Le calcul lambda

Marcel Crabbé
(Université Catholique de Louvain)

1. Le calcul λ est une théorie naïve des fonctions. La théorie naïve des ensembles basée sur le principe de compréhension — toute propriété $\varphi(x)$ détermine un ensemble $\{x|\varphi(x)\}$ — est, on le sait, contradictoire (paradoxe de Russell). Le calcul λ basé sur un principe analogue — toute expression $M[x]$ détermine une fonction $\lambda x M[x]$ — est non contradictoire (théorème de Church et Rosser). Bien plus, si l'on tente d'y reproduire le paradoxe ensembliste, on aboutit à un résultat simple mais puissant : le théorème du point fixe.

Le présent article est une introduction, non un survol. Nous avons omis notamment le calcul λ avec extensionnalité (la conversion-η), le calcul λ typé et/ou stratifié, la logique combinatoire ainsi que les modèles du calcul λ. Nous nous sommes bornés sur le plan technique à démontrer le théorème de Church et Rosser ainsi que la représentabilité des fonctions récursives partielles (via les machines de Cutland). En guise de référence bibliographique nous nous limiterons à la Bible en la matière : H.P. BARENDREGT, *The Lambda Calculus. Its Syntax and Semantics*, North-Holland, Amsterdam, 1981.

Nous présentons dans ce paragraphe quelques idées sous-jacentes au concept de fonction tel qu'il est mis en œuvre dans le calcul λ.

Une fonction f appliquée à un argument x produit une valeur $f(x)$. Elle effectue un certain travail que l'on peut évaluer soit en ne considérant que son résultat soit en tenant compte du travail lui-même. Le résultat ou l'ensemble des couples $(x, f(x))$ est appelé le **graphe** ou l'**extension** de la fonction (nous n'examinons pour l'instant que les fonctions à une variable). Selon le point de vue ensembliste ou extensionnel, la fonction est identifiée à son graphe. Qui plus est, une fonction est tout simplement un ensemble quelconque de couples vérifiant des conditions d'existence et d'unicité. Le concept de fonction devient donc extrêmement précis mais au prix d'une occultation de ce qui constitue la nature même de la fonction. Il devient impossible d'affirmer que deux fonc-

tions (distinctes) ont même graphe. Le concept de fonction s'est rétréci. Il s'est aussi élargi car cette définition ensembliste qui pose non seulement que toute fonction est un graphe, mais encore que tout graphe est une fonction, accorde le statut de fonction à des ensembles qui ne sont l'extension d'aucun processus effectif. Ce qui est appelé **intension** (avec "s") ou sens d'une fonction s'impose lorsqu'on prend en compte le type de travail effectué par la fonction. Cette notion est plus délicate que celle d'extension. D'aucuns la considèrent comme une idée vague et lui préfèrent pour cette raison le concept d'extension. Or les idées vagues sont fécondes. Elles sont sources de concepts multiples. Il en va ainsi de la notion d'intension. Une fonction comme celle désignée par l'expression x^2+x, par exemple, représente une correspondance qui s'établit en exécutant successivement deux instructions : élever l'argument au carré, ajouter l'argument au résultat. Tout un programme ! Si l'on identifie une fonction à son programme, on obtient un concept intensionnel très étroit. L'expression $x(x+1)$ décrit alors une autre fonction que celle décrite par x^2+x. Cette vision un peu trop sensible peut être "corrigée" en introduisant une relation d'équivalence entre programmes ayant même extension. Chaque relation de ce type fournit un concept d'intension. Les cas limites sont celui où une fonction est un programme et celui — trivial — où tous les programmes sont équivalents — ce qui redonne la définition extensionnelle. Le concept d'intension, quelle qu'en soit la détermination, prime celui d'extension. Outre que l'extension est une intension floue, l'intension permet en principe de trouver le graphe mais non l'inverse.

Une fonction s'applique à des objets. Ces objets peuvent à leur tour être des fonctions. L'intégrale définie et la dérivée en font foi. Plus radicalement, il n'est pas indispensable de distinguer les fonctions des entités qui ne sont pas des fonctions. L'artifice qui consiste à associer à chaque objet la fonction constante dont la valeur est cet objet quel qu'en soit l'argument, permet de faire l'économie de la distinction fonction–objet. Habituellement, les arguments d'une fonction sont d'un certain type et il en va de même des valeurs. On ne peut pas appliquer une fonction à n'importe quoi. Une fonction ne prend traditionnellement son sens que si l'on précise le domaine des arguments et celui des valeurs possibles. On devrait donc faire une différence entre une fonction comme *élever au carré dans les entiers* et la fonction *élever au carré dans les réels*, négligeant par là l'analogie existant entre les deux. De même ce n'est que par abus de langage que l'on pourra parler de la fonction triviale d'identité dont la valeur est identique à l'argument. Il y aurait donc autant

