

C. BERLINE

**Rétractions et interprétation interne du polymorphisme
: le problème de la rétraction universelle**

RAIRO. Informatique théorique et applications, tome 26, n° 1 (1992),
p. 59-91

http://www.numdam.org/item?id=ITA_1992__26_1_59_0

© AFCET, 1992, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Informatique théorique et applications » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

RÉTRACTIONS ET INTERPRÉTATION INTERNE DU POLYMORPHISME : LE PROBLÈME DE LA RÉTRACTION UNIVERSELLE (*)

par C. BERLINE (1)

Résumé. – *Le but de cet exposé est de synthétiser en un seul article tous les résultats connus d'existence de « rétractions universelles » (= r.u.) dans des sous-classes intéressantes de l'ensemble R des rétractions d'un domaine de Scott. Ce problème, purement algébrique, est lié à la modélisation du polymorphisme. La solution donnée par Berardi fournit comme sous-produit des modèles non triviaux d'un λ -calcul étendu qui n'est pas Church-Rosser. L'intérêt s'est porté sur des sous-classes de \mathbf{R} après qu'Ershov eût montré, en 1975, que \mathbf{R} elle-même n'avait pas d'objet universel dans $D = P_\omega$.*

Abstract. – *We give a condensed and uniform presentation of all known results (some of them unpublished) concerning the existence of "universal" retractions in interesting subclasses of the set of retractions of a Scott domain. This purely algebraic problem is linked to the modelisation of polymorphism. Berardi's solution provides, as a by-product, non trivial models to a non Church-Rosser extension of λ -calculus.*

INTRODUCTION

Soit D un modèle du λ -calcul pur qui est un domaine de Scott, par exemple P_ω . Les rétractions de D , qui sont les applications continues f de D dans D telles que $f \circ f = f$ (où \circ est la composition) s'identifient avec les points a de D tels que $a \circ a = a$, où cette fois \circ est l'interprétation dans D du terme $\lambda z \lambda y \lambda x z (yx)$ du λ -calcul pur.

On sait depuis Scott [10] interpréter les systèmes de types propositionnels du premier ordre par les rétractions de D , les termes de type A étant interprétés par des éléments de l'image de τ_A , où τ_A est la rétraction de D associée à A . Pouvoir interpréter le polymorphisme est lié au fait de trouver une rétraction « universelle » (r.u.) de D , *i. e.* une rétraction de D dont l'image est

(*) Reçu juin 1990, révisé en novembre 1990.

(1) Équipe de Logique Mathématique, Université Paris-VII, U.E.R. de Mathématiques, 2, place Jussieu, 75251 Paris Cedex 05, France.

l'ensemble \mathbf{R} des rétractions de D (ce qui permet de modéliser la notion de « type des types » (Scott [11]). La question de l'existence d'un D contenant une telle r.u. est ouverte depuis qu'Ersov a montré en 75 qu'il n'y en avait pas dans P_ω (cf. [10], [3], ex. 18.4, ou [4]).

En fait pour modéliser le polymorphisme il suffit de trouver une sous-classe non vide \mathbf{S} de \mathbf{R} stable pour l'interprétation de \rightarrow et \forall et ayant un élément $T \in \mathbf{S}$ -universel, *i.e.* tel que $T \in \mathbf{S}$ et $\mathbf{S} = \text{Im } T$.

Les classes suivantes de rétractions ont, dans de larges sous-classes de la classe des Domaines de Scott, un élément universel :

les clôtures (rétractions $\geq \text{id}$) (Scott [10]),

les rétractions finitaires (Scott [11], McCracken [9]),

les projections finitaires (Amadio, Bruce, Longo [1]),

et enfin :

les rétractions stables (Berardi [4]).

La solution « stable » a permis de plus de fournir le premier modèle non trivial de la théorie universelle LCR obtenue en ajoutant à la logique combinatoire les axiomes :

$$c \circ c = c, \quad \forall x (cx \circ cx = cx), \quad \forall x (x \circ x = x \rightarrow cx = x)$$

(les autres classes correspondent à des restrictions non équationnelles sur le \forall). LCR (le $\Lambda_{\beta\eta}$ de Amadio et Longo [2]) correspond à un λ -calcul qui n'est pas Church-Rosser.

Les classes des clôtures, projections et projections finitaires permettent quant à elles de modéliser une notion de sous-typage.

PRÉSENTATION

Les sections 1 et 2 contiennent tout ce qu'il faut savoir sur les domaines de Scott et les fonctions continues, la section 10 est l'analogue pour les DI-domaines de Berry et les fonctions stables. Toutes les affirmations sans démonstration sont des conséquences plus ou moins immédiates des définitions ou des faits précédents.

Dans la section 4, sont définies les rétractions, clôtures (rétractions $\geq \text{id}$) et projections (rétractions $\leq \text{id}$).

Dans la section 7, nous étudions les éléments compacts des rétracts $a(D)$ des domaines de Scott [$a(D)$ est l'image de D par la rétraction a] et donnons une première caractérisation des rétractions finitaires, qui sont par définition

les rétractions pour lesquelles $a(D)$ est Scott. En section 6 on montre que tout treillis complet admet une clôture universelle et nous étudions en parallèle le cas des projections : il y a une application qui a tous les attributs d'une projection universelle sauf un : la continuité ; la définition de cette « projection », et la raison pour laquelle elle ne marche pas vraiment, est une bonne introduction aux définitions des objets universels que nous verrons par la suite.

En sections 8 et 9 est montrée l'existence de projections et rétractions finitaires universelles et en section 11 celle d'une rétraction stable universelle. Ces trois cas présentent beaucoup de similarités, surtout les deux derniers. Le point crucial est que les rétractions finitaires sont exactement les sups dirigés de rétractions compactes (même chose pour les rétractions stables et les projections finitaires).

Après tant d'algèbre il est temps de revenir au λ -calcul.

Dans la section 3 nous introduisons les modèles continus du λ -calcul pur (leurs analogues stables seront introduits en section 10).

L'interprétation du polymorphisme dans les modèles du λ -calcul pur est faite en section 5. Pour l'interprétation des systèmes de typage du premier ordre et du sous-typage [10] est infiniment plus complet ; pour l'interprétation du polymorphisme dans des modèles du λ -calcul typé, voir [1].

Nous avons choisi de travailler ici dans des modèles du λ -calcul pur car l'interprétation du polymorphisme y est plus légère, puisqu'il n'y a pas d'espaces (explicites) de fonctions hiérarchisées : tout se passe à l'intérieur du même domaine ; de plus ils interviennent nécessairement pour modéliser LCR.

Cela dit, le problème de la rétraction universelle se pose au niveau plus général d'un domaine de Scott quelconque (ou d'un DI -domaine quelconque, suivant le cas). La r.u. est alors une rétraction de l'espace des fonctions continues (resp. stables) de D et n'est plus assimilable à un élément de D . Il devrait être clair dans les sections sur l'existence des objets universels, qu'elles peuvent être lues dans ce cadre plus général, et que l'hypothèse supplémentaire sur D n'est utilisée que dans un souci d'allègement typographique. Dans le même souci d'allègement typographique nous avons écrit ces sections dans le cas où D est extensionnel ; la modification pour passer au cas non extensionnel est minime : remplacer u par $\text{fun } u$ dans les ensembles dont les sups servent à définir les rétractions universelles.

1. DOMAINES DE SCOTT

Soit D un ensemble partiellement ordonné (p.o.).

Un sous-ensemble A de D est dit *dirigé* (ou filtrant à droite) si deux éléments quelconques de A sont majorés par un élément de A , en particulier toute chaîne est dirigée.

Un élément y de D est « fini » ou *compact* si pour tout sous-ensemble dirigé A de D on a : $y \leq \sup A \Rightarrow \exists x \in A \ y \leq x$. On notera D_c l'ensemble des éléments « finis » ou compacts de D et **dans tout l'article les lettres h, k, l désigneront uniquement des éléments de D_c .**

D est dit *algébrique* si pour tout $x \in D$ l'ensemble A_x des éléments « finis » de D qui sont plus petits que x est dirigé et $x = \sup A_x$.

DÉFINITION : Un *domaine de Scott* D est un ensemble partiellement ordonné, avec plus petit élément, noté \perp , et tel que :

- (i) tout sous-ensemble dirigé A de D admet une borne supérieure,
- (ii) tout sous-ensemble majoré C de D admet une borne supérieure (et par conséquent tout sous-ensemble non vide de D a une borne inférieure),
- (iii) D est algébrique.

[On parle de *dcpo* pour un D qui ne vérifie que la condition (i).]

Remarque : En présence de (ii) les ensembles A_x sont automatiquement dirigés et (iii) est équivalente à : $x = \sup A_x$ pour tout x .

Fait 1.0 : L'algébricité de D implique : $x \leq y \Leftrightarrow \forall h (h \leq x \rightarrow h \leq y)$.

Fait 1.1 : Si D est dénombrable (i) est équivalent à (i') : toute suite croissante d'éléments de D a une borne supérieure; de même, y est compact ssi pour toute suite croissante d'éléments x_n de D , $y \leq \sup(x_n) \Leftrightarrow \exists n \ y \leq x_n$. Il en est de même si D est algébrique et D_c dénombrable. Dans le cas non dénombrable il faut considérer des suites ordinales.

Fait 1.2 : Si D et D' sont Scott, $D \times D'$, muni de l'ordre produit, l'est également.

On peut penser chaque élément de D comme un ensemble d'informations à propos d'une entité à connaître (ex : entier, fonction de \mathbb{N} dans \mathbb{N}), l'ordre sur D étant l'inclusion, et voir un sous-ensemble majoré d'éléments de D comme un ensemble d'ensembles d'informations compatibles; nous garderons cette terminologie d'*éléments compatibles* de D dans le cas général. La notion de compacité est une modélisation de la notion d'information finie.

Fait 1.3 : Si h_1 et h_2 sont deux compacts compatibles d'un domaine de Scott, alors $h_1 \vee h_2$ est encore compact.

Exemples : Rappelons ici qu'un *treillis* est par définition un p.o. dont tout couple d'éléments a une borne sup et une borne inf, et qu'un treillis est dit *complet* si tout sous-ensemble (fini ou infini, et éventuellement vide) a une borne sup (et donc une borne inf). Tout treillis complet et algébrique est donc un domaine de Scott. C'est le cas en particulier de $P(\omega)$, ordonné par l'inclusion, ses éléments finis étant exactement les parties finies de ω . La classe suivante de treillis algébriques complets permet de visualiser tous les modèles de Scott connus qui sont des modèles du λ -calcul pur de façon analogue à $P(\omega)$ (Krivine [7]) :

D est l'ensemble $S(\Delta)$ des segments initiaux d'un ensemble pré-ordonné (Δ, \leq) (un préordre est une relation binaire réflexive et transitive). Un *segment initial* de Δ est une partie u de Δ telle que : $\forall x \in u \forall y \leq x \ y \in u$. D est ordonné par l'inclusion. $P(\omega)$ correspond au cas où Δ est ω muni du préordre trivial : l'égalité.

Tous ces $D = S(\Delta)$ sont des treillis complets ($\vee = \cup$ et $\wedge = \cap$) et algébriques, les éléments « finis » de D étant les segments initiaux de D engendrés par un nombre fini d'éléments :

$$u \in D_c \quad \text{ssi} \quad \exists n \exists \alpha_1 \dots \alpha_n \in \Delta \quad (u = \{x / \exists i \leq n \ x \leq \alpha_i\}).$$

Le lecteur intéressé par les modèles du λ -calcul *pur* pourra sans inconvénient se restreindre aux domaines de ce type, qui présentent cet avantage, parmi d'autres, que tous les éléments sont compatibles (et Δ est plus grand élément).

