V'_{k+2})W'_{k+2}$. Ainsi, par hypothèse de récurrence, pour tout $k$, $\Delta' \vdash v_k\sigma\sigma' \in [\![\Gamma' \vdash V'_k]\!]_{\Delta',\sigma\sigma',\xi''}$. Montrons que $V'_k$ $\mathbb{C}^*_{\Gamma'}$ $V_k\theta$. Soit $W_k = (z_{k+1} : V_{k+1}) \dots (z_q : V_q)W$ et $\theta_k = \{z_1 \mapsto v_1, \dots, z_{k-1} \mapsto v_{k-1}\}$. Montrons par récurrence sur $k$ que $V'_k$ $\mathbb{C}^*_{\Gamma'}$ $V_k\theta_{k-1}$ et $W'_k\{x_k \mapsto v_k\}$ $\mathbb{C}^*_{\Gamma'}$ $W_k\theta_k$.

- Cas $k = 1$. Comme $(\vec{z} : \vec{V})W$ $\mathbb{C}^*_{\Gamma'}$ $(x_1 : V'_1)W'_1$, par compatibilité avec le produit et $\alpha$-équivalence (on prend $x_1 = z_1$), $V'_1$ $\mathbb{C}^*_{\Gamma'}$ $V_1$ et $W'_1$ $\mathbb{C}^*_{\Gamma',z_1:V'_1}$ $W_1$. Donc, $W'_1\{x_1 \mapsto v_1\}$ $\mathbb{C}^*_{\Gamma'}$ $W_1\theta_1$.
- Cas $k > 1$. Par hypothèse de récurrence, nous avons $W'_{k-1}\{x_{k-1} \mapsto v_{k-1}\}$ $\mathbb{C}^*_{\Gamma'}$ $W_{k-1}\theta_{k-1}$. Or, $W'_{k-1}\{x_{k-1} \mapsto v_{k-1}\}$ $\mathbb{C}^*_{\Gamma'}$ $(x_k : V'_k)W'_k$ et $W_{k-1}\theta_{k-1} = (z_k : V_k\theta_{k-1})W_k\theta_{k-1}$. Par compatibilité avec le produit et $\alpha$-équivalence (on prend $x_k = z_k$), $V'_k$ $\mathbb{C}^*_{\Gamma'}$ $V_k\theta_{k-1}$ et $W'_k$ $\mathbb{C}^*_{\Gamma',z_k:V'_k}$ $W_k\theta_{k-1}$. Donc, $W'_k\{x_k \mapsto v_k\}$ $\mathbb{C}^*_{\Gamma'}$ $W_k\theta_k$.

En conclusion, $\Delta' \vdash v_k\sigma\sigma' \in [\![\Gamma' \vdash V'_k]\!]_{\Delta',\sigma\sigma',\xi''}$ et $V'_k$ $\mathbb{C}^*_{\Gamma'}$ $V_k\theta_k = V_k\theta$. Par (conv'), les types utilisés dans une conversion sont typables. On peut donc appliquer l'hypothèse de récurrence (b). Ainsi, $[\![\Gamma' \vdash V'_k]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash V_k\theta]\!]_{\Delta',\sigma\sigma',\xi''}$ et $\Delta' \vdash v_k\sigma\sigma' \in [\![\Gamma' \vdash V_k\theta]\!]_{\Delta',\sigma\sigma',\xi''}$.

(b) Comme pour (symb$^<$).

**(acc)** $\dfrac{\Gamma_0 \vdash_{\mathrm{c}} x\Gamma_0 : s}{\Gamma_0 \vdash_{\mathrm{c}} x : x\Gamma_0}$

(a) Puisque $(\sigma, \Gamma_0, \Delta')$ est valide par rapport à $\xi|_{\Delta'}$.

(b) $x$ est irréductible.

**(var)** $\dfrac{\Gamma' \vdash_{\mathrm{c}} T : s}{\Gamma', x : T \vdash_{\mathrm{c}} x : T}$

(a) Car $(\sigma\sigma', \Gamma', \Delta')$ est valide par rapport à $\xi''$.

(b) $x$ est irréductible.

**(weak)** $\dfrac{\Gamma' \vdash_{\mathrm{c}} t : T \quad \Gamma' \vdash_{\mathrm{c}} U : s}{\Gamma', x{:}U \vdash_{\mathrm{c}} t : T}$

(a) Par hypothèse de récurrence, $\Delta' \vdash t\sigma\sigma' \in [\![\Gamma' \vdash T]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma_0, \Gamma, x : U \vdash T]\!]_{\Delta',\sigma\sigma',\xi''}$.

(b) Comme $x \notin \mathrm{FV}(t)$, $[\![\Gamma', x{:}U \vdash t]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash t]\!]_{\Delta',\sigma\sigma',\xi''}$. Or, par hypothèse de récurrence, $[\![\Gamma' \vdash t]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash t']\!]_{\Delta',\sigma\sigma',\xi''}$.

**(prod)** $\dfrac{\Gamma', x{:}T \vdash_{\mathrm{c}} U : s}{\Gamma' \vdash_{\mathrm{c}} (x{:}T)U : s}$

(a) Nous devons montrer que $\Delta' \vdash (x : T\sigma\sigma')U\sigma\sigma' \in [\![\Gamma' \vdash s]\!]_{\Delta',\sigma\sigma',\xi''} = \overline{\mathbb{SN}}_{\Delta' \vdash s}$. Par inversion, $\Gamma' \vdash T : s'$. Par hypothèse de récurrence, $\Delta' \vdash T\sigma\sigma' \in [\![\Gamma' \vdash s']\!]_{\Delta',\sigma\sigma',\xi''}$. Montrons maintenant que $U\sigma\sigma' \in \mathcal{SN}$. Nous pouvons toujours supposer que $x \notin \mathrm{dom}(\Delta')$. Alors, $\sigma\sigma' : \Gamma', x{:}T \to \Delta''$ où $\Delta'' = \Delta, x{:}T\sigma\sigma'$. Soit $\xi''' = \xi''|_{\Delta''}$ si $x \in \mathcal{X}^\star$, et $\xi''' = \xi''|_{\Delta''} \cup \{x \mapsto \top_{\Delta'' \vdash x}\}$ si $x \in \mathcal{X}^\square$. Alors, $\xi'''$ est compatible avec $(\sigma\sigma', \Gamma', x : T, \Delta'')$ puisque, si $x \in \mathcal{X}^\square$ alors $x\xi''' \in \mathcal{R}_{\Delta'' \vdash x} = \mathcal{R}_{\Delta' \vdash x\sigma\sigma'}$ et, pour tout $X \in \mathrm{dom}^\square(\Gamma')$, $X\xi''' = X\xi'' \in \mathcal{R}_{\Delta' \vdash X\sigma\sigma'}$ puisque $\xi''$ est compatible avec $(\sigma\sigma', \Gamma', \Delta')$. De plus, $(\sigma\sigma', \Gamma', x : T, \Delta'')$ est valide par rapport à $\xi'''$ car, par le Lemme 115, $\Delta'' \vdash x\sigma\sigma' = \Delta'' \vdash x \in [\![\Gamma', x{:}T \vdash T]\!]_{\Delta'',\sigma\sigma',\xi'''}$. Donc, par hypothèse de récurrence, $\Delta'' \vdash U\sigma\sigma' \in [\![\Gamma', x{:}T \vdash s']\!]_{\Delta'',\sigma\sigma',\xi'''} = \overline{\mathbb{SN}}_{\Delta'' \vdash s'}$.

