Chapitre 3

Introduction à la Réductibilité

Ce chapitre est une introduction aux preuves de normalisation forte de λ-calcul typés. Le but principal est d'expliquer la notion de *réductibilité* qui est au centre de ces preuves. On essaye ensuite d'indiquer les idées essentielles et les points clés de la preuve de normalisation forte de CCI. À l'exception de sa partie 3.7, ce chapitre est sensé pouvoir être lu indépendamment du reste de la thèse. Le but de ce chapitre étant d'abord pédagogique, les parties techniques de certaines démonstrations sont données en petits caractères pour indiquer que leur lecture n'est pas indispensable pour saisir l'idée générale.

3.1 Rappels

Ces quelques lignes sont inutiles pour qui aura lu les chapitres précédents.

On considère un ensemble dénombrable de *variables*, que l'on désignera généralement par les lettres x, y, etc. Les λ-termes sont décrits par la grammaire habituelle :

$$\boxed{t \ := \ x \mid (t\ t) \mid \lambda x.t}$$

On définit l'opération de substitution, notée $t[x \setminus t']$ de la manière usuelle. Un *radical* est un terme de la forme $(\lambda x.t\ t')$. On définit la relation binaire $\rightarrow_\beta$ par :

$$(\lambda x.t\ t') \rightarrow_\beta t[x \setminus t']$$

et comme dans le chapitre précédent, on écrit $t \triangleright_\beta t'$ pour dire que l'on obtient t' par reécriture d'un sous-terme de t par $\rightarrow_\beta$. Un terme t est *normal* si aucun de ses sous-termes n'est un radical, ou de manière équivalente, s'il n'existe pas de terme t' tel que $t \triangleright_\beta t'$. Enfin un terme est dit *fortement normalisable*, s'il n'existe pas de chaîne infinie de réductions partant de t. On appelle $\mathcal{SN}$ l'ensemble des termes fortement normalisables.

> *Dans ce chapitre, on ne s'intéressera pas à la η-réduction. Aussi tous les systèmes considérés dans ce chapitre sont confluents et admettent la propriété de Church-Rosser indépendamment de la propriété de normalisation. On omettra systématiquement les preuves correspondantes.*

3.2 Normalisation et λ-calcul pur

Tout étudiant à qui l'on présente le λ-calcul apprend lors du premier cours, que certains termes n'admettent pas de formes normales. On donne en général l'exemple fameux :

$$(\lambda x.(x\ x)\ \lambda x.(x\ x)) \rhd_\beta (\lambda x.(x\ x)\ \lambda x.(x\ x)).$$

En simplifiant à peine, on peut dire que le but d'un système de type est d'isoler une classe aussi intéressante que possible de λ-termes qui vérifieront certaines propriétés, et en particulier celle de normalisation. La question posée ici est de prouver que tout terme bien typé est effectivement normalisable.

Avant de se lancer dans le typage, un bon exercice mental est d'oublier pour un temps le contre-exemple ci-dessus, et d'essayer de prouver naïvement la normalisation du λ-calcul pur. L'outil essentiel dont nous disposons pour prouver des propriétés sur les λ-termes est la récurrence structurelle. Essayons :

- Le terme x est fortement normalisable.
- Si le terme t est fortement normalisable, alors le terme $\lambda x.t$ l'est évidemment aussi.
- C'est le cas de l'application qui pose problème : même si les termes t_1 et t_2 sont fortement normalisables, le terme $(t_1\ t_2)$ ne l'est pas forcément. En effet, si $t_1 = \lambda x.t_3$, alors $(t_1\ t_2) \rhd_\beta t_3[x \setminus t_2]$ et nous ne pouvons rien dire sur ce dernier terme. Si l'on reprend l'exemple ci-dessus de terme non-normalisable, on voit bien que $\lambda x.(x\ x) \in \mathcal{SN}$.

On peut essayer de renforcer l'hypothèse de récurrence, en quantifiant par exemple sur toutes les substitutions possibles sur t en prouvant par récurrence sur t :

si l'on substitue des termes fortement normalisables à toutes les variables libres de t, on obtient un terme fortement normalisable.

Le lecteur curieux qui s'y essayera verra que cette fois encore, on ne peut terminer la preuve par récurrence à cause du cas de l'application. Dans un premier temps, c'est donc d'abord elle qui retiendra notre attention.

3.3 Types simples

On redonne rapidement la définition du λ-calcul simplement typé (voir aussi [83] par exemple). On se donne un ensemble de *types atomiques* que l'on n'a pas besoin de préciser. L'ensemble des types est alors défini inductivement :

- tout type atomique est un type
- si A et B sont des types, alors $A \to B$ est un type.

La présentation du λ-calcul simplement typé est plus concise si on utilise des variables explicitement typées : on considère simplement qu'à toute variable est associée un type, et pour tout type il existe une infinité de variables qui lui sont associées. On notera x^A, x^B, y^A, etc. À partir de là on définit la relation de typage par :

$$\boxed{x^A : A \qquad \frac{t : A}{\lambda x^B.t : A \to B} \qquad \frac{t_1 : A \to B \qquad t_2 : A}{(t_1\ t_2) : B}}$$

Considérons maintenant le problème de la normalisation forte des λ-termes simplement typés. Pour chaque terme formé par application, nous disposons d'une information essentielle : si $(t_1\ t_2)$ est bien formé, disons de type A, alors t_1 est bien formé d'un type $B \to A$ et t_2 est bien formé de type B. L'idée est de renforcer à nouveau l'hypothèse de récurrence, mais en se servant de l'information de type : pour chaque type U, on définit un prédicat $|U|$ sur les termes, de manière à ce que si t_1 vérifie $|B \to A|$ et t_2 vérifie $|B|$, alors $(t_1\ t_2)$ vérifie bien $|A|$. On se donne donc :

- $|A|(t) \Leftrightarrow t \in \mathcal{SN}$, si A est atomique.
- $|A \to B|(t) \Leftrightarrow \forall u.|A|(u) \Rightarrow |B|(t\ u)$.

C'est cette nouvelle condition de récurrence que l'on appelle *réductibilité*. On montre alors facilement la condition essentielle que pour tout type la notion de réductibilité est plus forte que la normalisation forte :

Lemme 3.1 *Pour tout type U et tout terme t, si $|U|(t)$, alors t est fortement normalisable.*

PREUVE Par récurrence structurelle sur U. On montre simultanément que $|U| \subset \mathcal{SN}$, et que pour toute variable x et toute séquence $\vec{t}$ de termes fortement normalisables on a $(x\ \vec{t}) \in |U|$. Le cas U atomique est immédiat. Si $U = A \to B$ et $t \in |A \to B|$, on a $x \in |A|$, et donc $(t\ x) \in |B|$. Comme $|B| \subset \mathcal{SN}$, on voit que t est fortement normalisable. Par ailleurs soit $(x\ \vec{t})$ et $u \in |A|$; alors $u \in \mathcal{SN}$ et aussi $(x\ \vec{t}\ u) \in |B|$; on a donc $(x\ \vec{t}) \in |A \to B|$. ■

Dans la suite on identifiera le prédicat $|U|$ et l'ensemble des termes qui le vérifient. On écrira donc indifféremment $|U|(t)$ et $t \in |U|$.

On va donc chercher à montrer que tout terme t de type A vérifie effectivement $|A|(t)$. Cette proposition n'est pas encore tout à fait assez forte, car il nous faut tenir compte des substitutions possibles sur les variables libres de t. Mais comme chacune de ces variables libres possède un type associé, on peut utiliser la notion de réductibilité pour *restreindre le domaine de quantification des substitutions* :

Définition 3.1 (Interprétations adaptées) *On appellera interprétation adaptée, toute substitution $\mathcal{I}$ qui à toute variable libre x^A de type A associera un terme reductible de type A :*

$$\mathcal{I}(x^A) \in |A|.$$

À partir de là, nous sommes capables de quantifier sur toutes les instanciations légales des variables libres d'un terme. Aussi sommes nous maintenant en état de prouver :

Théorème 3 *Si le jugement $t : U$ est dérivable, alors pour toute interprétation adaptée $\mathcal{I}$, on a $t[\mathcal{I}] \in |U|$.*

La preuve se fait par récurrence sur la structure de t, c'est-à-dire, à isomorphisme près, par récurrence sur la structure de la dérivation du jugement. Deux cas sont maintenant immédiats :

• Si $t = x^A$, on sait que $U = A$, et comme $\mathcal{I}$ est une interprétation adaptée, on a $t[\mathcal{I}] = \mathcal{I}(x^A) \in |A| = |U|$.

• Si $t = (t_1\ t_2)$, nous profitons de la définition du prédicat de réductibilité : on sait que $t_1 : A \to B$ et $t_2 : A$ avec $U = B$. Par hypothèse de récurrence on a $t_1[\mathcal{I}] \in |A \to B|$ et $t_2[\mathcal{I}] \in |A|$, alors la définition de $|A \to B|$ nous assure que

$$t[\mathcal{I}] = (t_1[\mathcal{I}]\ t_2[\mathcal{I}]) \in |B| = |U|.$$

Nous avons donc contourné la difficulté liée à l'application. En revanche le problème est repoussé au cas de l'abstraction. En effet l'introduction d'un λ augmente la fonctionnalité du type, et donc l'hypothèse à prouver est d'autant plus forte. C'est là qu'intervient le "miracle" de la réductibilité. On peut en effet généraliser la propriété $(x\ \vec{t}) \in |U|$ pour remarquer :

Lemme 3.2 *Si tous les réduits d'un terme $(t_1\ t_2)$ sont réductibles d'un type U, alors $(t_1\ t_2)$ l'est aussi.*

Ainsi que cet autre lemme qui nous sera également utile :

Lemme 3.3 *Soit un type U. L'ensemble $|U|$ est clos par β-réduction :*

$$t \in |U| \wedge t \rhd t' \Rightarrow t' \in |U|.$$

Preuve On prouve les deux lemmes conjointement par récurrence sur la structure de U. Le cas U atomique est évident. Si $U = A \to B$, alors on commence par remarquer que $(t_1\ t_2)$ est fortement normalisable puisque tous ses réduits le sont. Ensuite pour tout terme $u \in |A|$, on peut montrer par récurrence sur le nombre maximal de réductions dans u que $(t_1\ t_2\ u) \in |B|$. En effet, on peut appliquer l'hypothèse de récurrence et se contenter de prouver que tous les réduits de $(t_1\ t_2\ u)$ vérifient $|B|$. Or $(t_1\ t_2\ u)$ peut se réduire en :

- $(t'\ u)$ avec $(t_1\ t_2) \rhd t'$. Mais alors $t' \in |A \to B|$ et donc $(t'\ u) \in |B|$.
- $(t_1\ t_2\ u')$ avec $u \rhd u'$. Mais alors on a toujours $u' \in |A|$ (hypothèse de récurrence sur A) et le nombre de réductions possibles dans u a décru.

De même prenons $t \in |A \to B|$ avec $t \rhd t'$. Pour vérifier que $t' \in |A \to B|$, prenons $u \in |A|$. On a $(t\ u) \in |B|$, et donc par hypothèse de récurrence, $(t'\ u) \in |B|$, et donc $t' \in |A \to B|$. ■

À partir de là nous pouvons revenir à la preuve du théorème et traiter maintenant le cas de l'abstraction :

• Si $t = \lambda x^A.t_1$, on sait que $U = A \to B$, avec $t_1 : B$ et pour toute interprétation adaptée $\mathcal{I}$, $t_1[\mathcal{I}] \in |B|$. Montrer que $t[\mathcal{I}] \in |A \to B|$ c'est prouver que pour tout $u \in |A|$, $(t[\mathcal{I}]\ u) \in |B|$. Pour cela, d'après le lemme ci-dessus, il suffit de vérifier que tout réduit de $(t[\mathcal{I}]\ u)$ est bien élément de $|B|$. On raisonne par récurrence sur le nombre maximal de réductions faisables dans $t_1[\mathcal{I}]$ et u. Les réduits de $(t[\mathcal{I}]\ u)$ sont de l'une des formes suivantes :

$$\begin{array}{l} t_1[\mathcal{I}; x \leftarrow u] \\ (t_1'\ u) \text{ avec } t_1[\mathcal{I}] \rhd t_1' \\ (t_1[\mathcal{I}]\ u') \text{ avec } u \rhd u' \end{array}$$

Dans le premier cas, comme $\mathcal{I}; x \leftarrow u$ est bien une interprétation adaptée, $t_1[\mathcal{I}; x \leftarrow u]$ appartient bien à $|B|$. Dans les deux autres cas, il suffit de remarquer que le nombre maximal de réductions possibles dans les deux termes a décru et appliquer l'hypothèse de récurrence, en utilisant, dans le troisième cas, le lemme 3.3 pour assurer que $u' \in |A|$.

Ceci clôt la preuve du théorème.

On s'assure ensuite que le théorème suffit bien à prouver la normalisation forte en vérifiant que pour tout que type U, $x^U \in |U|$. Il s'ensuit que l'identité est une interprétation adaptée, et donc que si $t : A$, alors $t \in |A|$ et donc $t \in \mathcal{SN}$.

Cette preuve est due à Tait [140]. La formulation ci-dessus est très proche de [83]. Ce qui est essentiel à retenir, c'est le rôle des différentes étapes de la preuve, que nous retrouverons dans toutes les preuves de normalisation de tous les systèmes de types. À savoir, en simplifiant un peu :

- Chaque type U est interprété par un ensemble de termes, noté $|U|$.
- Preuve que $|U| \subset \mathcal{SN}$.
- Preuve d'un certain nombre de propriétés de clôture pour chaque ensemble $|U|$.
- Preuve que si $t : U$, alors $t[\mathcal{I}] \in |U|$, et donc en particulier $t \in |U|$ et $t \in \mathcal{SN}$.

On peut voir ce travail comme la construction d'un modèle syntaxique pour le système, chaque type étant interprété par un ensemble de termes. On peut également le voir comme les trois étapes d'un raisonnement par récurrence : définition de l'hypothèse de récurrence (construction de $|U|$), vérification que cette hypothèse est bien plus forte que le résultat final ($|U| \subset \mathcal{SN}$), et enfin la preuve par récurrence proprement dite, que si $t : U$ est dérivable, l'hypothèse de récurrence $t[\mathcal{I}] \in |U|$ est bien vérifiée.

Lors de l'étude de systèmes de types plus complexes, le problème essentiel sera la définition de l'interprétation des nouveaux types (inductifs, polymorphes, définis par récurrence, etc). Il faut toutefois remarquer que la définition de $|A \to B|$, qui est le cœur de la preuve, reste inchangée pour tous les systèmes.

