Cours de Lambda-calcul.[*]

Ch. Berline.

CHAPITRE 4 : Le Système E .[†]
et les Théorèmes de normalisation
du λ-calcul pur

Table des matières

[*]DEA de Logique et Fondements de l'Informatique (2000-2002), UFR de Mathématiques, Université de Paris 7.
[†]Version du 24 Février 2001, révisée le 22 Février 2002.

1 Introduction.

[[reste à faire : refondre les intros et l'appendice]]

Ce chapitre est un chapitre charnière entre la partie purement syntaxique qu'était la première partie de ce cours et les chapitres de constructions et de manipulations de modèles qui suivront. Nous allons en effet montrer les *théorèmes de normalisation* classiques du λ-calcul à partir d'un système de typage très simple, *le Système E*, qui est la version syntaxique du modèle le plus simple possible du λ-calcul pur, à savoir le *modèle d'Engeler*, et de la réductibilité, qui est une "sémantique syntaxique" du λ-calcul. C'est aussi au moyen de la réductibilité que nous démontrerons plus tard un *théorème d'approximation* pour une large famille d'extensions du système E qui s'avèrera un outil fondamental pour l'étude des théories équationnelles des modèles que nous présenterons par la suite. C'est enfin avec des techniques de réductibilité que nous montrerons les propriétés importantes du système F de Girard dans la suite de ce cours.

Le *système E* que nous présentons ici est le plus simple des systèmes de typage, ou plutôt "d'*assignation de types*", à tendance sémantique (en anglais : "Intersection type assignment systems").

D'un point de vue logique ces systèmes, dont le premier est dû à Coppo et Dezani, sont basés sur l'implication, et sur un usage restreint de la conjonction, dont le comportement devient alors semblable à celui de l'intersection (cf. Appendice).

Le système E restreint de plus l'usage de la conjonction dans la définition inductive de l'ensemble des formules. C'est le système que l'on obtient quand on enlève les redondances du système de [6] qui est présenté dans [1] sous le nom de D_Ω , sa restriction E_f correspondant, elle, au système D. L'utilisation systèmatique du système E et de ses restrictions simplifie nettement les preuves des Théorèmes de normalisation du λ-calcul ; de même l'utilisation systématique des extensions non redondantes de E simplifie nettement l'étude des modèles, comme nous le verrons par la suite. Le système E a été étudié aussi dans [3] sous le nom de "système strict", mais nous réserverons ici cette terminologie pour la restriction E^f de E.

La lettre E vient de "Engeler". Comme nous le verrons dans un chapitre suivant, le système E est en effet la lecture logique directe du *modèle d'Engeler* (et Plotkin). Ceci est un cas particulier de la dualité *Systèmes-de-typage / Modèles* dont nous verrons plusieurs avatars dans ce cours. Ces modèles duaux apparaissent comme une lecture sémantique directe des systèmes (et les systèmes comme une lecture syntaxique des modèles en question).

Comme nous l'avons déja mentionné, il existe une autre sémantique "syntaxique" de ces systèmes de typage. Elle est issue des techniques de réductibilité de Tait et de réalisabilité de Kleene. Elle interprète les termes par eux-mêmes et les types par des ensembles de termes, et elle permet de démontrer simultanément des propriétés des systèmes et des modèles, nous la verrons donc réapparaître plus tard. C'est la seule sémantique que nous utiliserons dans ce chapitre.

E^f et E^g sont deux restrictions simples de E.
E^f correspond à une version stricte du modèle d'Engeler.
Le but du présent chapitre est de montrer les théorèmes de normalisation suivants :

Théorème 1 *Il est équivalent pour un terme d'être normalisable de tête, résoluble, ou typable dans E.*

Prouvant ainsi l'équivalence des deux premières classes.

Théorème 2 *Il est équivalent pour un terme d'être normalisable à gauche, normalisable, ou typable dans E^g.*

Prouvant ainsi l'équivalence des deux premières classes.

Théorème 3 *Il est équivalent pour un terme d'être fortement normalisable et d'être typable dans E^f.*

Théorème 4 *Il est équivalent pour un λI-terme d'être normalisable et d'être fortement normalisable.*

Dans un prochain chapitre nous verrons de plus qu'être typable dans E est équivalent à : avoir une interprétation non vide dans le modèle d'Engeler.

Notations et conventions (rappels).
Etant donné un ensemble G, on désignera par G^* l'ensemble de ses parties finies et par $G^{<\omega}$ l'ensemble de ses suites finies, y compris la suite vide. Quand cela sera utile nous confondrons la suite $\bar{x}$ et le sous-ensemble sous-jacent et écrirons par exemple : $y \in \bar{x}$. Etant donnée une suite $\bar{x}$, on désignera par x_i le i-ème élément de la suite, et ça sous-entendra que $i \leq l(\bar{x})$. Cette convention sera utilisée aussi bien pour V et Λ que pour les ensembles F et F^* définis plus bas.

Les expressions de la forme $t[\bar{x} := \bar{u}]$, $\bar{x} : \bar{a}$ ou $\bar{t} : \bar{a}$, et plus généralement les expressions mettant deux suites finies en rapport direct, sous-entendront toujours que les deux suites ont la même longueur.

2 Le système E.

Le système E peut être vu comme restriction très simple, et de puissance équivalente, d'un système d'assignation de types dû à Coppo, Dezani, et Veneri [6], et qui est présenté dans [1] sous le nom de "système D_Ω". La correspondance de E avec le système D_Ω est explicitée dans l'appendice.

2.1 Présentation du système E.

L'ensemble des formules (ou *"types"*) du système E est noté F , ses éléments $\alpha, \beta, ...$ et les éléments de F^* : $a, b, ...$; L'ensemble fini a peut être pensé comme la conjonction de ses éléments.

L'ensemble F est défini par induction à partir d'un ensemble P non vide de variables propositionnelles comme le plus petit ensemble tel que :

$$F = P \cup \{a \to \alpha \ / \ (a, \alpha) \in F^* \times F\}$$

Convention. $a_1, ..., a_n \to \alpha$ est une abréviation pour $a_1 \to (a_2 \to (... \to (a_n \to \alpha)...))$ si $n \geq 1$ et α sinon ; elle sera elle-même abbrégée en $\bar{a} \to \alpha$.

Les *contextes du système* E sont les applications *partielles* Γ de V dans F^* qui sont de domaine fini (éventuellement vide). Le contexte de domaine vide sera appellé le *contexte vide.* Le contexte Γ défini par $dom(\Gamma) = \{x_1, ..., x_n\}$, où les x_i sont supposées distinctes, et $\forall i \ \Gamma(x_i) = a_i$, sera aussi désigné par :

$$\bar{x} : \bar{a} \quad \text{ou} \quad x_1 : a_1, ..., x_n : a_n$$

L'expression $\Gamma, \bar{x} : \bar{a}$ sous-entend que $\bar{x} \cap dom(\Gamma) = \emptyset$; elle désigne le prolongement évident de Γ à $\bar{x}$.