de fonctions d'identité qu'il y a d'ensembles. Par contre, selon le point de vue naïf, l'identité de *élever au carré* est *élever au carré* et l'identité de l'identité est l'identité. En effet, la notion de type ou de domaine disparaît si l'on prend en compte l'attitude naïve. Dans le calcul λ, tout sera une fonction et les fonctions s'appliqueront à n'importe quelle fonction, y compris à elles-mêmes. Ceci découle d'ailleurs partiellement du point de vue intensionnel. Car si une fonction est un programme — modulo une relation d'équivalence — on pourra le faire agir sur n'importe quel autre programme, quitte à ce qu'aucune valeur déterminée n'en résulte. Lorsqu'on définit la fonction comme étant son graphe, on se donne deux ensembles et on construit la fonction à partir d'eux. Il n'est donc pas possible, sans introduire un élément imprédicatif, d'appliquer la fonction à elle-même. Quand, par contre, on aborde le concept de fonction par la voie intensionnelle c'est la fonction qui est donnée d'abord et le domaine des arguments —et donc aussi l'ensemble des valeurs— en est déduit comme étant l'ensemble des entités qui donnent une valeur lorsque la fonction s'y applique.

Les fonctions à plusieurs arguments peuvent se ramener à des fonctions à un argument moyennant l'introduction de la notion de n-uple. Ainsi l'addition peut être envisagée non comme une fonction à deux arguments mais comme une fonction à un argument dont le domaine est constitué de couples. Toutefois dans le calcul λ un autre procédé de réduction lui est préféré. Il s'agit de l'astuce de Schönfinkel qui utilise l'élargissement de la notion de fonction tel qu'il est pratiqué du point de vue naïf. La fonction d'addition sera à nouveau considérée comme une fonction à un argument. Mais cette fois l'argument n'est plus nécessairement un couple. Par contre la valeur de la fonction est une autre fonction. A l'argument x, on associe la fonction $y \mapsto x + y$ qui à y fait correspondre $x+y$. Cette méthode est bien connue dans le contexte ensembliste — ou catégorique — où elle indique qu'une fonction de $A \times B$ à valeurs dans C n'est autre qu'une fonction de A à valeurs dans C^B : $C^{A\times B} \simeq (C^B)^A$. La fonction $x, y \mapsto x + y$ sera, dans l'esprit du calcul λ, remplacée par la fonction $x \mapsto (y \mapsto x + y)$.

La distinction fonction–objet est supprimée dans le calcul λ. Tout y est fonction. Elle est cependant remplacée par une autre distinction, celle de la fonction et de sa valeur pour un argument, valeur qui est elle-même une fonction. Ceci rejoint à nouveau le point de vue intuitif qui conduit à faire la différence entre la fonction qui à x associe $x^2 + x$, à savoir $x \mapsto x^2 + x$, et sa valeur pour l'argument x : $x^2 + x$. Le principe d'**abstraction**, corrélatif du principe de compréhension en théorie des ensembles, permettra d'extraire de

chaque expression $\ldots x \ldots x \ldots$ la fonction $x \mapsto \ldots x \ldots x \ldots$. L'opération $x \mapsto$ est notée λx.

D'autre part, chaque fonction peut s'**appliquer** à n'importe quelle autre fonction. L'abstraction et l'application sont des opérations qui s'annihilent. La fonction $\lambda x \ldots x \ldots x \ldots$ appliquée à l'argument z donne comme valeur $\ldots z \ldots z \ldots$. Ce phénomène appelé β-conversion est l'exemple privilégié d'une relation d'équivalence déterminant un concept d'intension, du moins si l'on considère les expressions, ou termes, du calcul λ comme des programmes.

---

2. Formalisation du calcul λ.

Le langage du calcul λ est constitué d'un alphabet et d'une syntaxe. L'alphabet comprend une infinité de variables, les symboles λ et $=$ ainsi que des parenthèses. La syntaxe énonce les règles qui permettent de construire les **termes** —sensés désigner des fonctions— et les formules.

Ces règles syntaxiques sont :

- toute variable est un terme;
- si M et N sont des termes, alors (MN) est un terme;
- si M est un terme et x une variable, alors $\lambda x\, M$ est un terme;
- si M et N sont des termes, alors $M = N$ est une formule.

Commentaire : $M = N$ signifie que les termes M et N désignent la même fonction au sens intensionnel. (MN) est sensé désigner le résultat de l'application de (la fonction) M à (l'argument) N. $\lambda x\, M$ est le pendant de la fonction $x \mapsto M$ qui à x fait correspondre M.