3.4 Le système T et son extension par des ordinaux

Le système T a été introduit par Gödel en 1958 [61], pour prouver la cohérence de l'arithmétique fonctionnelle du premier ordre. Il s'agit essentiellement du λ-calcul simplement typé étendu par le type inductif des entiers naturels. Tout comme nous l'avons vu, ceci correspond à l'introduction de deux constructeurs 0 et S, et d'un opérateur de récurrence structurelle, noté R. Les règles de typage sont les mêmes que pour le λ-calcul simplement typé, étendues par les règles suivantes :

$$\boxed{\begin{array}{c} 0 : \mathrm{Nat} \qquad S : \mathrm{Nat} \to \mathrm{Nat} \\ \dfrac{t_0 : T \qquad t_S : \mathrm{Nat} \to T \to T \qquad u : \mathrm{Nat}}{(R\ u\ t_0\ t_S) : T} \end{array}}$$

De plus, la β-réduction est étendue par :

$$\begin{array}{rcl} (R\ 0\ t_0\ t_S) & \rhd_\iota & t_0 \\ (R\ (S\ t)\ t_0\ t_S) & \rhd_\iota & (t_S\ t\ (R\ t\ t_0\ t_S)) \end{array}$$

La preuve de normalisation forte est alors une extension très simple de celle, donnée ci-dessus, pour le λ-calcul simplement typé. La définition de la réductibilité reste inchangée. En particulier, on prend :

$$|\mathrm{Nat}| \equiv \mathcal{SN}.$$

La preuve ne diffère alors pas beaucoup de celle du λ-calcul simplement typé. Le travail essentiel est d'adapter le lemme 3.2 ; pour cela, il est commode d'introduire une notion de *termes neutres.*

Définition 3.2 (termes neutres) *Un terme est dit* neutre *s'il n'est pas de l'une des formes suivantes :*

$$\lambda x.t \qquad 0 \qquad (S\ t).$$

On appelle $\mathcal{NT}$ l'ensemble des termes neutres.

On obtient alors une formulation des propriétés de clôture de $|T|$ plus synthétique que les lemmes 3.2 et 3.3.

Lemme 3.4 *Pour tout type T :*

- $|T| \subset \mathcal{SN}$
- $\forall t \in |T|.t \rhd t' \Rightarrow t' \in |T|$
- $\forall t \in \mathcal{NT}.(\forall t'.t \rhd t' \Rightarrow t' \in |T|) \Rightarrow t \in |T|$

La preuve est à peu près identique à celle des lemmes 3.2 et 3.3.

On peut ensuite conserver la définition d'interprétation adaptée du paragraphe précédent ainsi que la formulation du théorème :

Théorème 4 *Pout tout jugement $t : T$ dérivable dans le système T, pour toute interprétation adaptée $\mathcal{I}$ on a $t[\mathcal{I}] \in |T|$.*

PREUVE Là encore elle est par récurrence sur la dérivation du jugement, et diffère peu de la précédente. Le cas intéressant est celui de l'opérateur d'élimination : On sait que $u[\mathcal{I}] \in \mathcal{SN}$, $t_0[\mathcal{I}] \in |T|$, $t_S[\mathcal{I}] \in \mathcal{SN} \rightarrow |T| \rightarrow |T|$ et il faut vérifier $(R\ u\ t_0\ t_S)[\mathcal{I}] \in |T|$. La preuve utilise deux récurrences emboîtées : une première sur la *taille de la forme normale de u*, une deuxième sur le nombre maximal de réductions possibles dans $u[\mathcal{I}], t_0[\mathcal{I}]$ et $t_S[\mathcal{I}]$. Comme $(R\ u\ t_0\ t_S)[\mathcal{I}]$ est neutre, il suffit de prouver que tous ses réduits sont dans $|T|$, or les-dits réduits sont de l'une des formes suivantes :

- $(R\ u'\ t_0[\mathcal{I}]\ t_S[\mathcal{I}])$ avec $u[\mathcal{I}] \rhd u'$
- $(R\ u[\mathcal{I}]\ t_0'\ t_S[\mathcal{I}])$ avec $t_0[\mathcal{I}] \rhd t_0'$
- $(R\ u[\mathcal{I}]\ t_0[\mathcal{I}]\ t_S')$ avec $t_S[\mathcal{I}] \rhd t_S'$
- $t_0[\mathcal{I}]$ si $u[\mathcal{I}] = 0$
- $(t_S[\mathcal{I}]\ v\ (R\ v\ t_0[\mathcal{I}]\ t_S[\mathcal{I}]))$.

Les trois premiers cas sont aisément traités par la seconde hypothèse de récurrence. Le quatrième cas est facile : $t_0[\mathcal{I}] \in |T|$. Le dernier cas est le plus subtil : la forme normale de v est plus petite que celle de u, on a donc $(R\ v\ t_0[\mathcal{I}]\ t_S[\mathcal{I}])) \in |T|$ (première hypothèse de récurrence), comme on a également $v \in |\mathrm{Nat}|$, on peut conclure $(t_S[\mathcal{I}]\ v\ (R\ v\ t_0[\mathcal{I}]\ t_S[\mathcal{I}])) \in |T|$. ■

L'exemple du système T est essentiel, mais ne suffit pas tout à fait pour illustrer le principe des preuves par réductibilité adaptées aux types inductifs. En effet, une récurrence sur un entier correspond exactement à une récurrence structurelle sur la forme normale de sa représentation dans le système T [1]. C'est cette dernière propriété qui permet de choisir l'ensemble $\mathcal{SN}$ des termes fortement normalisables pour l'interprétation $|\mathrm{Nat}|$ du type des entiers naturels. Le cas général est

[1] Cette propriété est en fait vraie pour tout type de données.

plus complexe. Un bon exemple est à ce titre l'extension du système T par le type des ordinaux. Celui-ci est défini à partir du type des entiers naturels à l'aide de trois constructeurs :

$$\boxed{\begin{array}{lcl} 0_o & : & \mathrm{Ord} \\ S_o & : & \mathrm{Ord} \to \mathrm{Ord} \\ \lim & : & (\mathrm{Nat} \to \mathrm{Ord}) \to \mathrm{Ord} \end{array}}$$

Le schéma d'élimination obéit alors à la règle de typage suivante :

$$\boxed{\frac{t_0 : T \qquad t_S : \mathrm{Ord} \to T \to T \qquad t_l : (\mathrm{Nat} \to \mathrm{Ord}) \to (\mathrm{Nat} \to t) \to t \qquad u : \mathrm{Ord}}{(R_{\mathrm{Ord}}\ u\ t_0\ t_S\ t_l) : T}}$$

avec les nouvelles règles de réduction :

$$\boxed{\begin{array}{lll} (R_{\mathrm{Ord}}\ 0_o\ t_0\ t_S\ t_l) & \rhd_\iota & t_0 \\ (R_{\mathrm{Ord}}\ (S_o\ u)\ t_0\ t_S\ t_l) & \rhd_\iota & (t_S\ u\ (R_{\mathrm{Ord}}\ u\ t_0\ t_S\ t_l) \\ (R_{\mathrm{Ord}}\ (\lim\ f)\ t_0\ t_S\ t_l) & \rhd_\iota & (t_l\ f\ \lambda x^{\mathrm{Nat}}.(R_{\mathrm{Ord}}\ (f\ x)\ t_0\ t_S\ t_l) \end{array}}$$

C'est bien sûr la troisième de ces règles qui est nouvelle par rapport au cas des entiers naturels. Le point important est l'appel récursif sur $(f\ x)$, qui n'est pas un sous-terme de $(\lim f)$. Il faudra donc, au moins implicitement, exprimer le fait que, quel que soit l'entier x, $(f\ x)$ est inférieur à $(\lim\ f)$ pour un certain ordre bien-fondé. Ceci n'est pas surprenant, puisque $(\lim\ f)$ est justement censé représenter la borne supérieure de l'ensemble $\{f(n), n \in \mathbb{N}\}$. En termes de réductibilité, on exigera donc que $(\lim\ f)$ soit réductible de type Ord (élément de $|\mathrm{Ord}|$), si et seulement si pour tout terme u réductible de type Nat (élément de $|\mathrm{Nat}|$), le terme $(f\ u)$ est bien réductible de type Ord (c.à.d. est lui-même élément de $|\mathrm{Ord}|$). Plus précisément, on définit l'ensemble $|\mathrm{Ord}|$ inductivement comme le plus petit ensemble vérifiant les conditions suivantes :

- Si un terme u est fortement normalisable et n'admet pas de réduit de la forme $(S_o\ v)$ ou $(\lim\ f)$, alors $u \in |\mathrm{Ord}|$.
- Si un terme u est fortement normalisable et se réduit vers $(S_o\ v)$ avec $v \in |\mathrm{Ord}|$, alors $u \in |\mathrm{Ord}|$.
- Si un terme u est fortement normalisable et se réduit vers $(\lim\ f)$, avec pour tout $v \in |\mathrm{Nat}|$, $(f\ v) \in |\mathrm{Ord}|$, alors $u \in |\mathrm{Ord}|$.

La justification précise des définitions inductives en théorie des ensembles fait appel au lemme du point fixe de Tarski. C'est-à-dire que l'ensemble $|\mathrm{Ord}|$ est en fait défini comme le plus petit point fixe d'un opérateur croissant des ensembles de termes vers les ensembles de termes. Toutefois cette définition de $|\mathrm{Ord}|$ est un petit peu lourde à utiliser ; elle implique :

- de refaire la démonstration des lemmes 3.3, 3.2 et 3.1 car Ord est un type atomique qui n'est pas interprété par $\mathcal{SN}$
- de faire une preuve par récurrence sur *la structure de la preuve de* $t \in$ Ord pour vérifier que $(R_o\ t_0\ f_S\ f_l)$ est bien réductible.

Pour ces deux raisons, nous préférons introduire dès maintenant la notion de *candidats de réductibilité*, qui nous permettra de factoriser les preuves des lemmes 3.3, 3.2 et 3.1 et aussi d'utiliser une définition plus commode de $|\mathrm{Ord}|$. Par ailleurs, cette notion sera, c'est bien connu, indispensable lorsque nous passerons à des systèmes de type imprédicatifs.

Dans ce qui précède, nous avons utilisé les propriétés suivantes :

- Pour tout type T, $|T| \subset \mathcal{SN}$.
- Pour tout type T, l'ensemble $|T|$ est clos par réduction :
$$\forall t, t'.(t \in |T| \wedge t \rhd t') \Rightarrow t' \in |T|.$$
- Pour tout type T, si t est un terme neutre tel que pour tout terme t', $t \rhd t' \Rightarrow t' \in |T|$, alors $t \in |T|$.

Une remarque importante est que lorsque l'on prouve ces propriétés par récurrence sur la structure de T, on ne se sert *que* des hypothèses de récurrence. A partir de là on va pouvoir factoriser la plus grande partie de cette preuve, en distinguant les ensembles de termes vérifiant ces propriétés :

Définition 3.3 (Candidats de réductibilité) *Un ensemble de termes $\mathcal{C}$ est un candidat de réductibilité s'il vérifie les propriétés suivantes :*

$$\begin{array}{ll} \textbf{CR 1} & \mathcal{C} \subset \mathcal{SN} \\ \textbf{CR 2} & \forall t \in \mathcal{C}. \forall t'.(t \in \mathcal{C} \wedge t \rhd t') \Rightarrow t' \in \mathcal{C} \\ \textbf{CR 3} & \forall t \in \mathcal{NT}. (\forall t'. t \rhd t' \Rightarrow t' \in \mathcal{C}) \Rightarrow t \in \mathcal{C} \end{array}$$

Notre remarque précédente est alors illustrée par les trois lemmes :

Lemme 3.5 *L'ensemble $\mathcal{SN}$ est un candidat de réductibilité.*

Lemme 3.6 *Soient $\mathcal{C}$ et $\mathcal{D}$ deux candidats de réductibilité. Alors l'ensemble suivant est un candidat de réductibilité :*
$$\mathcal{C} \to \mathcal{D} \equiv \{t, \forall u \in \mathcal{C}.(t\ u) \in \mathcal{D}\}.$$

Preuve Soit $t \in \mathcal{C} \to \mathcal{D}$. $\mathcal{C}$ contient tous les termes normaux neutres, en particulier toutes les variables, on a donc $(t\ x) \in \mathcal{D}$ et donc $(t\ x)$ et t sont fortement normalisables. Si $(t\ u) \in \mathcal{D}$ et $t \rhd t'$, alors $(t\ u) \rhd (t'\ u)$ et donc $(t'u) \in \mathcal{D}$. On en déduit $t' \in \mathcal{C} \to \mathcal{D}$. Enfin supposons maintenant que t soit neutre et que tous ses réduits soient dans $\mathcal{C} \to \mathcal{D}$. Soit alors $u \in \mathcal{C}$, on prouve par récurrence sur le nombre maximal de réductions faisables dans $\mathcal{C}$ que $(t\ u) \in \mathcal{D}$; en effet, $(t\ u)$ est neutre et ses réduits sont de la forme :

- $(t'\ u)$ qui est dans $\mathcal{D}$ puisque $t' \in \mathcal{C} \to \mathcal{D}$
- $(t\ u')$ qui est dans $\mathcal{D}$ par hypothèse de récurrence.

Il n'y a pas d'autre cas, puisque t est neutre. ■

Lemme 3.7 *L'intersection d'un ensemble non-vide de candidats de réductibilité est un candidat de réductibilité.*

La preuve de ce dernier lemme est à peu près immédiate. Ce lemme est important, car il dit que tout ensemble de candidats de réductibilité admet une *borne inférieure* pour l'ordre d'inclusion. En d'autres termes, l'ensemble des candidats de réductibilité muni de l'ordre d'inclusion est un inf-demi-treillis. Une conséquence essentielle est que tout opérateur croissant sur les candidats de réductibilité admet un *plus petit point fixe*. Nous allons tout de suite faire usage de cette possibilité pour construire $|\mathrm{Ord}|$.