Les *séquents du système* E sont les expressions de la forme :

$$\Gamma \vdash t : \alpha$$

où Γ est un contexte, α une formule, et t un élément de Λ , c'est-à-dire un terme considéré au renommage près de ses variables liées. Les a_i sont les *prémisses* du séquent et α sa *conclusion.* Dans le cas où Γ est le contexte vide on écrira simplement : $\vdash t : \alpha$.

L'expression $\Gamma \vdash t : a$ est une *abbréviation* pour $\forall \alpha \in a \ \Gamma \vdash t : \alpha$.
L'expression $\Gamma \vdash \bar{t} : \bar{a}$ est une abbréviation pour $\forall i \leq l(\bar{t}) \ \Gamma \vdash t_i : a_i$.
En particulier $\Gamma \vdash t : \emptyset$ est toujours vrai.
On remarquera que ces notations sont compatibles en sens avec l'usage de la notation $x : a$ dès que l'on dispose de la règle 1. ci-dessous.

Les *règles du système* E sont les suivantes :

1)	(axiome / variable)	$\alpha \in \Gamma(x)$	$\Longrightarrow$	$\Gamma \vdash x : \alpha$
2)	($\to$-intro / abstraction)	$\Gamma, x : a \vdash t : \alpha$	$\Longrightarrow$	$\Gamma \vdash \lambda x.t : a \to \alpha$
3)	($\leftarrow$-élim / application)	$\left.\begin{array}{l}\Gamma \vdash t : a \to \beta \\ \Gamma \vdash u : a\end{array}\right\}$	$\Longrightarrow$	$\Gamma \vdash (t)u : \beta$

Un *axiome* est un séquent de la forme $\Gamma \vdash x : \alpha$ avec $\alpha \in \Gamma(x)$. Une *dérivation* est une suite finie de séquents dont chacun est un axiome ou est obtenu à partir de certains de ceux qui précèdent par application d'une instance de règle. Un *séquent est dérivable* dans le système si c'est le *séquent conclusion* d'une dérivation, c'est-à-dire son dernier séquent .

Notons que la nature de la dernière règle utilisée pour typer un terme est déterminée par la structure du terme ; dans le cas de l'abstraction l'instance de la règle est déterminée dès que l'on a choisi $x \notin dom(\Gamma) \cup Vlib(\lambda x.t)$.

L'expression "Si $\Gamma \vdash t : \alpha$ " est une abbréviation pour : "Si $\Gamma \vdash t : \alpha$ est dérivable dans le système" ; t est *typable de type* α dans le système si il existe Γ tel que $\Gamma \vdash t : \alpha$ soit dérivable, et il est *typable* si il existe un couple (Γ, α) tel que $\Gamma \vdash t : \alpha$.
Quand cela sera nécessaire nous préciserons "$\vdash_E$" et "E*-dérivation*".

Exemple 5 *$\vdash I : \alpha$ ssi il existe b, β tels que : $\alpha \equiv b \to \beta$ et $\beta \in b$.*
$\vdash V : \alpha$ ssi il existe b, c, β tels que : $\alpha \equiv b, c \to \beta$ et $\beta \in b$.
$\vdash F : \alpha$ ssi il existe b, c, γ tels que : $a \equiv b, c \to \gamma$ et $\gamma \in c$.

Notation 6 *$\Gamma \sqsubseteq \Gamma'$ signifie $dom(\Gamma) \subseteq dom(\Gamma')$ et $\forall x \in dom(\Gamma)$ $(\Gamma(x) \subseteq \Gamma'(x))$. Il est clair que deux contextes Γ et Γ' ont toujours un plus petit majorant pour l'ordre $\sqsubseteq$, que nous noterons : $\Gamma \cup \Gamma'$; il est tout aussi clair qu'il peut être défini alternativement comme le contexte qui satisfait :*
$dom(\Gamma \cup \Gamma') = dom(\Gamma) \cup dom(\Gamma')$ et, pour tout $x \in dom(\Gamma \cup \Gamma')$:
$(\Gamma \cup \Gamma')(x) = \Gamma(x) \cup \Gamma'(x)$ ou $\Gamma(x)$ ou $\Gamma'(x)$, suivant que x est dans l'intersection des domaines ou pas.

Exercice 7 *(*Typage dans E et η-réduction).
1. Montrer que si $b \supsetneq a$ on a :
$\nvdash_E \lambda x.x \quad : \{a \to \alpha\} \to (b \to \alpha)$ *, tandis que :*
$\vdash_E \lambda x \lambda y.xy : \{a \to \alpha\} \to (b \to \alpha)$.
2. Montrer que si t est une abstraction et si $x \notin Vlib(t)$, alors :
$\Gamma \vdash t : \alpha \Leftrightarrow \Gamma \vdash \lambda x.tx : \alpha$ [1]
ce qui est un cas particulier du fait que le typage dans E a un très bon comportement vis-à-vis de la β-équivalence, comme nous le verrons bientôt.

2.2 Les restrictions E^f et E^g.

Nous dirons qu'*une formule est stricte* si $\emptyset$ n'est pas utilisé dans sa construction, que a est strict si il est non vide et ne contient que des formules strictes, qu'un uple $\bar{a}$ est strict si il est composé d'ensembles stricts, et qu'*un séquent $\bar{x} : \bar{a} \vdash t : \alpha$ est strict* si α et $\bar{a}$ sont stricts; enfin *une dérivation est stricte* si tous les séquents qui la composent sont stricts.

Le système E^f est la version non redondante du système D de [1]. L'ensemble de ses formules est l'ensemble F^f des formules strictes de E et ses dérivations sont les dérivations strictes de E ; autrement dit les schémas de règles sont les mêmes que pour E mais il y en a moins d'instances car n'ont le droit d'intervenir que des formules strictes. La définition directe de E^f consisterait à donner la définition inductive de son ensemble de formules, puis à prendre les mêmes schémas de règles que pour E en excluant les instances où $a = \emptyset$.

Le système S de Curry consiste à se restreindre de plus au cas où tous les a sont des singletons (et à les identifier avec leurs éléments) .

Le système E^g est intermédiaire entre E et E^f : ses formules et ses règles sont celles du système E *mais* les seules dérivations admises sont celles dont la conclusion est un séquent strict. Récapitulons :

[1] pour le sens $\Leftarrow$ penser à renommer d'abord x à l'extérieur de $\Gamma \cup Vlib(t)$.

Une E^g-dérivation est une E-dérivation dont le séquent conclusion est strict.
Une E^f-dérivation est une dérivation stricte de E.