2. FONCTIONS CONTINUES

Soient D et D' deux dcpo (po + existence de sups dirigés). A partir de 2.3 D sera un domaine de Scott.

Toutes les fonctions considérées dans l'article seront des fonctions continues.

DÉFINITION : $f: D \rightarrow D'$ est *continue* si elle commute aux sups dirigés, i.e. :

$$\forall A \subseteq D, \quad A \text{ dirigé}, \quad A \neq \emptyset, \quad f(\sup A) = \sup f(A).$$

Ce qui implique que $f(A)$ a un sup si A est dirigé, que f est croissante (prendre $A = \{x, y\}$ avec $x \leq y$), et que $f(A)$ est dirigé (resp. borné) si A l'est; la restriction $A \neq \emptyset$ correspond au fait que l'on exige pas $f(\perp) = \perp$. Il revient

au même de dire que f est continue ssi elle commute avec tous les sups de suites croissantes (ordinales dans le cas non dénombrable). Enfin, si D est algébrique, f est continue ssi :

$$\forall x \in D, f(x) = \sup f(A_x) \text{ où } A_x \text{ est l'ensemble dirigé des éléments}$$

compacts h de D tels que $h \leq x$.

Noter que, même si D et D' sont Scott, f n'a aucune raison de commuter avec les sups d'ensembles bornés.

Les fonctions continues sont closes par composition.

On note $[D \rightarrow D']$ l'ensemble des fonctions continues de D dans D' , muni de l'ordre partiel suivant, qu'on appelle souvent *ordre extensionnel* :

$$f \leq h \quad \text{ssi} \quad \forall x \in D, f(x) \leq g(x).$$

Fait 2.1 : Soit D Scott, F un ensemble de fonctions continues compatibles de $[D \rightarrow D']$ et g la fonction définie par : $g(x) = \sup \{f(x) | f \in F\}$, alors g est continue et donc $g = \sup F$; il en est de même si F est un ensemble dirigé.

Fait 2.2 : $[D \rightarrow D']$ est un dcpo et, si D et D' sont Scott, $[D \rightarrow D']$ est Scott (le seul point non trivial est l'algébricité de $[D \rightarrow D']$, c'est une conséquence des faits 2.5 et 2.4 plus loin : nous préciserons d'abord quels sont les éléments compacts de $[D \rightarrow D']$).

Applications continues compactes

Soit D Scott et $\varepsilon_{h,k}$ la fonction définie sur D par : $\varepsilon_{h,k}(x) = k$ si $x \geq h$, et \perp sinon ($h, k \in D_c$). Il est facile de vérifier :

Fait 2.3 : $\varepsilon_{h,k}$ est continue et compacte.

Fait 2.4 : $\forall f \in [D \rightarrow D] \quad f = \sup \{ \varepsilon_{h,k} | (h, k \in D_c), k \leq f(h) \}$.

(le sup de droite existe et est $\leq f$ car tous ses éléments sont $\leq f$, f est égal au sup car D est algébrique).

Fait 2.5 : Les fonctions continues compactes sont exactement les sups finis possibles de fonctions $\varepsilon_{h,k}$ (le sup existe quand les k_i correspondant à des h_i compatibles sont également compatibles). En particulier :

Fait 2.6 : Si ε est une fonction compacte alors $\varepsilon(D)$ est fini et inclus dans D_c (réciproque fausse).

Démonstration de 2.5 : Dans un dcpo avec sups bornés les sups finis d'éléments compacts sont compacts (version plus précise de 1.3), donc les

sup_s finis de fonctions $\varepsilon_{hi, ki}$ compatibles sont compacts; réciproquement si ε est compacte, par 2.4 et le fait que les sup_s finis de fonctions $\varepsilon_{h, k}$ plus petites que ε forment un ensemble dirigé, ε est l'un de ces sup_s finis.

Fait 2.7: Les fonctions compactes sont closes par composition (exercice, utiliser 2.5).

Traces

2.8. On définit la trace de f comme le sous-ensemble suivant de $D_c \times D_c$:

$$\text{Tr}f = \{ (h, k) / k \leq f(h) \} = \{ (h, k) / \varepsilon_{h, k} \leq f \}.$$

$\text{Tr}f$ détermine f (car D est Scott): un avatar de 2.4 est:

$$f(x) = \sup \{ k / \exists h \leq x (h, k) \in \text{Tr}f \}.$$

2.9. Il est facile de vérifier que:

$$\begin{aligned} f \leq g &\Leftrightarrow \text{Tr}f \subseteq \text{Tr}g, \text{ et} \\ f = \sup H &\Leftrightarrow \text{Tr}f = \bigcup \text{Tr}g, \quad g \in H, \quad (H \text{ dirigé}). \end{aligned}$$

2.10.

$$\text{Tr}g \circ f = \{ (h, k) / (h, l) \in \text{Tr}f \text{ et } (l, k) \in \text{Tr}g \}$$

Remarques: Comme dans tout cet article les lettres h, k, l désignent des éléments « finis », mais h fini n'implique pas $f(h)$ fini, ce qui explique le choix de \leq au lieu de $=$ dans la définition de la trace.

La trace que nous utilisons ici n'est pas tout à fait la trace de Scott, où l'on demande que k soit une sorte de singleton (vrai singleton si $D = P_\omega$, segment initial engendré par un seul élément pour les autres modèles, dans la présentation de Krivine). nous avons choisi cette trace-ci parce qu'elle se comporte simplement pour la composition.

2.10.5. Soit $A \subseteq \text{Tr}f$. Il y a une plus petite fonction continue g telle que $A \subseteq \text{Tr}g$, à savoir: $g = \sup \{ \varepsilon_{h, k} / (h, k) \in A \}$. Si A est fini g est compacte; en particulier si $\text{Tr}f$ est finie f est compacte; la réciproque est fautive (considérer la trace de $\varepsilon_{h, k}$).

Fonctions n -aires

DÉFINITION : $f: D^n \rightarrow D'$ est continue si elle est continue par rapport à chaque argument. Exemple : $\circ: [D \rightarrow D]^2 \rightarrow [D \rightarrow D]$ est continue, et en particulier croissante en ses deux arguments.

2. 11. f , binaire, est continue ssi pour tous ensembles dirigés A et B on a : $f(\sup A, \sup B) = \sup \{f(a, b) / a \in A, b \in B\}$; ceci implique facilement que si A et B sont indexés par le même ensemble d'indices I ,

$$f(\sup A, \sup B) = \sup \{f(a_i, b_i) / i \in I\}.$$

En conséquence :

2. 12. Les fonctions continues sont closes par substitutions.

Exemple : les fonctions $p_n: f \rightarrow f \circ \dots \circ f$ (n fois) sont continues.

2. 13. Inf: \wedge est continue sur D^2 : si x et $y = \sup(y_n)$ sont compatibles et la suite y_n croissante, alors : $x \wedge y = \sup(x \wedge y_n)$: \geq vient de la croissance évidente de \wedge , pour montrer \leq il suffit de voir que tout compact h inférieur au premier membre, l'est aussi au second (1. 0). Soit donc $h \leq x, y$; comme h est compact, il est plus petit que l'un des y_n et donc que $x \wedge y_n$ pour cet n , donc $h \leq \sup(x \wedge y_n)$.

2. 14. Infs: Il est facile de déduire de 2. 13 l'analogie de 2. 1 pour les infs finis : si f_1 et f_2 sont deux fonctions continues, la fonction f , définie point par point par $f(x) = f_1(x) \wedge f_2(x)$, est continue et donc égale à : $f_1 \wedge f_2$.

3. Modèles du λ -calcul pur

Les modèles du λ -calcul pur que nous considérons ici (et jusqu'à la section 9) sont essentiellement des domaines de Scott tels que toutes les fonctions continues de D dans D sont codées de façon continue (et compatible avec le λ -calcul) par des éléments de D . Ces modèles sont des λ -modèles au sens de [Barendregt p. 95], et nous les appellerons c - λ -modèles. Nous verrons qu'on peut interpréter dans ces modèles non seulement les termes du λ -calcul mais aussi les types, termes typés, sortes (types de types), etc. Leurs analogues « stables » (ou s - λ -modèles) seront introduits en 10. 15.

DÉFINITION : Un c - λ -modèle est un triplet (D, app, λ) où :

- (i) D est un domaine de Scott,
- (ii) app est une fonction continue de $D \times D$ dans D (l'« application »),
- (iii) λ est une fonction continue de $[D \rightarrow D]$ dans D .

Nous désignerons $\text{app}(a, b)$ par ab (ou des variantes avec parenthèses si cela s'avère nécessaire) et $\lambda(f)$ par $\lambda x.f(x)$ (x est donc variable muette dans cette expression). Les deux fonctions λ et app sont reliées par la relation :

$$(iv) \quad \forall f \in [D \rightarrow D], \quad \forall b \in D, \quad (\lambda x.f(x))b = f(b),$$

Un c - λ -modèle modélise donc clairement le λ_β -calcul. C'est un modèle *extensionnel* (ou modèle du $\lambda_{\beta\eta}$ -calcul) si il vérifie de plus :

$$(v) \quad \forall a \in D, \quad \lambda x.ax = a.$$

3.0. Il résulte de (i) que pour tout $a \in D$ la fonction $\varphi_a: x \rightarrow ax$ est dans $[D \rightarrow D]$ et que la fonction $\varphi: a \rightarrow \varphi_a$ est continue de $[D \rightarrow D]$ dans D ; (iv) s'écrit alors $\varphi \circ \lambda = \text{id}_{[D \rightarrow D]}$ (ainsi λ est injective et réalise un isomorphisme de $[D \rightarrow D]$ sur son image) et (v) s'écrit $\lambda \circ \varphi = \text{id}_D$ (et λ est bijective).

3.1. Disons que $a \in D$ *représente, ou code, la fonction* $f \in [D \rightarrow D]$ si $a = \lambda x.f(x)$. On a alors: $ay = (\lambda x.f(x))y = f(y)$ pour tout y de D , donc $f = \varphi_a$ et on a: $a = \lambda x.ax$. Réciproquement: si $a \in D$ est tel que $a = \lambda x.ax$, alors a représente bien évidemment la fonction φ_a . En résumé les éléments a de D qui représentent des éléments de $[D \rightarrow D]$ sont exactement ceux qui vérifient l'équation $a = \lambda x.ax$ (et donc tous les éléments de D dans le cas extensionnel). J'appellerai ces éléments «*fonctions de D*» par la suite.

3.2. Définissons dans $D: a \circ b = \lambda x.a(bx)$. Si a et b sont deux fonctions qui représentent f et g respectivement, alors $a \circ b$ est la fonction de D qui représente $f \circ g$.

3.3. *Extensionnalité faible*: Si a et b sont deux fonctions et si $\forall x \in D, ax = bx$, alors $a = b$ (évident). Autrement dit D satisfait l'axiome d'extensionnalité restreint aux fonctions.

3.4. Si a et b sont des fonctions alors $a \leq b \Leftrightarrow \forall x ax \leq bx \Leftrightarrow \varphi_a \leq \varphi_b$.

Cette remarque relie l'ordre de $[D \rightarrow D]$ et l'ordre des fonctions de D ; c'est ici qu'apparaît la différence avec les s - λ -modèles (où l'on perd \Leftarrow ; cf. 10.15), cela dit 3.4 ne sera pas utilisée dans la section 7, ce qui nous permettra d'utiliser les résultats de cette section dans le cadre stable.