Dans $\lambda x M$ les occurrences de la variable x sont **liées** (on dit aussi apparentes, muettes ou locales). On identifiera, comme c'est l'usage, des termes qui ne diffèrent que par les variables liées. Par exemple $\lambda x(xy)$ et $\lambda z(zy)$. On aurait pu utiliser les carrés de Bourbaki et concevoir $\lambda x(xy)$ comme une écriture populaire pour $\lambda(\underset{\sqcup}{\square}\, y)$. $M \equiv N$ signifiera que M et N sont identiques aux variables liées près —ou que M et N sont des écritures pour un même terme avec carrés de Bourbaki.

Si M, N sont des termes et x une variable, $M[x := N]$ est le résultat de la **substitution** de N aux occurrences libres de x dans M, ou dans un terme $\equiv M$, si une variable libre de N risquait d'être liée. $\lambda y(yx)[x := z]$ est donc $\lambda y(yz)$, mais $\lambda y(yx)[x := y]$ est $\lambda z(zy)$ et non $\lambda y(yy)$ ($\underbrace{\lambda(\square}\, x)[x := y]$ est $\underbrace{\lambda(\square}\, y)$). Moyennant ces conventions usuelles, on pourra traiter la substitution comme s'il n'y avait pas de variables liées (voir l'exercice ci-dessous).

Axiomes et règles du calcul λ

Un terme de la forme $(\lambda x MN)$ est une **coupure** et le terme $M[x := N]$ est sa **réduite**. Les axiomes et règles du calcul λ exprimeront que $M = N$ est vrai si et seulement si N peut s'obtenir à partir de M en remplaçant successivement certaines (occurrences de) coupures par leurs réduites ou certaines réduites par les coupures correspondantes :

- $(\lambda x\, MN) \;=\; M[x := N]$,
- $M = N$,
- $M = N \Longrightarrow N = M$,
- $M = N,\ N = P \Longrightarrow M = P$,
- $M = M',\ N = N' \Longrightarrow (MN) = (M'N')$,
- $M = M' \Longrightarrow \lambda x\, M = \lambda x\, M'$.

En vue de clarifier les écritures, on admettra les notations suivantes :

- $M_1 M_2 \ldots M_n$ pour $(\ldots(M_1 M_2)\ldots M_n)$ —association des parenthèses à gauche;
- $\lambda x_1 \ldots x_n . MN_1 \ldots N_m$ pour $\lambda x_1 \ldots \lambda x_n(\ldots(MN_1)\ldots N_m)$;
- $(\lambda x\, M)$ pour $\lambda x\, M$;
- $M[x]$ pour M et $M[N]$ pour $M[x := N]$, s'il n'y a pas d'ambiguïté à craindre.

Exercice

Corriger la "démonstration" suivante :

$$\begin{aligned}(\lambda z((\lambda yz.y)zy))x &= (\lambda yz.y)xy \\ &= (\lambda z.x)y \\ &= x.\end{aligned}$$

$$\begin{aligned}(\lambda z((\lambda yz.y)zy))x &= (\lambda z((\lambda z.z)y))x \\ &= (\lambda z.y)x \\ &= y.\end{aligned}$$

Donc, $x = y$.

---

3. Nous donnons maintenant un premier aperçu de la puissance d'expression du calcul λ en répertoriant quelques termes importants.

Le terme $\lambda x\, x$ — ou $\lambda\,\square$ — est noté I et représente l'identité. On a $IM = M$ pour tout terme M, en particulier $II = I$.

Le terme $\lambda xy.x$ — ou $\lambda\lambda\,\square$ — est noté K et représente le producteur de fonctions constantes. KM représente la fonction constante associée à M, car $KMN = M$.

Le terme $\lambda xyz.xz(yz)$ — ou $\lambda\lambda\lambda\,\square\square(\square\square)$ — est noté S. Il permet de distribuer λ par rapport à l'application : $S(\lambda z\, M)(\lambda z\, N) = \lambda z.MN$.

Si P, M et N sont des termes, le terme PMN s'écrit encore :

$$M \text{ si } P \text{ et } N \text{ sinon.}$$

Pour rendre cette notation efficace, on convient que le terme K représente également le vrai et que le terme $\lambda xy.y$ — ou $\lambda\lambda\,\square$ — qui est noté F représente le faux. On a donc :

$$\begin{aligned}M \text{ si } P \text{ et } N \text{ sinon} &= M, \quad \text{si } P = K; \\ M \text{ si } P \text{ et } N \text{ sinon} &= N, \quad \text{si } P = F.\end{aligned}$$

Les n-uples ($n \geq 0$) peuvent être codés comme suit : $\langle M_1, \ldots, M_n \rangle$ est le terme $\lambda x.xM_1 \ldots M_n$ — ou $\lambda\,\square\, M_1 \ldots M_n$ — la variable x n'étant pas libre

dans l'un des M_i $(1 \leq i \leq n)$. Cette définition est acceptable, car on peut coder les projections correspondantes. En effet, si $1 \leq i \leq n$, soit p_i^n le terme $\lambda x_1 \ldots x_n.x_i$ — ou $\lambda \ldots \underbrace{\lambda \ldots \lambda}_{i} \square$, le lien étant issu du $i^{\text{ème}}$ λ. On constate que :

$$\langle M_1, \ldots, M_n \rangle p_i^n = M_i.$$

Par conséquent, $\langle M_1, \ldots, M_n \rangle = \langle N_1, \ldots, N_m \rangle$ entraîne que $M_i = N_i$ $(1 \leq i \leq n)$.