Remarque 8 *D'une manière générale, le fait qu'un terme soit typable dans un certain système limite, plus ou moins, ses possibilités d'application et limite en particulier l'auto-application. L'exemple le plus simple en est l'exercice suivant :*

Exercice 9 *Montrer que xx n'est pas typable dans le système S mais qu'il l'est dans E^f, et donc dans E et E^g.*

2.3 Propriétés élémentaires.

Toutes les propriétés qui suivent sont de la forme $\forall t \forall \Gamma \forall \alpha \, P(\Gamma, \alpha, t)$ et se démontrent très facilement par induction sur la complexité de t (ou directement). Dans le cas où $t = \lambda x.u$ il faut parfois penser à renommer x avant de passer à u.

1. Si $\Gamma \vdash_{E^f} t : \alpha$, alors Γ mentionne toute variable libre x de t et de plus $\Gamma(x) \neq \emptyset$.
   Faux pour E .
2. Si $y \notin dom(\Gamma)$, alors $\Gamma \vdash_E t : \alpha$ ssi $\Gamma, y : \emptyset \vdash_E t : \alpha$.
   Cette assertion n'a évidemment pas de sens pour E^f.
   Démonstration. Le seul pas de l'induction qui n'est pas trivial est celui où t est de la forme $\lambda x.u$: il faut penser à choisir x à l'extérieur de $dom(\Gamma) \cup \{y\}$ avant de remonter à u.
3. Si $\Gamma \vdash_E t : \alpha$, alors $\forall \Gamma' \sqsupseteq \Gamma \; (\Gamma' \vdash_E t : \alpha)$ et $\forall \Gamma' (\Gamma \cup \Gamma' \vdash_E t : \alpha)$.
   Vrai pour E^f ; noter que dans ce cas on ne quantifie que sur des Γ, Γ' stricts.
   En particulier, si t et t' sont typables, alors il existe Γ tel que tous deux soient typables dans le contexte Γ.
4. $\Gamma \vdash_E t : \alpha$ ssi $\Gamma_t \vdash_E t : \alpha$, où Γ_t est la restriction de Γ à l'ensemble des variables libres de t .
   Vrai pour aussi pour E^f. En particulier tout terme clos typable est typable dans le contexte vide.
5. Si t est typable à l'aide d'une E^f-dérivation, alors tous les sous-termes de t sont typables par une E^f-dérivation (et sont typés au cours de toute E^f-dérivation aboutissant à un typage de t). C'est faux pour E car les instances de la règle 3 où $a = \emptyset$ n'induisent pas de typage de u. Toutefois :
6. Si t est typable dans E et $t = \lambda x.v$ ou $t = (v)u$, alors v est typable dans E.
7. t est typable dans E ou E^f ssi $\lambda x_1 ... \lambda x_m.t$ l'est ; de plus les types de ce dernier terme sont tous de la forme $a_1, ..., a_m \to \alpha$, où α est un type de t.
8. Toute abstraction typable est typable d'une infinité de façons dans le même contexte.

9. Plus précisément, si $u \equiv \lambda x.t$ est de type $a \to \alpha$ dans le contexte Γ, alors il est dans ce contexte de type $b \to \alpha$ pour tout b contenant a. Ceci est faux en général, par exemple si u est une variable.
10. *Règle dérivée.* Pour E et E^f. Il est souvent commode d'utiliser directement la règle dérivée $(\bar{3})$ suivante, dont la validité se montre facilement par induction sur $l(\bar{a})$:

$$\left.\begin{array}{c}\Gamma \vdash t : \bar{a} \to \beta \\ \Gamma \vdash \bar{u} : \bar{a}\end{array}\right\} \Longrightarrow \Gamma \vdash t\bar{u} : \beta$$

et que réciproquement si la conclusion de l'implication est vraie, alors il existe $\bar{a}$ tel que l'hypothèse soit vraie.

3 Typage et β-réduction.

Notations. On associe à tout système de typage F le préordre : $t \leq_F t'$ ssi $\forall\Gamma\forall\alpha\,(\,\Gamma \vdash_F t : \alpha \Longrightarrow \Gamma \vdash_F t' : \alpha\,)$, et on note $\simeq_F$ la relation d'équivalence associée : $t \simeq_F t'$ ssi $(\,t \leq_F t'$ et $t' \leq_F t\,)$.

Il est facile de vérifier que $\leq_E$ et $\leq_{E^f}$, et donc $\simeq_E$ et $\simeq_{E^f}$, passent au contexte, mais pas $\leq_{E^g}$ (exercice 14). On a d'autre part $\leq_E \subseteq \leq_{E^g}$ et finalement la proposition suivante exprime que $\simeq_\beta \subseteq \simeq_E$et que $\to_\beta \subseteq \leq_{E^f}$.

Proposition 10 *Conservation du type par β-réduction ou β-équivalence.*

1.	$\Gamma \vdash_E t : \alpha$	*et*	$t \simeq_\beta t'$	$\Longrightarrow$	$\Gamma \vdash_E t' : \alpha$
1*g*.	$\Gamma \vdash_{E^g} t : \alpha$	*et*	$t \simeq_\beta t'$	$\Longrightarrow$	$\Gamma \vdash_{E^g} t' : \alpha$
2.	$\Gamma \vdash_{E^f} t : \alpha$	*et*	$t \to_\beta t'$	$\Longrightarrow$	$\Gamma \vdash_{E^f} t' : \alpha$

Remarque. Nous allons donner ici une démonstration syntaxique de cette proposition mais on peut noter que (1) est aussi un corollaire immédiat du théorème de dualité système/modèle que nous montrerons dans un chapitre suivant ; (2) a également une démonstration sémantique, via une version stricte du modèle d'Engeler. Quant à (1g) c'est une conséquence immédiate de (1) et de ce que le fait qu'un séquent soit strict ou non ne dépend que des formules du séquent.

Lemme 11 *1.* (Adéquation syntaxique) *Dans E et E^f il est vrai que : pour tous $\Gamma, v, \bar{x}, \bar{u}, \alpha$,*

$$\exists\bar{a}\quad \left.\begin{array}{c}\Gamma, \bar{x} : \bar{a} \vdash v : \alpha \\ \Gamma \vdash \bar{u} : \bar{a}\end{array}\right\} \Longrightarrow \Gamma \vdash v[\bar{x} := \bar{u}] : \alpha$$

2. La réciproque est vraie si on sait de plus qu'il existe $\bar{b}$ que $\Gamma \vdash \bar{u} : \bar{b}$.

Remarquons que dans le cas de E la condition exigée sur les u_i est automatiquement vérifiée, avec $b_i = \emptyset$ pour tout i, et que dans le cas de E^f elle exprime seulement que les u_i doivent être typables dans le contexte Γ.

Démonstration. Par induction sur la complexité de v pour les deux directions. L'hypothèse supplémentaire pour 2. sert à démarrer l'induction dans le cas où $t \equiv y$; penser à distinguer suivant que y figure dans $\bar{x}$ ou non.

Exercice vivement conseillé : la rédiger.