Définition 3.4 *On définit $|\mathrm{Ord}|$ comme le plus petit candidat de réductibilité vérifiant la condition suivante : un terme t appartient à $|\mathrm{Ord}|$ si et seulement si pour tout candidat de réductibilité $\mathcal{C}$, pour tous termes $t_0 \in \mathcal{C}$, $t_S \in |\mathrm{Ord}| \to \mathcal{C} \to \mathcal{C}$, $t_l \in (|Nat| \to |\mathrm{Ord}|) \to (|Nat| \to \mathcal{C}) \to \mathcal{C}$, on a*

$$(R_{\mathrm{Ord}}\ t\ t_0\ t_S\ t_l) \in \mathcal{C}.$$

Il faut bien sûr justifier l'existence de l'ensemble $|\mathrm{Ord}|$. Grâce a la remarque précédente, nous allons pouvoir le définir comme le plus petit point fixe d'un opérateur croissant sur les candidats :

Soit S un ensemble de termes et $\mathcal{C}$ un candidat de réductibilité. On peut définir l'ensemble de termes suivant :

$$G(S,\mathcal{C}) \equiv \{t, \forall (t_0, t_S, t_l) \in \mathcal{C} \times S \to \mathcal{C} \to \mathcal{C} \times (\boldsymbol{\mathcal{SN}} \to S) \to (\boldsymbol{\mathcal{SN}} \to \mathcal{C}) \to \mathcal{C}.(R_o\ t\ t_0\ t_S\ t_l) \in \mathcal{C}\}$$

et aussi l'application F des ensembles de termes vers les ensembles de termes, définie par :

$$F(S) \equiv \bigcap_{\mathcal{C} \in \mathcal{CR}} G(S,\mathcal{C}).$$

On peut alors définir $|\mathrm{Ord}|$ comme le plus petit point fixe de F à condition de vérifier les deux conditions suivantes :

- F est croissante : si $S_1 \subset S_2$ alors $F(S_1) \subset F(S_2)$
- F est effectivement un opérateur sur les candidats : si $S \in \mathcal{CR}$ alors $F(S) \in \mathcal{CR}$.

Le premier point est facile ; il suffit de remarquer $\mathcal{C} \subset \mathcal{C}' \Rightarrow \mathcal{C}' \to \mathcal{D} \subset \mathcal{C} \to \mathcal{D}$ et le reste suit aisément. Pour le deuxième point, $\mathcal{CR}$ étant clos pour l'intersection, il suffit de vérifier que $G(S,\mathcal{C}) \in \mathcal{CR}$. Les preuves des conditions **CR 1** et **CR 2** sont immédiates. La preuve est similaire à celle du lemme 3.6, par récurrence sur le nombre maximal de réductions possibles sur les termes t_0, t_S et t_l.

À partir de là, il est immédiat que :

Lemme 3.8 *Soit T un type quelconque du système avec ordinaux. L'ensemble $|T|$ est un candidat de réductibilité.*

On peut expliquer ainsi la définition de $|\mathrm{Ord}|$: comme tous les $|T|$ sont des candidats de réductibilité, *il suffit de quantifier sur tous les candidats de réductibilité pour quantifier en particulier sur toutes les interprétations $|T|$ possibles.* C'est aussi exactement ce qui sera fait dans la section suivante pour interpréter la quantification polymorphe du système F.

Le reste de la preuve est très facile. Comme précédemment on prouve que si $t : T$, alors pour toute interprétation adaptée $\mathcal{I}$ on a $t[\mathcal{I}] \in |T|$. On détaille les cas des nouvelles règles :

- introduction de 0_o : si $\mathcal{C} \in \mathcal{CR}$, $t_0 \in \mathcal{C}$ et t_S et t_l sont fortement normalisables, on vérifie facilement, comme d'habitude maintenant par récurrence sur le nombre maximal de réductions possibles dans t_0, t_S et t_l que $(R_o\ 0_o\ t_0\ t_S\ t_l) \in \mathcal{C}$.
- les cas des règles d'introduction de S_o et lim sont tout à fait similaires [2].
- introduction de R_o : c'est immédiat ; par définition, si $u[\mathcal{I}] \in |\mathrm{Ord}|$, $t_o[\mathcal{I}] \in |T|$, $t_S[\mathcal{I}] \in |\mathrm{Ord} \to T \to T|$ et $t_l \in |(\mathrm{Nat} \to \mathrm{Ord}) \to (\mathrm{Nat} \to T) \to T|$ alors
$$(R_o\ u[\mathcal{I}]\ t_o[\mathcal{I}]\ t_S[\mathcal{I}]\ t_l[\mathcal{I}]) \in |T|.$$

3.5 Le système F

Le système F, également appelé λ-calcul du second ordre, a été introduit par Girard en 1970 pour interpréter les preuves de l'arithmétique fonctionnelle du second ordre et prouver ainsi sa cohérence [58, 59]. En arithmétique fonctionnelle du second ordre, on peut définir une proposition ou un prédicat en quantifiant sur toutes les propositions ou prédicats ; c'est ce que l'on appelle l'imprédicativité. Sa contre-partie en λ-calcul est précisément la présence de types polymorphes. En fait, le système F est le λ-calcul polymorphe le plus simple qui soit. Les types de ce système sont décrits par l'algèbre :
$$T\ :=\ \alpha \mid T \to T \mid \forall\alpha.T$$
où α désigne un élément d'un ensemble dénombrable et donné de *variables de types*.

Nous considérerons la version du système où les applications de types ne sont pas données explicitement dans les termes. Du fait de la présence de variables liées dans les types, une présentation utilisant des contextes est plus commode ; ceci rend par ailleurs inutiles les annotations de types sur les variables. Les termes sont décrits par :
$$t\ :=\ x \mid (t\ t) \mid \lambda x.t$$
et les règles de typage sont :

$$\text{(Var)}\ \frac{(x,T) \in \Gamma}{\Gamma \vdash x : T}$$
$$\text{(Lam)}\ \frac{\Gamma :: (x,T_2) \vdash t : T_1}{\Gamma \vdash \lambda x.t : T_2 \to T_1} \qquad \text{(App)}\ \frac{\Gamma \vdash t_1 : T_2 \to T_1 \qquad \Gamma \vdash t_2 : T_2}{\Gamma \vdash (t_1\ t_2) : T_1}$$
$$(\forall\text{-I})\ \frac{\Gamma \vdash t : T \qquad \forall (x,T_0) \in \Gamma.\alpha \notin FV(T_0)}{\Gamma \vdash t : \forall\alpha.T} \qquad (\forall\text{-E})\ \frac{\Gamma \vdash t : \forall\alpha.T}{\Gamma \vdash t[\alpha \setminus T_1] : T[\alpha \setminus T_1]}$$

La définition des contextes et de FV ainsi que la signification de $(x,T) \in \Gamma$ sont évidentes ; on peut reprendre les définitions formelles du chapitre 2. De même les définitions des termes neutres et fortement normalisables ainsi que des candidats de réductibilité restent quasiment inchangées.

La différence essentielle avec les systèmes précédents est bien sûr la présence de variables de types et de la quantification du second ordre. Le travail consistera essentiellement à réussir à interpréter ces derniers. Il est clair que l'interprétation d'une variable de type dépend de la manière dont elle

[2]Notons en particulier que l'on n'a pas besoin d'induction transfinie pour vérifier que lim $\in |(\mathrm{Nat} \to \mathrm{Ord}) \to \mathrm{Ord}|$

sera ensuite instanciée. Pour celà, on commence par se donner l'interprétation de chaque variable pour définir les interprétations des types. De cette manière, on va être capable de quantifier sur toutes les instanciations possibles. Comme dans la définition de Ord dans le système précédent, une quantification universelle sur tous les candidats de réductibilité permettera de capturer une quantification universelle sur tous les types.

Définition 3.5 (Interprétation) *On appelle interprétation une application associant un candidat de réductibilité à toute variable de type et un λ-terme à toute variable de terme.*

Une interprétation sera généralement désignée par la lettre $\mathcal{I}$. Cette notion étend la notion de substitution utilisée jusqu'alors dans la dernière partie de la preuve. On pourrait ici distinguer interprétations des variables de type et interprétations des variables de terme, mais dans les systèmes avec types dépendants il nous faudra de toute façons les mélanger.

On peut alors définir, pour tout type T, le candidat de réductibilité $|T|_\mathcal{I}$ pour toute interprétation $\mathcal{I}$. On désignera par $\mathcal{I}; \alpha \leftarrow \mathcal{C}$ l'interprétation valant $\mathcal{C}$ en α et égale à $\mathcal{I}$ partout ailleurs.

$$\begin{array}{rcl} |\alpha|_\mathcal{I} & \equiv & \mathcal{I}(\alpha) \\ |A \rightarrow B|_\mathcal{I} & \equiv & |A|_\mathcal{I} \rightarrow |B|_\mathcal{I} \\ |\forall \alpha . A|_\mathcal{I} & \equiv & \bigcap_{\mathcal{C} \in \mathcal{CR}} |A|_{\mathcal{I};\alpha \leftarrow \mathcal{C}} \end{array}$$

D'après les lemmes précédents, il est clair que $|T|_\mathcal{I}$ est effectivement un candidat de réductibilité. Par ailleurs une propriété essentielle est que l'interprétation est compatible avec la substitution :

Lemme 3.9 *Pour tous types T et T_1, toute variable de type α et tout interprétation $\mathcal{I}$, on a :*

$$|T[\alpha \setminus T_1]|_\mathcal{I} = |T|_{\mathcal{I};\alpha \leftarrow |T_1|_\mathcal{I}}.$$

PREUVE On raisonne bien sûr par récurrence structurelle sur T ; les trois cas sont quasi-immédiats. ■

Il nous suffit alors d'adapter la définition 3.1 à la notion d'interprétation pour terminer notre preuve :

Définition 3.6 (Interprétations adaptées) *Une interprétation $\mathcal{I}$ est dite adaptée à un contexte Γ , si pour tout couple (x, T) apparaissant dans Γ, on a bien $\mathcal{I}(x) \in |T|_\mathcal{I}$.*

Théorème 5 *Si un jugement $\Gamma \vdash t : T$ est dérivable, alors pour toute interprétation $\mathcal{I}$ adaptée à Γ, on a $t[\mathcal{I}] \in |T|_\mathcal{I}$.*

La preuve est bien sûr par récurrence sur la structure de la dérivation du jugement de typage ; elle ressemble énormément à la preuve correspondante pour le λ-calcul simplement typé. Les cas supplémentaires ne sont pas difficiles.

- Le cas de la règle VAR est immédiat : par définition, $\mathcal{I}(x) \in |T|_\mathcal{I}$.
- Le cas de APP est identique au cas des types simples : $t_1[\mathcal{I}] \in |T_2 \rightarrow T_2|_\mathcal{I}$ et $t_2[\mathcal{I}] \in |T_2|_\mathcal{I}$. Aussi : $(t_1\ t_2)[\mathcal{I}] = (t_1[\mathcal{I}]\ t_2[\mathcal{I}]) \in |t_1|_\mathcal{I}$.

- Le cas de Lam n'est pas vraiment nouveau non plus. On sait que pour toute interprétation $\mathcal{I}$ adaptée à $\Gamma :: (x, T_2)$, $t[\mathcal{I}] \in |T_1|_{\mathcal{I}}$. Soit maintenant $t_2 \in |T_2|_{\mathcal{I}}$; il nous faut vérifier

$$(\lambda x^{T_2}.t[\mathcal{I}; x^{T_2} \leftarrow x^{T_2}]\ t_2) \in |T_1|_{\mathcal{I}}$$

  pour prouver $(\lambda x^{T_2}.t)[\mathcal{I}] \in |T_1 \to T_2|_{\mathcal{I}}$. Comme précédemment, on raisonne par récurrence sur le nombre maximal de réductions possibles dans $t[\mathcal{I}]$ et t_2. Le terme $(\lambda x^{T_2}.t[\mathcal{I}]\ t_2)$ peut se réduire en :

  - $(\lambda x^{T_2}.t'\ t_2')$, cas traité par l'hypothèse de récurrence
  - $t[\mathcal{I}; x^{T_2} \leftarrow t_2]$, mais alors comme $t_2 \in |T_2|_{\mathcal{I}}$, $\mathcal{I}; x^{T_2} \leftarrow t_2$ est une interprétation adaptée à $\Gamma :: (x, T_2)$, et donc $t[\mathcal{I}; x^{T_2} \leftarrow t_2] \in |T_1|_{\mathcal{I}}$.

- Soit $\mathcal{I}$ une interprétation adaptée à Γ et $\mathcal{C}$ un candidat de réductibilité quelconque. Il faut vérifier que $\mathcal{I}; \alpha \leftarrow \mathcal{C}$ est bien une interprétation adaptée à Γ. C'est évident, puisque pour tout (x, T_0) apparaissant dans Γ, on a $\alpha \notin FV(T_0)$, et donc $|T_0|_{\mathcal{I}} = |T_0|_{\mathcal{I}; \alpha \leftarrow \mathcal{C}}$.

- L'ensemble $|T_1|_{\mathcal{I}}$ est un candidat de réductibilité ; donc

$$|T[\alpha \setminus T_1]|_{\mathcal{I}} = |T|_{\mathcal{I}; \alpha \leftarrow |T_1|_{\mathcal{I}}} \subset |\forall \alpha.T|_{\mathcal{I}}$$

  ce qui implique $t[\mathcal{I}] \in |T[\alpha \setminus T_1]|_{\mathcal{I}}$.

Enfin pour vérifier que tous les termes typables sont fortement normalisables, il suffit de remarquer que l'interprétation $\mathcal{I}$ telle que $\mathcal{I}(\alpha) = \mathcal{SN}$ pour tout α et $\mathcal{I}(x) = x$ pour tout x est adaptée à tout contexte.

Il faut remarquer que la concision et la simplicité de présentation du système F se retrouve au niveau de la preuve de normalisation ; cette dernière est plus courte et plus élégante que, par exemple, la preuve pour le système avec ordinaux. Une remarque bien connue est que comme le système F est imprédicatif, sa preuve de normalisation doit l'être aussi. Plus précisément, c'est le fait de quantifier sur toutes les interprétations $\mathcal{I}$ qui est effectivement un raisonnement d'ordre supérieur : une interprétation définit en effet un prédicat sur les λ-termes pour chaque variable de type. On quantifie donc bien sur une classe de prédicats.

Il faut aussi noter que prouver la normalisation du système F étendu par les entiers primitifs du système T, ainsi que des ordinaux comme présenté dans la section précédente ne pose maintenant aucun problème. Il suffit de suivre la preuve pour le système F, en reprenant les définitions de $|\mathrm{Nat}|$ et $|\mathrm{Ord}|$ données précédemment. Cette remarque est importante puisque nous cherchons ultimement à prouver la normalisation forte pour un système incorporant polymorphisme et types inductifs.

3.6 Les types dépendants

La notion de types dépendants correspond, comme indiqué dans le chapitre 1, à la possibilité de faire dépendre un type d'un terme, ce qui permet de représenter les prédicats de manière interne au calcul. Il connu que, pour les λ-calculs dits du "Cube de Barendregt", on peut assez facilement ramener la normalisation forte du système avec types dépendants à celui du système sans types dépendants correspondant (F_ω pour le Calcul des Constructions, types simples pour LF...). Nous ne ferons toutefois pas usage de cette facilité, car nous n'en aurons plus la possibilité lorsque nous ajouterons l'élimination forte.