3.5. Pour toute fonction $a \in D$ on définit :

$$\text{Im } a = aD = \{ ax/x \in D \}$$

et

$$\text{Fix } a = \{ x/ax = x \}.$$

$\text{Im } a$ et $\text{Fix } a$ sont exactement l'image et l'ensemble de points fixes de la fonction de $[D \rightarrow D]$ représentée par a . La continuité de cette fonction implique que $\text{Fix } a$ est clos par les sups dirigés (pour l'ordre induit par celui de D) et en particulier par les sups de chaînes. Si D est un treillis complet $\text{Fix } a$ en est un aussi (théorème de Tarski) mais pour les sous-ensembles non dirigés de $\text{Fix } a$ il n'y a aucune raison que les sups dans D et dans $\text{Fix } a$ coïncident.

3.6. *Traces*: Pour toute fonction a de D , $\text{Tr } a$ sera par définition la trace de la fonction codée par a .

3.7. *Notation*. xyz est par définition $(xy)z$.

3.8. *Fonctions binaires*: Si g est une fonction binaire continue sur D , elle est représentée dans D par l'élément $a = \lambda x. \lambda y. g(x, y)$ (on utilise d'abord qu'à x fixé $g(x, y)$ est continue, puis que $x \rightarrow \lambda y. g(x, y)$ est continue), et on a pour tous x, y : $g(x, y) = (ax)y$.

3.9. *Puissances n -ièmes*: Rappelons que p_n est la fonction continue de $[D \rightarrow D]$ dans $[D \rightarrow D]$ qui élève une fonction à sa puissance n -ième. Soit $\pi_n = \lambda \circ p_n \circ \varphi$; π_n est une fonction continue de D dans D (composée de trois fonctions continues) et

$$\forall a \in D$$

$$\pi_n(a) = \lambda(p_n(\varphi_a)) = \lambda(\varphi_a^n) = \lambda x. \varphi_a^n(x) = \lambda x. a(a(\dots(ax))\dots)$$

$$(n \text{ fois}) = a \circ a \circ \dots \circ a \text{ (} n \text{ fois),}$$

que nous abrègerons en a^n . Ainsi $\pi_n(a) = a^n$.

4. RÉTRACTIONS (ET SOUS CLASSES)

D est un c - λ -modèle.

DÉFINITION : $a \in D$ est une *rétraction* si $a \circ a = a$; a est donc en particulier une fonction de D et réciproquement une fonction est une rétraction ssi $aD = \text{Im } a \subseteq \text{Fix } a$ (et alors $aD = \text{Fix } a$).

Fait 4.1: aD est un dcpo; de plus: $\sup_{aD} A = \sup_D A$ pour tout sous-ensemble dirigé A de aD .

aD a un plus petit élément : $a \perp$.

aD a des sups bornés et si C est un ensemble d'éléments compatibles de aD on a : $\sup_{aD} C = a \sup_D C$. Mais :

aD n'est en général pas Scott : il vérifie seulement une forme faible d'algébricité : la « continuité » (un dcpo D' est *continu* si tout élément x de D' est sup dirigé des éléments y de D' tels que $y \ll x$, où $y \ll x$ signifie : si A est dirigé et $\sup A \geq x$ alors $\exists z \in A \ y \leq z$; noter que $y \ll x$ implique $y \leq x$ et que pour y compact il y a équivalence).

Les rétractions a de D telles que aD est Scott sont appelées *finitaires* et seront étudiées en sections 7 et 10; on verra que ce sont exactement les sups dirigés de rétractions compactes.

Exemples : $\varepsilon_{h,k}$ est une rétraction ssi $k \geq h$, et $\varepsilon_{h,k} D = \{\perp, k\}$ (nous gardons la même notation pour ces fonctions et leurs codes).

$\text{id} = \lambda x. x$ est un exemple trivial de rétraction finitaire non compacte et $\text{id} D = D$; $\text{fun} = \lambda u. \lambda x. ux$ est une rétraction dont l'image est exactement l'ensemble des fonctions de D (et D dans le cas extensionnel), qui est lui-même isomorphe à $[D \rightarrow D]$, fun est donc finitaire.

Une rétraction $\geq \text{id}$ est appelée une *clôture*; nous verrons plus tard que toute clôture est finitaire (démonstration directe facile). Une rétraction $\leq \text{id}$ est appelée une *projection*. Les clôtures et les projections ont des comportements en partie symétriques, la différence apparaissant au niveau de la continuité et de l'algébricité, dont les définitions sont liées de façon essentielle à l'orientation de l'ordre (en particulier les projections n'ont pas de raison d'être toujours finitaires).

4.2. L'ensemble \mathbf{R} des rétractions est l'ensemble des points fixes de la fonction $\lambda x. x \circ x$, donc \mathbf{R} est clos par sups dirigés (3.5) et il en est de même de l'ensemble \mathbf{P} des projections et de l'ensemble \mathbf{C} des clôtures.

Si D est un treillis complet \mathbf{R} en est un également. Le problème de la rétraction universelle commence donc par la question :

\mathbf{R} est-il dans ce cas un treillis continu?

Ersov a montré que c'était faux pour P_ω et que la raison en était que P_ω contenait des rétractions non extensives (cf. [10]).

Une rétraction a est *extensive* si $a = \lambda x. \perp$ ou $\exists h \in D_c \ h \neq \perp$ et $h \leq ah$. En particulier toutes les clôtures sont extensives. Nous verrons que toutes les rétractions finitaires et dont les rétractions stables sont aussi extensives (cf. 7.3).

Sous-rétractions

Les rétractions ne sont en général pas déterminées par leurs images ($a \rightarrow aD$ n'est en général pas injective). C'est pourtant le cas si on se restreint, soit aux projections, soit aux clôtures. Cela découle des faits suivants dont la démonstration, immédiate, n'utilise pratiquement que la croissance de \circ .

Faits 4.3. (1) Soient a et b deux rétractions; on a :

$$\begin{aligned} \text{Im } a \subseteq \text{Im } b &\Leftrightarrow b \circ a = a \\ a \leq b &\Rightarrow b \circ a \geq a \quad \text{et} \quad a \circ b \leq b. \end{aligned}$$

(2) Si a et b sont des projections,

$$a \leq b \Leftrightarrow b \circ a = a \quad (\Leftrightarrow b \circ a = a \text{ et } a \circ b = a).$$

(3) Si a et b sont des clôtures,

$$a \leq b \Leftrightarrow a \circ b = b \quad (\Leftrightarrow a \circ b = b \text{ et } b \circ a = b).$$

COROLLAIRE 4.4 : Si a et b sont deux projections, on a : $a \leq b \Leftrightarrow \text{Im } a \subseteq \text{Im } b$.

Si a et b sont deux clôtures, on a : $a \leq b \Leftrightarrow \text{Im } a \supseteq \text{Im } b$.

Dans les deux cas : $a = b \Leftrightarrow \text{Im } a = \text{Im } b$.

L'ordre partiel sur \mathbf{R} adapté à la notion de sous-typage est l'ordre $\circ \leq$ défini par : $a \circ \leq b$ ssi $a = a \circ b = b \circ a$ ($a \circ \leq b$ signifie exactement que $\text{Im } a \subseteq \text{Im } b$ et que a est une rétraction de $\text{Im } b$); ce qu'on vient de voir c'est que $\circ \leq$ coïncide avec \leq sur les projections, et avec \geq sur les clôtures! donc que ces deux classes sont particulièrement bien adaptées à la modélisation du sous-typage (*mais voir section 5 plus loin*).

4.4. *Unicité de « la » rétraction universelle?*: Il résulte de 4.3 que les objets universels qu'on va trouver dans les classes des clôtures, projections, projections finitaires, seront uniques. Qu'en est-il des rétractions finitaires ou stables?

5. INTERPRÉTATION DU POLYMORPHISME

L'existence d'une rétraction universelle à l'intérieur d'une sous-classe \mathbf{S} de \mathbf{R} permet de modéliser [Type : Type] («être un type» est un type) en interprétant les types par des éléments de \mathbf{S} (par leurs images en fait). Il en résulte qu'on peut modéliser F , F_ω , et le calcul des constructions (*cf.* par exemple [1] ou [4]).

Nous nous contenterons ici de modéliser le système F de Girard, dont nous donnerons une présentation assez succincte : pour les démonstrations des lemmes syntaxiques nous renvoyons à [7], et pour une présentation générale du système, avec une sémantique différente, à [6].

L'intérêt de considérer des rétractions au lieu de simples fonctions vient de ce que leur image a une bonne structure, proche de celle de D . Il est à signaler d'autre part que le pouvoir expressif des rétractions est grand, plus grand que ce qui est traité ici puisque les connecteurs \wedge , \vee et les types de données, même définis de façon récursive, sont interprétables par des rétractions, et qu'elles reflètent une notion de sous-typage, le tout de manière parfaitement cohérente ([10] pour le premier ordre).

En revanche on n'a pas correction des données (la réciproque de 5.3 plus loin, pour les types de données) : Pour avoir cette correction les données il faut utiliser une classe plus large que celle des rétractions : celle des relations d'équivalence partielles (Plotkin, [10], p. 564, pour P_ω et le type des entiers). Les relations d'équivalence partielles apportent également la solution d'un autre problème, posé par les concepteurs de langages, car elles permettent de modéliser une notion de sous-typage qui rend la flèche contravariante en son premier argument, alors qu'avec les rétractions on a covariance (comme il est facile de le vérifier avec 5.3).

Le système F

Les types (encore appelés types variables)

Ce sont les formules du calcul propositionnel du 2nd ordre, construites avec \rightarrow et \forall sur des variables de types α, β, \dots . Les types seront notés σ, τ, \dots ; $\sigma[\alpha]$ signale qu'on s'intéresse aux apparitions libres de α dans σ et $\sigma[\tau/\alpha]$ désigne le type obtenu en remplaçant toutes les occurrences libres de α dans σ par τ , après renommage éventuel des variables liées de σ de manière qu'aucune variable libre de τ ne soit liée (par \forall) dans σ .

Les termes typés

Ils sont construits par induction à partir de variables typées x^σ, y^τ, \dots , et des variables de type, à l'aide de quatre règles et possèdent tous un type unique que nous préciserons au moment de la construction : « $M:\tau$ » abrègera : « M est un terme typé et son type est τ ». $M[x^\sigma]$ et $M[\alpha]$ signalent qu'on s'intéresse aux apparitions libres de x^σ ou α dans M , $M[N/x^\sigma]$ désigne le terme typé obtenu en remplaçant dans M toutes les occurrences libres de x^σ par le terme typé N de type σ , et $M[\sigma/\alpha]$ le terme obtenu en remplaçant les

occurrences libres de α par σ , après renommage éventuel des variables liées de M de manière qu'aucune variable libre de N ou σ ne soit liée dans M (par \forall ou λ).

Construction des termes typés :

$$x^\sigma : \sigma$$

$$M : \tau \Rightarrow \lambda x^\sigma . M : \sigma \rightarrow \tau$$

$$M : \sigma \rightarrow \tau \text{ et } N : \sigma \Rightarrow MN : \tau$$

$M : \sigma \Rightarrow \forall \alpha M : \forall \alpha \sigma$ (sous la condition supplémentaire que si x^α est libre dans M , alors α n'est pas variable libre dans τ).

$$M : \forall \alpha \tau \Rightarrow M \sigma : \tau[\sigma/\alpha].$$

5.0. LEMME : $M[x^\sigma] : \tau$ et $N : \sigma \Rightarrow M[N/x^\sigma] : \tau$, de même

$$M[\alpha] : \tau \Rightarrow M[\sigma/\alpha] : \tau[\sigma/\alpha].$$

5.1. Sur les termes typés on a la relation de réduction suivante :

$$(\lambda x^\sigma . M) N \text{ red } M[N/x^\sigma] \quad \text{si } N \text{ est de type } \sigma,$$

et,

$$(\forall \alpha M) \sigma \text{ red } M[\sigma/\alpha]$$

qui préserve le type, comme il est très facile de le vérifier en utilisant 5.0, et qui donne un sens opérationnel au λ -calcul typé. Nous noterons \equiv la relation d'équivalence engendrée par red, c'est-à-dire la plus petite relation d'équivalence qui contient les couples de red.