(b) Il y a deux cas :

– $T \to T'$. Montrons que $[\![\Gamma' \vdash (x{:}T)U]\!]_{\Delta',\sigma\sigma',\xi''} \subseteq [\![\Gamma' \vdash (x{:}T')U]\!]$. L'inclusion inverse se prouve de manière similaire. Soit $\Delta'' \vdash u \in [\![\Gamma' \vdash (x{:}T)U]\!]_{\Delta',\sigma\sigma',\xi''}$, $\Delta''' \vdash t \in [\![\Gamma' \vdash T']\!]_{\Delta'',\sigma\sigma',\xi''|_{\Delta''}}$ et $\sigma'' = \sigma\sigma' \cup \{x \mapsto t\}$. Par hypothèse de récurrence, $[\![\Gamma' \vdash T]\!]_{\Delta'',\sigma\sigma',\xi''|_{\Delta''}} = [\![\Gamma' \vdash T']\!]_{\Delta'',\sigma\sigma',\xi''|_{\Delta''}}$. Donc, $\Delta''' \vdash t \in [\![\Gamma' \vdash T]\!]_{\Delta'',\sigma\sigma',\xi''|_{\Delta''}}$, $\Delta''' \vdash ut \in [\![\Gamma', x : T \vdash U]\!]_{\Delta''',\sigma'',\xi''|_{\Delta'''}}$ et $\Delta'' \vdash u \in [\![\Gamma' \vdash (x : T')U]\!]_{\Delta',\sigma\sigma',\xi''}$.

– $U \to U'$. Montrons que $[\![\Gamma' \vdash (x{:}T)U]\!]_{\Delta',\sigma\sigma',\xi''} \subseteq [\![\Gamma' \vdash (x{:}T)U']\!]$. L'inclusion inverse se prouve de manière similaire. Soit $\Delta'' \vdash u \in [\![\Gamma' \vdash (x{:}T)U]\!]_{\Delta',\sigma\sigma',\xi''}$, $\Delta''' \vdash t \in [\![\Gamma' \vdash T]\!]_{\Delta'',\sigma\sigma',\xi''|_{\Delta''}}$ et $\sigma'' = \sigma\sigma' \cup \{x \mapsto t\}$. Alors, $\Delta''' \vdash ut \in [\![\Gamma', x : T \vdash U]\!]_{\Delta''',\sigma'',\xi''|_{\Delta'''}}$. Par hypothèse de récurrence, $[\![\Gamma', x{:}T \vdash U]\!]_{\Delta''',\sigma'',\xi''|_{\Delta'''}} = [\![\Gamma', x : T \vdash U']\!]_{\Delta''',\sigma'',\xi''|_{\Delta'''}}$. Donc, $\Delta'' \vdash ut \in [\![\Gamma', x : T \vdash U']\!]_{\Delta''',\sigma'',\xi''|_{\Delta'''}}$ et $\Delta'' \vdash u \in [\![\Gamma' \vdash (x{:}T)U']\!]_{\Delta',\sigma\sigma',\xi''}$.

**(abs)** $\dfrac{\Gamma', x{:}T \vdash_{\mathrm{c}} u : U \quad \Gamma' \vdash_{\mathrm{c}} (x{:}T)U : s}{\Gamma' \vdash_{\mathrm{c}} [x{:}T]u : (x{:}T)U}$

(a) Soit $\Gamma'' = \Gamma', x{:}T$. Nous devons montrer que $\Delta' \vdash [x : T\sigma\sigma']u\sigma\sigma' \in [\![\Gamma' \vdash (x : T)U]\!]_{\Delta',\sigma\sigma',\xi''}$. Soit $\Delta'' \vdash t \in [\![\Gamma' \vdash T]\!]_{\Delta',\sigma\sigma',\xi''}$ et $S \in \mathcal{R}_{\Delta'' \vdash t}$ si $x \in \mathcal{X}^\square$. Soit $\sigma'' = \sigma\sigma' \cup \{x \mapsto t\}$, $\xi''' = \xi''|_{\Delta''}$ si $x \in \mathcal{X}^\star$ et $\xi''' = \xi''|_{\Delta''} \cup \{x \mapsto S\}$ si $x \in \mathcal{X}^\square$. Nous avons $\sigma'' : \Gamma'' \to \Delta''$, $\xi'''$ est compatible avec $(\sigma'', \Gamma'', \Delta'')$ et $(\sigma'', \Gamma'', \Delta'')$ est valide par rapport à $\xi''$. Donc, par hypothèse de récurrence, $\Delta'' \vdash u\sigma'' \in R = [\![\Gamma'' \vdash U]\!]_{\Delta'',\sigma'',\xi'''}$. Alors, pour montrer que $v = [x : T\sigma\sigma']u\sigma\sigma'\ t \in R$, il nous suffit de montrer que $T\sigma\sigma', u\sigma\sigma' \in \mathcal{SN}$. En effet, dans ce cas, il est facile de montrer que tous les réduits immédiats du terme neutre $v$ appartiennent à $R$ par récurrence sur $(T\sigma\sigma', u\sigma\sigma', t)$ avec $\to_{\mathrm{lex}}$ comme ordre bien fondé. Par

hypothèse de récurrence, nous avons $\Delta' \vdash (x : T\sigma\sigma')U\sigma\sigma' \in [\![\Gamma' \vdash s]\!]_{\Delta',\sigma\sigma',\xi''}$. Donc, $T\sigma\sigma' \in \mathcal{SN}$. Enfin, si on prend $t = x$, ce qui est possible d'après le Lemme 115, nous avons vu que, par hypothèse de récurrence, $u\sigma'' = u\sigma\sigma' \in \mathcal{SN}$.

(b) Il y a deux cas :
- $T \to T'$. Comme $\overline{\mathbb{T}}_{\Delta' \vdash T\sigma\sigma'} = \overline{\mathbb{T}}_{\Delta' \vdash T'\sigma\sigma'}$, $[\![\Gamma' \vdash [x : T]u]\!]_{\Delta',\sigma\sigma',\xi''}$ et $[\![\Gamma' \vdash [x : T']u]\!]$ ont le même domaine et sont égales.
- $u \to u'$. Soit $\Delta'' \vdash t \in \overline{\mathbb{T}}_{\Delta' \vdash T\sigma\sigma'}$, $S \in \mathcal{R}_{\Delta'' \vdash t}$ si $x \in \mathcal{X}^{\square}$, $\sigma'' = \sigma\sigma' \cup \{x \mapsto t\}$, $\xi''' = \xi''|_{\Delta''}$ si $x \in \mathcal{X}^{\star}$ et $\xi''' = \xi''|_{\Delta''} \cup \{x \mapsto S\}$ si $x \in \mathcal{X}^{\square}$. Par hypothèse de récurrence, $[\![\Gamma', x : T \vdash u]\!]_{\Delta'',\sigma'',\xi'''} = [\![\Gamma', x : T \vdash u']\!]_{\Delta'',\sigma'',\xi'''}$. Donc, $[\![\Gamma' \vdash [x : T]u]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash [x : T]u']\!]_{\Delta',\sigma\sigma',\xi''}$.