Un terme de la forme $MN_1 \ldots N_n$ peut être compris comme le résultat de l'application de la fonction M aux arguments $N_1, \ldots, N_n$. Le terme $\langle M \rangle$ est, plus ou moins, le représentant de la fonction n-aire au sens habituel :

$$\langle M \rangle \langle N_1, \ldots, N_n \rangle = MN_1 \ldots N_n.$$

En général,

$$\langle M_1, \ldots, M_m \rangle \langle N_1, \ldots, N_n \rangle = M_1 N_1 \ldots N_n M_2 \ldots M_m.$$

Notons que

$$\langle\rangle = I = p_1^1, \quad K = p_1^2, \quad F = KI = p_2^2 \quad \text{et} \quad M \text{ si } P \text{ et } N \text{ sinon } = \langle M, N \rangle P.$$

En utilisant les relations suivantes :

- $\lambda x\, x = I = SKK$,
- $\lambda x\, M = KM$, si x n'est pas libre dans M,
- $\lambda x.MN = S(\lambda x\, M)(\lambda x\, N)$,

on peut montrer que tout terme clos (sans variable libre) se ramène à un terme qui est une juxtaposition de S, de K et de parenthèses. Soit alors $G \equiv \langle S, K \rangle$. On a

$$GG = SSKK = I,$$
$$G(GG) = GI = ISK = F,$$
$$G(G(GG)) = GF = K,$$
$$\text{et} \quad G(G(G(GG))) = GK = S.$$

Tout terme clos se ramène donc à une juxtaposition de G et de parenthèses.

Si M est un terme, ω_M est le terme $\lambda x.M(xx)$ — $\lambda \underbrace{M(\square \square)}_{1}$ — où x est non libre dans M. On remarque que $\omega_M \omega_M = M(\omega_M \omega_M)$. Cela montre le

Théorème du point fixe

Pour tout terme M, il existe un terme N tel que $MN = N$.

---

4. La réduction et les concepts apparentés

Un terme est dit **normal** s'il ne contient pas de coupures. Autrement dit, dans un terme normal il n'y a pas de sous-termes de la forme $(\lambda x M)N$.

Il est aisé de montrer par induction sur la structure des termes que tout terme du calcul λ est de la forme:

$$\lambda x_1 \ldots x_n . M M_1 \ldots M_m, \qquad n, m \geq 0,$$

où M est soit une variable, soit une coupure. Donc, les termes normaux sont ceux où M est une variable et où les M_i $(0 \leq i \leq m)$ sont normaux.

Exemples

Les variables, les termes I, K, F, $\lambda x.xxx$, yII sont normaux. Par contre, les termes $y(II)$ et $(\lambda x.xxx)(\lambda x.xxx)$ ne sont pas normaux.

Remarquons que $y(II) = yI$ et que yI est normal. Par contre, il ne semble pas y avoir de terme normal N tel que $\Omega\Omega = N$, où Ω est le terme $\lambda x.xxx$. On constate en effet que $\Omega\Omega = \Omega\Omega\Omega = \ldots = \Omega\Omega \ldots \Omega$ etc.

Ceci conduit à la définition suivante: un terme M **a une forme normale** s'il existe un terme normal N tel que $M = N$. En ce cas, N est une **forme normale** de M.

Dans un terme normal, il n'y a pas de coupures. On peut donc espérer obtenir une forme normale d'un terme en y remplaçant les coupures par leurs réduites. Cependant, ainsi qu'en témoignent les exemples ci-dessus, de nouvelles coupures peuvent apparaître au cours de cette opération. Et il peut en être de même si on poursuit le processus de normalisation. D'ailleurs dans certains cas la tentative échoue. Aucune forme normale de $\Omega\Omega$, par exemple, ne peut en résulter.

Définition de la relation de réduction

M **se réduit immédiatement à** N, en abrégé $M \xrightarrow{1} N$, si N est le résultat de l'élimination d'exactement une (occurence de) coupure de M.

M **se reduit à** N, en abrégé $M \longrightarrow N$, s'il existe une suite finie de termes $M_0, \ldots, M_k$ telle que $M_0 \equiv M$, $M_k \equiv N$ et $M_i \xrightarrow{1} M_{i+1}$, si $0 \leq i < k$.