Sémantique dénotationnelle de F au moyen des rétractions

On travaille dans c - λ -modèle D , les modifications pour adapter ce qui suit au cadre stable à l'aide des sections 10 et 11 sont minimales.

On suppose que c est une rétraction « S -universelle» de $\mathbf{S} \subseteq \mathbf{R}$: $c \in \mathbf{S}$ et $\mathbf{S} = cD$. Nous ne supposons pas systématiquement dans cette section que D est extensionnel : ça n'offre ici aucun intérêt; cependant l'extensionnalité est, en partie, nécessaire si on choisit pour \mathbf{S} la classe des projections ou celle des clôtures, nous verrons pourquoi en 5.4.

Nous interpréterons les types et les termes typés de F dans des environnements. Un *environnement* ρ est une application de $V \cup X$ dans D , où V est l'ensemble des variables de type et X celui des variables typées, et qui est

soumise à des restrictions naturelles. La première de ces restrictions :

$$\text{env (i)} \quad \forall \alpha \in V, \quad \rho(\alpha) \in S$$

nous permettra d'interpréter tout type σ par un élément σ_ρ^* de S (5.7). La deuxième condition peut alors s'énoncer :

$$\text{env (ii)} \quad \forall x^\sigma \in X, \quad \rho(x^\sigma) \in \sigma_\rho^* D$$

et nous permettra d'interpréter tout terme typé $M : \sigma$ par un élément M_ρ^* de $\sigma_\rho^* D$ (5.10). Enfin notre interprétation sera dénotationnelle : deux termes équivalents pour \equiv auront même valeur dans D (5.11). En résumé nous aurons :

THÉORÈME 5.2 : *Pour tout environnement ρ , pour tout type σ , et tous termes typés M, N :*

- (1) $\sigma_\rho^* \in \mathbf{S} = cD \subseteq \mathbf{R}$
- (2) $M : \sigma \Rightarrow M_\rho^* \in \sigma_\rho^* D$
- (3) $M \equiv N \Rightarrow M_\rho^* = N_\rho^*$.

Les deux premiers résultats se démontrent par induction sur la construction du type (respectivement : du terme typé); on doit donc pouvoir disposer dans D de reflets de \rightarrow et \forall , ce qui conduit aux définitions et notations suivantes, qui apparaîtront naturelles dès le lemme 5.4 :

5.3. DÉFINITIONS : Pour $a, b \in \mathbf{R}$, $F \in [cD \rightarrow cD]$ on définit :

- (i) $a \rightarrow b = \lambda z. b \circ z \circ a$,
- (ii) $\forall x : c F(x) = \lambda z. \lambda x. (F(cx) \circ z \circ c) x$.

Pour être plus suggestif on peut introduire les notations suivantes, pour $u \in D$ et $f \in [D \rightarrow D]$:

- (iii) $u : a$ si $u \in aD$ et $a \in cD$ (lire : « u est de type (sémantique) a »),
- (iv) $\lambda x : a. f(x) = \lambda x. f(ax)$,

et réécrire (ii) en :

- (ii') $\forall x : c F(x) = \lambda z. \lambda x : c. (F(x))(zx)$.

LEMME 5.4 : *Soient $a, b \in \mathbf{R}$ et $F \in [cD \rightarrow cD]$.*

- (a) $a \rightarrow b$ et $\forall x : c F(x)$ sont des rétractions,
- (b) $(a \rightarrow b)D$ est isomorphe à $[aD \rightarrow bD]$ et $(\forall x : c F(x))D$ est isomorphe au sous-espace E de $[cD \rightarrow cD]$ défini par $E = \{g \in [cD \rightarrow cD] / \forall u \ g(u) \in F(u)D\}$,
- (c) Si a, b, c et tous les éléments de cD sont des rétractions finitaires, alors $a \rightarrow b$ et $\forall x : c F(x)$ sont aussi finitaires,

(d) Si D est extensionnel, si a, b, c et tous les éléments de cD sont des projections (respectivement des clôtures), alors $a \rightarrow b$ et $\forall x : cF(x)$ en sont aussi.

Démonstration: (a) $(a \rightarrow b)z = b \circ z \circ a = b \circ b \circ z \circ a \circ a = (a \rightarrow b)z$ donc $a \rightarrow b$ est bien une rétraction; la vérification du \forall est laissée en exercice [penser à utiliser que $F(cx)$ est une rétraction].

(b) Soit f_z la fonction codée par l'élément z de $\text{fun } D$.

Comme a et b sont des rétractions, ainsi que $d = a \rightarrow b$, on a : $dD = \text{Fix } d = \{z/b \circ z \circ a = z\} = \{z/z = b \circ z \text{ et } z = z \circ a\}$. La première condition est équivalente à $f_z(D) \subseteq bD$ et la seconde à $f_z(x) = f_z(ax) \forall x$. A partir de ces considérations on démontre aisément que l'application $z \rightarrow f_z/aD$ est un isomorphisme (une bijection bicontinue) entre dD et $[aD \rightarrow bD]$.

Soit $e = \forall x : cF(x) = \lambda z . \lambda x : c.(F(x))(zx)$. On a :

$$eD = \{z/z = \lambda x . (F(cx) \circ z \circ c) x\}$$

et

$$z \in eD \quad \text{ssi} \quad f_z(cx) \in F(cx) . D \text{ pour tout } x \text{ de } D.$$

Il en résulte que, toujours modulo des détails laissés au lecteur, $z \rightarrow f_z/cD$ est un isomorphisme entre eD et E .

(d) Si a, b, c et tous les éléments de cD [et donc tous les éléments de $F(cD)$] sont des projections, alors :

$$a \rightarrow b = \lambda z . b \circ z \circ a \leq \lambda z . \text{id} \circ z \circ \text{id} = \lambda z . \lambda x . zx = \lambda z . z = \text{id}.$$

$$\forall x : cF(x) = \lambda z . \lambda x . (F(cx) \circ z \circ c) x \leq \lambda z . \lambda x . (\text{id} \circ z \circ \text{id}) x = \text{id}.$$

Le cas des clôtures se traite de façon symétrique.

(c) Supposons maintenant que tout le monde est finitaire.

$(a \rightarrow b)D$ est Scott par (b) et 2. 2.

Reste à montrer que le rétract E est algébrique. Anticipant sur 7. 1 nous savons que les éléments compacts de $F(t)D$ sont les éléments de la forme $F(t)k$, $k \in D_c$ et $k \leq F(t)k$. Considérons les fonctions $\delta_{h,k}$, $h, k \in D_c$, définies pour $t \in cD$ par : $\delta_{h,k}(t) = F(t)k$ si $t \in P$ et $F(t) \perp$ sinon, où $P = \{t \in cD / t \geq h \text{ et } k \leq F(t)k\}$. Il est immédiat que $\delta_{h,k}$ est croissante et que $\delta_{h,k}(t) \in F(t)D$ pour tout t de cD . Pour voir que $\delta_{h,k} \in E$ il suffit donc de vérifier qu'elle est continue, ce qui se fait en quelques lignes sans surprises, une fois remarqué que, du fait de la compacité de h et k et de la continuité (et croissance) de F

et app, on a, pour toute suite croissante d'éléments de $cD : (\sup t_n) \in P$ ssi $\exists m t_m \in P$ ssi $\exists m \forall n \geq m t_n \in P$.

Montrons que $\delta_{h,k}$ est un élément compact de E . Soit f_n une suite croissante d'éléments de E telle que : $\delta_{h,k} \leq \sup f_n$. On a : $k \leq F(h)k = \delta_{h,k}(h) \leq \sup (f_n(h))$ et il y a donc un entier m tel que $k \leq f_m(h)$. Montrons que $\delta_{h,k} \leq f_m$. Si $t \in P$

$$\delta_{h,k}(t) = F(t)k \leq F(t)f_m(h) \leq F(t)f_m(t) = f_m(t);$$

sinon

$$\delta_{h,k}(t) = F(t) \perp \leq F(t)f_m(t) = f_m(t).$$

C.Q.F.D.

Considérons maintenant une fonction quelconque f de E . Utilisant le fait que $F(t)D$ est algébrique, on a :

$$f(t) = F(t) f(t) = \sup_{h,k} \{ F(t)k / h \leq t \text{ et } k \leq F(t)k \leq f(t) \},$$

donc $f = \sup \{ \delta_{h,k} / \delta_{h,k} \leq f \}$ et E est algébrique lui aussi.

Remarque 5.5. (conséquence immédiate des définitions 5.3; incluse dans 5.4) :

$$\begin{aligned} u \in (a \rightarrow b)D \quad \text{et} \quad v \in aD &\Rightarrow uv \in bD \\ u \in (\forall x : c F(x))D \quad \text{et} \quad v \in cD &\Rightarrow uv \in F(v)D. \end{aligned}$$

Interprétation des types variables

Soit ρ une application de V dans D satisfaisant $\text{env}(i) : \forall \alpha \in V \rho(\alpha) \in S = cD$. On désignera par $\rho[\alpha \rightarrow u]$ l'environnement ρ' égal à ρ partout sauf en α où il prend la valeur $u \in cD$. On définit par induction sur la complexité de σ son interprétation dans tous les environnements possibles ρ par :

$$\begin{aligned} (\alpha_i)_\rho^* &= \rho(\alpha_i) \\ (\sigma \rightarrow \tau)_\rho^* &= \sigma_\rho^* \rightarrow \tau_\rho^* \\ (\forall \alpha \sigma)_\rho^* &= \forall u : c \sigma_{\rho[\alpha \rightarrow u]}^*. \end{aligned}$$

Pour que cette définition ait un sens il faut montrer à chaque étape σ que :

5.6. Pour tout ρ l'application $u \rightarrow \sigma_{\rho[\alpha \rightarrow u]}^*$ est une fonction continue (exercice).

LEMME 5.7 : Si S est la classe des clôtures, des rétractions finitaires ou des projections finitaires, alors : $\forall \rho, \sigma \sigma_\rho^* \in S$.

Démonstration: Par induction sur σ , en utilisant 5.4 (c) ou (d).

Interprétation des termes typés

Supposons maintenant que ρ est une application de $V \cup X$ dans D qui satisfait env (i) et env (ii): $\forall x^\sigma \in X \rho(x^\sigma) \in \sigma_\rho^* D$. On définit, par induction sur la construction de M , l'interprétation M_ρ^* de M dans tous les environnements ρ possibles par :

$$\begin{aligned} (x^\sigma)_\rho^* &= \rho(x^\sigma) \\ (MN)_\rho^* &= M_\rho^* N_\rho^* \\ (M\sigma)_\rho^* &= M_\rho^* \sigma_\rho^* \\ (\lambda x^\sigma. M[x^\sigma])_\rho^* &= \lambda u : \sigma_\rho^* . M_\rho^*_{[x^\sigma \rightarrow u]} \\ (\forall \alpha M[\alpha])_\rho^* &= \lambda u : c . M_\rho^*_{[\alpha \rightarrow u]} \end{aligned}$$

Ici encore il faudrait vérifier à chaque étape que :

5.8. Les applications $u \rightarrow M_\rho^*_{[\alpha \rightarrow u]}$ et $u \rightarrow M_\rho^*_{[x^\sigma \rightarrow u]}$ sont continues.