**(app)** $\dfrac{\Gamma' \vdash_{\mathrm{c}} t : (x : U)V \quad \Gamma' \vdash_{\mathrm{c}} u : U}{\Gamma' \vdash_{\mathrm{c}} tu : V\{x \mapsto u\}}$

(a) Par hypothèse de récurrence, $\Delta' \vdash t\sigma\sigma' \in [\![\Gamma' \vdash (x : U)V]\!]_{\Delta',\sigma\sigma',\xi''}$ et $\Delta' \vdash u\sigma\sigma' \in [\![\Gamma' \vdash U]\!]_{\Delta',\sigma\sigma',\xi''}$. Soit $S = [\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''}$ si $x \in \mathcal{X}^{\square}$. Alors, par définition de $[\![\Gamma' \vdash (x : U)V]\!]_{\Delta',\sigma\sigma',\xi''}$, $\Delta' \vdash t\sigma\sigma'u\sigma\sigma' \in R = [\![\Gamma', x : U \vdash V]\!]_{\Delta',\sigma'',\xi'''}$ où $\sigma'' = \sigma\sigma' \cup \{x \mapsto u\sigma\sigma'\}$, $\xi''' = \xi''$ si $x \in \mathcal{X}^{\star}$, et $\xi''' = \xi'' \cup \{x \mapsto S\}$ si $x \in \mathcal{X}^{\square}$. Par substitution de candidats, $R = [\![\Gamma' \vdash V\{x \mapsto u\}]\!]_{\Delta',\sigma\sigma',\xi''}$.

(b) Il y a trois cas :
- $t \to t'$. Par hypothèse de récurrence, $[\![\Gamma' \vdash t]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash t']\!]_{\Delta',\sigma\sigma',\xi''}$. Donc, $[\![\Gamma' \vdash tu]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash t]\!]_{\Delta',\sigma\sigma',\xi''}(\Delta' \vdash u\sigma\sigma', [\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''}) = [\![\Gamma' \vdash t']\!](\Delta' \vdash u\sigma\sigma', [\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''}) = [\![\Gamma' \vdash t'u]\!]_{\Delta',\sigma\sigma',\xi''}$.
- $u \to u'$. Par hypothèse de récurrence, $[\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash u']\!]_{\Delta',\sigma\sigma',\xi''}$. Par **(P1)**, $[\![\Gamma' \vdash t]\!](\Delta' \vdash u\sigma\sigma', [\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''}) = [\![\Gamma' \vdash t]\!](\Delta' \vdash u'\sigma\sigma', [\![\Gamma' \vdash u']\!]_{\Delta',\sigma\sigma',\xi''})$. Donc, $[\![\Gamma' \vdash tu]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash tu']\!]_{\Delta',\sigma\sigma',\xi''}$.
- $t = [x : U']v$ et $t' = v\{x \mapsto u\}$. Soit $\sigma'' = \{x \mapsto u\}\sigma\sigma'$ et $\xi''' = \xi'' \cup \{x \mapsto [\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''}\}$. Par substitution de candidats, $[\![\Gamma' \vdash t']\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma', x : U' \vdash v]\!]_{\Delta',\sigma'',\xi'''}$. D'un autre côté, $[\![\Gamma' \vdash tu]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash t]\!]_{\Delta',\sigma\sigma',\xi''}(\Delta' \vdash u\sigma\sigma', [\![\Gamma' \vdash u]\!]_{\Delta',\sigma\sigma',\xi''}) = [\![\Gamma', x : U' \vdash v]\!]_{\Delta',\sigma\sigma' \cup \{x \mapsto u\sigma\sigma'\},\xi'''} = [\![\Gamma', x : U' \vdash v]\!]_{\Delta',\sigma'',\xi'''}$.

**(conv)** $\dfrac{\Gamma' \vdash_{\mathrm{c}} t : T \quad \Gamma' \vdash_{\mathrm{c}} T : s \quad T \downarrow T' \quad \Gamma' \vdash_{\mathrm{c}} T' : s}{\Gamma' \vdash_{\mathrm{c}} t : T'}$

(a) Soit $U$ le réduit commun à $T$ et $T'$. Par hypothèse de récurrence, $\Delta' \vdash t\sigma\sigma' \in [\![\Gamma' \vdash T]\!]_{\Delta',\sigma\sigma',\xi''}$, $[\![\Gamma' \vdash T]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash U]\!]_{\Delta',\sigma\sigma',\xi''}$ et $[\![\Gamma' \vdash T']\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash U]\!]_{\Delta',\sigma\sigma',\xi''}$. Donc, $[\![\Gamma' \vdash T]\!]_{\Delta',\sigma\sigma',\xi''} = [\![\Gamma' \vdash T']\!]_{\Delta',\sigma\sigma',\xi''}$ et $\Delta' \vdash t\sigma\sigma' \in [\![\Gamma' \vdash T']\!]_{\Delta',\sigma\sigma',\xi''}$.

(b) Par hypothèse de récurrence. ■

**Lemme 147 (Réductibilité des symboles d'ordre supérieur)** Soit $f \in \mathcal{F}_{\omega}$, $\tau_f = (\vec{x} : \vec{T})U$, $\Delta \in \mathbb{E}$, $\theta : \Gamma_f \to \Delta$ et $\xi$ compatible avec $(\theta, \Gamma_f, \Delta)$. Si $(\theta, \Gamma_f, \Delta)$ est valide par rapport à $\xi$ alors $\Delta \vdash f(\vec{x})\theta \in [\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi}$.

**Preuve.** Par récurrence sur $((f, \Delta, \theta, \xi), \vec{x}\theta)$ avec $(\sqsupset, \to)_{\mathrm{lex}}$ comme ordre bien fondé.

Posons $t_i = x_i\theta$ et $t = f(\vec{t})$. Comme dans le lemme sur les symboles de premier ordre, il suffit de montrer que, pour tout réduit immédiat $t'$ de $t$, $\Delta \vdash t' \in [\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi}$.

Si la réduction a lieu dans un $t_i$ alors on peut conclure par hypothèse de récurrence car les candidats de réductibilité sont stables par réduction **(R2)** et $\sqsupset$ est compatible avec la réduction.