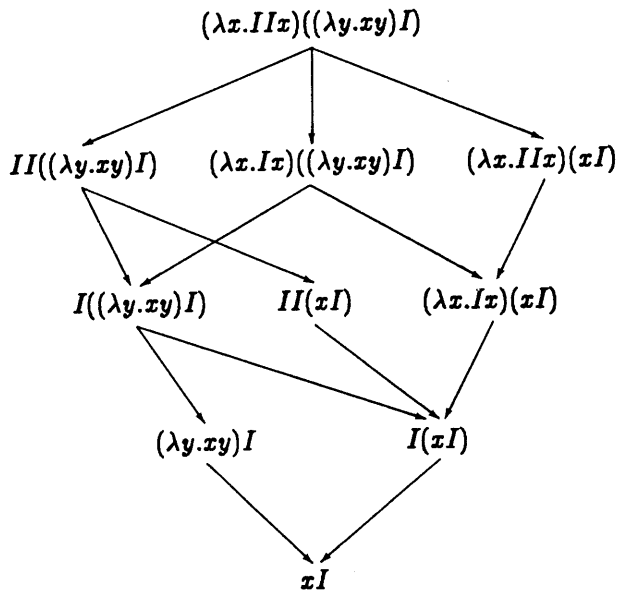


Figure 1

Il est clair que si $M \to N$ alors $M = N$. Réciproquement, on vérifie que $M = N$ si et seulement si il existe une suite finie de termes $M_0, \ldots, M_k$ telle que $M_0 \equiv M$, $M_k \equiv N$ et $M_i \xrightarrow{1} M_{i+1}$ ou $M_1 \xrightarrow{1} M_i$ $(0 \leq i < k)$.

Si $M \xrightarrow{1} N$, alors M n'est pas un terme normal. En particulier la relation $\xrightarrow{1}$ n'est pas réflexive. Mais il y a des termes qui se réduisent immédiatement à eux-mêmes. $(\lambda x.xx)(\lambda x.xx)$ en est un exemple.

La relation $\xrightarrow{1}$ n'est pas transitive bien qu'il y ait des termes M, N, P, tels que $M \xrightarrow{1} N$, $N \xrightarrow{1} P$ et $M \xrightarrow{1} P$. L'exemple précédent en fournit. Plus simplement: $(\lambda y.x)(II) \xrightarrow{1} (\lambda y.x)I \xrightarrow{1} x$.

Une **réduction** de M est une suite de termes $M_0, M_1, \ldots$ telle que $M_0 \equiv M$ et $M_i \xrightarrow{1} M_{i+1}$, si $0 \leq i$ et M_i n'est pas normal. Un même terme peut donner naissance à des réductions distinctes (voir Figure 1).

Un terme est **normalisable** s'il se réduit à un terme normal. Cela revient

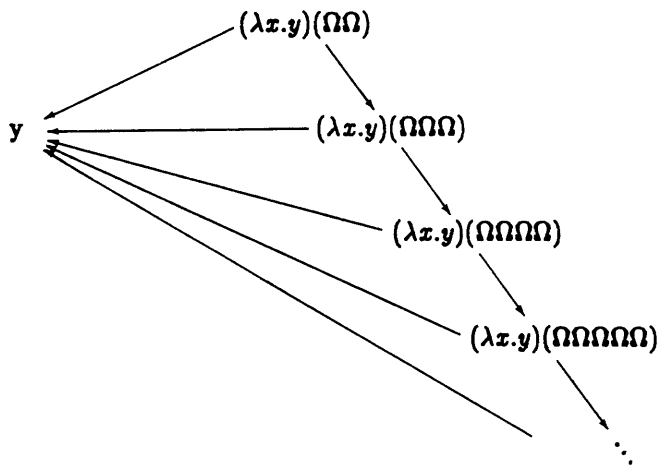


Figure 2

à dire qu'il a une réduction finie. Signalons, pour mémoire, qu'un terme absolument normalisable est un terme dont toutes les réductions sont finies. La Figure 2 montre qu'un terme normalisable n'est pas nécessairement absolument normalisable. Tout terme normalisable a une forme normale. La question de savoir si tout terme ayant une forme normale est normalisable sera tranchée dans le paragraphe suivant.

Nous concluons ce paragraphe par une proposition, démontrable sans peine, qui sera utilisée tacitement dans la suite.

Proposition 0

1. Si $M \longrightarrow N$, alors toute variable libre de N est une variable libre de M.
2. $M\,[x := N]\,[y := P] \equiv M\,[y := P]\,[x := N[y := P]]$, si $x \not\equiv y$ et si x n'est pas libre dans P.
3. Si $M \longrightarrow M'$, alors $M[x := N] \longrightarrow M'[x := N]$.
4. Si $N \longrightarrow N'$, alors $M[x := N] \longrightarrow M[x := N']$.
5. Si $M \longrightarrow M'$ et $N \longrightarrow N'$, alors $M[x := N] \longrightarrow M'[x := N']$.