5.9. LEMME DE SUBSTITUTION :

- (a) $(\tau[\sigma/\alpha])_\rho^* = \tau_\rho^*_{[\alpha \rightarrow \sigma_\rho^]}$
- (b) $(M[\sigma/\alpha])_\rho^* = M_\rho^*_{[\alpha \rightarrow \sigma_\rho^]}$
- (c) $(M[N/x^\sigma])_\rho^* = M_\rho^*_{[x^\sigma \rightarrow N_\rho^]}$.

Démonstration: Par induction sur la complexité de τ ou M .

LEMME 5.10 : Pour tout environnement, $M : \sigma \Rightarrow M_\rho^* \in \sigma_\rho^* D$.

Démonstration: Par induction sur la construction de M .

(a) Si M est une variable typée, on utilise env (ii).

(b) Si $M = NP$, $N : \sigma \rightarrow \tau$, $P : \sigma$, $M : \tau$, on a, par hypothèse de récurrence: $N_\rho^* \in (\sigma \rightarrow \tau)_\rho^* D = (\sigma_\rho^* \rightarrow \tau_\rho^*) D$ et $P_\rho^* \in \sigma_\rho^* D$, donc $M_\rho^* = N_\rho^* P_\rho^* \in \tau_\rho^* D$ (5.5).

(c) Si $M = N\sigma$, $N : \forall \alpha \tau$, $M : \tau[\sigma/\alpha]$, on a, par hypothèse d'induction: $N_\rho^* \in (\forall \alpha \tau)_\rho^* D = (\forall u : c \tau_\rho^*_{[\alpha \rightarrow u]}) D$, donc $M_\rho^* = N_\rho^* \sigma_\rho^* \in \tau_\rho^*_{[\alpha \rightarrow \sigma_\rho^]} D = (\tau[\sigma/\alpha])_\rho^* D$ (5.5 et 5.9 a).

(d) Si $M = \lambda x^\sigma. N$, $N : \tau$, $M : \sigma \rightarrow \tau$, alors :

$$M_\rho^* = \lambda u : \sigma_\rho^* . N_\rho^*_{[x^\sigma \rightarrow u]} = \lambda u . N_\rho^*_{[x^\sigma \rightarrow \sigma_\rho^* u]}.$$

Clairement $M_\rho^* \circ \sigma_\rho^* = M_\rho^*$; d'autre part $\tau_\rho^* \circ M_\rho^* = \lambda u . \tau_\rho^* (N_\rho^*_{[x^\sigma \rightarrow \sigma_\rho^* u]})$. En appliquant l'hypothèse d'induction à N pour $\rho' = \rho[x^\sigma \rightarrow \sigma_\rho^* u]$ et en notant que $\tau_\rho^* = \tau_{\rho'}^*$ puisque ρ et ρ' coïncident sur V , on a :

$\tau_\rho^* \circ M_\rho^* = \lambda u. N_\rho^* [x^\sigma \rightarrow \sigma_\rho^* u] = M_\rho^*$. Ainsi $\tau_\rho^* \circ M_\rho^* \circ \sigma_\rho^* = M_\rho^*$ et
 $M_\rho^* \in (\sigma_\rho^* \rightarrow \tau_\rho^*) D = (\sigma \rightarrow \tau)_\rho^* D$.

(e) Si $M = \forall \alpha N$, $N : \tau$, $M : \forall \alpha \tau$, alors :

$M_\rho^* = \lambda u : c. N_\rho^* [\alpha \rightarrow u] = \lambda u : c. \tau_\rho^* [\alpha \rightarrow u] N_\rho^* [\alpha \rightarrow u]$
 (hypothèse d'induction) $= (\forall \alpha \tau)_\rho^* z$,

où $z = \lambda u. N_\rho^* [\alpha \rightarrow u]$ (5.8), donc $M_\rho^* \in (\forall \alpha \tau)_\rho^* D$.

LEMME 5.11 : $M \equiv N \Rightarrow M_\rho^* = N_\rho^*$.

Démonstration : Il suffit de remarquer que 5.9 (b) et (c) impliquent :
 $M \text{ red } N \Rightarrow M_\rho^* = N_\rho^*$.

6. CLÔTURES ET PROJECTIONS

D est un c - λ -modèle qui est de plus un treillis complet, ce qui revient à dire que D a un plus grand élément, qu'on note \top .

Rappelons que les clôtures sont les rétractions $\geq \text{id}$, et les projections les rétractions $\leq \text{id}$. Les exemples triviaux de projection et de clôture sont respectivement : $\lambda x. \perp$ et $\lambda x. \top$.

Pour simplifier l'écriture nous supposerons de plus, comme il l'a été dit dans l'introduction, que D est extensionnel; le passage au cas non extensionnel est simple : il suffit de remplacer u par $\text{fun } u$ dans les ensembles dont le sup sert à définir les clôture et projection universelles.

LEMME 6.1 : *Les projections sont closes par sups quelconques, Les clôtures sont closes par infs quelconques et sups dirigés.*

Démonstration : Soit A un ensemble de projections et $a = \text{sup } A$. Clairement $a \leq \text{id}$ donc $a \circ a \leq a \circ \text{id} = a$. D'autre part $\forall x \in A \ x = x \circ x \leq a \circ a$, donc $a \leq a \circ a$. En conclusion $a \circ a = a$ et a est une projection.

Le fait que les clôtures soient closes par infs quelconques se démontre de façon symétrique; elles sont closes par sups dirigés parce que les rétractions le sont (4.2).

THÉORÈME 6.2 : *Soit C la fonction de D dans D définie par :*

$$C(u) = \inf \{ v/v \text{ clôture } \geq u \} = \min \{ v/v \text{ clôture } \geq u \},$$

C est continue et définit une clôture de D , dont l'image est l'ensemble des clôtures de D .

Démonstration: Le fait que l'inf soit en fait un minimum vient de 6.1.

Il est immédiat que C est croissante, que $C \geq \text{id}$, que $C(u)$ est une clôture pour tout u , et que, si u est une clôture, $C(u) = u$. Il en résulte que $C \circ C = C$ et $\text{Im } C = \{\text{clôtures}\}$. Reste à voir la continuité: soit (x_n) une suite croissante d'éléments de D et x son sup. Puisque C est croissante $C(x) \geq \sup C(x_n)$; enfin $\sup C(x_n)$ est une clôture (6.1), qui est plus grande que chaque $C(x_n)$, donc que chaque x_n , donc que x , donc que $C(x)$.

C.Q.F.D.

PROPOSITION 6.3 : *La fonction P définie sur D par :*

$$P(u) = \sup \{v/v \text{ projection } \leq u\} = \max \{v/v \text{ projection } \leq u\}$$

est une fonction croissante, telle que : $P \leq \text{id}$, $P \circ P = P$ et l'image de P est l'ensemble des projections de D .

La démonstration est symétrique de celle des faits correspondants pour C .

En résumé, il y a dans les treillis complets une clôture universelle, et il n'y a pas, *a priori*, de projection universelle (problème de la continuité de P). On récupère la continuité de P en se restreignant aux projections compactes $\leq u$ (cf. section 8), mais $\text{Im } P$ n'est plus alors l'ensemble complet des projections de D .

Nous allons maintenant donner deux autres définitions possibles de P et C ; le fait que la fonction C définie en 6.4 et 6.5 est une clôture universelle, est traité en détail dans [Scott 76], le fait que ce soit la même que celle qui est définie dans 6.1, suit de 4.4; les démonstrations sont symétriques pour P .

PROPOSITION 6.4 :

$$P(u)(x) = \sup \{z/z \leq x \text{ et } z \leq uz\}$$

$$C(u)(x) = \inf \{z/z \geq x \text{ et } z \geq uz\}$$

et il est facile de vérifier que les sup et inf sont en fait des max et min.

PROPOSITION 6.5 :

$$P(u, x) = \sup \{z/z = \inf(x, uz)\}$$

$$C(u, x) = \inf \{z/z = \sup(x, uz)\}.$$

$P(u)(x)$ et $C(u)(x)$ apparaissent ainsi respectivement comme le plus grand et le plus petit point fixe des fonctions continues $\lambda z. \inf(x, uz)$ et $\lambda z. \sup(x, uz)$; on retrouve ici sous un autre angle le problème de la continuité de P : alors que la fonction qui associe à une fonction continue son plus petit

point fixe est continue, celle qui associe le plus grand point fixe ne l'est en général pas.

7. RÉTRACTIONS FINITAIRES

NOTATIONS: $uA = \{ux/x \in A\}$, $A_c = A \cap D_c$ est l'ensemble des éléments compacts de D qui sont dans A . Si A est un retract de D , A_K est l'ensemble des éléments compacts du domaine continu A . h, k, l désignent toujours des éléments de D_c .

LEMME 7.1 : Si u est une rétraction on a :

- (1) $(uD)_K = \{uh/h \leq uh\} = \{uh/(h, h) \in \text{Tr } u\}$,
- (2) $(uD)_c \subseteq (uD)_K$.

Démonstration: (1) Soit $h \in D_c$ tel que $h \leq uh$. Supposons $uh \leq \sup uA$ avec uA dirigé, alors $h \leq \sup uA$ et, comme h est compact, $h \leq ux$ pour un $x \in A$. Donc $uh \leq u(ux) = ux \in uA$ et $uh \in (uD)_K$. Réciproquement soit $ux \in (uD)_K$. On a $ux = \sup A$, avec $A = \{h/h \leq ux\}$, donc $ux = u(ux) = \sup uA$ (et uA est dirigé puisque A l'est et u est continue); comme ux est un compact de uD , on a $ux \leq uh$ pour un h de A , et, par conséquent, $ux = uh$ et $h \leq ux = uh$.

- (2) Si $h \in uD$, on a $uh = h$; on conclut avec (1).

COROLLAIRE 7.2 : (1) Si u est une projection, $(uD)_K = (uD)_c$,

(2) Si u est une clôture, $(uD)_K = uD_c$,

(3) Si u est compacte, $(uD)_K = uD = (uD)_c$ (u rétraction).

LEMME 7.3 : u , rétraction, est finitaire, si et seulement si :

$$\forall (h, k) \in \text{Tr } u \quad \exists l \quad (h, l), (l, l), (l, k) \in \text{Tr } u.$$

Démonstration: u est finitaire ssi tout élément ux de uD est sup d'éléments de $(uD)_K$; comme D est Scott, $x = \sup \{h/h \leq x\}$, et il suffit de savoir que les uh sont sups d'éléments de $(uD)_K$, c'est-à-dire, compte tenu de 7.1, que

$$\forall k \leq uh \quad \exists l \quad k \leq ul \quad \text{et} \quad l \leq ul \leq uh.$$

COROLLAIRE 7.4 : Toute clôture est finitaire.

Toute rétraction compacte est finitaire.

Démonstration: Si u est une clôture on a $uh \geq h$ pour tout h , il suffit donc de prendre $l = h$. Si u est compacte uh est compact pour tout h , on prendra donc $l = uh$ (ou, plus directement, si u est compacte uD est fini donc Scott).

8. PROJECTIONS FINITAIRES

D est un c - λ -modèle (extensionnel).

Il résulte de 7.2 qu'une projection u est finitaire ssi tout élément x de uD est sup des éléments compacts de D qui sont dans uD . Les deux lemmes suivants permettent de caractériser les projections finitaires comme les sups d'ensembles de projections compactes. Le lemme 8.1 est un cas particulier d'un résultat plus général que nous verrons plus loin (9.3).

LEMME 8.1 : *Si A est un ensemble de projections compactes, alors $\sup A$ est une projection finitaire.*

Démonstration : On sait déjà que $v = \sup A$ est une projection (6.1), d'autre part $vx = \sup \{ \varepsilon x / \varepsilon \in A \}$ par 2.1. Pour tout ε de A on a : $\varepsilon \leq v$ et donc $\text{Im } \varepsilon \subseteq \text{Im } v$ (4.4); tous les εx sont compacts par 2.6, donc vx est sup d'éléments compacts qui sont dans vD et par conséquent v est finitaire.