Supposons maintenant qu'il existe une règle $(l \to r, \Gamma_0, \rho)$ et une substitution $\sigma$ telles que $t = l\sigma$. Supposons de plus que $l = f(\vec{l})$ et $\gamma_0 = \{\vec{x} \mapsto \vec{l}\}$. Alors, $\theta = \gamma_0\sigma$. Par **(S5)**, pour tout $x_i \in \mathrm{FV}^{\square}(\vec{T}U)$, $x_i\gamma_0\sigma \downarrow x_i\gamma_0\rho\sigma$.

Montrons alors que $[\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi} = [\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi}$ et $[\![\Gamma_f \vdash T_i]\!]_{\Delta,\theta,\xi} = [\![\Gamma_f \vdash T_i]\!]_{\Delta,\gamma_0\rho\sigma,\xi}$. Par **(S4)**, $\sigma : \Gamma_0 \to \Delta$. Par définition d'une règle bien formée, $\Gamma_0 \vdash l\rho : U\gamma_0\rho$. Donc, par inversion, $\gamma_0\rho : \Gamma_f \to \Gamma_0$ et $\gamma_0\rho\sigma : \Gamma_f \to \Delta$. Montrons maintenant que $\xi$ est compatible avec $(\gamma_0\rho\sigma, \Gamma_f, \Delta)$. Soit $x_i \in \mathrm{FV}^{\square}(\vec{T}U)$. Comme $\xi$ est compatible avec $(\gamma_0\sigma, \Gamma_f, \Delta)$, $x_i\xi \in \mathcal{R}_{\Delta\vdash x_i\gamma_0\sigma}$. Comme $x_i\gamma_0\sigma \downarrow x_i\gamma_0\rho\sigma$, d'après le Lemme 114 (b), $x_i\xi \in \mathcal{R}_{\Delta\vdash x_i\gamma_0\rho\sigma}$. Donc, d'après le Lemme 121 (c), $[\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi} = [\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi}$ et $[\![\Gamma_f \vdash T_i]\!]_{\Delta,\theta,\xi} = [\![\Gamma_f \vdash T_i]\!]_{\Delta,\gamma_0\rho\sigma,\xi}$.

Définissons maintenant $\xi'$ de telle sorte que $[\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi} = [\![\Gamma_0 \vdash U\gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$ et $[\![\Gamma_f \vdash T_i]\!]_{\Delta,\gamma_0\rho\sigma,\xi} = [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$. Soit $Y \in \mathrm{dom}^{\square}(\Gamma_0)$. Par **(b)**, pour tout $X \in \mathrm{FV}^{\square}(\vec{T}U)$, $X\gamma_0\rho \in \mathrm{dom}^{\square}(\Gamma_0)$ et, pour tout $X, X' \in \mathrm{FV}^{\square}(\vec{T}U)$, $X\gamma_0\rho = X'\gamma_0\rho$ implique $X = X'$. Alors, s'il existe $X$ (nécessairement unique) tel que $Y = X\gamma_0\rho$, nous prenons $Y\xi' = X\xi$. Sinon, nous prenons $Y\xi' = \top_{\Delta\vdash Y\sigma}$. Vérifions que $\xi'$ est compatible avec $(\sigma, \Gamma_0, \Delta)$. Soit $Y \in \mathrm{dom}^{\square}(\Gamma_0)$. Si $Y = X\gamma_0\rho$ alors $Y\xi' = X\xi$. Puisque $\xi$ est compatible avec $(\gamma_0\rho\sigma, \Gamma_0, \Delta)$, $X\xi \in \mathcal{R}_{\Delta\vdash X\gamma_0\rho\sigma}$. Comme $X\gamma_0\rho = Y$, $Y\xi' \in \mathcal{R}_{\Delta\vdash Y\sigma}$. Enfin, si $Y \neq X\gamma_0\rho$, $Y\xi' = \top_{\Delta\vdash Y\sigma} \in \mathcal{R}_{\Delta\vdash Y\sigma}$. Donc, $\xi'$ est compatible avec $(\sigma, \Gamma_0, \Delta)$. Par substitution de candidats, il existe $\xi''$ tel que $[\![\Gamma_0 \vdash U\gamma_0\rho]\!]_{\Delta,\sigma,\xi'} = [\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi''}$ et, pour tout $X \in \mathrm{dom}^{\square}(\Gamma_f)$, $X\xi'' = [\![\Gamma_0 \vdash X\gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$. Si $X \in \mathrm{FV}^{\square}(\vec{T}U)$ alors, par **(b)**, $X\gamma_0\rho = Y \in \mathrm{dom}^{\square}(\Gamma_0)$ et $X\xi'' = Y\xi' = X\xi$. Comme $\xi''$ et $\xi$ coïncident sur $\mathrm{FV}^{\square}(U)$, $[\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi''} = [\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi}$. Ainsi, $[\![\Gamma_f \vdash U]\!]_{\Delta,\gamma_0\rho\sigma,\xi} = [\![\Gamma_0 \vdash U\gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$. On montrerait de même que $[\![\Gamma_f \vdash T_i]\!]_{\Delta,\gamma_0\rho\sigma,\xi} = [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$.

Prouvons maintenant que $(\sigma, \Gamma_0, \Delta)$ est valide par rapport à $\xi'$. Par définition du Schéma Général, $(l \to r, \Gamma_0, \rho)$ est bien formée : $\Gamma_0 \vdash f(\vec{l})\rho : U\gamma_0\rho$, $\mathrm{dom}(\rho) \cap \mathrm{dom}(\Gamma_0) = \emptyset$ et, pour tout $x \in \mathrm{dom}(\Gamma_0)$, il existe $i$ tel que $l_i : T_i\gamma_0$ $(\rhd_1^{\rho})^*$ $x : x\Gamma_0$. Par inversion, $\Gamma_0 \vdash l_i\rho : T_i\gamma_0\rho$. Comme $\Delta \vdash l_i\sigma \in [\![\Gamma_0 \vdash T_i\gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$, par correction de l'accessibilité, $\Delta \vdash x\sigma \in [\![\Gamma_0 \vdash x\Gamma_0\rho]\!]_{\Delta,\sigma,\xi'}$. Comme $\mathrm{dom}(\rho) \cap \mathrm{dom}(\Gamma_0) = \emptyset$, $x\Gamma_0\rho = x\Gamma_0$ et $\Delta \vdash x\sigma \in [\![\Gamma_0 \vdash x\Gamma_0]\!]_{\Delta,\sigma,\xi'}$.

Ainsi, par correction de la clôture calculable, $\Delta \vdash r\sigma \in [\![\Gamma_0 \vdash U\gamma_0\rho]\!]_{\Delta,\sigma,\xi'} = [\![\Gamma_f \vdash U]\!]_{\Delta,\theta,\xi}$. ■

**Lemme 148 (Réductibilité des termes bien typés)** Pour tout $\Gamma \vdash t : T$, $\Delta \in \mathbb{E}$, $\theta : \Gamma \to \Delta$, $\xi$ compatible avec $(\theta, \Gamma, \Delta)$, si $(\theta, \Gamma, \Delta)$ est valide par rapport à $\xi$ alors $\Delta \vdash t\theta \in [\![\Gamma \vdash T]\!]_{\Delta,\theta,\xi}$.