Corollaire 12 *1.* $\Gamma \vdash_E (\lambda x.v)u : \alpha \Longleftrightarrow \Gamma \vdash_E v[x := u] : \alpha$
2. $\Gamma \vdash_{E^f} (\lambda x.v)u : \alpha \Longrightarrow \Gamma \vdash_{E^f} v[x := u] : \alpha$ *et la réciproque est vraie si u est typable dans Γ.*

Démonstration. Pour tous Γ, α on a : $\Gamma \vdash (\lambda x.v)u : \alpha \Longleftrightarrow \exists a\ [\Gamma \vdash u : a\ et\ \Gamma \vdash \lambda x.v : a \to \alpha] \Longleftrightarrow \exists a\ [\Gamma \vdash u : a\ et\ \Gamma, x : a \vdash v : \alpha] \Longrightarrow \Gamma \vdash v[x := u] : \alpha$, et l'implication est une équivalence dans le cas du système E et, dans le cadre du système E_f , si u est typable dans le contexte Γ (lemme d'adéquation).

Démonstration des points 1 et 2 de la Proposition 10.
Le cas où t est un rédexe et t' est son réduit est traité par le Corollaire précédent. On règle ensuite les cas $t \to_{\beta_0} t'$ et $t' \to_{\beta_0} t$ puis le cas général en utilisant le fait que les relations $\leq_{E,E_f}$ passent au contexte et sont transitives.

Exercice 13 *Exemple de deux termes fortement normalisables, β-équivalents et non E_f-équivalents. Le terme $t \equiv \lambda x.(\lambda d.x)(x)x$ est tel que $t \to_h I$. Montrer que $\not\vdash_{E_f}\ t : \{\alpha\} \to \alpha$.*

Exercice 14 *Non-contextualité de $\leq_{E^g}$.*
Esquisse d'un contre-exemple. *Le Théorème 2, que nous n'avons pas encore montré, implique que $\lambda x.x\omega V \leq_{E^g} \lambda x.x\omega F$, cependant $(\lambda x.x\omega V)F \not\leq_{E^g} (\lambda x.x\omega F)F$.*

4 Typabilité des termes normalisables.

Proposition 15 *Tout terme résoluble est typable dans E .*

Démonstration. Compte tenu de la remarque 7, section 2.3, il suffit de le montrer pour tout terme t qui est clos et résoluble. Pour un tel terme il existe $n \geq 0$ et $u_1, ..., u_n$ tels que $tu_1...u_n \simeq_\beta x$. Toute variable est typable et la typabilité se conserve par β-équivalence donc $tu_1...u_n$ est typable ; il suffit alors d'utiliser la remarque 6, section 2.3, pour conclure que t lui-même est typable.

Exercice 16 *Typer directement les fnt, en utilisant le moins de variables propositionnelles possibles.*

Lemme 17 *Si $\bar{t}$ est typable dans E_f, alors $\lambda\bar{y}.x\bar{t}$ l'est aussi.*

Démonstration. Soit $t \equiv \lambda\bar{y}.x\bar{t}$. Compte tenu de la remarque 7 de la section 2.3, et du fait que toute variable est typable, il suffit de le montrer pour $t = xt_1...t_n$, $n \geq 1$. Par hypothèse il existe pour tout i un séquent dérivable $\Gamma_i \vdash t_i : \alpha_i$. Soit $\Gamma = \cup\{\Gamma_i\ /\ i \in N\} \cup \{x : \{\{\alpha_1\}, ..., \{\alpha_n\} \to \alpha\}\}$ où α est une formule stricte quelconque. Il est facile de montrer par induction sur $p \leq n$ que $\Gamma \vdash t : \alpha$. De plus cette dérivation de t est stricte dès que celles des t_i le sont.$\diamond$

Corollaire 18 *Tout terme normal est typable dans E^f.*

Proposition 19 *Tout terme normalisable est typable dans E^g.*

Démonstration. C'est une conséquence immédiate du Corollaire précédent et de la Proposition 10 (1g).

Proposition 20 *Tout terme fortement normalisable est typable dans E^f.*

Démonstration. Elle se fait par induction sur le couple $\Sigma'(t) = (\Sigma(t), l(t))$ ordonné lexicographiquement, où $l(t)$ est la longueur du terme et $\Sigma(t)$ est la somme des longueurs des normalisations de t (= réductions de t qui aboutissent sur la forme normale de t). Ce nombre est bien défini car un terme fortement normalisable n'a qu'un nombre fini de normalisations (Lemme de König). Notons que les réduits et les sous-termes d'un terme fortement normalisable sont tous fortement normalisables et ont un Σ' strictement plus petit.

Le cas $\Sigma(t) = 0$ correspond au cas déja réglé où t est normal.

On suppose que $\Sigma(t) \neq 0$ et on distingue deux cas :

a) $t = \lambda\bar{x}.y\bar{t}$, $l(\bar{t}) \geq 1$. On applique l'hypothèse d'induction aux t_i, et le Lemme 17.

b) $t = \lambda\bar{x}.R\bar{t}$, où $R \equiv (\lambda x.v)u$ est un rédex et $l(\bar{t}) \geq 0$. Il est facile de se ramener au cas où $t = R\bar{t}$. Soit $t' \equiv v[x := u]\bar{t}$; on applique l'hypothèse d'induction à t' et u. Ces termes sont donc typables et on peut supposer que c'est dans le même contexte strict Γ (Remarque 3 de la section 2.3) ; en particulier il existe $\bar{a}$ et α stricts tels que $\Gamma \vdash \bar{t} : \bar{a}$ et $\Gamma \vdash v[x := u] : \bar{a} \to \alpha$. D'après le Corollaire 12 on a aussi $\Gamma \vdash R : \bar{a} \to \alpha$, et donc $\Gamma \vdash t : \alpha$, ce qui montre la typabilité de t.

Proposition 21 *Tout λI-terme normalisable à gauche est typable dans E^f.*

Démonstration. La démonstration suit exactement les mêmes lignes que la précédente, sauf que l'induction a lieu cette fois sur le couple $L'(t) = (L(t), l(t))$, où $L(t)$ est la longueur de la normalisation gauche de t, et que dans le cas (b) la typabilité de u dans le contexte Γ ne vient plus de l'hypothèse d'induction mais de ce que u est sous-terme du terme $v[x := u]$ (Remarque 5 de la section 2.3). En effet nous sommes dans le cas où R est un λI-terme, donc x est variable libre de v.

5 Réductibilité.

La méthode consiste à interpréter les types par des parties de Λ de telle façon qu'un terme soit dans l'interprétation de son type, ce qui va nous forcer à ne considérer que des parties saturées, en un sens à préciser.

Dans tout le reste du chapitre les lettres U, W, N, S désignent des parties quelconques de Λ, et V dénote toujours l'ensemble des variables de Λ.

5.1 Interprétation de l'implication.

On commence par définir l'opérateur binaire sur $P(\Lambda)$ qui interprétera la flèche ; on le note par $\Rightarrow$.