5. Le théorème de Church et Rosser

L'auto-référence est souvent source de contradiction. Dans le calcul λ, il est permis d'appliquer un terme à n'importe quel terme y compris à lui-même. Ne peut-on pas dès lors par un argument habile montrer que $M = N$ quels que soient M et N ? Il suffirait pour cela de prouver, par exemple, que $K = F$, car alors $M = KMN = FMN = N$.

Si M et N se réduisent à un même terme, alors $M = N$. Donc, pour montrer que $M = N$, il suffit de trouver un terme auquel M et N se réduisent. Or K et F sont des termes normaux distincts. On ne peut pas par conséquent déduire $K = F$ par cette méthode. Le théorème de Church et Rosser affirme qu'il n'y en a pas vraiment d'autre. Autrement dit, si $M = N$, alors M et N se réduisent à un même terme. On pourra en conclure que $K \neq F$ et en général que $M \neq N$ si M et N sont deux termes normaux distincts.

Il est remarquable que les démonstrations habituelles de ce théorème sont combinatoires, en ce sens qu'aucune référence n'est faite à une représentation intuitive du concept de fonction tel qu'il est manipulé dans le calcul λ. Les fameux modèles de Scott, qui sont de telles représentations, ont été découverts bien après que la cohérence du calcul λ ait été établie par la voie du théorème de Church et Rosser.

En vue de donner une démonstration simplifiée de ce théorème, nous associerons à chaque terme M un terme noté $\varrho(M)$. Si nous comprenons une réduction comme un (essai de) calcul de la valeur d'un terme, $\varrho(M)$ peut-être compris comme étant une première valeur approchée de M obtenue en éliminant progressivement les coupures à partir de l'intérieur.

Définition de ϱ

- $\varrho(x)$ est x,
- $\varrho(PQ)$ est $\varrho(P)\varrho(Q)$, si PQ n'est pas une coupure,
- $\varrho(\lambda x P)$ est $\lambda x\, \varrho(P)$,
- $\varrho((\lambda x P)Q)$ est $\varrho(P)[x := \varrho(Q)]$.

Lemme 1

1. $\varrho(P)\ \varrho(Q) \longrightarrow \varrho(PQ)$.
2. $\varrho(M)\ [x := \varrho(N)] \longrightarrow \varrho(M[x := N])$.

Démonstration

1. Si PQ n'est pas une coupure, alors $\varrho(PQ) \equiv \varrho(P)\varrho(Q)$.
   Si PQ est une coupure: $(\lambda x P')Q$, alors

$$\varrho(P)\varrho(Q) \equiv (\lambda x \varrho(P'))\varrho(Q) \xrightarrow{1} \varrho(P')[x := \varrho(Q)] \equiv \varrho(PQ).$$

2. Par induction sur la structure de M. On suppose que x n'est pas liée dans M.

   (a) M est une variable

   i. $M \equiv x$: en ce cas

$$\begin{aligned} \varrho(M)[x := \varrho(N)] &\equiv \varrho(N) \\ &\equiv \varrho(M[x := N]). \end{aligned}$$

   ii. $M \not\equiv x$: alors

$$\begin{aligned} \varrho(M)[x := \varrho(N)] &\equiv \varrho(M) \\ &\equiv \varrho(M[x := N]). \end{aligned}$$

   (b) $M \equiv PQ$ n'est pas une coupure.
   Par hypothèse d'induction et 1 :

$$\begin{aligned} \varrho(M)[x := \varrho(N)] &\equiv \varrho(P)[x := \varrho N)]\ \varrho(Q)[x := \varrho(N)] \\ &\to \varrho(P[x := N])\ \varrho(Q[x := N]) \\ &\to \varrho(M[x := N]). \end{aligned}$$

   (c) $M \equiv \lambda y P$.
   Par hypothèse d'induction :

$$\begin{aligned} \varrho(M)[x := \varrho(N)] &\to \lambda y\ \varrho(P[x := N]) \\ &\equiv \varrho M([x := N]). \end{aligned}$$

   (d) $M \equiv (\lambda y P)Q$.
   On suppose que y est non libre dans N. De plus, x n'étant pas liée dans M, $x \not\equiv y$. Par hypothèse d'induction :

$$\begin{aligned}
\varrho(M)\,[x := \varrho(N)] &\equiv \varrho(P)\,[y := \varrho(Q)]\,[x := \varrho(N)] \\
&\equiv \varrho(P)\,[x := \varrho(N)]\,[y := \varrho(Q)[x := \varrho(N)]] \\
&\to \varrho\,(P[x := N])\,[y := \varrho(Q[x := N])] \\
&\equiv \varrho\,(M[x := N]).
\end{aligned}$$

□

Proposition 1

1. $M \to \varrho(M)$.
2. Si $M \xrightarrow{1} N$, alors $N \to \varrho(M)$.
3. Si $M \to N$, alors $\varrho(M) \to \varrho(N)$.