**Preuve.** Par récurrence sur $\Gamma \vdash t : T$. On procède comme dans le lemme de correction de la clôture calculable sauf dans le cas (symb) où on utilise les lemmes de réductibilité des symboles de premier ordre et d'ordre supérieur. ■

**Théorème 149 (Normalisation forte)** Toutes les termes bien typés sont fortement normalisables.

**Preuve.** Supposons que $\Gamma \vdash t : T$. Soit $\theta$ la substitution identité. C'est une substitution bien typée de $\Gamma$ dans $\Gamma$. Soit $\xi$ l'assignement de candidats défini par $X\xi = \top_{\Gamma\vdash X}$. C'est un assignement compatible avec $(\theta, \Gamma, \Gamma)$ puisque, pour tout $X \in \mathrm{dom}^{\Box}(\Gamma)$, $X\xi \in \mathcal{R}_{\Gamma\vdash X}$. Enfin, $(\theta, \Gamma, \Gamma)$ est valide par rapport à $\xi$ puisque, pour tout $x \in \mathrm{dom}(\Gamma)$, $\Gamma \vdash x \in [\![\Gamma \vdash x\Gamma]\!]_{\Gamma,\theta,\xi}$. Donc, d'après le lemme de réductibilité des termes bien typés, $\Gamma \vdash t \in [\![\Gamma \vdash T]\!]_{\Delta,\theta,\xi}$ et, puisque $[\![\Gamma \vdash T]\!]_{\Delta,\theta,\xi} \subseteq \mathcal{SN}$, $t$ est fortement normalisable. ■

# Chapitre 9

# Futures directions de recherche

Dans ce chapitre, nous énumérons plus ou moins par ordre d'importance certaines de nos conditions de normalisation forte qu'il conviendrait d'affaiblir ou des extensions qu'il conviendrait d'étudier. Mise à part la récriture modulo, tous ces problèmes nous paraissent assez difficiles.

- **Récriture modulo.** Dans notre travail, nous n'avons pas considéré de récriture modulo certaines théories équationnelles extrêmement utiles comme l'associativité et la commutativité. Il est important de pouvoir étendre nos résultats à ce type de récriture. Mais, alors que cela ne semble pas poser trop de difficultés pour ce qui est de la récriture au niveau objet, et nous avons déjà quelques résultats préliminaires dans cette direction, cela est moins clair dans le cas de la récriture au niveau type.

- **Types quotients.** Nous avons vu que la récriture permet la formalisation de types quotients de manière interne, en autorisant des règles sur des "constructeurs". Cependant, prouver des propriétés par "induction" sur de tels types nécessite de connaître quelles sont les formes normales [70] et peut également nécessiter une stratégie de réduction particulière [37] ou de la récriture conditionnelle.

- **Confluence.** Parmi nos conditions de normalisation forte, nous demandons non seulement que la récriture soit confluente mais également que sa combinaison avec la $\beta$-réduction le soit. C'est là une condition assez forte dans la mesure où on ne peut pas s'appuyer sur la normalisation forte pour montrer la confluence [95, 20]. Et mis à part les cas des systèmes de récriture du premier ordre en l'absence de types dépendants [26] ou des systèmes d'ordre supérieur linéaires-gauches [92, 116], peu de résultats sont connus sur la modularité de la confluence pour la combinaison récriture d'ordre supérieur et $\beta$-réduction. Il serait donc intéressant d'étudier de près cette question tout à fait générale.

- **Confluence locale.** Nous pensons que, peut-être, la confluence locale est suffisante pour établir notre résultat. En effet, confluence locale et normalisation forte impliquent confluence. Cependant, dans ce cas, il semble nécessaire de montrer de

nombreuses propriétés en même temps que la normalisation forte, en particulier la correction de la $\beta$-réduction (*"subject reduction"*), ce qui semble difficile car beaucoup de définitions reposent sur cette propriété.

- **Simplicité.** Pour les symboles de prédicat non primitifs, nous demandons que les régles qui les définissent n'aient pas de paires critiques avec les autres régles. Ces conditions très fortes permettent de définir de manière simple une interprétation valide pour ces symboles de prédicats. Il conviendrait d'étudier dans quelle mesure ces conditions peuvent être affaiblies.

- **Cohérence logique.** Contrairement à ce qui se passe dans le Calcul des Constructions pur, dans le Calcul des Constructions Algébriques, il n'est pas impossible que les symboles et les règles de récriture permettent de former une preuve de $\bot = (P : \star)P$ en forme normale dans l'environnement vide. Dans ce cas, la normalisation forte ne suffit plus à assurer la cohérence logique. Il convient donc de rechercher des conditions syntaxiques sur les symboles et les règles qui assurent la cohérence du système (à l'instar des environnements "fortement cohérents" de J. Seldin [106]). Et, de manière plus générale, il convient de rechercher et d'étudier des modèles du Calcul des Constructions Algébriques.

- **Conservativité.** [1] Nous avons vu que, dans le Calcul des Constructions, l'ajout de règles de récriture permet de typer davantage de termes. Il conviendrait d'étudier dans quelle mesure une proposition prouvable en utilisant un ensemble de règles de récriture $\{l_1 \to r_1, \ldots\}$ peut aussi être prouvée en utilisant un ensemble d'hypothèses $\{l_1 = r_1, \ldots\}$. Cela constitue une autre voix de recherche pour établir la cohérence logique et mieux comprendre l'impact de la récriture sur le typage et la prouvabilité.

- **Définitions locales.** Dans notre travail, nous nous sommes restreints à des symboles définis de manière globale. Cependant, en pratique, au cours de l'élaboration d'une preuve formelle dans un système comme Coq [52], il peut être utile d'introduire des symboles et des règles de récriture permettant de simplifier celle-ci. Il conviendrait de réfléchir aux problèmes que cela peut poser d'avoir des définitions locales et d'étudier dans quelle mesure nos travaux peuvent être utiles pour les résoudre. Le problème des abréviations locales a déjà été résolu par E. Poll et P. Severi [101]. Celui des définitions locales par récriture est abordé par J. Chrzaszcz [27].

- **HORPO.** Pour assurer la normalisation forte des définitions d'ordre supérieur, nous avons fait le choix d'étendre la méthode du Schéma Général de J.-P. Jouannaud et M. Okada [73]. Or, l'ordre HORPO de J.-P. Jouannaud et A. Rubio [74], qui étend aux termes du $\lambda$-calcul simplement typé l'ordre RPO de N. Dershowitz pour les termes du premier ordre, est naturellement plus puissant puisqu'il est défini de manière récursive. De plus, D. Walukiewicz a récemment étendu cet

1. Nous remercions Henk Barendregt pour nous avoir suggéré d'approfondir cette question.

ordre aux termes du Calcul des Constructions avec symboles au niveau objet [117]. La combinaison de nos deux travaux devrait permettre d'aboutir à une extension de HORPO aux termes du Calcul des Constructions avec symboles au niveau type également.