Le système que nous considérerons ici est le système F étendu par des types dépendants. Dans la classification du Cube de Barendregt, il est connu sous le nom de λP_2. Il se distingue du Calcul des Constructions uniquement en ce qu'il interdit la formation de fonctions des types vers les types (ou plus généralement des prédicats vers les prédicats) : pour tout type A on peut définir $list_A$, le type des listes d'éléments de type A, mais on ne peut pas construire le type "générique" *list*, qui à A associe $list_A$.

Contrairement à ce que nous avons fait dans le chapitre précédent pour le Calcul des Constructions inductives, nous donnons une présentation stratifiée des preuves (t), des prédicats (T) et des ordres (K) :

$$\boxed{\begin{array}{lll} K & ::= & \mathrm{Set} \mid (x:T)K \\ T & ::= & \alpha \mid [x:T]T \mid (T\ t) \mid (\alpha:K)T \mid (x:T_1)T_2 \\ t & ::= & x \mid [x:T]t \mid (t\ t) \mid [\alpha:K]t \mid (t\ T) \end{array}}$$

Pour ces trois classes de termes, on peut sans difficultés définir les notions de substitution, β-réduction normalisation et β-équivalence de la manière usuelle. Les contextes sont définis comme suit :

$$\Gamma \ ::= \ [] \mid \Gamma :: (x,T) \mid \Gamma :: (\alpha, K).$$

La complication de la présentation se comprend bien si l'on songe qu'une variable de type α peut effectivement représenter un type (c'est-à-dire $\alpha : \mathrm{Set}$), ou par exemple un prédicat sur un type donné ($\alpha : \mathrm{Nat} \to \mathrm{Set}$). Il est donc nécessaire de faire apparaître les liaisons des variables de type (ou plus exactement des variables de prédicat) dans le contexte. Les ordres servent à déterminer le domaine d'un prédicat : si $T : \mathrm{Set}$, alors T est effectivement un type, si $T : T_0 \to \mathrm{Set}$, T sera un prédicat sur les objet de type T_0, etc. Bien sûr, si un jugement $\Gamma \vdash t : T$ est dérivable, alors T sera un type et on pourra aussi dériver $\Gamma \vdash T : \mathrm{Set}$.

Les jugements peuvent être de trois formes :

- $\Gamma \vdash t : T$ pour dire que le terme t est bien formé et habite le type T.
- $\Gamma \vdash T : K$ pour dire que T est un prédicat bien-formé d'ordre K.
- $\Gamma \vdash K$ pour dire que K est un ordre bien-formé.

On écrit $\Gamma \vdash \mathcal{J}$ pour désigner indiféremment les trois formes de jugement. Les règles de dérivation sont alors :

$$
\text{(Ax)}\quad [\,] \vdash \text{Set} \qquad \text{(K-Prod)}\ \frac{\Gamma :: (x,T) \vdash K}{\Gamma \vdash (x:T)K}
$$

$$
\text{(T-Lam)}\ \frac{\Gamma :: (x,T_1) \vdash T : K}{\Gamma \vdash [x:T_1]T : (x:T_1)K} \qquad \text{(T-App)}\ \frac{\Gamma \vdash T : (x:T_1)K \qquad \Gamma \vdash t : T_1}{\Gamma \vdash (T\ t) : K[x \setminus t]}
$$

$$
\text{(T-Prod1)}\ \frac{\Gamma :: (x,T_1) \vdash T : \text{Set}}{\Gamma \vdash (x:T_1)T : \text{Set}} \qquad \text{(T-Prod2)}\ \frac{\Gamma :: (\alpha,K) \vdash T : \text{Set}}{\Gamma \vdash (\alpha:K)T : \text{Set}}
$$

$$
\text{(t-Lam1)}\ \frac{\Gamma :: (x,T_1) \vdash t : T}{\Gamma \vdash [x:T_1]t : (x:T_1)T} \qquad \text{(t-Lam2)}\ \frac{\Gamma :: (\alpha,K) \vdash t : T}{\Gamma \vdash [\alpha:K] : (\alpha:K)T}
$$

$$
\text{(t-App1)}\ \frac{\Gamma \vdash t_1 : (x:T_2)T_1 \qquad \Gamma \vdash t_2 : T_2}{\Gamma \vdash (t_1\ t_2) : T_1[x \setminus t_2]} \qquad \text{(t-App2)}\ \frac{\Gamma \vdash t : (\alpha:K)T \qquad \Gamma \vdash T_1 : K}{\Gamma \vdash (t\ T_1) : T[\alpha \setminus T_1]}
$$

$$
\text{(Var1)}\ \frac{\Gamma \vdash T : \text{Set} \qquad x \notin \Gamma}{\Gamma :: (x,T) \vdash x : T} \qquad \text{(Var2)}\ \frac{\Gamma \vdash K \qquad \alpha \notin \Gamma}{\Gamma :: (\alpha,K) \vdash \alpha : K}
$$

$$
\text{(Weak1)}\ \frac{\Gamma \vdash \mathcal{J} \qquad \Gamma \vdash T : \text{Set}}{\Gamma :: (x,T) \vdash \mathcal{J}} \qquad \text{(Weak2)}\ \frac{\Gamma \vdash \mathcal{J} \qquad \Gamma \vdash K}{\Gamma :: (\alpha,K) \vdash \mathcal{J}}
$$

$$
\text{(Conv)}\ \frac{\Gamma \vdash t : T_1 \qquad \Gamma \vdash T_1 : \text{Set} \qquad T_1 =_\beta T_2 \qquad \Gamma \vdash T_2 : \text{Set}}{\Gamma \vdash t : T_2}
$$

Dans ce chapitre, nous nous intéresserons uniquement à la normalisation forte des λ-termes purs sous-jacents aux preuves. On passe des unes aux autres par l'opération "d'oubli" suivante :

$$
\begin{array}{rcl}
\|x\| & \equiv & x \\
\|[x:T]t\| & \equiv & \lambda x.\|t\| \\
\|[\alpha:K]t\| & \equiv & \|t\| \\
\|(t_1\ t_2)\| & \equiv & (\|t_1\|\ \|t_2\|) \\
\|(t\ T)\| & \equiv & \|t\|
\end{array}
$$

On peut alors prouver facilement la propriété essentielle suivante :

Lemme 3.10 *Soit un terme t. Si $t \rhd_\beta^* t'$, alors $\|t\| \rhd_\beta^* \|t'\|$. Aussi, pour tous termes t et t', on a*

$$t =_\beta t' \Rightarrow \|t\| =_\beta \|t'\|.$$

L'adaptation de la méthode de réductibilité au présent système se comprend intuitivement bien. Comme précédemment, nous allons interpréter tout type (c.à.d. tout prédicat d'ordre Set) par un candidat de réductibilité, c.à.d. un ensemble de λ-termes ; comme les prédicats d'arité 1 (c.à.d. d'un ordre de la forme $T \to \text{Set}$) sont essentiellement des fonctions des preuves vers les types, ils seront interprétés par des applications associant un candidat de réductibilité à tout terme pur, et ainsi de suite.

Formellement, on commence par définir *l'arité* d'un ordre comme le nombre de produits qui le forment :

$$\begin{aligned}\mathsf{Ar}(\mathrm{Set}) &= 0\\ \mathsf{Ar}((x:T)K) &= \mathsf{Ar}(K)+1\end{aligned}$$

on peut alors définir pour chaque entier n l'ensemble $\rho(n)$ des *valuations d'arité* n :

- $\rho(0)$ est l'ensemble des parties de Λ
- $\rho(n+1)$ est l'ensemble des applications de Λ vers $\rho(n)$.

Nous pouvons maintenant définir une notion de candidat de réductibilité pour chaque arité :

Définition 3.7 *Pour tout entier naturel n, on définit $\mathcal{CR}(n) \subset \rho(n)$, l'ensemble des* candidats de réductibilité de degré *n :*

- *$\mathcal{CR}(0)$ est l'ensemble des ensembles de termes $\mathcal{C}$ vérifiant les trois conditions* **CR 1**, **CR 2** *et* **CR 3** *déjà énoncées (définition 3.3)*
- *$\mathcal{CR}(n+1)$ est l'ensemble des applications f de Λ vers $\mathcal{CR}(n)$ vérifiant :*

$$\forall t, t'. t =_\beta t' \Rightarrow f(t) = f(t').$$

On a déjà vu que $\mathcal{SN}$ était un candidat de réductibilité d'arité 0. On peut généraliser ceci en définissant pour tout n un élément canonique $Can(n)$ de $\mathcal{CR}(n)$:

- $Can(0) \equiv \mathcal{SN}$
- $Can(n+1)$ est l'application qui à tout λ-terme associe $Can(n)$.

Cette remarque nous permet en particulier de quantifier sur l'ensemble $\mathcal{CR}(n)$, qui est donc non vide.

On veut maintenant définir les interprétations $\mathcal{I}$ et les valuations $|T|_\mathcal{I}$. Mais pour cela, il nous faut d'abord nous assurer que $\mathcal{I}$ interprétera chaque variable α par une valuation de la bonne arité. Pour cela, nous utilisons une notion de prédicats *bien-construits*.

Définition 3.8 (Arité-interprétation) *Une arité-interprétation est une application $\mathcal{A}$ qui associe un entier naturel $\mathcal{A}(\alpha)$ à toute variable de prédicat. On désigne par $\mathcal{A}; \alpha \leftarrow n$ l'arité-interprétation valant n en α et égale à $\mathcal{A}$ partout ailleurs.*

Définition 3.9 (Bonne-construction) *Soit T un prédicat et $\mathcal{A}$ une arité-interprétation. On dit que T est bien construit d'arité n sous $\mathcal{A}$ si :*

- *$T = \alpha$ et $\mathcal{A}(\alpha) = n$*
- *$T = (x:T_1)T_2$, $n = 0$ et T_1 et T_2 sont bien construits d'arité 0 sous $\mathcal{A}$.*
- *$T = (\alpha : K)T_0$, $n = 0$, et T_0 est bien construit d'arité 0 sous $\mathcal{A}; \alpha \leftarrow \mathsf{Ar}(K)$.*

- $T = (T_0\ t)$ *et* T_0 *est bien-construit d'arité* $n+1$ *sous* $\mathcal{A}$.
- $T = [x : T_1]T_2$ *et* T_2 *est bien-construit d'arité* $n-1$ *sous* $\mathcal{A}$.

Si un prédicat est bien-construit d'arité n, on peut alors l'interpréter par un candidat de réductibilité d'arité n, à condition de disposer de valuations de la bonne arité pour toute variable de prédicat.

Définition 3.10 (Interprétation) *Une interprétation* $\mathcal{I}$ *est un application qui :*

- *associe un* λ-terme pur $\mathcal{I}(x)$ *à toute variable de preuve* x.
- *associe un couple* $\mathcal{I}(\alpha) = (n, \mathcal{C})$ *à toute variable de prédicat* α, *où* n *est un entier naturel et* $\mathcal{C}$ *un candidat de réductibilité d'arité* n.

Toute interprétation définit évidemment une arité-interprétation associée. Étant donné un terme preuve t, *on désigne par* $t[\mathcal{I}]$ *le terme pur* $\|t\|$ *dont toutes les variables libres ont été substitués par leurs valuation* $\mathcal{I}(x)$. *Par exemple :*

$$[\alpha : K][x : T \to T](x\ y)[\mathcal{I}] = \lambda x.(x\ \mathcal{I}(y)).$$

Définition 3.11 *Soit* $\mathcal{I}$ *une interprétation et* T *une prédicat bien-construit d'arité* n *sous l'arité-interprétation associée à* $\mathcal{I}$. *On définit alors* $|T|_{\mathcal{I}}$, *candidat de réductibilité d'arité* n *par récurrence sur la structure de* T *:*

- $|\alpha|_{\mathcal{I}} \equiv \mathcal{C}$ *avec* $\mathcal{I}(\alpha) = (n, \mathcal{C})$
- $|(T\ t)|_{\mathcal{I}} \equiv |T|_{\mathcal{I}}(t[\mathcal{I}])$
- $|[x : T_1]T_2|_{\mathcal{I}}$ *est l'application qui à tout terme preuve* u *associe* $|T_2|_{\mathcal{I};x\leftarrow u}$
- $|(\alpha : K)T|_{\mathcal{I}} \equiv \bigcap_{\mathcal{C}\in\mathcal{CR}(\mathsf{Ar}(K))} |T|_{\mathcal{I};\alpha\leftarrow(\mathsf{Ar}(K),\mathcal{C})}$
- $|(x : T_1)T_2|_{\mathcal{I}}$ *est l'ensemble des* λ*-termes purs* u, *tel que pour tout* λ*-terme* $v \in |T_1|_{\mathcal{I}}$, *on a* $(u\ v) \in |T_2|_{\mathcal{I};x\leftarrow v}$

Il faut souligner que la définition de $|(\alpha : K)T|_{\mathcal{I}}$ est simplement la généralisation de la définition de $|\forall\alpha.T|_{\mathcal{I}}$ utilisée dans le cadre du système F, alors que la définition de $|(x : T_1)T_2|_{\mathcal{I}}$ correspond à la définition de $|A \to B|_{\mathcal{I}}$ utilisée jusqu'à maintenant, à la nuance près que l'interprétation de T_2 dépend maintenant de la valeur de x.

Pour que la définition ci-dessus soit rigoureuse, il nous faut vérifier que $|T|_{\mathcal{I}}$ est effectivement toujours un candidat de réductibilité de la bonne arité. La preuve n'est pas difficile ; il faut juste renforcer légèrement l'hypothèse de récurrence.

PREUVE En même temps que l'on définit $|T|_{\mathcal{I}}$ par récurrence sur la structure de T, on prouve également les deux hypothèses de récurrence :

- Si T est bien construit d'ordre n sous $\mathcal{I}$, alors $|T|_{\mathcal{I}}$ est un candidat de réductibilité d'ordre n
- Soit $\mathcal{I}'$ une interprétation telle que $\mathcal{I}(\alpha) = \mathcal{I}'(\alpha)$ pour tout α et $\mathcal{I}(x) =_\beta \mathcal{I}'(x)$ pour tout x. Alors $|T|_{\mathcal{I}} = |T|_{\mathcal{I}'}$.

Le cas $T = \alpha$ est immédiat et $T = (T'\ t)$ et très facile ainsi que $T = [\alpha : K]T'$. On détaille un peu plus les autres cas :

- $T = [x : T_1]T_2$. Pour tout terme pur u, $|T_2|_{\mathcal{I};x\leftarrow u}$ est un candidat de réductibilité par hypothèse de récurrence. De plus si $u =_\beta u'$, alors $|T_2|_{\mathcal{I};x\leftarrow u} = |T_2|_{\mathcal{I};x\leftarrow u'} = |T_2|_{\mathcal{I}';x\leftarrow u}$.