Exemples de projections compactes : les $\varepsilon_{h,h}$.

LEMME 8.2 : *Soit u une projection finitaire, alors $u = \sup \{ \varepsilon / \varepsilon \text{ projection compacte } \leq u \}$.*

Démonstration : Il suffit de montrer que $u = \sup E$, avec

$$E = \{ \varepsilon_{h,h} / h = uh \} \\ (= \{ \varepsilon_{h,h} / h \leq uh \} = \{ \varepsilon_{h,h} / \varepsilon_{h,h} \leq u \}).$$

Soit $y = ux \in uD$; par 7.3 et le fait que u est une projection on a :

$$y = \sup \{ h / uh = h \text{ et } h \leq ux \} = \sup \{ \varepsilon_{h,h}(x) / uh = h \text{ et } h \leq ux \} \\ \leq \sup \{ \varepsilon_{h,h}(x) / uh = h \text{ et } h \leq x \} = (\sup E) x.$$

En résumé $u \leq \sup E \leq u$.

C.Q.F.D.

THÉORÈME 8.3 : *La fonction P de D dans D , définie par :*

$$P(u) = \sup \{ \varepsilon / \varepsilon \text{ projection compacte } \leq u \},$$

est continue et définit une projection finitaire de D , qui est en fait une projection finitaire universelle.

Remarques. — Ici bien sûr le sup n'est plus un max, puisque les projections compactes sont seulement stables par sups finis, mais, compte tenu de 8.1 et

8.2:

$$P(u) = \max \{ \text{projections finitaires } \leq u \}.$$

Démonstration de 8.3 : $P(u)$ est une projection finitaire pour tout u par 6.1 et 8.1. Si u est une projection finitaire, $P(u) = u$ par 8.2; par conséquent $P \circ P = P$ et $\text{Im } P = \{ \text{projections finitaires} \}$. $P \leq \text{id}$ car $P(u) \leq u$ pour tout u . Enfin la continuité de P suit de ce que, pour un ε compact, $\varepsilon \leq \sup(u_n)$ équivaut à $\exists n \varepsilon \leq u_n$.

Une autre définition possible de P est :

PROPOSITION 8.4 : $P(u, x) = \sup \{ h/h \leq x \text{ et } h \leq uh \}$ (rappel : $h \in D_c$).

Démonstration : Soit $P_1(u, x)$ le second membre de cette égalité. Il est aisé de voir que $P_1(u, \cdot) = \sup \{ \varepsilon_{n, h}/h \leq uh \}$. Comme $h \leq uh$ équivaut à $\varepsilon_{n, h} \leq u$, on a : $P_1(u, \cdot) \leq P(u)$ pour tout u . Réciproquement : $P(u) = P(P(u)) = \sup \{ \varepsilon_{n, h}/h \leq P(u)h \}$ [forme forte de 8.2 donnée dans sa démonstration, appliquée à la projection finitaire $P(u)$]. Comme $P(u) \leq u$ on a donc

$$P(u) \leq \sup \{ \varepsilon_{n, h}/h \leq uh \} = P_1(u, \cdot), \quad \text{et} \quad P(u) = P_1(u, \cdot).$$

9. RÉTRACTION FINITAIRE UNIVERSELLE

D est un c - λ -modèle (extensionnel).

La définition de la rétraction finitaire universelle est un peu moins simple que celle de la projection finitaire universelle, quoique analogue dans les grandes lignes. La difficulté vient de ce qu'un sup de rétractions, même bornées, n'est en général pas une rétraction [rappelons qu'en revanche un sup dirigé de rétractions est une rétraction (4.2)]. On est ainsi conduit à considérer les prérétractions.

DÉFINITION 9.1 : $a \in D$ est une *prérétraction* si a est une fonction et $a \circ a \geq a$. Si a est une prérétraction, la suite (a^n) est croissante et $\Delta a = \sup(a^n)$ est la plus petite rétraction $\geq a$, où $a^n = a \circ \dots \circ a$ n fois. Soit encore : $\Delta a = \sup(\pi_n(a))$ (cf. 3.9). Noter que les π_n sont des fonctions continues (croissantes) mais que la suite π_n n'est pas croissante; en revanche la suite $\pi_n(a)$ est croissante dès que a est une prérétraction.

LEMME 9.2 : *L'ensemble des prérétractions de D est clos par sups dirigés ou bornés.*

Démonstration: Soit A un ensemble de prérétractions admettant un sup, et soit $a = \sup A$. Pour tout x de A on a $x \leq x \circ x \leq a \circ a$, donc $a \leq a \circ a$; a est une fonction car, comme tout rétract, fun D est clos par sup dirigés ou bornés.

LEMME 9.3 : *Toute rétraction u qui est un sup de prérétractions compactes est finitaire.*

Démonstration: Soit A un ensemble de prérétractions compactes tel que $u = \sup A$ existe et est une rétraction. Quitte à remplacer A par l'ensemble des sups finis des éléments de A , on peut supposer que A est dirigé. Soit $x \in D$ et $k \leq ux$, il nous faut trouver un l (compact) tel que $l \leq ul \leq ux$ et $k \leq ul$. Comme k est compact et A dirigé, il y a un $v \in A$ tel que $k \leq vx$. Je dis que $l = vx$ convient. En effet l est compact parce que v est compacte, de plus : $ux = u(ux) \geq u(vx) = ul$ et $ul = u(vx) \geq v(vx) \geq vx = l$, enfin $k \leq vx = l$.

LEMME 9.4 : *Si u est une rétraction finitaire, $u = \sup \{ \varepsilon / \varepsilon \text{ rétraction compacte } \leq u \}$.*

Commençons par montrer que $u = \sup A_u$ où A_u est l'ensemble des prérétractions compactes $\leq u$. Soit $(h, k) \in \text{Tr } u$. Comme u est finitaire on peut trouver un l tel que : $(h, l), (l, l), (l, k)$ soient dans $\text{Tr } u$ (7.3). Soit ε la plus petite fonction dont la trace contient $(h, k), (h, l), (l, l), (l, k)$; c'est une fonction compacte (2.10.5), et $\leq u$. D'autre part ces quatre couples sont aussi dans la trace de $\varepsilon \circ \varepsilon$: par exemple, $(h, k) \in \text{Tr } \varepsilon \circ \varepsilon$ parce que (h, l) et (l, k) sont dans $\text{Tr } \varepsilon$. Puisque ε est minimale, on a donc $\varepsilon \leq \varepsilon \circ \varepsilon$ et ε est une prérétraction. Il en résulte que u est bien le sup des prérétractions compactes qui sont sous lui (A_u est dirigé et $\text{Tr } u = \bigcup \text{Tr } \varepsilon, \varepsilon \in A_u$).

Il suffit maintenant de montrer que pour toute prérétraction compacte ε , la rétraction $\Delta \varepsilon$ est aussi compacte (elle est évidemment plus petite que u si ε l'est puisque u est ici une rétraction). Soit r la cardinalité de εD (2.6); pour tout $x \in D$ la suite $\varepsilon^n x$ est croissante (car la suite ε^n l'est) donc on a nécessairement $\varepsilon^{r+1}(x) = \varepsilon^r(x)$; donc $\varepsilon^{r+1} = \varepsilon^r = \Delta \varepsilon$ et $\Delta \varepsilon$, composée d'un nombre fini de fonctions compactes, est aussi compacte.

Nous sommes maintenant en mesure de définir notre candidat.

DÉFINITION 9.5 : Soient, pour un élément u quelconque de D :

$$\sigma(u) = \sup \{ \varepsilon / \varepsilon \text{ prérétraction compacte } \leq u \} = \sup A_u,$$

et

$$R(u) = \sup \{ \varepsilon^n / \varepsilon \text{ prérétraction compacte } \leq u, n \in \mathbb{N} \},$$

Il est immédiat que :

$$R(u) = \sup \{ \Delta \varepsilon / \varepsilon \text{ prérétraction compacte } \leq u \}.$$

9.6. On a : $R(u) = \Delta \sigma(u) = \sup \{ \sigma(u)^n \}$.

En effet

$$R(u) = \sup \{ \varepsilon^n / \varepsilon \in A_u, n \in \mathbb{N} \} = \sup_n \sup \{ \varepsilon^n / \varepsilon \in A_u \} = \sup_n \{ \pi_n(\varepsilon) / \varepsilon \in A_u \};$$

comme π_n est continue (3.9) et A_u dirigé,

$$R(u) = \sup_n (\pi_n(\sup A_u)) = \sup_n \sigma(u)^n = \Delta \sigma(u).$$

THÉORÈME 9.7 : R est une rétraction finitaire universelle.

Démonstration : (a) $\sigma(u)$ est une prérétraction pour tout u (9.2) et $R(u)$ est une rétraction (9.6), qui est finitaire par (9.5), (9.3), et le fait que les ε^n sont compacts.

(b) Si u est une rétraction finitaire, $\sigma(u) = u$ (9.4), $\sigma(u)$ est une rétraction et $R(u) = \sigma(u) = u$.

(c) Comme d'habitude, on en conclut que $R \circ R = R$ et $\text{Im } R = \{ \text{rétractions finitaires} \}$.

(d) σ est continue parce que pour un ε compact, être $\leq \sup(u_n)$ revient à être $\leq u_n$ pour un certain n .

(e) $R_1 = \sup(\pi_n \circ \sigma)$ est une (vraie) fonction continue, parce que c'est un sup croissant de (vraies) fonctions continues, et $R(u) = R_1(u)$ pour tout u , donc $R = R_1$, R est continue, et on peut maintenant l'assimiler, de même que σ , à un élément de D .

(f) R est finitaire. En effet, puisque R est continue et A_u dirigé, $Ru = \sup \{ R\varepsilon / \varepsilon \in A_u \}$. Il suffit alors de remarquer que $R\varepsilon = \Delta\varepsilon$ est compacte (cf. fin dem. 9.4).

10. LES FONCTIONS STABLES

DI-domaines [Berry]

10.1. Un *DI-domaine* est, par définition, un domaine D de Scott tel que de plus :

- (i) pour tout compact h , $E_h = \{ x / x \leq h \}$ est fini, et,
- (ii) $\forall x, yz$ compatibles $(x \vee y) \wedge z = (x \wedge z) \vee (y \wedge z)$.

Il est facile de voir que si (ii) est vraie alors elle est vraie pour tout z [remplacer z par $z' = z \wedge (x \vee y)$].

Remarques: (a) D algébrique implique: (i) $\Leftrightarrow E_h \cap D_c$ est fini et (i) $\Rightarrow E_h \subseteq D_c$. (b) D algébrique et (ii) impliquent: si $\sup A$ existe, alors $(\sup A) \wedge y = \sup \{x \wedge y / x \in A\}$. Montrons le sens non trivial. Soit $h \leq \sup A$, y ; comme h est compact on a: $h \leq \sup B$, où B est un sous-ensemble fini de A ($\sup A$ est aussi le sup dirigé des sups des parties finies de A), donc $h \leq \sup B \wedge y = \sup \{x \wedge y / x \in B\} \leq \sup \{x \wedge y / x \in A\}$.

Exemples: $(P(\omega), \subseteq)$, les domaines qualitatifs de Girard, les espaces cohérents, $D \times D'$ où D et D' sont des DI -domaines.

Fonctions stables

10.2 DÉFINITION: f est *stable* si elle est continue et commute aux infis finis d'éléments compatibles: $\forall x, y$ compatibles $f(x \wedge y) = f(x) \wedge f(y)$.

Exemples: Les $\varepsilon_{h,k}$ de la section 2 sont stables, mais leurs sups finis ne le sont en général pas; si f_1 et f_2 sont stables, la fonction continue $f_1 \wedge f_2$, définie en 2.14, est stable.