- **$\eta$-Réduction.** Parmi nos conditions, nous l'avons vu, nous demandons la confluence de $\rightarrow_{=}\rightarrow_{\mathcal{R}} \cup \rightarrow_{\beta}$. Tels quels, nos résultats ne peuvent donc pas s'appliquer à la $\eta$-réduction, bien connue pour poser des difficultés supplémentaires [56], puisque $\rightarrow_{\beta} \cup \rightarrow_{\eta}$ n'est pas confluente sur les termes mal typés. Là encore, il conviendrait d'étudier dans quelle mesure cette condition de confluence peut-être relâchée.

- **Prédicats positifs non-stricts.** Les ordres utilisés dans la définition du Schéma Général pour comparer deux séquences d'arguments permettent d'accepter des définitions par récursion sur des types basiques ou strictement positifs comme c'est le cas dans le système Coq [52]. Cependant, N. P. Mendler [89] a montré que les définitions par récursion sur des types positifs non-stricts [87] étaient également fortement normalisantes. Il serait intéressant de parvenir à définir un schéma permettant d'accepter de telles définitions. Cependant, dans ce cas, on se heurte à une difficulté importante, comme celle que nous avons rencontré avec la $\iota$-réduction du Calcul des Constructions Inductives (CIC) (voir discussion avant la Définition 94) : la présence d'appels récursifs avec des variables liées qui, bien sûr, ne peuvent être instanciées que par des sous-termes stricts du membre gauche.

# Bibliographie


[1] M. Abadi, L. Cardelli, P.-L. Curien, and J.-J. Lévy. Explicit substitutions. *Journal of Functional Programming*, 1(4) :375–416, 1991.

[2] T. Altenkirch. *Constructions, Inductive Types and Strong Normalization*. PhD thesis, Edinburgh University (UK), 1993.

[3] F. Baader and T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[4] F. Barbanera. Adding algebraic rewriting to the Calculus of Constructions : strong normalization preserved. In *Proc. of the 2nd Int. Work. on Conditional and Typed Rewriting Systems*, LNCS 516, 1990.

[5] F. Barbanera and M. Fernández. Combining first and higher order rewrite systems with type assignment systems. In *Proc. of the 1st Int. Conf. on Typed Lambda Calculi and Applications*, LNCS 664, 1993.

[6] F. Barbanera and M. Fernández. Modularity of termination and confluence in combinations of rewrite systems with $\lambda_\omega$. In *Proc. of the 20th Int. Colloq. on Automata, Languages and Programming*, LNCS 700, 1993.

[7] F. Barbanera, M. Fernández, and H. Geuvers. Modularity of strong normalization and confluence in the algebraic-$\lambda$-cube. In *Proc. of the 9th IEEE Symp. on Logic in Computer Science*, 1994. Extended version in [8].

[8] F. Barbanera, M. Fernández, and H. Geuvers. Modularity of strong normalization in the algebraic-$\lambda$-cube. *Journal of Functional Programming*, 7(6) :613–660, 1997.

[9] H. Barendregt. Introduction to generalized type systems. *Journal of Functional Programming*, 1(2) :125–154, 1991.

[10] H. Barendregt. Lambda calculi with types. In S. Abramski, D. Gabbay, and T. Maibaum, editors, *Handbook of logic in computer science*, volume 2. Oxford University Press, 1992.

[11] B. Barras. *Auto-validation d'un système de preuves avec familles inductives*. PhD thesis, Université Paris VII (France), 1999.

[12] G. Barthe. The relevance of proof-irrelevance. In *Proc. of the 25th Int. Colloq. on Automata, Languages and Programming*, LNCS 1443, 1998.

[13] G. Barthe and H. Geuvers. Congruence types. In *Proc. of the 9th Int. Work. on Computer Science Logic*, LNCS 1092, 1995.

[14] G. Barthe and H. Geuvers. Modular properties of algebraic type systems. In *Proc. of the 2nd Int. Work. on Higher-Order Algebra, Logic and Term Rewriting*, LNCS 1074, 1995.

[15] G. Barthe and P.-A. Melliès. On the subject reduction property for algebraic type systems. In *Proc. of the 10th Int. Work. on Computer Science Logic*, LNCS 1258, 1996.

[16] G. Barthe and F. van Raamsdonk. Termination of algebraic type systems : the syntactic approach. In *Proc. of the 6th Int. Conf. on Algebraic and Logic Programming*, LNCS 1298, 1997.

[17] P. Bendix and D. Knuth. *Computational problems in abstract algebra*, chapter Simple word problems in universal algebra. Pergamon Press, 1970.

[18] S. Berardi. Towards a mathematical analysis of the Coquand-Huet Calculus of Constructions and the other systems in Barendregt's Cube. Technical report, Carnegie-Mellon University (USA) and Universita di Torino (Italy), 1988.

[19] F. Blanqui. Definitions by rewriting in the Calculus of Constructions. In *Proc. of the 16th IEEE Symp. on Logic in Computer Science*, 2001.

[20] F. Blanqui. Termination and confluence of higher-order rewrite systems. In *Proc. of the 11th Int. Conf. on Rewriting Techniques and Applications*, LNCS 1833, 2000.

[21] F. Blanqui, J.-P. Jouannaud, and M. Okada. The Calculus of Algebraic Constructions. In *Proc. of the 10th Int. Conf. on Rewriting Techniques and Applications*, LNCS 1631, 1999.

[22] F. Blanqui, J.-P. Jouannaud, and M. Okada. Inductive-data-type systems. *Theoretical Computer Science*, 277, 2001.

[23] V. Breazu-Tannen. Combining algebra and higher-order types. In *Proc. of the 3rd IEEE Symp. on Logic in Computer Science*, 1988.

[24] V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic strong normalization. In *Proc. of the 16th Int. Colloq. on Automata, Languages and Programming*, LNCS 372, 1989. Extended version in [25].

[25] V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic strong normalization. *Theoretical Computer Science*, 83(1) :3–28, 1991.

[26] V. Breazu-Tannen and J. Gallier. Polymorphic rewriting conserves algebraic confluence. *Information and Computation*, 114(1) :1–29, 1994.

[27] J. Chrzaszcz. Modular rewriting in the Calculus of Constructions, 2000. Presented at the Int. Work. on Types for Proofs and Programs.

[28] A. Church. A simple theory of types. *Journal of Symbolic Logic*, 5 :56–68, 1940.

[29] L. Colson and D. Fredholm. System T, call-by-value and the minimum problem. *Theoretical Computer Science*, 206 :301–315, 1998.

[30] E. Contejean, C. Marché, B. Monate, and X. Urbain. CiME, 2000. Available at `http://www.lri.fr/~demons/cime.html`.

[31] T. Coquand. Pattern matching with dependent types. In *Proc. of the 1992 Int. Work. on Types for Proofs and Programs. See* `http://www.lfcs.informatics.ed.ac.uk/research/types-bra/proc/`.

[32] T. Coquand. An algorithm for testing conversion in type theory. In G. Huet and G. Plotkin, editors, *Logical Frameworks*, pages 255–279. Cambridge University Press, 1991.