Définition 22 $U \Rightarrow W \;\; =_{def} \{t \in \Lambda \;/\; \forall u \in U \;\; (t)u \in W\}$. *Cette flèche est clairement croissante en* W *et décroissante en* U *(pour l'inclusion). On a :* $\emptyset \Rightarrow W = \Lambda$ *et* $U \Rightarrow \Lambda = \Lambda$.

On dit aussi que la flèche ensembliste est *contravariante* par rapport à son premier argument et *covariante* par rapport au second.[2]

5.2 Saturation et auto-saturation.

La notion suivante est cruciale.

Définition 23 *Soient* $W, S \in P(\Lambda)$. W est S-saturée, $S \subseteq \Lambda$ *si*

$$\forall u, \bar{t} \in \Lambda^{<\omega} \; \forall v \in S \;\; (u[x := v]\bar{t} \in W \Rightarrow (\lambda x.u)v\bar{t} \in W)$$

On dit que W est saturée *si elle est* Λ*-saturée.*
On dira que W *est* auto-saturée *si elle est* W*-saturée.*

L'ensemble des parties S-saturées est clairement clos par intersection quelconque ; il contient $\emptyset$ et il contient Λ (dont on peut considérer que c'est l'intersection de la famille vide d'ensembles S-saturés). Enfin il est clos par $\Rightarrow$, en fait $U \Rightarrow W$ est S-saturé dès que W l'est. Par ailleurs si W est S-saturée et si $S' \subseteq S$ alors W est clairement S'-saturée, en particulier tout ensemble saturé est auto-saturé.

Exemple 24 (Ensembles saturés). *Tout ensemble clos par* β*-expansion de tête*[3], *et a fortiori tout ensemble clos par* β*-expansion ou par* β*-équivalence, est saturé; en particulier les ensembles* $\emptyset, \Lambda, N_g$, *et* N_h *sont saturés, où* N_g *et* N_h *sont respectivement l'ensemble des termes normalisables à gauche et de tête. Ils sont donc aussi auto-saturés. L'ensemble* N_f *des termes fortement normalisables n'est pas saturé.*

Lemme 25 *L'ensemble* N_f *des termes fortement normalisables est auto-saturé.*

Preuve. Supposons que $u[x := v]\bar{t}$ est fortement normalisable; ceci implique que u et tous les t_i le sont. Supposons de plus que v est fortement normalisable et qu'il y a une réduction infinie σ de $(\lambda x.u)v\bar{t}$. Puisque u, v et $\bar{t}$ sont fortement normalisables, σ est forcée de réduire un jour le rédex de tête : il existe des β-réduits $u', v', \bar{t}'$ de $u, v, \bar{t}$ tels que σ se factorise en : $(\lambda x.u)v\bar{t} \rightarrow_\beta (\lambda x.u')v'\bar{t}' \rightarrow_{\beta_0} u'[x := v']\bar{t}' \rightarrow_{\beta_0} \ldots$. Or une suite de réductions partant de $u'[x := v']\bar{t}'$ ne peut pas être infinie, sinon, comme $u[x := v]\bar{t} \rightarrow_\beta u'[x := v']\bar{t}'$, il y aurait une suite infinie de réductions partant de $u[x := v]\bar{t}$; contradiction.

Définition 26 *On dira que* $\mathcal{S} \subseteq P(\Lambda)$ *est* auto-saturé [4] *si pour tous* $S, S' \in \mathcal{S}$ *on a que* S *est* S'*-saturé.*

[2]Cette interprétation correspond à l'interprétation (informelle) intuitioniste de l'implication par Heyting, à savoir : une preuve de $A \rightarrow B$ est une "méthode" (disons, nous, un algorithme) permettant de construire une preuve de B à partir de toute preuve de A.

[3]$U \subseteq \Lambda$ est *clos par* β*-expansion (resp.* β*-expansion de tête)* si $t \in U$ et $t' \rightarrow_\beta t$ (resp. $t' \rightarrow_h t$) impliquent $t' \in U$.

[4]L'introduction de cette notion est intéressante car elle permet une présentation plus uniforme des théorèmes d'adéquation des divers systèmes que nous rencontrerons, ainsi que des preuves de normalisation.

Il est clair que tout sous-ensemble d'un ensemble auto-saturé de $P(\Lambda)$ est auto-saturé, que si $\mathcal{S}$ possède un élément maximum $S_{\max}$ il suffit de demander que tout S de $\mathcal{S}$ soit $S_{\max}$-saturé. On peut noter qu'un ensemble $\mathcal{S} \subseteq \Lambda$ qui est auto-saturé ne contient que des parties auto-saturées de Λ, mais qu'un ensemble de parties auto-saturées n'est évidemment pas forcément auto-saturé. On remarquera enfin que $\mathcal{S} =_{def} \{N\}$ est auto-saturé si et seulement N est auto-saturé.

Exemple 27 *Les ensembles $\mathcal{S}$ suivants sont auto-saturés et tels que $\Lambda, \emptyset \in \mathcal{S}$.*
$P^{sat}(\Lambda)$, l'ensemble de toutes les parties saturées de Λ,
et tous ses sous-ensembles ; en particulier :
$P^{\beta}(\Lambda)$, l'ensemble des parties de Λ closes par β-équivalence
$P^{\exp\beta}(\Lambda)$, l'ensemble des parties de Λ closes par β-expansion.
$P^{\beta\eta}(\Lambda)$, l'ensemble des parties de Λ closes par $\beta\eta$-équivalence.

Exemple 28 *Pour tout $N \subseteq \Lambda$ qui est auto-saturé (ex : $N = N_f$), l'ensemble $P^{N sat}(N)$ des parties N-saturées de N est auto-saturé.*

5.3 Interprétation des types de E_f et E.

Définition 29 *Soit $\mathcal{S} \subseteq P(\Lambda)$. Une* interprétation *$I$ des variables propositionnelles de E dans $\mathcal{S}$ (on parle aussi d'*environnement à valeur dans *$\mathcal{S}$) est une application I de P dans $\mathcal{S}$.*

Définition 30 *$\mathcal{S}$ étant fixé, ainsi que I à valeurs dans $\mathcal{S}$, on étend I à F par induction :*

$$\begin{array}{llll} I(a \to \alpha) & =_{def} & I(a) \to I(\alpha) & \textit{avec :} \\ I(a) & =_{def} & \cap\{I(\alpha) \ / \ \alpha \in a\} \textit{ si } a \neq \emptyset & \textit{et} \\ I(\emptyset) & =_{def} & \Lambda & \end{array}$$

Par convention, $I(\bar{a}) \equiv (I(a_1), ..., I(a_n))$ où $n =_{def} l(\bar{a})$.