10.3. *Propriété caractéristique des fonctions stables:* Une fonction continue f est stable si et seulement si:

$$\forall h, k \quad [k \leq f(h) \Rightarrow \exists h_0 \text{ minimum } \leq h \text{ tel que } k \leq f(h_0)].$$

Esquisse de démonstration: \Rightarrow : si f est stable $h_0 = \inf \{1 \leq h/k \leq f(l)\}$ est l'élément cherché. \Leftarrow : Par continuité il suffit de démontrer que pour tous h, h' compatibles $f(h \wedge h') \geq f(h) \wedge f(h')$; or, si $k \leq f(h), f(h')$, et si h et h' sont compatibles, on a $k \leq f(h_0)$ pour un $h_0 \leq h, h'$ (unicité de l'élément minimal), donc $k \leq f(h \wedge h')$.

10.4. *Traces stables:* Toute fonction stable est donc déterminée par le sous-ensemble suivant de sa trace: $\text{Tr}_s f = \{(h, k) / h \text{ minimum tel que } k \leq f(h)\}$. Clairement:

$$\begin{aligned} \text{Tr} f &= \{(h', k) / \exists h \leq h' (h, k) \in \text{Tr}_s f\}, \\ \forall x \quad f(x) &= \sup \{k / \exists h \leq x (h, k) \in \text{Tr}_s f\}, \end{aligned}$$

enfin: si (h, k) et (h', k) sont dans $\text{Tr}_s f$ et si h et h' sont compatibles, alors $h = h'$ (considérer $h \wedge h'$).

$$\text{Exemple: } \text{Tr}_s(\varepsilon_{h,k}) = \{(\perp, \perp)\} \cup \{(h, k') / \perp \neq k' \leq k\}.$$

10.5. $\text{Tr}_s g \circ f = \{ (h, k) / \exists l (h, l) \in \text{Tr}_s f \text{ et } (l, k) \in \text{Tr}_s g \}$, et cette fois le l est unique; en effet, si (h, l) et (h, l') sont dans $\text{Tr}_s f$ alors l et l' sont compatibles, donc (l, k) et $(l', k) \in \text{Tr}_s g$ implique $l = l'$.

L'inclusion des traces stables ne correspond plus à l'ordre point par point sur les fonctions, elle correspond à l'ordre de Berry, ou ordre stable: par définition, pour deux fonctions continues f et g ,

$$f \leq_s g \Leftrightarrow \forall x \leq y, f(x) = f(y) \wedge g(x)$$

(les fonctions et leurs arguments croissent ainsi de façon orthogonale).

Clairement, $f \leq_s g \Rightarrow f \leq g$.

10.6. $f \leq_s g \Rightarrow f^2 = f \circ g \wedge g \circ f$ (appliquer la définition à $x = f(z)$ et $y = g(z)$). Cette remarque nous servira à caractériser de manière particulièrement simple les clôtures et les projections dans le cadre stable.

10.7. Si f et g sont stables $f \leq_s g \Leftrightarrow \text{Tr}_s f \subseteq \text{Tr}_s g$.

Démonstration: Supposons $f \leq_s g$ et soit $(h, k) \in \text{Tr}_s f$; clairement $(h, k) \in \text{Tr}_s g$. Soit $h_0 \leq h$ (h_0, k) $\in \text{Tr}_s g$. On a $k \leq g(h_0), f(h)$, donc $k \leq f(h_0)$ puisque $f \leq_s g$, donc $h_0 = h$ et $(h, k) \in \text{Tr}_s g$. Réciproquement supposons $\text{Tr}_s f \subseteq \text{Tr}_s g$; on a: $\text{Tr} f \subseteq \text{Tr} g$ et $f \leq g$; soit $h \leq h'$; on veut montrer $f(h) \geq g(h) \wedge f(h')$, l'inégalité inverse étant évidente. Soit $l \leq g(h), f(h')$ et $h'_0 \leq h'$ tel que $(h'_0, l) \in \text{Tr}_s(f)$; on a $(h'_0, l) \in \text{Tr}_s g$ et h'_0 et h sont compatibles; de $l \leq g(h'_0), g(h)$ et de la stabilité de g on déduit $l \leq g(h'_0 \wedge h)$, donc $h'_0 \wedge h = h'_0, h'_0 \leq h$ et $l \leq f(h'_0) \leq f(h)$.

C.Q.F.D.

Le lemme suivant est une conséquence immédiate des définitions.

LEMME 10.8 : (a) $f \leq_s g$ et g stable $\Rightarrow f$ stable.

(b) $f \leq_s g'$ et $f \leq g \leq g' \Rightarrow f \leq_s g$.

On note $[D \rightarrow D']_s$ l'ensemble des fonctions stables de D dans D' , ordonné par \leq_s . Toutes les notions relatives à $[D \rightarrow D']_s$ seront distinguées des notions analogues dans $[D \rightarrow D]$ par un «s»; en particulier un ensemble de fonctions stables sera s -majoré si il est majoré pour \leq_s par une fonction stable; s -majoré implique évidemment majoré (de même s -dirigé implique dirigé), il est donc clair que si le sup_s d'une famille H de fonctions stables existe alors $\text{sup} H$ existe aussi; *a priori* $\text{sup} H \leq \text{sup}_s H$ mais en fait ils coïncident :

LEMME 10.9. (sups stables et sups extensionnels) : Soit H un ensemble de fonctions stables,

(a) H dirigé $\Rightarrow g = \text{sup} H$ est stable,

(b) H s -dirigé $\Rightarrow \sup H = \sup_s H$,

(c) H s -majoré $\Rightarrow \sup H = \sup_s H$.

Autrement dit $[D \rightarrow D]_s$ admet des sups dirigés et des sups bornés, qui coïncident avec les sups extensionnels (mais la seule existence du sup extensionnel ne garantit évidemment pas celle du \sup_s).

Démonstration :

(a) $g(x \wedge y) = \sup_f (f(x \wedge y)) = \sup_f (f(x) \wedge f(y)) = \sup_f (f(x)) \wedge \sup_f (f(y))$

(continuité de \wedge et le fait que les ensembles $H_z = \{f(z) | f \in H\}$ sont tous dirigés) $= g(x) \wedge g(y)$.

(b) $g = \sup H$ est stable par (a). Montrons que g s -majoré H . Soit $f \in H$ et $H_f = \{h \in H | h \geq_s f\}$. Puisque H est s -dirigé H_f l'est aussi et $g = \sup H_f$. Supposons $x \leq_s y$; on a $f(x) = h(x) \wedge f(y)$ pour toute $h \in H_f$ et, par suite, $f(x) = g(x) \wedge f(y)$ (continuité et distributivité de \wedge). Il nous reste à montrer que $g \leq_s g'$ pour tout s -majorant stable g' de H . Si $x \leq y$ on a

$$g(x) = \sup_f (f(x)) = \sup_f (f(y) \wedge g'(x)) = \sup_f (f(y)) \wedge g'(x) = g(y) \wedge g'(x)$$

(car H_y est dirigé et \wedge continu).

(c) Soit $g = \sup H$. On démontre de manière analogue au (b) que $g \leq_s g'$ pour tout s -majorant stable g' de H . Cette fois il existe un tel g' par hypothèse; on en déduit que g est stable et s -majoré H (10.8 (a) et (b)). Il en résulte que $g = \sup_s H$.

10.10. Soit H un ensemble s -dirigé de fonctions stables et g stable, alors :

$$g = \sup_s H \Leftrightarrow \text{Tr}_s g = \bigcup \text{Tr}_s f, f \in H \quad (\text{et l'union est évidemment dirigée}).$$

En effet il est facile de montrer que le second membre est équivalent à $\text{Tr} g = \bigcup \text{Tr} f$; on conclut avec 2.9 et 10.9.

10.11. Soit $A \subseteq D_c \times D_c$. Si il existe une fonction stable dont la trace stable contient A , alors il en existe une qui est s -minimale, soit g . Si A est fini g est s -compacte et $\text{Tr}_s g$ est finie.

Démonstration : $[D \rightarrow D]_s$ admet des sups bornés par 10.9 donc des inf pour toute famille non vide, et il suffit de prendre $g = \inf_s \{f | A \subseteq \text{Tr}_s f\}$. Supposons maintenant A fini et $g \leq \sup H$, H ensemble s -dirigé de fonctions stables. On a : $A \subseteq \text{Tr}_s g \subseteq \bigcup \text{Tr}_s f, f \in H$. Comme l'union est dirigée et A fini, il y a une $f \in H$ telle que $A \subseteq \text{Tr}_s f$; par minimalité de g , $g \leq_s f$ et g est s -compacte. Soit

$$A' = \{(h'_1 \vee \dots \vee h'_n, k'_1 \vee \dots \vee k'_n) | n \in \mathbb{N}, h'_1, \dots, h'_n \text{ compatibles}, (h'_i, k'_i) \in \text{Tr}_s g, \exists (h_i, k_i) \in A (h'_i \leq h_i \text{ et } k'_i \leq k_i)\};$$

A' est fini, $A' \subseteq \text{Tr}_s g$ et la fonction $g'(x) = \sup \{ k/\exists h \leq x(h, k) \in A' \}$ est stable et de trace A' (exercice), donc $g' \leq_s g$, $g' = g$ et $\text{Tr}_s g'$ est finie.

- 10.12. (a) Toute fonction stable est \sup_s de fonctions stables s -compactes,
- (b) f stable est s -compacte ssi $\text{Tr}_s f$ est finie,
- (c) en particulier les fonctions compactes sont closes par \circ .

Démonstration: (a) Soit g stable; $\text{Tr}_s g$ est l'union dirigée de ses parties finies p ; pour chaque p on peut trouver une fonction f_p stable, minimale, et s -compacte telle que: $p \subseteq \text{Tr}_s f_p \subseteq \text{Tr}_s g$ (10.11); manifestement $\text{Tr}_s g = \bigcup \text{Tr}_s f_p$. On conclut avec 10.10 que $g = \sup H$, où H est l'ensemble (dirigé) des f_p .

(b) Si $\text{Tr}_s g$ est finie, g est compacte par 10.11; réciproquement si g est compacte elle est nécessairement égale à l'une des f_p , et $\text{Tr}_s f_p$ est finie par 10.11.

(c) est une conséquence immédiate de (b) et de 10.5.

THÉORÈME 10.13 : $[D \rightarrow D]_s$ est un *DI-domaine*.

Démonstration: Tout découle de 10.9-12, sauf la distributivité de l'inf, par rapport au \sup_s . Mais $f \vee_s g = f \vee g$ dès que f et g sont s -compatibles (10.9); si l'on montre que, sous cette même hypothèse, $f \wedge_s g = f \wedge g$, la distributivité suivra de la propriété analogue de D et du fait que \wedge et \vee sont définis sur des fonctions de manière extensionnelle. On sait déjà (ex. 10.2) que $f \wedge g$ est stable; montrons que, si f et g sont s -compatibles, $f \wedge g \leq_s f$. Soit F un s -majorant de f et g ; si $x \leq y$, on a :

$$\begin{aligned} (f \wedge g)(x) &= f(x) \wedge g(x) = F(x) \wedge f(y) \wedge g(y) \\ &= F(x) \wedge f(y) \wedge (f \wedge g)(y) = f(x) \wedge (f \wedge g)(y). \end{aligned}$$

C.Q.F.D.

De même $f \wedge g \leq_s g$, comme enfin $f \wedge g$ majore tout minorant commun à f et g , on a bien $f \wedge g = f \wedge_s g$.