[33] T. Coquand and J. Gallier. A proof of strong normalization for the Theory of Constructions using a Kripke-like interpretation, 1990. Paper presented at the 1st Int. Work. on Logical Frameworks but not published in the proceedings. Available at `ftp://ftp.cis.upenn.edu/pub/papers/gallier/sntoc.dvi.Z`.

[34] T. Coquand and G. Huet. A theory of constructions, 1984. Paper presented at the Int. Symp. on Semantics of Data Types but not published in the proceedings. Extended version in [35].

[35] T. Coquand and G. Huet. The Calculus of Constructions. *Information and Computation*, 76(2–3) :95–120, 1988.

[36] T. Coquand and C. Paulin-Mohring. Inductively defined types. In *Proc. of the 1988 Int. Conf. on Computer Logic*, LNCS 417.

[37] P. Courtieu. Normalized types. In *Proc. of the 15th Int. Work. on Computer Science Logic*, LNCS 2142, 2001.

[38] H. B. Curry and R. Feys. *Combinatory Logic*. North-Holland, 1958.

[39] N. de Bruijn. The mathematical language AUTOMATH, its usage, and some of its extensions. In *Proc. of the Symp. on Automatic Demonstration*, Lecture Notes in Mathematics. Springer, 1968. Reprinted in [58].

[40] N. Dershowitz. Hierarchical termination. In *Proc. of the 4th Int. Work. on Conditional and Typed Rewriting Systems*, LNCS 968, 1994.

[41] N. Dershowitz. Orderings for term rewriting systems. In *20th IEEE Symposium on Foundations of Computer Science*, 1979. Extended version in [42].

[42] N. Dershowitz. Orderings for term rewriting systems. *Theoretical Computer Science*, 17 :279–301, 1982.

[43] N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, chapter 6. North-Holland, 1990.

[44] D. Dougherty. Adding algebraic rewriting to the untyped lambda calculus. In *Proc. of the 4th Int. Conf. on Rewriting Techniques and Applications*, LNCS 488, 1991. Extended version in [45].

[45] D. Dougherty. Adding algebraic rewriting to the untyped lambda calculus. *Information and Computation*, 101(2) :251–267, 1992.

[46] G. Dowek. La part du calcul, 1999. Mémoire d'habilitation. Available at `http://logical.inria.fr/~dowek/`.

[47] G. Dowek, T. Hardin, and C. Kirchner. Theorem proving modulo. Technical Report 3400, INRIA Rocquencourt (France), 1998.

[48] G. Dowek and B. Werner. Proof normalization modulo. In *Proc. of the 1998 Int. Work. on Types for Proofs and Programs*, LNCS 1657.

[49] G. Dowek and B. Werner. An inconsistent theory modulo defined by a confluent and terminating rewrite system, 2000. Available at `http://-logical.inria.fr/~dowek/`.

[50] K. Drosten. *Termersetzungssysteme*. PhD thesis, Passau University (Germany), 1989.

[51] M. Fernández. *Modèles de calculs multiparadigmes fondés sur la réécriture*. PhD thesis, Université Paris-Sud (France), 1993. See [8].

[52] J.-C. Filliâtre, C. Paulin, and *et al*. *The Coq Proof Assistant Reference Manual*. INRIA Rocquencourt (France), 2000. Available at `http://coq.inria.-fr/`.

[53] K. Futatsugi, J. Goguen, J.-P. Jouannaud, and J. Meseguer. Principles of OBJ-2. In *Proc. of the 12th ACM Symp. on Principles of Programming Languages, 1985*.

[54] J. Gallier. On Girard's "Candidats de Réductibilité". In P.-G. Odifreddi, editor, *Logic and Computer Science*. North-Holland, 1990.

[55] G. Gentzen. *Investigations into logical deduction*. PhD thesis, Göttingen, Germany, 1933. See [109].

[56] H. Geuvers. *Logics and Type Systems*. PhD thesis, Nijmegen University (Netherlands), 1993.

[57] H. Geuvers and M.-J. Nederhof. A modular proof of strong normalization for the Calculus of Constructions. *Journal of Functional Programming*, 1(2) :155–189, 1991.

[58] H. Geuvers, R. Nederpelt, and R. de Vrijer, editors. *Selected Papers on Automath*, volume 133 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1994.

[59] E. Giménez. Structural recursive definitions in type theory. In *Proc. of the 25th Int. Colloq. on Automata, Languages and Programming*, LNCS 1443, 1998.

[60] E. Giménez. *Un Calcul de Constructions infinies et son application à la vérification de systèmes communiquants*. PhD thesis, ENS Lyon (France), 1996.

[61] J.-Y. Girard. Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types. In J. Fenstad, editor, *Proc. of the 2nd Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1971. See [62].

[62] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. Cambridge University Press, 1988.

[63] K. Gödel. Über einer bisher noch nicht benützte Erweiterung des finiten standpunktes. *Dialectica*, 12, 1958. Reprinted in [64].

[64] K. Gödel. *Collected works – vol. 2 : publications 1938-1974*. Oxford University Press, 1990.

[65] J. A. Goguen and J. J. Tardo. An introduction to OBJ : a language for writing and testing formal algebraic specifications. In *Proc. of the Specification and Reliable Software Conference*, 1979.

[66] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. In *Proc. of the 2nd IEEE Symp. on Logic in Computer Science*, 1987.

[67] W. A. Howard. The formulae-as-types notion of construction (1969). In J. P. Seldin and J. R. Hindley, editors, *To H. B. Curry : Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press, 1980.

[68] J. Hsiang. *Topics in Automated Theorem Proving and Program Generation*. PhD thesis, University of Illinois (USA), 1982.

[69] J.-P. Jouannaud, C. Kirchner, H. Kirchner, and A. Megreliss. Programming with equalities, subsorts, overloading and parametrization in OBJ. *Journal of Logic Programming*, 12(1 and 2) :257–279, 1992.

[70] J.-P. Jouannaud and E. Kounalis. Proof by induction in equational theories without constructors. In *Proc. of the 1st IEEE Symp. on Logic in Computer Science*, 1986. Extended version in [71].

[71] J.-P. Jouannaud and E. Kounalis. Automatic proofs by induction in theories without constructors. *Information and Computation*, 82(1) :1–33, 1989.

[72] J.-P. Jouannaud and M. Okada. Executable higher-order algebraic specification languages. In *Proc. of the 6th IEEE Symp. on Logic in Computer Science*, 1991. Extended version in [73].

[73] J.-P. Jouannaud and M. Okada. Abstract Data Type Systems. *Theoretical Computer Science*, 173(2) :349–391, 1997.

[74] J.-P. Jouannaud and A. Rubio. The Higher-Order Recursive Path Ordering. In *Proc. of the 14th IEEE Symp. on Logic in Computer Science*, 1999. Extended version in [76].

[75] J.-P. Jouannaud and A. Rubio. A recursive path ordering for higher-order terms in eta-long beta-normal form. In *Proc. of the 7th Int. Conf. on Rewriting Techniques and Applications*, LNCS 1103, 1996.