- $T = (\alpha : K)T'$. On sait alors que $n = 0$, et $|T|_\mathcal{I}$ étant défini comme une intersection de candidats est bien un candidat. La preuve de l'autre hypothèse de récurrence est immédiate.

- $T = (x : T_1)T_2$. Ce cas est similaire au précédent. Il faut toutefois adapter le lemme 3.6 au cas des types dépendants.

■

On vérifie par ailleurs facilement que tout prédicat bien-formé est bien-construit. Il faut juste auparavant définir une adéquation entre contextes et arité-interprétations :

Définition 3.12 (arité-interprétation adaptée) *Une arité-interprétation $\mathcal{A}$ est adaptée à un contexte Γ, si et seulement si pour tout couple (α, K) apparaissant dans Γ, on a $\mathcal{A}(\alpha) = \mathsf{Ar}(K)$.*

Lemme 3.11 *Soit un jugement dérivable $\Gamma \vdash T : K$ et $\mathcal{A}$ une arité-interprétation adaptée à Γ. Alors T est bien construit d'arité $\mathsf{Ar}(K)$ sous $\mathcal{A}$.*

PREUVE Par récurrence sur la structure de la dérivation du jugement. ■

À partir de là, on définit une adéquation entre contextes et interprétations. C'est la notion *d'interprétation adaptée* 3.1, revue et corrigée dans le cadre de systèmes avec types dépendants.

Définition 3.13 (interprétation adaptée) *Soit Γ un contexte bien-formé (c.à.d. tel qu'il existe un jugement $\Gamma \vdash J$ dérivable). Une interprétation $\mathcal{I}$ est adaptée à Γ, si :*

- *l'arité-interprétation correspondant par $\mathcal{I}$ est adaptée à Γ*

- *pour tout couple (x, T) apparaissant dans Γ, on a $\mathcal{I}(x) \in |T|_\mathcal{I}$.*

Pour s'assurer que cette définition est correcte, il faudrait vérifier les propriétés suivantes :

- si un contexte $\Gamma :: (x, T)\Gamma'$ est bien formé, alors $\Gamma \vdash T : \mathrm{Set}$.
- si $\mathcal{A}$ est adaptée à Γ, alors $\mathcal{A}$ est adaptée à toute sous-partie de Γ.

La seconde propriété ne pose aucun problème. La première est facile à vérifier par récurrence sur la structure de la preuve de bonne-formation de Γ (elle est faite dans le cadre de CCI dans le chapitre précédent).

Nous avons maintenant construit tous les outils nécessaires pour vérifier la normalisation. Il nous reste un point très important à traiter : ce système se distingue des précédents par la présence de la règle de *conversion* :

$$(\textsc{Conv})\ \frac{\Gamma \vdash t : T_1 \qquad \Gamma \vdash T_1 : \mathrm{Set} \qquad T_1 =_\beta T_2 \qquad \Gamma \vdash T_2 : \mathrm{Set}}{\Gamma \vdash t : T_2}$$

Lorsque nous prouverons que $t : T \Rightarrow t[\mathcal{I}] \in |T|_\mathcal{I}$, il nous faudra traiter le cas de cette règle. On utilisera alors la propriété que l'interprétation des prédicats est *invariante par conversion.*

Lemme 3.12 (invariance par conversion) *Soit T un prédicat bien construit d'ordre n sous une-interprétation $\mathcal{I}$. Si $T \triangleright_\beta T'$ alors T' est bien-construit d'ordre n sous $\mathcal{I}$, et $|T|_\mathcal{I} = |T'|_\mathcal{I}$.*

Cette propriété est la clé de l'adaptation de la méthode de réductibilité à des systèmes avec types dépendants. C'est elle qui nous posera le plus de difficultés dans des systèmes plus complexes, comme dans la partie suivante et surtout pour le Calcul des Constructions Inductives dans le chapitre suivant. Toutefois dans le cadre du système $\lambda P2$, la preuve est facile :
PREUVE Par récurrence sur la structure de T. On ne détaille que les cas intéressants où T est le radical réduit ou bien la réduction a lieu dans un terme preuve.

- Si $T = ([x : T_1]T_2\ t)$ et $T' = T_2[x \setminus T_1]$, c'est une conséquence immédiate de la propriété $|T_2[x \setminus t]|_{\mathcal{I}} = |T_2|_{\mathcal{I};x \leftarrow |t|_{\mathcal{I}}}$.
- Si $T = ([\alpha : K]T_1\ T_2)$ et $T' = T_1[\alpha \setminus T_2]$, c'est une conséquence immédiate de la propriété $|T_1[\alpha \setminus T_2]|_{\mathcal{I}} = |T_1|_{\mathcal{I};\alpha \leftarrow |T_2|_{\mathcal{I}}}$.
- Si $T = (T_1\ t)$ et $T' = (T_1\ t')$ avec $t \rhd t'$, l'hypothèse de récurrence nous assure que $|T_1|_{\mathcal{I}}$ est un candidat de réductibilité. Aussi comme $|t|_{\mathcal{I}} =_\beta |t'|_{\mathcal{I}}$, on a bien $|T_1|_{\mathcal{I}}(|t|_{\mathcal{I}}) = |T_1|_{\mathcal{I}}(|t'|_{\mathcal{I}})$.

■

Nous pouvons maintenant passer à la preuve de normalisation proprement dite :

Théorème 6 *Soit $\Gamma \vdash t : T$ un jugement dérivable et $\mathcal{I}$ une interprétation adaptée à Γ ; on a alors $t[\mathcal{I}] \in |T|_{\mathcal{I}}$.*

La preuve est une récurrence sur la structure de la dérivation du jugement, sur le modèle de ce qui a été fait précédemment. Il ne semble pas nécessaire de détailler.

Corollaire 3.1 *Si le jugement $\Gamma \vdash t : T$ est dérivable dans $\lambda P2$, alors $\|t\|$ est fortement normalisable.*

3.7 Types définis par récurrence

Il nous reste un point important à illustrer avant de conclure ce chapitre. C'est celui des types définis par récurrence sur des objets inductifs, c'est-à-dire la normalisation de systèmes comportant ce que nous avons appelé des opérateurs d'élimination forte. Il va être facile d'étendre le système $\lambda P2$ en ce sens, puisqu'il possède déjà des types dépendants. Nous considérons les deux types inductifs primitifs déjà étudiés précédemment : les entiers naturels et les ordinaux. L'ensemble des prédicats de $\lambda P2$ est donc étendu par les objets Nat et Ord, et on se donne les nouvelles règles d'inférence suivantes :

$$\boxed{[] \vdash \mathrm{Nat} : \mathrm{Set} \qquad [] \vdash \mathrm{Ord} : \mathrm{Set}}$$

De même l'algebre des *termes de preuves* est étendue par les constructeurs et les schémas d'élimination déja utilisés dans la partie 3.4. Les termes de preuves sont donc maintenant définis par :

$$\boxed{t \quad ::= \quad x \mid [x : T]t \mid (t\ t) \mid [\alpha : K]t \mid (t\ T)}$$

Les règles d'introduction des constructeurs restent essentiellement inchangées. Par contre, et comme indiqué dans le chapitre 1, nous pouvons maintenant typer de manière plus précise les schémas d'élimination :

$$\begin{array}{|c|}\hline
(\textsc{0-N-Intro})\ \ [] \vdash 0 : \mathrm{Nat} \qquad (\textsc{S-N-Intro})\ \ [] \vdash S : \mathrm{Nat} \rightarrow \mathrm{Nat} \\
(\textsc{0-O-Intro})\ \ [] \vdash 0_o : \mathrm{Ord} \qquad (\textsc{S-O-Intro})\ \ [] \vdash S_o : \mathrm{Ord} \rightarrow \mathrm{Ord} \\
(\textsc{lim-Intro})\ \ [] \vdash \lim : (\mathrm{Nat} \rightarrow \mathrm{Ord}) \rightarrow \mathrm{Ord} \\
(\textsc{Nat-Elim})\ \dfrac{\begin{array}{c}\Gamma \vdash T : \mathrm{Nat} \rightarrow \mathrm{Set} \qquad \Gamma \vdash t : \mathrm{Nat} \qquad \Gamma \vdash t_0 : (T\ 0) \\ \Gamma \vdash t_S : (m : \mathrm{Nat})(T\ m) \rightarrow (T\ (S\ m))\end{array}}{\Gamma \vdash (R_N\ t\ t_0\ t_S) : (T\ nt)} \\
(\textsc{Ord-Elim})\ \dfrac{\begin{array}{c}\Gamma \vdash T : \mathrm{Ord} \rightarrow \mathrm{Set} \qquad \Gamma \vdash t : \mathrm{Ord} \qquad \Gamma \vdash t_0 : (T\ 0_o) \\ \Gamma \vdash t_S : (m : \mathrm{Ord})(T\ m) \rightarrow (T\ (S_o\ m)) \\ \Gamma \vdash t_l : (f : \mathrm{Nat} \rightarrow \mathrm{Ord})((n : \mathrm{Nat})(T\ (f\ n))) \rightarrow (T\ (\lim\ f))\end{array}}{\Gamma \vdash (R_O\ t\ t_0\ t_S\ t_l) : (T\ t)} \\ \hline
\end{array}$$

Les règles de réduction des opérateurs R_N et R_O sont les mêmes que celles utilisées dans la section 3.4. En particulier, elles étendent la règle de conversion. Comme nous utilisons des *termes purs*, non-typés, dans notre preuve de normalisation, il nous faut là aussi tenir compte des constructeurs de schémas d'élimination. Les termes purs sont donc maintenant définis par :

$$\boxed{t \ := \ x \mid \lambda x.t \mid (t\ t) \mid 0 \mid S \mid 0_O \mid S_O \mid \ \lim \ \mid R_N \mid R_O}$$

et l'opération d'oubli est bien sûr étendue par :

$$\begin{array}{rcl}
\|0\| & \equiv & 0 \\
\|S\| & \equiv & S \\
\|R_N\| & \equiv & R_N \\
& \text{etc} &
\end{array}$$

avec les règles de réduction maintenant habituelles sur les termes purs.

À ce stade, nous avons défini une extension cohérente de $\lambda P2$. Il serait en revanche peu interessant d'étudier sa normalisation : il suffit littéralement de combiner la preuve du système T (définitions de $|\mathrm{Nat}|$ et $|\mathrm{Ord}|$) avec celle de $\lambda P2$. Nous commençons donc par introduire les opérateurs d'élimination forte. Comme indiqué dans le chapitre 2, ces derniers sont construits sur le modèle des opérateurs d'élimination habituels, mais permettent de construire des types. Nous appellerons respectivement Rec et Rec_O les opérateurs d'élimination forte des naturels et des ordinaux. Les prédicats sont finalement décrits par :

$$\begin{array}{|lll|}\hline
T := & \alpha & \mid [x : T]T \mid (T\ t) \mid (\alpha : K)T \mid (x : T_1)T_2 \\
& & \mid \mathrm{Nat} \mid \mathrm{Ord} \mid \mathsf{Rec}(t, T, \alpha.T) \mid \mathsf{Rec}_O(t, T, \alpha.T, \alpha.T) \\ \hline
\end{array}$$

Remarquons que Rec et Rec_O lient des variables dans certains de leurs sous-termes. Par exemple dans le terme $\mathsf{Rec}(t, T_0, \alpha.T_S)$, la variable de prédicat α est liée dans le sous-terme T_S, avec les conséquences habituelles pour la substitution. Les règles de réduction sur les prédicats de la section précédente sont étendues par :

$$\begin{aligned}
&\mathsf{Rec}_N(0, T_0, \alpha.T_S) \rhd_\iota T_0\\
&\mathsf{Rec}_N((S\ t), T_0, \alpha.T_S) \rhd_\iota (T_S[\alpha \setminus \mathsf{Rec}_N(t, T_0, \alpha.T_S)]\ t)\\
&\mathsf{Rec}_O(0_O, T_0, \alpha.T_S, \beta.T_l) \rhd_\iota T_0\\
&\mathsf{Rec}_O((S\ t), T_0, \alpha.T_S, \beta.T_l) \rhd_\iota (T_S[\alpha \setminus \mathsf{Rec}(t, T_0, \alpha.T_S, \beta.T_l)]\ t)\\
&\mathsf{Rec}_O((\lim\ t), T_0, \alpha.T_S, \beta.T_l) \rhd_\iota (T_l[\beta \setminus [x : \mathrm{Nat}]\mathsf{Rec}((t\ x), T_0, \alpha.T_S, \beta.T_l)]\ t)
\end{aligned}$$

Remarque On voit que les comportements calculatoires respectifs de Rec et Rec_O sont essentiellement identiques à ceux de R et R_O. Les différences sont simplement dues à l'impossibilité, dans ce système, de construire des fonctions des prédicats vers les prédicats : $[\alpha : K]T$ est interdit. Dans le Calcul des Constructions étendu par Extern, cette restriction n'existe pas, et les opérateurs d'élimination faible et forte se comportent de manière identique.

Passons enfin aux règles d'inférence correspondant aux schémas d'élimination forte. On simplifie la présentation du système en autorisant uniquement la construction d'objets de type Set par élimination forte, contrairement aux Calcul des Construction Inductives, dans lequel il est possible de construire des prédicats d'ordre arbitraire.

$$(\textsc{Nat-S-Elim})\ \frac{\Gamma \vdash t : \mathrm{Nat} \qquad \Gamma \vdash T_0 : \mathrm{Set} \qquad \Gamma :: (\alpha, \mathrm{Set}) \vdash T_S : \mathrm{Nat} \to \mathrm{Set}}{\Gamma \vdash \mathsf{Rec}_N(t, T_0, \alpha.T_S) : \mathrm{Set}}$$

$$(\textsc{Ord-S-Elim})\ \frac{\begin{array}{c}\Gamma \vdash t : \mathrm{Ord} \qquad \Gamma \vdash T_0 : \mathrm{Set} \qquad \Gamma :: (\alpha, \mathrm{Set}) \vdash T_S : \mathrm{Ord} \to \mathrm{Set}\\ \Gamma :: (\beta, \mathrm{Nat} \to \mathrm{Set}) \vdash T_l : (\mathrm{Nat} \to \mathrm{Ord}) \to \mathrm{Set}\end{array}}{\Gamma \vdash \mathsf{Rec}_O(t, T_0, \alpha.T_S, \beta.T_l) : \mathrm{Set}}$$

On renvoie par exemple au chapitre 1 pour les exemples d'utilisation de cette règle.