10.14. *Fonctions n-aires:* La bonne notion de stabilité, pour les fonctions n -aires, est la suivante: (pour $n=2$) $F: D \times D' \rightarrow D''$ est *stable* si elle est continue et: $F(x \wedge x', y \wedge y') = F(x, y) \wedge F(x', y')$ toutes les fois que x et x' sont compatibles, ainsi que y et y' (ce qui est exactement dire que F est stable sur $D \times D'$ muni de l'ordre produit); ceci implique que F est stable composante par composante, mais la réciproque est fautive. Les fonctions stables n -aires sont closes par substitution.

Exemples : $\circ : [D \rightarrow D']_s^2 \rightarrow [D \rightarrow D']_s$ est une fonction binaire stable (exercice), et par conséquent les p_n sont des fonctions stables de $[D \rightarrow D]_s$ dans $[D \rightarrow D]_s$.

***s*- λ -modèles**

10.15. Les modèles du λ -calcul pur basés sur les *DI*-domaines et les fonctions stables (ou *s*- λ -modèles), se définissent de manière tout à fait analogue à ce qui a été fait en section 3, la seule différence est la suivante : seules sont codées les fonctions stables, et on les code de façon stable. Plus précisément : on exige cette fois que *app* soit une fonction *stable* et que l'application $\lambda : f \rightarrow \lambda x.f(x)$ soit une fonction stable de $[D \rightarrow D]_s$ dans D . Ceci a pour conséquence (insistons lourdement) que :

1. les fonctions représentées dans D sont exactement les fonctions stables; on appellera encore leurs codes : les fonctions de D ;
2. $f \rightarrow \lambda x.f(x)$ réalise un isomorphisme entre $[D \rightarrow D]_s$ et l'ensemble des fonctions de D ordonné par l'ordre de D ; on aura donc, pour toutes fonctions a et b de D :

$$a \leq b \quad \text{ssi} \quad \varphi_a \leq_s \varphi_b \quad \text{ssi} \quad \forall x \leq y \quad ax = ay \wedge bx.$$

Le reste des remarques et définitions de la section 3 vaut sans changement mais on peut ajouter que les π_n définies en 3.9 sont des fonctions stables de D dans D .

Rétractions

Une rétraction est toujours une fonction de D telle que $a \circ a = a$. Tout ce que nous avons dit en section 7 reste évidemment valable, en particulier la caractérisation des rétractions finitaires, mais cette fois nous allons être capable de montrer que toutes les rétractions sont finitaires.

LEMME 10.16 : *Toute rétraction u est finitaire, et de plus :*

$$\forall (h, k) \in \text{Tr}_s u \quad \exists l \text{ (unique)} \quad (h, l), (l, l), (l, k) \in \text{Tr}_s u.$$

Démonstration : Il suffit de montrer que

$$\forall (h, k) \in \text{Tr}_s u \quad \exists l \quad (h, l), (l, l), (l, k) \in \text{Tr}_s u.$$

En effet on en déduira aisément la propriété analogue sur les Tr en utilisant : $\text{Tr}_s u \subseteq \text{Tr} u$ et $\forall (h, k) \in \text{Tr} u \quad \exists h' \leq h \quad (h', k) \in \text{Tr}_s u$.

De $u = u \circ u \circ u$ et de 10.5 on déduit qu'il existe l, l' uniques tels que $(h, l), (l, l'), (l', k)$ sont dans $\text{Tr}_s u$. Si $l \neq l'$ on obtient deux décompositions de $(h, k) \in \text{Tr}_s u \circ u$, à savoir $(h, k), (l, k)$ et $(h, l'), (l', k)$, contradiction.

10.17. *Projections et clôtures :*

u projection $\Leftrightarrow u$ fonction et $u \leq \text{id}$,

u clôture $\Leftrightarrow u = \text{id}$.

Démonstration : On applique 10.6 aux fonctions stables $f = \varphi_u$, $g = \text{id}_D$, puis à : $g = \varphi_u$ et $f = \text{id}_D$.

Les deux résultats qui suivent peuvent être passés par le lecteur intéressé par le seul problème de l'existence de la r.u.

LEMME 10.18 : *Si u est une rétraction $(uD)_K = \{uh/(h, h) \in \text{Tr}_s u\}$.*

Démonstration : On sait déjà par 7.1 que $(uD)_K = \{uh/(h, h) \in \text{Tr} u\}$. Il suffit donc de montrer que $(h, h) \in \text{Tr} u \Rightarrow \exists l (l, l) \in \text{Tr}_s u$ et $uh = ul$. Mais $(h, h) \in \text{Tr}_s u$ implique (10.16) : $\exists l (h, l), (l, l), (l, h) \in \text{Tr}_s u$, et on a : $l \leq uh \leq u(l) = ul$, donc $ul \leq uh \leq ul$, et donc $uh = ul$.

PROPOSITION 10.19 : *Si u est une rétraction, uD est clos par inf d'éléments compatibles et uD est un DI-domaine.*

Démonstration : uD est clos par inf tout simplement parce que u est stable. Les remarques faites en 4.1 et le lemme précédent impliquent que uD est un domaine de Scott. L'inf et la sup dans uD de deux éléments compatibles de uD sont respectivement : $x \wedge y$ et $u(x \vee y)$ et il est facile de vérifier que uD hérite de D la distributivité de l'inf par rapport au sup. Il reste à voir que sous tout élément de $(uD)_K$ il n'y a qu'un nombre fini d'éléments de $(uD)_K$; grâce à 10.1 et au lemme 10.18, il suffit de voir que $uk \leq uh$, $(k, k) \in \text{Tr}_s u$, et $(h, h) \in \text{Tr}_s u$, impliquent $k \leq h$. Mais $k \leq uk \Rightarrow k \leq uh \Rightarrow \exists l \leq h (l, k) \in \text{Tr}_s u$; on a $l \leq uh$ car $uh \leq h$, donc k et l sont compatibles et, comme $(k, k) \in \text{Tr}_s u$, on a $l = k$.

C.Q.F.D.

11. RÉTRACTION STABLE UNIVERSELLE

D est un s - λ -modèle, i.e. un modèle stable du λ -calcul, au sens de 10.15. En particulier le rétract $\text{fun } D$ de D est isomorphe à $[D \rightarrow D]_s$, l'espace des fonctions stables de D muni de l'ordre stable. Les rétractions et les prérétractions sont définies comme dans le cas continu, et codent automatiquement des fonctions stables.

Et en fait, comme précédemment, nous supposons de plus D extensionnel.

Fait 11.1 : L'ensemble des prérétractions est clos par sups bornés et dirigés, et par infis finis.

Démonstration : Le cas des sups est réglé par 9.2. Soient x et y deux prérétractions compatibles, $x \wedge y$ est une fonction (10.2 et 10.13), d'autre part $(x \wedge y)^2 = x^2 \wedge y^2$ (stabilité de π_2) $\geq x \wedge y$.

LEMME 11.2 : *Pour toute rétraction u on a :*

$$u = \sup \{ \varepsilon / \varepsilon \text{ rétraction compacte } \leq u \}.$$

Démonstration : Même démonstration que pour 9.4, en utilisant cette fois les traces stables et 10.16.

Noter : tous les sups dont il est question ici sont pris dans (D, \leq) ; quand il arrivera qu'il en soit autrement nous le préciserons.

DÉFINITIONS 9.5. (*rappel*) :

$$\sigma(u) = \sup \{ \varepsilon / \varepsilon \text{ prérétraction compacte } \leq u \} = \sup A_u,$$

$$R(u) = \sup \{ \varepsilon^n / \varepsilon \in A_u \} = \sup \{ \Delta \varepsilon / \varepsilon \in A_u \} = \Delta \sigma(u).$$

THÉORÈME 11.3 : *R est une rétraction (stable) universelle.*

Démonstration : (a) $\sigma(u)$ est une prérétraction (11.1) et $R(u)$ est une rétraction.

(b) si u est une rétraction $\sigma(u) = u$ (11.2) et $R(u) = \sigma(u) = u$.

(c) $R \circ R = R$ et $\text{Im } R = \{ \text{rétractions} \}$.

(d) σ est une fonction continue (9.7 d), mais aussi stable. Supposons en effet u et v compatibles. Par seule croissance de σ on a $\sigma(u \wedge v) \leq \sigma(u) \wedge \sigma(v)$. Pour voir l'inégalité inverse il suffit de montrer, puisque D est algébrique, que tout h plus petit que $\sigma(u)$ et $\sigma(v)$ est plus petit que $\sigma(u \wedge v)$. Or $h \leq \sigma(u)$, $\sigma(v)$ implique : $\exists \varepsilon \in A_u$, $\exists \varepsilon' \in A_v$ $h \leq \varepsilon, \varepsilon'$ (compacité de h), donc $h \leq \varepsilon \wedge \varepsilon'$; comme $\varepsilon \wedge \varepsilon'$ est une prérétraction (11.1), manifestement plus petite que $u \wedge v$, on a gagné.

(e) On sait que, dans $[D \rightarrow D]$, $R = \sup(\pi_n \circ \sigma)$ (9.7 e). R est donc le sup continu d'une suite croissante de fonctions stables, donc R est stable (10.9. a).

COROLLAIRE 11.4 : *(D, R) est un modèle de LCR.*

Remarque 11.5: Il y a aussi une projection universelle, donnée par :

$$P(u) = \sup \{ \varepsilon \text{ projection compacte } \leq u \} \\ = \sup \{ \text{fonctions compactes } \leq \text{id} \wedge u \},$$

et pas de clôture universelle.

Démonstration (esquisse): Avec 10.17 on a l'équivalence entre les deux valeurs de $P(u)$, et le fait que id est la seule clôture; le fait que $P \circ P = P$ et $\text{Im } P = \{ \text{projections} \}$ vient de la section 8. Il reste deux choses à montrer: que P est stable (même démonstration que pour σ) et que $P \leq_s \text{id}$, ce qui est trivial. Il ne nous reste plus qu'à remarquer que notre seule clôture, id , a pour image D et non $\{ \text{id} \}$.

REMERCIEMENTS

Je remercie Thomas Ehrard pour l'appui qu'il m'a donné pour l'ordre stable et Xavier Gouy pour ses remarques pertinentes à la lecture du manuscrit.

BIBLIOGRAPHIE

1. R. AMADIO, K. BRUCE et G. LONGO, The Finitary Projection Model for 2nd Order λ -Calculus, *IEEE Conference on Logic in Computer science*, Boston, juin 1986.
2. R. AMADIO et G. LONGO, Type-Free Compiling of Parametric Types, Formal Description of Programming Concepts III, M. WIRSING éd., *Elsevier Science Publishers*, I.F.I.P., 1987, p. 337-398.
3. H. BARENDREGT, The λ -Calculus, its Syntax and Semantics, revised edition, *North Holland*, 1984.
4. S. BERARDI, Retractions on dI-Domains as a Model for Type: Type, *Inform. and Comput.*, 1991, 94, p. 204-231.
5. G. BERRY, Modèles complètement adéquats et stables des λ -calculs typés, *Thèse d'État*, Université Paris-VII, 1979.
6. J. Y. GIRARD, The System F of Variable Types Fifteen Years Later, *Theoret. Comput. Sci.*, 1986, 45, p. 159-192.
7. J. L. KRIVINE, λ -Calcul: Types et Modèles, *Masson*, Paris, 1990.
8. N. McCracken, An Investigation of a Programming Language with a Polymorphic Type Structure, *Ph. D. Thesis*, Syracuse University, 1979.
9. N. McCracken, A Finitary Retract Model for the Polymorphic λ -Calculus, 1984, unpublished.
10. D. SCOTT, Data Types as Lattices, *S.I.A.M. J. Comput.*, 1976, 5, p. 522-587.
11. D. SCOTT, A Space of Retracts, manuscript, 1980.