[76] J.-P. Jouannaud and A. Rubio. Higher-order recursive path orderings " la carte", 2001. Journal submission. Available at `http://www.lri.fr/-~jouannau/`.

[77] C. Kirchner and *et al*. *ELAN user manual*. INRIA Nancy (France), 2000. Available at `http://elan.loria.fr/`.

[78] J. W. Klop, V. van Oostrom, and F. van Raamsdonk. Combinatory reduction systems : introduction and survey. *Theoretical Computer Science*, 121 :279–308, 1993.

[79] C. Loria-Saenz and J. Steinbach. Termination of combined (rewrite and $\lambda$-calculus) systems. In *Proc. of the 3rd Int. Work. on Conditional and Typed Rewriting Systems*, LNCS 656, 1992.

[80] Z. Luo. *An Extended Calculus of Constructions*. PhD thesis, Edinburgh University (UK), 1990.

[81] Z. Luo and R. Pollack. *LEGO Proof Development System : User's manual*. Edinburgh University (UK), 1992. Available at `http://www.dcs.ed.ac.uk-/home/lego/`.

[82] O. Lysne and J. Piris. A termination ordering for higher order rewrite systems. In *Proc. of the 6th Int. Conf. on Rewriting Techniques and Applications*, LNCS 914, 1995.

[83] P. Martin-Löf. Haupsatz for the intuitionnistic theory of iterated inductive definitions. In J. Fenstad, editor, *Proc. of the 2nd Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1971. See [84].

[84] P. Martin-Löf. *Intuitionistic type theory*. Bibliopolis, Napoli, Italy, 1984.

[85] R. Matthes. Monotone fixed-point types and strong normalization. In *Proc. of the 12th Int. Work. on Computer Science Logic*, LNCS 1584, 1998.

[86] R. Matthes. *Extensions of System F by Iteration and Primitive Recursion on Monotone Inductive Types*. PhD thesis, München University (Germany), 1998.

[87] R. Matthes. Lambda calculus : A case for inductive definitions. Available at `http://www.tcs.informatik.uni-muenchen.de/~matthes/`, 2000.

[88] R. Mayr and T. Nipkow. Higher-order rewrite systems and their confluence. *Theoretical Computer Science*, 192(2) :3–29, 1998.

[89] N. P. Mendler. *Inductive Definition in Type Theory*. PhD thesis, Cornell University (USA), 1987.

[90] J. Meseguer and *et al*. *Maude : Specification and Programming in Rewriting Logic*. SRI International's Computer Science Laboratory (USA), 1999. Available at `http://maude.csl.sri.com/`.

[91] D. Miller. A logic programming language with lambda-abstraction, function variables, and simple unification. In *Proc. of the 1989 Int. Work. on Extensions of Logic Programming*, LNCS 475.

[92] F. Müller. Confluence of the lambda calculus with left-linear algebraic rewriting. *Information Processing Letters*, 41(6) :293–299, 1992.

[93] R. Nederpelt. *Strong normalization in a typed lambda calculus with lambda structured types*. PhD thesis, Eindhoven University (Netherlands), 1973.

[94] M. Newman. On theories with a combinatorial definition of equivalence. *Annals of Mathematics*, 43(2) :223–243, 1942.

[95] T. Nipkow. Higher-order critical pairs. In *Proc. of the 6th IEEE Symp. on Logic in Computer Science*, 1991. Extended version in [88].

[96] M. J. O'Donnell. Computing in systems described by equations. In *LNCS 58*. Springer, 1977.

[97] M. Okada. Strong normalizability for the combined system of the typed lambda calculus and an arbitrary convergent term rewrite system. In *Proc. of the 1989 Int. Symp. on Symbolic and Algebraic Computation*, ACM Press.

[98] C. Paulin-Mohring. Extracting Fω's programs from proofs in the Calculus of Constructions. In *Proc. of the 16th ACM Symp. on Principles of Programming Languages, 1989.*

[99] C. Paulin-Mohring. Inductive definitions in the system Coq - rules and properties. In *Proc. of the 1st Int. Conf. on Typed Lambda Calculi and Applications*, LNCS 664, 1993.

[100] D. A. Plaisted. A recursively defined ordering for proving termination of term rewriting systems. Technical report, University of Illinois (USA), 1978.

[101] E. Poll and P. Severi. Pure Types Systems with definitions. In *Proc. of the 3rd Int. Symp. on Logical Foundations of Computer Science*, LNCS 813, 1994.

[102] R. Pollack. *The theory of LEGO, a proof checker for the Extended Calculus of Constructions.* PhD thesis, Edinburgh University (UK), 1994.

[103] J. Reynolds. Towards a theory of type structure. In *Programming Symposium, LNCS 19*, 1974.

[104] M. Rusinowitch. On termination of the direct sum of term-rewriting systems. *Information Processing Letters*, 1987.

[105] H. Schwichtenberg. Definierbare Funktionen im λ–Kalkül mit Typen. *Archive for Mathematical Logic*, 17 :113–114, 1976.

[106] J. Seldin. Excluded middle without definite descriptions in the theory of constructions. In *Proc. of the 1st Montreal Workshop on Programming Language Theory*, 1991.

[107] M. P. A. Sellink. Verifying process algebra proofs in type theory. In *Proc. of the Int. Work. on Semantics of Specification Languages*, Workshops in Computing, 1993.

[108] M. Stefanova. *Properties of Typing Systems.* PhD thesis, Nijmegen University (Netherlands), 1998.

[109] M. E. Szabo, editor. *Collected papers of Gerhard Gentzen.* Studies in Logic and the Foundations of Mathematics. North-Holland, 1969.

[110] W. W. Tait. Infinitely long terms of transfinite type. In J. Crossley and M. Dummet, editors, *Formal Systems and Recursive Functions*, Studies in Logic and the Foundations of Mathematics. North-Holland, 1965.

[111] W. W. Tait. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic*, 32(2) :198–212, 1967.

[112] J. Terlouw. Een nadera bewijstheoretische analyse van GSTT's. Technical report, Nijmegen University (Netherlands), 1989.

[113] Y. Toyama. Counterexamples to termination for the direct sum of term rewriting systems. *Information Processing Letters*, 25(3) :141–143, 1987.

[114] L. S. van Benthem Jutting. Typing in pure type systems. *Information and Computation*, 105(1) :30–41, 1993.

[115] J. van de Pol. *Termination of higher-order rewrite systems.* PhD thesis, Utrecht University (Nederlands), 1996.

[116] V. van Oostrom. *Confluence for Abstract and Higher-Order Rewriting*. PhD thesis, Vrije University (Netherlands), 1994.

[117] D. Walukiewicz. Termination of rewriting in the Calculus of Constructions. In *Proc. of the 2000 Workshop on Logical Frameworks and Meta-languages*.

[118] B. Werner. *Une Théorie des Constructions Inductives*. PhD thesis, Université Paris VII (France), 1994.

[119] A. Whitehead and B. Russell. *Principia Mathematica*. Cambridge University Press, 1911.