3.7.1 La preuve de normalisation

On aura compris que celle-ci reprend la plupart des éléments de la preuve de normalisation de $\lambda P2$, et que la nouveauté essentielle est l'interprétation des types définis par élimination forte. À ce stade, il n'est sans doute pas inutile de redonner la définition précise des candidats de réductibilité, combinaison de 3.2 et de 3.7 :

Définition 3.14 *Un terme pur est neutre (c'est-à-dire élément de $\mathcal{NT}$) s'il n'est pas de la forme :*

$$\lambda x.t \quad 0 \quad (S\ t) \quad 0_O \quad (S_O\ t) \quad (\lim\ t).$$

Définition 3.15 *Pour tout entier naturel n, on définit $\mathcal{CR}(n) \subset \rho(n)$, l'ensemble des* candidats de réductibilité de degré *n :*

- *$\mathcal{CR}(0)$ est l'ensemble des ensembles de termes $\mathcal{C}$ vérifiant les trois conditions* **CR 1**, **CR 2** *et* **CR 3** *déjà énoncées (définition 3.3)*

- *$\mathcal{CR}(n+1)$ est l'ensemble des applications f de Λ vers $\mathcal{CR}(n)$ vérifiant :*

$$\forall t, t'.t =_{\beta\iota} t' \Rightarrow f(t) = f(t').$$

Les notions d'arité-interprétation 3.8 et d'interprétation 3.10 restent inchangées. Comme nous avons restreint l'élimination forte aux prédicats d'arité 0, il est facile d'étendre la notion de bonne-construction :

Définition 3.16

- *Le prédicat $\mathsf{Rec}_N(t, T_0, \alpha.T_S)$ est bien construit d'arité n sous une arité-interprétation $\mathcal{A}$ si et seulement si $n = 0$, T_0 est bien construit d'arité 0 sous $\mathcal{A}$ et T_S est bien construit d'arité 1 sous $\mathcal{A}; \alpha \leftarrow 0$.*

- *Le prédicat $\mathsf{Rec}_O(t, T_0, \alpha.T_S, \beta.T_l)$ est bien construit d'arité n sous une arité-interprétation $\mathcal{A}$ si et seulement si $n = 0$, T_0 est bien construit d'arité 0 sous $\mathcal{A}$, T_S est bien construit d'arité 1 sous $\mathcal{A}; \alpha \leftarrow 0$ et T_l est bien construit d'arité 1 sous $\mathcal{A}; \beta \leftarrow 1$.*

Les autres clauses définissant la bonne-construction restent inchangées par rapport à 3.9.

On vérifie comme précédemment que si $\Gamma \vdash T : K$ est dérivable, et si $\mathcal{A}$ est une arité interprétation adaptée à Γ, alors T est bien-construit d'arité $\mathsf{Ar}(K)$ sous $\mathcal{A}$.

Nous arrivons maintenant au point crucial : la définition de $|T|_\mathcal{I}$. Les définitions de $|\alpha|_\mathcal{I}, |(x : T_1)T_2|_\mathcal{I}, |(\alpha : K)T_1|_\mathcal{I}, |(T\ t)|_\mathcal{I}$ et $|[x : T_1]T_2|_\mathcal{I}$ sont bien sûr identiques à celles données en 3.11. Les ensembles $|\mathrm{Nat}|_\mathcal{I}$ et $|\mathrm{Ord}|_\mathcal{I}$ sont identiques à $|\mathrm{Nat}|$ et $|\mathrm{Ord}|$ définis dans la partie 3.4 (ils ne dépendent donc pas de $\mathcal{I}$). La difficulté réside bien sûr dans les définitions de $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_\mathcal{I}$ et $|\mathsf{Rec}_O(t, T_0, \alpha.T_S, \beta.T_l)|_\mathcal{I}$. Dans un premier temps nous détaillerons seulement le cas des entiers (interprétation de Rec). La difficulté réside dans le traitement de la règle de conversion ; on peut dire que nous devons respecter les "spécifications" suivantes :

- Si $\Gamma \vdash T_0 : \mathrm{Set}$, $\Gamma :: (\alpha, \mathrm{Set}) \vdash T_S : \mathrm{Nat} \to \mathrm{Set}$, et si $\mathcal{I}$ est adaptée à Γ, alors

  $$|\mathsf{Rec}_N(0, T_0, \alpha.T_S)|_\mathcal{I} = |T_0|_\mathcal{I}.$$

  Plus généralement, si $\Gamma \vdash t : \mathrm{Nat}$ et $|t|_\mathcal{I} \rhd^* 0$, alors

  $$|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_\mathcal{I} = |T_0|_\mathcal{I}.$$

- Si de plus $\Gamma \vdash t : \mathrm{Nat}$, alors

  $$|\mathsf{Rec}_N((S\ t), T_0, \alpha.T_S)|_\mathcal{I} = |(T_S[\alpha \setminus \mathsf{Rec}_N(t, T_0, \alpha.T_S)]\ t)|_\mathcal{I}$$

Comme le deuxième cas est équivalent à

$$|\mathsf{Rec}_N((S\ t), T_0, \alpha.T_S)|_\mathcal{I} = |T_S|_{\mathcal{I};\alpha\leftarrow|\mathsf{Rec}_N(t,T_0,\alpha.T_S)|_\mathcal{I}}(|t|_\mathcal{I})$$

on peut voir ces égalités comme une définition récursive de l'application qui à $|t|_\mathcal{I}$ et $\mathcal{I}$ associe $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_\mathcal{I}$. Il nous reste à déterminer :

1. dans quel cas cette définition est bien fondée
2. que faire dans les cas où $|t|_{\mathcal{I}}$ ne se réduit ni vers 0 ni vers $(S\ u)$.

La réponse à la deuxième question est simple : si $|t|_{\mathcal{I}}$ ne se réduit pas vers une forme de constructeur, on peut considérer que $\mathsf{Rec}_N(t, T_0, \alpha.T_S)$ est un type atomique, et prendre pour $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}}$ un candidat de réductibilité arbitraire (typiquement $\boldsymbol{\mathcal{SN}}$).

Pour ce qui est de la première question, on voit que la récurrence considérée est bien fondée, si $|t|_{\mathcal{I}}$ est élément de $|\mathrm{Nat}|_{\mathcal{I}}$ (respectivement $|\mathrm{Ord}|_{\mathcal{I}}$ si on considère Rec_O). Dans le cas des entiers naturels tout d'abord. Supposons que $|t|_{\mathcal{I}}$ admette une forme normale u. Cette forme normale est bien sûr élément de $|\mathrm{Nat}|_{\mathcal{I}}$ qui est égal à $\boldsymbol{\mathcal{SN}}$. On définit alors formellement la fonction f qui à une interprétation $\mathcal{I}$ et un terme normal v associe un ensemble de termes :

$$\begin{array}{rcl} f(\mathcal{I}, 0) & \equiv & |T_0|_{\mathcal{I}} \\ f(\mathcal{I}, (S\ v)) & \equiv & |T_S|_{\mathcal{I};\alpha \leftarrow f(\mathcal{I},v)}(v) \\ f(\mathcal{I}, v) & \equiv & \boldsymbol{\mathcal{SN}} \text{ si n'est pas de la forme 0 ou } (S\ v). \end{array}$$

Et on définit alors :

Définition 3.17 *Si $|t|_{\mathcal{I}} \in \boldsymbol{\mathcal{SN}}$, et si u est sa forme normale, alors $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}}$ est défini comme $f(\mathcal{I}, u)$.*

Il nous reste à décider de la valeur de $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}}$ dans le cas où $|t|_{\mathcal{I}}$ n'est pas normalisable. On sait qu'ultimement, si $\mathsf{Rec}_N(t, T_0, \alpha.T_S)$ est bien formé, et si $\mathcal{I}$ est une interprétation adaptée au contexte correspondant, ceci ne sera jamais le cas. Mais nous ne pouvons, bien sûr, pas encore invoquer cet argument.

Un problème similaire apparaît dans les preuves de normalisation de système de types de Martin-Löf avec univers. La solution adoptée par Per Martin-Löf lui-même [97] et par Catarina Coquand [139], est de ne pas définir $|T|_{\mathcal{I}}$ dans les cas correspondants. Il faut alors également définir le domaine sur lequel l'interprétation est définie. Ensuite on prouve que $|T|_{\mathcal{I}}$ est effectivement définie si T est bien typé et $\mathcal{I}$ vérifie certaines conditions bien-choisies.

La solution que nous proposons est légèrement différente : $|T|_{\mathcal{I}}$ est toujours défini. Dans le cas $|t|_{\mathcal{I}}$ on assigne à $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}}$ une valeur arbitraire :

Définition 3.17 (suite) *Si $|t|_{\mathcal{I}}$ n'est pas normalisable, alors :*

$$|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}} \equiv \boldsymbol{\mathcal{SN}}.$$

On vérifie alors facilement que $|T|_{\mathcal{I}}$ et toujours un candidat de réductibilité.

Nous simplifions ainsi la définition de l'interprétation, mais ne faisons essentiellement que repousser le problème. En effet, il n'est maintenant plus vrai que si $T =_\beta T'$ alors $|T|_{\mathcal{I}} = |T'|_{\mathcal{I}}$. Considérons par exemple :

$$\begin{array}{rcl} T & \equiv & [x : \mathrm{Nat}]\mathsf{Rec}((S\ x), T_0, \alpha.[y : nat]T_1) \\ T' & \equiv & [x : nat]T_1 \end{array}$$

où T_1 est un type ne dépendant ni de x ni de y. Il est évident que $T \triangleright T'$. Pourtant $|T'|_{\mathcal{I}}$ est l'application constante qui à tout terme pur associe $|T_1|_{\mathcal{I}}$, alors que si on applique un terme pur non-normalisable u à $|T|_{\mathcal{I}}$ on obtient $\mathcal{SN}$ (à cause de la définition précédente et parce que $(S\ u)$ est non-normalisable). Or il est clair qu'il n'y a aucune raison particulière pour que $|T_1|_{\mathcal{I}} = \mathcal{SN}$. Par contre, pour tout terme pur $v \in |\mathrm{Nat}|_{\mathcal{I}}$, on aura bien :

$$|T|_{\mathcal{I}}(v) = |T'|_{\mathcal{I}}(v) = |T_1|_{\mathcal{I}}.$$

Dans le cadre du système étudié ici, la solution la plus simple et la plus élégante m'a été indiquée par Milena Stefanova. Il suffit de modifier l'interprétation de la λ-abstraction des types dépendants et de prendre :

$$\begin{aligned} |[x:T_0]T|_{\mathcal{I}} &\equiv u \mapsto |T|_{\mathcal{I};x\leftarrow u} \text{ pour } u \in |T_0|_{\mathcal{I}} \\ &\equiv u \mapsto |T|_{\mathcal{I};x\leftarrow x} \text{ pour } u \notin |T_0|_{\mathcal{I}}. \end{aligned}$$

On ne change pas ainsi les autres étapes de la preuve, et on retrouve bien, pour tous prédicats bien formés T et T' la propriété

$$T =_\beta T' \Rightarrow |T|_{\mathcal{I}} = |T'|_{\mathcal{I}}.$$

Malheureusement, il semble que cette modification ne soit pas compatible avec la η-réduction pour des raisons évidentes. Comme notre but ici est d'illustrer les techniques utilisées dans le chapitre suivant, nous conservons notre définition première de l'interprétation des prédicats, et introduisons la propriété pour un prédicat d'être "invariant sur son domaine". Il s'agit simplement de formaliser le fait que lorsque l'on applique à un prédicat uniquement des termes qui sont éléments des interprétations de son domaine de quantification, on obtient un type dont l'interprétation est invariante par réduction. Cette propriété, qui rappelle beaucoup la réductibilité, sera prouvée lors de la récurrence finale.

Définition 3.18 *Soit un jugement dérivable $\Gamma \vdash K$ et une interprétation $\mathcal{I}$ adaptée à Γ. On définit $dom(K,\Gamma,\mathcal{I})$ comme un ensemble de $\mathsf{Ar}(K)$-uplets de termes purs par récurrence sur la structure de K :*

- *si $K = \mathrm{Set}$, alors $dom(K,\Gamma,\mathcal{I})$ est l'unique 0-uplet*
- *si $K = (x : T)K'$, alors $(u, u_1, \ \dots \ , u_n)$ est élément de $dom(K,\Gamma,\mathcal{I})$ si et seulement si $u \in |T|_{\mathcal{I}}$, et $(u_1, \ \dots \ , u_n) \in dom(K, \Gamma :: (x,T), \mathcal{I}; x \leftarrow u)$.* [3]

Définition 3.19 (invariant sur son domaine) *Soit un jugement dérivable $\Gamma \vdash T : K$ et une interprétation $\mathcal{I}$ adaptée à Γ. On dit que $(T,K,\Gamma,\mathcal{I})$ est invariant sur son domaine si pour tout $\mathcal{A} \in dom(K,\Gamma,\mathcal{I})$, si $T \rhd^* T'$, alors*

$$|T|_{\mathcal{I}}(\mathcal{A}) = |T'|_{\mathcal{I}}(\mathcal{A}).$$

En particulier si $K = \mathrm{Set}$ alors $|T|_{\mathcal{I}} = |T'|_{\mathcal{I}}$.

Nous sommes maintenant parés pour la preuve finale :

Théorème 7 *Pour tout jugement dérivable $\Gamma \vdash t : T$, $\|t\|$ est fortement normalisable.*

[3] Pour être tout à fait rigoureux, il faudrait en fait prouver que si $\Gamma \vdash (x : T)K$ est dérivable, alors $\Gamma \vdash T : \mathrm{Set}$ et $\Gamma :: (x,T) \vdash K$ le sont aussi. Ce résultat est en fait démontré dans le chapitre 2 dans un cadre plus général, et sa preuve nous intéresse peu ici.

PREUVE On reste bien sûr fidèle à la récurrence sur la structure de la dérivation. L'hypothèse de récurrence étant :

- si $\Gamma \vdash t : T$ est dérivable, pour toute interprétation $\mathcal{I}$ adaptée à Γ, on a $|t|_{\mathcal{I}} \in |T|_{\mathcal{I}}$
- si $\Gamma \vdash T : K$ est dérivable, pour toute interprétation $\mathcal{I}$ adaptée à Γ, $(T, K, \Gamma, \mathcal{I})$ est invariant sur son domaine.