Démonstration

1. Par induction sur M.
   (a) $x \to x \equiv \varrho(x)$.
   (b) Par hypothèse d'induction et le lemme 1.1 :
   $$PQ \to \varrho(P)\varrho(Q) \to \varrho(PQ).$$
   (c) Par hypothèse d'induction :
   $$\lambda x\, P \to \lambda x\, \varrho(P) \equiv \varrho(\lambda x\, P).$$

2. Si $M \xrightarrow{1} N$, M contient une coupure. On procède par induction sur M en tenant compte des cas suivants :
   (a) $M \equiv (\lambda x\, P)Q$ est la coupure éliminée (base de l'induction). Donc,
   $$N \equiv P[x := Q].$$
   En vertu de 1,
   $$N \to \varrho(P)\,[x := \varrho(Q)] \equiv \varrho(M).$$
   (b) $M \equiv PQ$ n'est pas la coupure éliminée.
   Alors $N \equiv P'Q$ avec $P \xrightarrow{1} P'$ ou $N \equiv PQ'$ avec $Q \xrightarrow{1} Q'$.
   Par l'hypothèse d'induction, le lemme 1.1 et le point 1 :
   $$N \longrightarrow \varrho(P)\,\varrho(Q) \longrightarrow \varrho(M).$$

(c) $M \equiv \lambda x\, P$.
Donc $N \equiv \lambda x P'$, où $P \xrightarrow{1} P'$. Par l'hypothèse d'induction :

$$\lambda x\, P' \to \lambda x\, \varrho(P) \equiv \varrho(M).$$

3. Il suffit de montrer que $M \xrightarrow{1} N$ entraîne $\varrho(M) \to \varrho(N)$. Les cas de l'induction à envisager sont, à peu de choses près, semblables à ceux du point précédent.

(a) $M \equiv (\lambda x\, P)Q$ et $N \equiv P[x := Q]$.
Par le lemme 1.2 :

$$\begin{array}{rcl} \varrho(M) & \equiv & \varrho(P)[x := \varrho(Q)] \\ & \to & \varrho(P[x := Q]) \\ & \equiv & \varrho(N). \end{array}$$

(b) $M \equiv (\lambda x P)Q$ et $N \equiv (\lambda x P')Q$ avec $P \xrightarrow{1} P'$, ou $N \equiv (\lambda x P)Q'$ avec $Q \xrightarrow{1} Q'$.
Par hypothèse d'induction :

$$\begin{array}{rcl} \varrho(M) & \equiv & \varrho(P)[x := \varrho(Q)] \\ & \to & \varrho(P')[x := \varrho(Q)] \\ & & \text{ou} \\ & \to & \varrho(P)[x := \varrho(Q')] \\ & \equiv & \varrho(N). \end{array}$$

(c) $M \equiv PQ$ n'est pas une coupure. Donc, $N \equiv P'Q$ avec $P \xrightarrow{1} P'$ ou $N \equiv PQ'$ avec $Q \xrightarrow{1} Q'$.
Par hypothèse d'induction et le lemme 1.1 :

$$\begin{array}{rcccl} \varrho(M) & \longrightarrow & \varrho(P')\varrho(Q) & \to & \varrho(N) \\ & & \text{ou} & & \\ & \longrightarrow & \varrho(P)\varrho(Q') & \longrightarrow & \varrho(N). \end{array}$$

(d) $M \equiv \lambda x P$, $N \equiv \lambda x P'$ et $P \xrightarrow{1} P'$.
Par hypothèse d'induction :

$$\varrho(M) \equiv \lambda x \varrho(P) \longrightarrow \lambda x \varrho(P') \equiv \varrho(N).$$

□

Théorème de Church et Rosser

Si $M = N$, il existe un terme P tel que $M \rightarrow P$ et $N \rightarrow P$.

Démonstration

Nous montrons que pour toute suite de termes $M_0, \ldots, M_k \quad (k \geq 0)$ telle que $M_i \xrightarrow{1} M_{i+1}$ ou $M_{i+1} \xrightarrow{1} M_i$, lorsque $0 \leq i < k$, on a $M_k \rightarrow \varrho^k(M_0)$. $\varrho^k(M)$ désigne ici le terme $\underbrace{\varrho \ldots \varrho}_{k \text{ fois}}(M)$. Le théorème en découlera immédiatement en vertu de la proposition 1.1.

Nous raisonnons par récurrence sur k.