Remarque 31 *Si $\mathcal{S}$ est clos par $\Rightarrow$ et par intersection finie, alors $I(\alpha) \in \mathcal{S}$, pour toute formule stricte α et toute I à valeur dans $\mathcal{S}$. Si de plus $\mathcal{S}$. est clos par l'opérateur $\Lambda \Rightarrow [-]$, alors c'est vrai pour tout $\alpha \in F$.*

Remarque 32 *L'interprétation $I(\alpha)$ ne dépend que de α, de $\mathcal{S}$, et de la valeur de I sur les variables propositionnelle qui figurent dans α.*

Notation 33 *$P(N/U)$ dénote l'ensemble des parties de N qui contiennent U, où $N, U \subseteq \Lambda$.*

Le lemme suivant permet de construire des $P(N/U)$, $U \supseteq V$, qui sont clos par intersection finie, par $\Rightarrow$, et eventuellement par l'opérateur $\Lambda \Rightarrow [-]$. Ici V désigne toujours l'ensemble des variables de Λ.

Définition 34 *$N_0 \subseteq \Lambda$ est* bien adaptée *(resp.* fortement adaptée*)*[5] *à $N \subseteq \Lambda$ si elle vérifie les propriétés suivantes :*

1) $V \subseteq N^0 \subseteq N$
2) $(N^0 \Rightarrow N) \subseteq N$
3) $N^0 \subseteq (N \Rightarrow N^0)$ *(resp.* $N^0 \subseteq (\Lambda \Rightarrow N^0)$)

[5]Contrairement à [1], nous ne demandons pas ici de propriété de saturation sur N.

Lemme 35 *Si N^0 est bien adapté à N, alors l'ensemble $P(N/N^0) \subseteq P(N/V)$ et $P(N/N^0)$ est clos par $\Rightarrow$ et par intersection quelconque. Si N^0 est fortement adapté à N, alors $P(N/N^0)$ est de plus clos par $\Lambda \Rightarrow [-]$.*

Preuve. Ce sont des conséquences triviales de l'adaptation (ou de la forte adaptation) de N^0 à N et des propriétés de variance et covariance de $\Rightarrow$.

5.4 Sémantiques ensemblistes adéquates de E et E^f.

Définition 36 *J'appellerai* ensemble adéquat *(sous-entendu :* de candidats de réductibilité) pour E^f *tout $\mathcal{S} \subseteq P(\Lambda)$ qui est clos par $\Rightarrow$ et par intersection finie et qui est auto-saturé. $\mathcal{S}$* sera adéquat pour E *(et E^g) si il contient de plus Λ.*

Lemme 37 *(adéquation). Soit $\mathcal{S} \subseteq P(\Lambda)$ un ensemble adéquat pour E (resp. E^f) et I une interprétation quelconque, à valeur dans $\mathcal{S}$. Alors :*
1. Pour tout type α de E (resp. E^f) on a $I(\alpha) \in \mathcal{S}$.
2. Pour tout t et pour tous $\bar{x}, \bar{u}, \bar{a}, \alpha$ on a :

$$\left.\begin{array}{c} \bar{x} : \bar{a} \vdash t : \alpha \\ \bar{u} \in I(\bar{a}) \end{array}\right\} \Longrightarrow t[\bar{x} := \bar{u}] \in I(\alpha)$$

où $\vdash \; = \; \vdash_E$ (resp. $\vdash_{E_f}$)
3. Si t est clos, ou si $\forall i \; x_i \in I(a_i)$, alors : $\bar{x} : \bar{a} \vdash t : \alpha \Longrightarrow t \in I(\alpha)$

Preuve.
1. suit des propriétés de clôture de $\mathcal{S}$ et du fait que I est à valeur dans $\mathcal{S}$.
3. suit immédiatement de 2.
2. est par induction sur la longueur du terme (ou d'une dérivation du séquent ; ici c'est équivalent).

• Si t est une variable, alors nécessairement $t \equiv x_i \in \bar{x}$ et $\alpha \in a_i$. On a donc $t[\bar{x} := \bar{u}] = u_i \in I(a_i) \subseteq I(\alpha)$.

• Si $t \equiv vw$, alors il existe b tel que $\bar{x} : \bar{a} \vdash v : b \to \alpha$ et $\bar{x} : \bar{a} \vdash_I w : b$. Par hypothèse d'induction $v[\bar{x} := \bar{u}] \in I(b \to \alpha) \equiv I(b) \to I(\alpha)$ et $w[\bar{x} := \bar{u}] \in I(b)$ (l'hypothèse d'induction n'est utilisée que si $b \neq \emptyset$; si $b = \emptyset$ c'est trivial puisque $I(\emptyset) \equiv \Lambda$). Il en résulte que $(vw)[\bar{x} := \bar{u}] \equiv v[\bar{x} := \bar{u}]w[\bar{x} := \bar{u}] \in I(\alpha)$.

• Si $t \equiv \lambda y.v$, alors α est de la forme $b \to \beta$. Supposons que $y \notin \bar{x}$ et que y n'est pas libre dans $\bar{u}$. On a $\bar{x} : \bar{a}, y : b \vdash v : \beta$, donc, par hypothèse d'induction, pour tout $w \in I(b)$ on a $v[\bar{u}/\bar{x}, w/y] \in I(\beta)$. Maintenant $v[\bar{u}/\bar{x}, w/y] = v[\bar{x} := \bar{u}][y := w]$ puisque y n'est pas libre dans $\bar{u}$, et, puisque $\mathcal{S}$ est auto-saturé (et qu'il contient Λ si on s'occupe du système E), $I(\beta)$ est $I(b)$-saturée ; donc $(\lambda y.v[\bar{x} := \bar{u}])w \in I(\beta)$. Comme ceci a lieu pour tout $w \in I(b)$, on a bien $t \in I(b \to \beta) \equiv I(\alpha)$. $\diamond$

Tous les exemples d'ensembles auto-saturés que nous avons donnés dans la Section 5.2 sont trivialement clos par intersection quelconque. Ceux qui sont adéquats pour E^f (resp. E) sont donc exactement ceux qui sont en plus clos par $\Rightarrow$ (resp. et contiennent Λ).

Exemple 38 (ensembles $\mathcal{S}$ adéquats).

Tous les $\mathcal{S}$ de l'Example 27 sont clairement adéquats pour E (et E^f), et ont Λ pour élément maximal.

6 Normalisabilité des termes typables.

Lemme 39 *1. $N_h^0 =_{def} \{x\bar{t} \;/\; x \in V,\ \bar{t} \in \Lambda^{<\omega}\}$ est fortement adapté à N_h.*
2. $N_g^0 =_{def} \{x\bar{t} \;/\; x \in V,\ \bar{t} \in N_g^{<\omega}\}$ est adapté à N_g.
3. $N_f^0 =_{def} \{x\bar{t} \;/\; x \in V,\ \bar{t} \in N_f^{<\omega}\}$ est adapté à N_f.