On ne détaille que les cas des règles correspondant au second cas, ainsi que la règle de conversion :

- La règle VAR2. C'est immédiat, puisque α est en forme normale.
- Les règles WEAK1 et WEAK2. Il suffit de remarquer que si x (respectivement α) n'est pas libre dans K, alors $\mathrm{dom}(K, \Gamma, \mathcal{I}) = \mathrm{dom}(K, \Gamma :: (x, T_1), \mathcal{I})$ (respectivement $\mathrm{dom}(K, \Gamma :: (\alpha, K_1), \mathcal{I})$).
- La règle T-APP. La conclusion de la règle est $\Gamma \vdash (T\ t) : K[x \setminus t]$. On sait par hypothèse de récurrence que $|t|_{\mathcal{I}} \in |T_1|_{\mathcal{I}}$ et que $(T, (x : T_1)K, \Gamma, \mathcal{I})$ est invariant sur son domaine. Alors pour tout $\mathcal{A} \in \mathrm{dom}(K[x \setminus t], \Gamma, \mathcal{I})$ on a $(|t|_{\mathcal{I}}, \mathcal{A}) \in \mathrm{dom}((x : T)K, \Gamma, \mathcal{I})$. Le résultat suit.
- La règle T-LAM. La conclusion est $\Gamma \vdash [x : T_1]T : (x : T_1)K$. Si $T \rhd^* T_0$, alors $T' = [x : T_1']T'$. Soit $(u_1, u_2, \ \ldots \ , u_n) \in \mathrm{dom}((x : T_1)K, \Gamma, \mathcal{I})$; alors $u_1 \in |T_1|_{\mathcal{I}}$ et donc $(T, K, \Gamma :: (x, T_1), \mathcal{I}; x \leftarrow u_1)$ est invariant sur son domaine par hypothèse de récurrence. Comme $(u_2, \ \ldots \ , u_n) \in \mathrm{dom}(K, \Gamma :: (x, T_1), \mathcal{I}; x \leftarrow u_1)$, on en déduit :

$$|T|_{I;x\leftarrow u_1}(u_2, \ \ldots \ , u_n) = |T'|_{\mathcal{I};x\leftarrow u_1}(u_2, \ \ldots \ , u_n)$$

  c'est-à-dire aussi

$$|[x : T_1]T|_{\mathcal{I}}(u_1, u_2, \ \ldots \ , u_n) = |[x : T_1']T'|_{\mathcal{I}}(u_1, u_2, \ \ldots \ , u_n).$$

- La règle PROD1. La conclusion est $\Gamma \vdash (x : T_1)T_2 : \mathrm{Set}$. Les réduits de $(x : T_1)T_2$ sont forcément de la forme $(x : T_1')T_2'$. Par hypothèse de récurrence on sait alors que $|T_1|_{\mathcal{I}} = |T_1'|_{\mathcal{I}}$, et si $u \in |T_1|_{\mathcal{I}}$, alors $|T_2|_{\mathcal{I};x\leftarrow u} = |T_2'|_{\mathcal{I};x\leftarrow u}$. on en déduit :

$$|(x : T_1)T_2|_{\mathcal{I}} = (x : T_1')T_2'|_{\mathcal{I}}.$$

- La règle PROD2. Ce cas est similaire au précédent.
- La règle NAT-S-ELIM. La conclusion est

$$\Gamma \vdash \mathsf{Rec}_N(t, T_0, \alpha.T_S) : \mathrm{Set}.$$

  On sait par hypothèse de récurrence que $|t|_{\mathcal{I}}$ est fortement normalisable ; on raisonne par récurrence sur sa forme normale.

  – Si $|t|_{\mathcal{I}} \rhd^* 0$, alors on sait que $|\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}} = |T_0|_{\mathcal{I}}$. Comme si $T_0 \rhd^* T_0'$, et que $\mathsf{Rec}_N(t, T_0, \alpha.T_S)$ se réduit soit vers T_0', soit vers $\mathsf{Rec}_N(t', T_0', \alpha.T_S')$, le résultat suit.
  – Si $|t|_{\mathcal{I}} \rhd^* (S\ v)$, les réduits de $\mathsf{Rec}_N(t, T_0, \alpha.T_S)$ sont de la forme :

  * $\mathsf{Rec}_N(t', T_0', \alpha.T_S')$ auquel cas $|t'|_{\mathcal{I}} \rhd^* (S\ u)$. L'hypothèse de récurrence assure alors $|T_0|_{\mathcal{I}} = |T_0'|_{\mathcal{I}}$, et $|\mathsf{Rec}(u, T_0, \alpha.T_S)|_{\mathcal{I}} = |\mathsf{Rec}(u, T_0', \alpha.T_S')|_{\mathcal{I}}$. Le résultat suit.
  * un réduit de $(T_S'[\alpha \setminus T_2]\ t')$, avec $t \rhd^* (S\ t')$ et $\mathsf{Rec}(t', T_0, \alpha.T_S) \rhd^* T_2$. Comme substituer α par T_2 dans T_S', on sait que $((T_S'[\alpha \setminus T_2]\ t'), \mathrm{Set}, \Gamma, \mathcal{I})$ est invariant sur son domaine. Par ailleurs les hypothèses de récurrence assurent que $|(T_S'[\alpha \setminus T_2]\ t')|_{\mathcal{I}} = |\mathsf{Rec}_N(t, T_0, \alpha.T_S)|_{\mathcal{I}}$.

- Le cas de la règle Conv. La conclusion est $\Gamma \vdash t : T_2$. On sait que $|t|_{\mathcal{I}} \in |T_1|_{\mathcal{I}}$. Comme $T_1 =_\beta T_2$, T_1 et T_2 admettent un réduit commun T_3. Par ailleurs l'hypothèse de récurrence assure alors $|T_1|_{\mathcal{I}} = |T_3|_{\mathcal{I}}$ et $|T_2|_{\mathcal{I}} = |T_3|_{\mathcal{I}}$, et donc $|t|_{\mathcal{I}} \in |T_2|_{\mathcal{I}}$.

■

On voit que le travail nécessaire pour prouver l'invariance de $|T|_{\mathcal{I}}$ par réduction est non-trivial. Dans le cas du système étudié ci-dessus, il semble toutefois que la solution présentée ici est plus simple que celle consistant à définir $|T|_{\mathcal{I}}$ de manière partielle. C'est un peu moins vrai pour le Calcul des Constructions Inductives complet : la preuve détaillée dans le chapitre 4 est certainement compliquée à cause de ce problème, et il est possible, mais pas sûr, qu'une preuve plus proche de celle de Martin-Löf soit plus claire.

3.8 Petit bestiaire des Candidats de Réductibilité

Nous avons présenté en 3.4 la notion de *candidats de réductibilité*, puis vu comment cette notion était indispensable pour prouver la normalisation de tout système au moins aussi expressif que F. Depuis l'introduction de cette notion par Jean-Yves Girard, un certain nombre de variantes ont été présentées. Il nous a semblé intéressant, quitte à céder au démon de la taxinomie, de faire le point sur ces définitions alternatives et, à travers ce tour d'horizon, de caractériser un peu plus précisément cette notion.

Dans ce chapitre, on considèrera, à titre d'exemple, le système F. On garde les notations de la partie 3.5, à savoir $\mathcal{I}$ pour les interprétations, $|T|_{\mathcal{I}}$ pour l'interprétation du type T sous $\mathcal{I}$, etc.

3.8.1 Caractérisation des candidats

La classe des candidats de réductibilité est un ensemble d'ensembles de termes. Comme nous l'avons mentionné précédemment, l'interprétation de tout type T du système considéré (c'est-à-dire l'ensemble des termes réductibles de type T) sera un candidat de réductibilité. C'est nécessaire puisque nous voulons être capable de quantifier sur toutes les interprétations possibles de types en quantifiant sur l'ensemble des candidats. C'est l'origine de la dénomination de "candidats de réductibilité" : un candidat est susceptible de correspondre à l'ensemble des termes réductibles d'un certain type.

Pour que cette condition soit remplie, il faut et il suffit que $\mathcal{CR}$, l'ensemble des candidats soit clos pour les différentes clauses de la définition des interprétations des types. Dans le cas du système F il faut que :

1. Si $\mathcal{C}_1$ et $\mathcal{C}_2$ sont des candidats, alors $C_1 \to \mathcal{C}_2$ est un candidat.
2. L'intersection d'un ensemble non-vide de candidats de réductibilité est un candidat de réductibilité.

De plus, pour être capable de quantifier sur tous les candidats de réductibilité, $\mathcal{CR}$ ne doit pas être vide. Comme on veut être capable de construire une interprétation adaptée à tout contexte dans laquelle les variables de termes sont interprétées par elles-même ($\mathcal{I}(x) = x$), on peut écrire :

3. Il existe un candidat de réductibilité.
4. Tout candidat de réductibilité contient toute variable de terme.

Une autre condition moins intéressante est que tout élément d'un candidat de réductibilité est fortement normalisable. En effet, une variable de type α peut-être interprétée par un candidat de réductibilité quelconque, et il nous faut garantir que les éléments de l'interprétation de α vérifient bien la condition que nous voulons prouver ultimement : la normalisation forte. On a donc :

5. $\forall \mathcal{C} \in \mathcal{CR} \ . \ \mathcal{C} \subset \boldsymbol{\mathcal{SN}}$.

L'autre condition clé est à nouveau celle qui nous permettera de traiter le cas de l'abstraction :

$$\frac{\Gamma :: (x, A) \vdash t : B}{\Gamma \vdash \lambda x.t : A \to B}$$

Étant donnée une interprétation $\mathcal{I}$ adaptée à Γ, il est facile de ramener l'hypothèse de récurrence à l'énoncé :

$$\forall u \in |A|_{\mathcal{I}}.t[\mathcal{I}; x \leftarrow u] \in |B|_{\mathcal{I}}.$$

Il nous faut alors prouver que $\lambda x.t[\mathcal{I}] \in |A \to B|$, c'est-à-dire :

$$\forall u \in |A|_{\mathcal{I}}.(\lambda x.t[\mathcal{I}] \ u) \in |B|_{\mathcal{I}}.$$

En d'autres termes il nous faut être capable de déduire $(\lambda x.t[\mathcal{I}] \ u) \in |B|_{\mathcal{I}}$ de $t[\mathcal{I}; x \leftarrow u] \in |B|_{\mathcal{I}}$. Si l'on considère que les seules informations dont nous disposons sont :

- $|A|_{\mathcal{I}}$ et $|B|_{\mathcal{I}}$ sont des candidats de réductibilité,
- $t[\mathcal{I}; x \leftarrow u] \in |B|_{\mathcal{I}}$,
- $u \in |A|_{\mathcal{I}}$,

on peut en déduire la dernière condition que doivent vérifier les candidats :

6. $\forall A, B \in \mathcal{CR}.\forall u \in A.\forall t.t[x \setminus u] \in B \Rightarrow (\lambda x.t \ u) \in B$.

Un ensemble d'ensembles de termes qui vérifie les six conditions ci-dessus peut donc être choisi comme définition des candidats de réductibilité (est candidat à la candidature).

3.8.2 Une preuve de normalisation abstraite

Pour nous convaincre que les six critèreres ci-dessus sont suffisants, nous pouvons refaire la preuve de normalisation forte du système F. Dans cette partie, on supposera donc simplement que l'on dispose d'un ensemble d'ensembles de termes $\mathcal{CR}$ qui vérifie nos six conditions.

Pour toute interprétation condidat $\mathcal{I}$ et tout type T, on définit l'interprétation $|T|_{\mathcal{I}}$, un ensemble de termes, comme en 3.5 par récurrence sur la structure de T :

$$\begin{array}{rcl} |\alpha|_{\mathcal{I}} & \equiv & \mathcal{I}(\alpha) \\ |A \to B|_{\mathcal{I}} & \equiv & |A|_{\mathcal{I}} \to |B|_{\mathcal{I}} \\ |\forall\alpha.A|_{\mathcal{I}} & \equiv & \bigcap_{\mathcal{C}\in\mathcal{CR}} |A|_{\mathcal{I};\alpha\leftarrow\mathcal{C}}. \end{array}$$

Les conditions 1, 2 et 3 suffisent évidemment à assurer :

Lemme 3.13 *Pour tout type T et toute interprétation candidat $\mathcal{I}$, $|T|_{\mathcal{I}}$ est un candidat de réductibilité.*

La preuve du lemme de substitution reste inchangée :

Lemme 3.14 *Pour tous T, T', α et $\mathcal{I}$:*

$$|T[\alpha \setminus T']|_{\mathcal{I}} = |T|_{\mathcal{I};\alpha\leftarrow|T'|_{\mathcal{I}}}.$$

Il n'est pas non plus nécessaire de changer :

Définition 3.20 *Une interprétation $\mathcal{I}$ est dite adaptée à un contexte Γ si pour tout couple (x, T) de Γ on a $\mathcal{I}(x) \in |T|_{\mathcal{I}}$.*

Et comme d'habitude :

Théorème 8 *Pour tout jugement dérivable $\Gamma \vdash t : T$ et toute interprétation candidat $\mathcal{I}$ adaptée à Γ, on a $|t|_{\mathcal{I}} \in |T|_{\mathcal{I}}$.*

PREUVE Par récurrence sur la dérivation du jugement. Tous les cas sont identiques à la preuve de 3.5 à l'exception du cas de la règle LAM qui est traité grâce à la condition 6 comme indiqué. ■

Corollaire 3.2 *Si $\Gamma \vdash t : T$ est dérivable, il éxiste une interprétation candidat $\mathcal{I}$ telle que $t \in |T|_{\mathcal{I}}$.*

PREUVE Comme il existe au moins un candidat de réductibilité $\mathcal{C}$ (condition 3), il suffit de prendre $\mathcal{I}(\alpha) = \mathcal{C}$ pour tout α et $\mathcal{I}(x) = x$ pour tout x. La condition 4 garantit que cette interprétation est adaptée à tout contexte. On a alors $|t|_{\mathcal{I}} = t$ et donc $t \in |T|_{\mathcal{I}}$. ■

Corollaire 3.3 *Si $\Gamma \vdash t : T$ est dérivable, alors t est fortement normalisable.*

PREUVE On prend $\mathcal{I}$ telle que définie précédemment. On a $t \in |T|_{\mathcal{I}}$, et comme $|T|_{\mathcal{I}}$ est un candidat de réductibilite, on a $t \in \mathcal{SN}$. ■

3.8.3 Autres réductions

Il est évident qu'il faut adapter les conditions énoncées lorsque l'on considère d'autres réductions que β comme par exemple l'élimination des entiers du système T ou plus généralement la ι-réduction des types inductifs. Une exception est la η-réduction (essentiellement à cause du lemme de retard de η) qui pour une fois ne pose pas de difficultés. En général il suffira d'étendre la condition 6 au cas où d'autres réductions se produisent en tête.

En λ-calcul typé, la λ-abstraction correspond à l'introduction du type flèche ; l'application à son élimination. La réduction correspond à l'élimination conjointe de ces deux opérateurs. Cette symétrie se retrouve dans toutes les extensions que nous considérons dans cette thèse : un constructeur permet l'introduction d'un type inductif et l'opérateur d'élimination permet précisément d'éliminer ce type ; la rencontre d'un constructeur et d'un schéma d'élimination donne lieu à un nouveau type de radical.