Démonstration. Il suffit essentiellement de remarquer que pour les trois normalisations considérées, tout terme t tel que tx est normalisable pour une variable x, est lui même normalisable (ceci implique en effet que $N^0 \Rightarrow N \subseteq N$, le reste étant trivial). Pour la forte normalisation c'est évident, et pour les deux autres c'est facile (et déja vu dans le Chapitre 2) [6].

Corollaire 40 *$\mathcal{S}_f =_{def} P^{N_f Sat}(N_f/N_f^0) =_{def} P^{N_f Sat}(\Lambda) \cap P(N_f/N_f^0)$ est adéquat pour E_f.*

Proposition 41 *1. Tout terme typable dans E est normalisable de tête.*
2. Tout terme typable dans E^g est normalisable à gauche.
3. Tout terme typable dans E^f est fortement normalisable.

Démonstration.
• 1. Soit I_h l'interprétation à valeurs dans $\mathcal{S} = P^{sat}(\Lambda)$ qui envoie toute variable propositionnelle sur N_h. Comme $P^{sat}(\Lambda)$ est adéquat pour E le lemme d'adéquation correspondant est valable pour I_h. Par ailleurs, par les Lemmes 39.1 et 35 on aura $I(\alpha) \in P(N_h/N_h^0)$ pour toute formule α, donc $V \subseteq I(\alpha) \subseteq N_h$; il en résulte en particulier que $V \subseteq I(a)$ pour tout $a \in F^*$ (on distingue le cas $a = \emptyset$, pour lequel c'est évident). On conclut avec le lemme d'adéquation.
• 2. Soit I_g l'interprétation à valeurs dans $\mathcal{S} = P^{sat}(\Lambda)$ qui envoie toute variable propositionnelle sur N_g. Comme $P^{sat}(\Lambda)$ est adéquat pour E le lemme d'adéquation correspondant est valable pour I_g. Par les Lemmes 39.2 et 35 on aura pour toute α *stricte* $I(\alpha) \in P(N_g/N_g^0)$, donc $V \subseteq I(\alpha) \subseteq N_g$, et ceci sera encore vrai pour tout $I(a)$, a strict. On conclut avec le lemme d'adéquation.
• 3. On choisit cette fois l'interprétation I_f à valeurs dans $\mathcal{S}_f =_{def} P^{N_f sat}(N_f)$ qui envoie toute variable propositionnelle sur N_f. Par les Lemmes 39.3 et 35 on aura pour toute α stricte $I(\alpha) \in P(N_f/N_f^0)$, donc $V \subseteq I(\alpha) \subseteq N_f$, et ceci sera encore vrai pour tout $I(a)$, a strict. On conclut avec le lemme d'adéquation.

Compte tenu des résultats de la section 4 nous pouvons maintenant conclure :

Théorème 42 *1. Il est équivalent pour un terme d'être normalisable de tête, résoluble, ou typable dans E.*
2. Il est équivalent pour un terme d'être normalisable à gauche, être normalisable, être typable dans E^g.
3. Il est équivalent pour un terme d'être fortement normalisable et d'être typable dans E^f.

[6] Rappel : on distingue deux cas, suivant que t se réduit en tête à une abstraction ou non.

Corollaire 43 *Tout λI-terme normalisable est fortement normalisable.*

Démonstration. En effet, d'après ce qui précède il est normalisable à gauche, et donc typable dans E_f par la Proposition 21. Et donc fortement normalisable d'après ce qui précède.

Exercice 44 *(*Le Théorème de normalisation quasi-gauche).

Une réduction infinie *quasi-gauche* d'un terme t est une suite infinie $t \equiv t_0 \to_{\beta_0} t_1 \to_{\beta_0} \ldots$ telle que pour tout $n \geq 0$ il existe $p \geq n$ tel que t_{p+1} est obtenu en réduisant le redex le plus à gauche de t_p. L'exercice consiste à montrer qu'un terme normalisable n'a pas de réduction quasi-gauche infinie. La démonstration est similaire à celle du Théorème de normalisation gauche, la différence étant qu'on part du couple (N^0_{qg}, N_{qg}) où N_{qg} est l'ensemble de termes qui n'admettent pas de réduction infinie quasi-gauche et $N^0_{qg} \equiv \{x\bar{t}\,/\; x \in V,\; \bar{t} \in N_{qg}^{<\omega}\}$. Il suffit de vérifier que N_{qg} est saturé et que N^0_{qg} lui est bien adapté.

7 La version paresseuse de E.

Le *λ-calcul paresseux* considère qu'il est inutile de calculer une fonction si elle n'est appliquée à aucun argument ; en λ-calcul paresseux toute abstraction est donc considérée comme une forme normale, et dans cette optique il faut impérativement différentier les termes du λ-calcul qui n'ont pas le même ordre de fonctionalité.

L'ordre de fonctionalité d'un terme étant le nombre maximum d'abstraction qui peuvent venir en tête d'un terme au cours de sa réduction de tête (ou d'une réduction quelconque) si un tel nombre existe, et ∞ sinon. Par exemple Ω et YK sont respectivement d'ordre 0 et ∞. Notons qu'un terme d'ordre infini est nécessairement non résoluble.

Le système de typage que nous introduisons maintenant est l'analogue paresseux du système E, c'est aussi le dual d'un modèle du λ-calcul (paresseux ou non), le modèle de Longo [10]. Nous noterons ce système E_L.

Le système E_L a le même ensemble de formules que E mais il distingue l'une des variables propositionnelles, que nous noterons f. La notion de séquents est la même que pour E mais on ajoute la règle :

• $2'$) (funct/abstraction) $\vdash \lambda x.t : f$

qui exprime que toute abstraction est typable de type f (dans le contexte vide). Ce système partage la pluspart des propriétés élémentaires de E (je vous laisse vérifier lesquelles), mais évidemment pas la Propriété 1. de la section 2.3 : le contexte ne mentionne plus nécessairement toutes les variables libres des termes typables dans ce contexte.

Les théorèmes de normalisation prouvables à partir du système E_L sont analogues à ceux que l'on prouve avec E, sauf qu'ils prennent en compte l'ordre de fonctionalité des termes :

Exercice 45 *Le système E_L.*

Montrer que les seuls termes non typables sont les non résolubles d'ordre 0.

Montrer que α est le type d'un non résoluble d'ordre $n \geq 1$ ssi

$\alpha \equiv \bar{a} \rightarrow f$ *pour un $\bar{a}$ tel que $l(\bar{a}) \leq n-1$.*

Montrer que t est d'ordre de fonctionalité n ssi il est typable par tous les types de la forme précédente.

Aide : *on utilisera la technique de réductibilité avec*

$$N_{h,L} \equiv \{\, t \;/\; \exists x \exists u \in \Lambda\ (t \rightarrow_h \lambda x.u)\ \textit{ou}\ \exists x\, \exists \bar{v} \in \Lambda^{<\omega}\ (t \rightarrow_h x\bar{v}) \,\}$$

8 Appendice : Curry-Howard et Intersection de types.

Tout système de typage du λ-calcul, par exemple E et D_Ω, a une logique sous-jacente obtenue en retirant toutes les indications de termes, dont les variables du λ-calcul. Cette logique correspond à un "*calcul de séquents*" qui fait penser au Calcul des Séquents (usuel) de Gentzen en Théorie de la Démonstration. En fait il s'agit ici d'un calcul de séquents dont le style de démonstration sous-jacent n'est pas le Calcul de Séquents mais la "Déduction Naturelle" : pour chaque connecteur les règles seprésentent sous une dualité de la forme intro/élim au lieu de la dualité intro-droite/intro-gauche que l'on trouve dans Gentzen. De plus tout terme typable peut être vu comme le code d'une démonstration de l'un quelconque de ses types ; plus précisément le séquent $\Gamma \vdash t : \alpha$ est dérivable ssi t code une démonstration de α ayant pour hypothèses les prémisses de Γ (Correspondance de Curry-Howard).

Ce codage est *structurel* (le terme est construit inductivement en suivant les étapes de la démonstration), mais il n'est pas forcément injectif : des preuves différentes de α dans le même contexte peuvent très bien être codées par le même terme ; par ailleurs un terme peut avoir plusieurs types dans le même contexte.

Les systèmes de typage d'inspiration sémantique (*Intersection type assignment systems* en anglais), rentrent dans ce cadre ... si on est tolérant au niveau de la logique ; en effet l'usage de la conjonction dans les preuves y est extrêmement limité.

Plus précisément dans ces systèmes on ne peut "démontrer" $\alpha \wedge \beta$ que si α et β admettent des démonstrations suffisamment semblables (codables de la même manière) : il ne suffit pas qu'elles soient démontrables chacune de son côté. En effet on ne peut dériver $\Gamma \vdash t :\ \alpha \wedge \beta$ que si l'on a pû dériver $\Gamma \vdash t : \alpha$ et $\Gamma \vdash t : \beta$.

Dans le cas de E on peut bien sûr arguer en plus qu'on fait un usage déja trop restrictif de la conjonction dès la définition des formules.

En fait le système E tel qu'il est présenté dans ce cours fait seulement un *usage implicite* de la conjonction : $a \neq \emptyset$ devant être compris comme $\bigwedge_{\alpha \in a} \alpha$, et un *usage restreint* :on n'utilise jamais la conjonction à droite d'une flèche ; il s'agit de plus d'une *conjonction finitaire globale* (au lieu de binaire) et *non* *ordonnée*. De même $\emptyset$ joue ici essentiellement le rôle du Ω de [1] et du ω de [6] sauf qu'on ne l'utilise jamais à droite d'une flèche.

Quasiment tous les systèmes d'intersection de type utilisent au contraire l'intersection de façon explicite, complete, binaire et ordonnée. La motivation principale en est de présenter un système qui ressemble le plus possible, au niveau formel, à de la logique usuelle, mais ceci se paye syntaxiquement dans les caractérisations et dans les preuves.

9 Notes bibliographiques.

La bibliographie des systèmes d'intersection de types est extrêmement riche et nous nous contenterons ici de mentionner les articles en rapport direct avec le contenu du chapitre.

Comme nous l'avons déja indiqué dans l'introduction le premier système d'intersection de types est dû à Coppo et Dezani [7], et coincide avec le système E^f de ce cours. le système E coïncide, lui, avec le "système strict" de [3]. Et finalement E et E^f sont la charpente des systèmes D_Ω et D de [1].

Parmi les propriétés des systèmes E et autres qui ne sont pas abordées dans ce cours on peut mentionner les questions de décidabilité et l'existence éventuelle du *typage principal d'un terme* (c'est à dire d'un typage à partir duquel on peut retrouver tous les autres types du terme). On en trouvera l'étude dans [4], ainsi ce qu'une comparaison détaillée entre les systèmes d'intersection de types classiques ou "historiques" ; l'existence d'un typage principal pour le système D est aussi traitée dans [1] et il suffit d'adapter à E_f .

L'article paradigmatique sur la *dualité* des systèmes d'intersection de types et les modèles de filtres est [5, mettre plus tard?] mais le système de typage qui y est étudié est une "η-clôture" de D_Ω. La version non redondante E_η du système de [5] (qui est équivalent au "restricted system" de [6], est étudiée dans [3] sous le nom de "système essentiel". Il est à noter que dans la pluspart des systèmes d'intersection de types introduits dans la littérature tout type d'un terme type aussi tous ses η-réduits, ces systèmes sont donc des extensions de E_η. En fait cette condition est une condition quasiment nécessaire pour que l'on puisse utiliser facilement la technique de réductibilité pour la démonstration d'un théorème d'approximation (fort) correspondant [8].

Les *arguments de réductibilité* actuellement donnés dans la littérature sont issus de la technique de "computabilité" de Tait [11], elle-même issue de la réalisabilité de Kleene (cf. [9]). Elles font toutes intervenir un prédicat ternaire $Comp(\Gamma, t, \alpha)$. La présentation qui en est donnée ici suit la version simplifiée de Krivine [1], où il suffit de jouer avec $\Gamma \vdash t : \alpha$ et $t \in I(\alpha)$.

Références

[1] J.L. Krivine, λ-calcul : Types et Modèles, Masson, 1990.

[2] J.L. Krivine, λ-calculus : Types and Models, Ellis et Horwood, 1993.

[3] S. van Bakel, Complete restrictions of the Intersection Type Discipline, TCS vol.102, pp. 135-163, 1992.

[4] S. van Bakel, Intersection Type Assignment Systems, Theor. Comput. Science 151(2), pp. 385-435, 1995.

[5] H. Barendregt, M.Coppo and M. Dezani-Ciancaglini, A filter λ-model and the completeness of type assignment, JSL vol.48 (4), pp. 931-940, 1983.

[6] M. Coppo, M. Dezani-Ciancaglini and B. Veneri, Functional characters of solvable terms, Zeitschrift für Matematische Logik und Grundlagen der Mathematik, vol. 27, pp.45-58, 1981.

[7] M. Coppo and M. Dezani-Ciancaglini, An extension of the basic functionality theory for the λ-calculus, Notre-Dame Journal of Formal Logic, vol.21 (4), pp.685-693, 1980.

[8] M. Dezani and I.Margaria, A characterization of F-complete type assignments, TCS 45, p.121-157, 1986.

[9] S.C. Kleene, Realizability : a retrospective survey, In Proceedings of the Cambridge Summer School in Math. Logic (Cambridge 1971), Lecture Notes in Mathematics 337, p.95-112, Springer-Verlag.

[10] G. Longo, Set-theoretical models of λ-calculus : theories, expansions and isomorphisms, Annals of Pure and Applied Logic 24, p.153-188, 1983.

[11] W. Tait, Intensional interprétation of functional finite types, SL. 32, 1967 *[compléter]*.