3.8.4 La définition de Girard

C'est la définition originelle de la Thèse d'État de Girard, simplifiée dans [83]. C'est aussi celle que nous avons présentée dans ce chapitre et que nous utilisons dans le chapitre suivant, après l'avoir adaptée aux définitions inductives génériques. Rappellons les trois conditions de Girard :

CR 1 $\quad \mathcal{C} \subset \mathcal{SN}$

CR 2 $\quad \forall t \in \mathcal{C}. \forall t'. (t \in \mathcal{C} \wedge t \rhd t') \Rightarrow t' \in \mathcal{C}$

CR 3 $\quad \forall t \in \mathcal{NT}. (\forall t'. t \rhd t' \Rightarrow t' \in \mathcal{C}) \Rightarrow t \in \mathcal{C}$

La condition 5 est exactement **CR 1** et la condition 4 une conséquence immédiate de **CR 3**. Il est aussi immédiat de vérifier que $\mathcal{SN}$ est un candidat au sens de Girard ce qui permet de vérifier la condition 3. On a déja vu comment prouver les conditions 1 et 2 ; ce qu'il faut remarquer, c'est que **CR 2** est utilisé pour prouver la condition 1.

La preuve de la condition 6 est également implicitement présente dans la première preuve de normalisation de F. On remarque que là encore, **CR 2** est uniquement utilisée pour prouver 6, en conjonction, bien sûr avec **CR 3**.

Cette définition présente deux avantages : tout d'abord elle est remarquablement concise et élégante. Ensuite, l'utilisation de la notion de termes neutres permet de l'étendre de manière très uniforme à de nouvelles réductions, comme les types inductifs. Il suffit en effet d'étendre la notion de terme neutre aux formes de termes qui correspondent aux nouveaux radicaux, comme par exemple $R(t, t', t'')$.

On peut en revanche lui reprocher de n'être pas vraiment minimale et de cacher ainsi certains mécanismes de la preuve. En particulier, la condition **CR 2** n'est pas vraiment nécessaire, en ce qu'elle ne correspond vraiment à aucune des 6 conditions que doivent vérifier les candidats. C'est en fait le prix à payer pour avoir transformé la condition 6 en **CR 3** : on a du coup besoin de **CR 2** pour prouver 1 et surtout 6. De plus ces conditions sont prouvées par des récurrences emboitées (ou grace à l'introduction d'un ordre lexicographique) un peu lourdes.

Un autre "défaut" de cette définition est que les candidats de réductibilité ainsi définis ne sont pas clos par réunion. On ne peut donc pas les utiliser pour prouver la normalisation de systèmes avec types union.

3.8.5 Les ensembles saturés de Tait

Cette première alternative à la définition de Girard à été d'abord introduite par Tait [140]. Elle est très facile à comprendre si l'on considère les 6 conditions que nous avons énoncées :

Définition 3.21 (Ensembles saturés) *Un ensemble de λ-termes S est dit saturé si :*

> **SAT 1** *tout élément de S est fortement normalisable*
>
> **SAT 2** $(x\ \vec{N_i}) \in S$ *si tout N_i est fortement normalisable*
>
> **SAT 3** *si* $(M[x \setminus N]\ \vec{N}) \in S$ *et N est fortement normalisable, alors* $(\lambda x.M\ N\ \vec{N}) \in S$.

On reconnaît, en gros, trois des six conditions : **SAT 1** est exactement la condition 5, **SAT 2** et **SAT 3** sont respectivement des généralisations de nos conditions 4 et 6. On montre facilement que $\mathcal{SN}$ est saturé ce qui vérifie la condition 3. La preuve de la condition 2 est immédiate.

La preuve de la condition 1 est plus intéressante : elle explique pourquoi les clauses **SAT 2** et **SAT 3** sont plus fortes que les conditions 4 et 6.

Lemme 3.15 *La famille des ensembles saturés de Tait vérifie la condition 1.*

Preuve Soient S_1 et S_2 deux ensembles saturés. On a :

- Si $t \in S_1 \to S_2$, comme $x \in S_1$ on a $(t\ x) \in S_2$ ce qui implique (**SAT 1**) que $(t\ x) \in \mathcal{SN}$ et donc t est fortement normalisable. $S_1 \to S_2$ vérifie **SAT 1**.
- Soit $(x\ \vec{N}) \in \mathcal{SN}$ et $N \in S_1$, d'après **SAT 2** pour S_2 et **SAT 1** pour S_1 on a $(x\ \vec{N}\ N) \in S_2$. Donc $S_1 \to S_2$ vérifie **SAT 2**.
- Soit $t[x \setminus u] \in S_1 \to S_2$ avec $u \in S$ pour un certain ensemble saturé S. Alors u est fortement normalisable. Soit $v \in S_1$, on a $(t[x \setminus u]\ v) \in S_2$. La clause **SAT 3** pour S_2 assure que $(\lambda x.t\ u\ v) \in S_2$. Donc $S_1 \to S_2$ vérifie **SAT 3** (le cas où l'on inverse S et $S_1 \to S_2$ dans l'énoncé est immédiat).

■

On remarque tout de suite que cette définition est plus proche de notre caractérisation des candidats que la définition de Girard ; aussi les preuves sont elles peut-être très légèrement plus simples. On peut aussi remarquer (nous ne sommes pas les premiers à le faire) que la notion d'ensemble saturé est moins restrictive que la définition de Girard : un candidat de réductibilité est saturé alors que la réciproque est fausse.

Lemme 3.16 *Soit $\mathcal{SAT}$ l'ensemble des ensembles saturés. On a : $\mathcal{CR} \subsetneq \mathcal{SAT}$.*

Preuve Soit $\mathcal{C} \in \mathcal{CR}$, $\mathcal{C}$ vérifie évidemment **SAT 1** et **SAT 2** (respectivement identique et conséquence de **CR 1** et **CR 3**). La clause **SAT 3** se démontre de la manière usuelle, par récurrence sur le nombre maximal de réductions faisables sur les arguments ; c'est quasi-identique à la preuve que $\mathcal{CR}$ vérifie la condition 6.

Il nous reste à construire un ensemble saturé qui ne soit pas un candidat de réductibilité au sens de Girard. Considérons l'ensemble des termes fortement normalisables, tel que leur forme normale par réduction de tête faible soit de la forme $(x\ \vec{N})$ ou égale à $\lambda x.(\lambda z.zy)$. Plus précisément il s'agit de l'ensemble S_0 définit inductivement par :

- $(x\ \vec{N}) \in \mathcal{SN} \Rightarrow (x\ \vec{N}) \in S_0$
- $\lambda x.(\lambda z.zy) \in S_0$
- $\forall t.\forall \vec{N}.\forall u \in \mathcal{SN}.\forall x.(t[x \setminus u]\ \vec{N}) \in S_0 \Rightarrow (\lambda x.t\ u\ \vec{N}) \in S_0$.

Il est très facile de voir que S_0 est saturé. Par induction, on vérifie également que $\lambda x.(\lambda z.zy)$ est le seul élément de S_0 qui soit une λ-abstraction, et donc $\lambda x.y \notin S_0$. Or comme

$$\lambda x.(\lambda z.zy) \rhd_\beta \lambda x.y$$

S_0 ne vérifie pas **CR 2**. ■

Un avantage objectif des ensembles saturés est que leur famille est close par union arbitraire. On peut donc les utiliser pour prouver la normalisation de systèmes avec types union.

Lemme 3.17 *Soit $(S_i)_{i\in I}$ une famille arbitraire d'ensembles saturés. Alors $\bigcup_{i\in I} S_i$ est un ensemble saturé.*

Preuve Les conditions **SAT 1** et **SAT 3** sont trivialement vérifiées. Si $(M[x\backslash N]\ \vec{N}) \in \bigcup_{i\in I} S_i$ et N est fortement normalisable, alors il existe $i \in I$ tel que $(M[x\backslash N]\ \vec{N}) \in S_i$ et donc $(\lambda x.M\ N\ \vec{N}) \in S_i$. ■

3.8.6 Une variante des ensembles saturés

La définition suivante est une légère variante de la précédente. Elle est due à David McAllester. On ajoute une clause à la définition des ensembles saturés :

SAT 1 tout élément de S est fortement normalisable

SAT 2 $(x\ \vec{N_i}) \in S$ si tout N_i est fortement normalisable

SAT 3 si $(M[x \setminus N]\ \vec{N}) \in S$ et N est fortement normalisable, alors $(\lambda x.M\ N\ \vec{N}) \in S$

SAT 4 si $M \in S$ et $M \rhd^{whd} M'$, alors $M' \in S$.

Où $\rhd^{whd}$ désigne la réduction faible de tête, définie inductivement par :

- $(\lambda x.t\ u) \rhd^{whd} t[x \setminus u]$
- $t \rhd^{whd} t' \Rightarrow (t\ u) \rhd^{whd} (t'\ u)$.

La réduction faible de tête est en fait une stratégie de réduction, en ce sens qu'elle est *déterministe* : tout terme peut se réécrire au plus d'une manière par $\rhd^{whd}$. Les termes qui n'admettent pas de réduit pour $\rhd^{whd}$ sont ceux de la forme $\lambda x.t$ ou $(x\ \vec{N})$; on les appelle les *valeurs*. Si la réduction de tête faible termine sur un terme t, elle termine forcément sur une valeur qu'on appelle la valeur de t. La particularité essentielle de $\mathcal{SAT}$ tel que redéfini ci-dessus est :

Lemme 3.18 *Soit $\mathcal{V}$ un ensemble de termes fortement normalisables et de la forme $\lambda x.t$. Alors il existe un et un seul ensemble saturé dont les valeurs soient exactement la réunion de $\mathcal{V}$ et de l'ensemble des termes $(x\ \vec{N})$ fortement normalisables.*

En d'autres termes, tout ensemble saturé est alors caractérisé exactement par l'ensemble de ses valeurs. La structure des ensembles saturés est ainsi plus homogène. On met également mieux en valeur le rôle particulier joué par la réduction de tête.

Nous laisserons le lecteur motivé vérifier que les 6 conditions sont bien vérifiées. On peut remarquer que cette définition est strictement plus restrictive que la précédente et strictement moins que celle de Girard (on peut reprendre pour cela le contre-exemple précédent). Par ailleurs, cette version de $\mathcal{SAT}$ est également close pour l'union.

3.8.7 La définition de Parigot

Dans toutes les définitions considérées jusqu'ici, on définit essentiellement les candidats à l'aide de variantes des conditions 4, 5 et surtout 6 (c'est-à-dire celles qui servent dans la dernière partie de la preuve), puis on vérifie que les conditions 1 et 2 sont aussi vérifiées (c'est-à-dire que l'ensemble des candidats est clos par les constructions de l'interprétation des types). Michel Parigot c'est rendu compte que l'on pouvait simplement inverser le problème et prendre les conditions 1 et 2 pour définir les candidats de réductibilité puis vérifier que les autres conditions étaient alors respectées. On peut s'étonner que personne n'ait eu cette idée plus tôt ; mais il y a une différence essentielle entre cette définition et les précédentes : il s'agit d'une définition *inductive* de l'ensemble $\mathcal{CR}$ et non plus juste d'un prédicat du premier ordre sur les ensembles de termes.

Cette définition a été inventée par Parigot pour une preuve directe de normalisation forte du système F étendu avec des opérateurs classiques (continuations) [117].

Définition 3.22 (Candidats de Réductibilité de Parigot) *On définit l'ensemble $\mathcal{CR}$ d'ensembles de termes comme le plus petit vérifiant :*

- $\boldsymbol{\mathcal{SN}} \in \mathcal{CR}$
- $\forall \mathcal{C}_1, \mathcal{C}_2 \in \mathcal{CR}\ .\ \mathcal{C}_1 \to \mathcal{C}_2 \in \mathcal{CR}$
- *pour toute famille $(\mathcal{C}_i)_{i\in I}$ d'éléments de $\mathcal{CR}$, $\bigcap_{i\in I} \mathcal{C}_i \in \mathcal{CR}$.*

Il est facile de voir que ces trois clauses correspondent à une définition inductive correcte. Bien sûr, les conditions 1, 2 et 3 sont vérifiées par cette nouvelle version de $\mathcal{CR}$. Pour vérifier que tous les éléments d'un candidat au sens de Parigot sont fortement normalisables, il suffit de vérifier que ces candidats sont non-vides (pour passer à la deuxième clause). Or, étant donné un candidat $\mathcal{C}$ de Parigot, on prouve aisément, par récurrence sur la preuve que $\mathcal{C}$ est un candidat, que

$$(x\ \vec{N}) \in \boldsymbol{\mathcal{SN}} \Rightarrow (x\ \vec{N}) \in \mathcal{C}.$$

Ce qui permet également de vérifier la condition 4.

La condition 6 est également prouvée par récurrence sur la preuve que $\mathcal{C} \in \mathcal{CR}$. Plus exactement on renforce l'hypothèse de récurrence et prouve que S vérifie la clause **SAT 3** de Tait :

- Elle est évidemment vérifiée par $\boldsymbol{\mathcal{SN}}$
- Si $u \in S'$ et $t[x \setminus u] \in S$ avec $S = S_1 \to S_2$, alors pour tout $v \in S_1$ on a

$$(t[x \setminus u]\ v) \in S_2$$

et par hypothèse de récurrence sur S_2

$$(\lambda x.t\ u\ v) \in S_2$$

ce qui prouve

$$(\lambda x.t\ u) \in S.$$

- Le cas de l'intersection est immédiat.

Il est clair que cette définition est la plus restrictive qui soit, modulo le choix du candidat canonique (ici $\mathcal{SN}$), puisque les deux autres clauses de la définition inductive sont des conditions que doivent nécéssairement vérifier toute famille de candidats.

3.8.8 La définition de Parigot étendue par l'union

Tout comme la définition de Girard, celle de Parigot n'est pas close pour l'union. On peut toutefois facilement y remédier cette fois en l'ajoutant simplement à la définition :

- $\mathcal{SN} \in \mathcal{CR}$
- $\forall \mathcal{C}_1, \mathcal{C}_2 \in \mathcal{CR}. \mathcal{C}_1 \rightarrow \mathcal{C}_2 \in \mathcal{CR}$
- pour toute famille $(\mathcal{C}_i)_{i \in I}$ d'éléments de $\mathcal{CR}$, $\bigcap_{i \in I} \mathcal{C}_i \in \mathcal{CR}$
- pour toute famille $(\mathcal{C}_i)_{i \in I}$ d'éléments de $\mathcal{CR}$, $\bigcup_{i \in I} \mathcal{C}_i \in \mathcal{CR}$.

Nous laissons le lecteur vérifier que cette variante vérifie bien les six conditions. Ce n'est pas étonnant puisque $\mathcal{SAT}$ est déja clos pour l'union, ce qui montre que la condition 6 "passe à l'union".