1. Si $k = 0$, $M_0 \rightarrow \varrho^0(M_0) \equiv M_0$.
2. $k > 0$. Examinons les deux situations possibles.
   (a) Si $M_{k-1} \xrightarrow{1} M_k$, on a $M_k \rightarrow \varrho(M_{k-1})$ (proposition 1.2). En outre, par hypothèse, $M_{k-1} \rightarrow \varrho^{k-1}(M_0)$. Donc, par la proposition 1.3, $\varrho(M_{k-1}) \rightarrow \varrho^k(M_0)$. D'où la conclusion.
   (b) Si $M_k \xrightarrow{1} M_{k-1}$, alors par hypothèse et la proposition 1.1, $M_{k-1} \rightarrow \varrho^{k-1}(M_0) \rightarrow \varrho^k(M_0)$. Et donc, $M_k \rightarrow \varrho^k(M_0)$.

□

Corollaire

1. Si M et N sont normaux et distincts, alors $M \neq N$. En particulier, le calcul λ est cohérent.
2. Un terme a au plus une forme normale. On peut donc parler de **la** forme normale s'il en existe.
3. Tout terme ayant une forme normale est normalisable.
4. M a une forme normale si et seulement si il existe un nombre naturel k tel que $\varrho^k(M)$ est normal. Il y a donc une **stratégie** pour construire une réduction débouchant sur la forme normale, si elle existe.

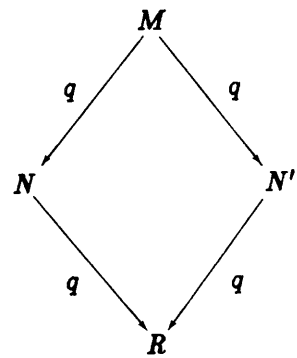
Remarques sur la démonstration de Tait et Martin-Löf

La démonstration du théorème de Church et Rosser, réalisée par Martin-Löf selon une idée de Tait, consiste à définir une relation de réduction quasi immédiate que nous noterons $\xrightarrow{q}$ telle que :

1. $M \xrightarrow{q} N$ entraîne $M \to N$,

2. $M \xrightarrow{1} N$ entraîne $M \xrightarrow{q} N$,

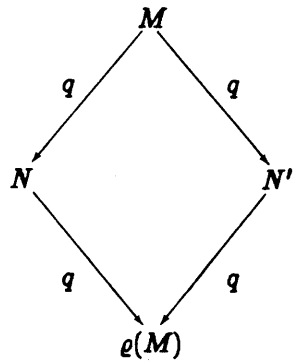
3. Si $M \xrightarrow{q} N$ et $M \xrightarrow{q} N'$, alors il existe un terme R tel que $N \xrightarrow{q} R$ et $N' \xrightarrow{q} R$:



(La propriété 3 est appelée propriété du losange.) On en déduit la propriété du losange pour la réduction ordinaire et, de là, le théorème de Church et Rosser.

La définition de la relation $\xrightarrow{q}$ est donnée à l'aide d'axiomes et de règles. Nous montrons ici que le même résultat s'obtient directement à partir de la proposition 1. Définissons pour cela $M \xrightarrow{q} N$ par : $M \to N$ et $N \to \varrho(M)$.

Cette relation vérifie les propriétés 1 et 2 (proposition 1.2). Elle vérifie également la propriété du losange. En effet, si $M \xrightarrow{q} N$, alors $N \to \varrho(M)$ et $\varrho(M) \to \varrho(N)$ (proposition 1.3). Dès lors :



6. Une des réussites les plus fameuses de la logique contemporaine est la caractérisation de la notion de calculabilité indépendamment de tout système formel. La notion de calculabilité acquiert par là un caractère **absolu**. Ceci signifie que les diverses définitions existantes de la calculabilité sont toutes équivalentes en ce qu'elles décrivent la même classe de fonctions, du moins au point de vue extensionnel. Dans ce paragraphe, nous montrerons une partie de ce résultat, à savoir que les fonctions calculables sont représentables dans le calcul λ.

Nous prendrons comme définition de référence de la calculabilité la définition particulièrement éclairante, donnée par Cutland (dans *Computability*, Cambridge University Press, 1980). Cette définition présente les fonctions calculables comme étant celles qui sont programmables sur un ordinateur théorique appelé URM (Unlimited Register Machine). Ces fonctions sont exactement les fonctions récursives partielles ou programmables sur une machine de Turing. Pour représenter ces fonctions dans le calcul λ, il est indispensable avant tout de coder les nombres naturels de manière à pouvoir représenter le successeur et la fonction caractéristique de l'égalité. Nous pourrons ensuite, après avoir rappelé la définition de Cutland — en la manipulant quelque peu —, représenter l'ensemble des fonctions récursives partielles. Pour cela, nous utiliserons un théorème du point fixe.

Codage des naturels

Les nombres naturels sont codés dans le calcul λ par des termes appelés

numéraux. Il existe différents systèmes de numéraux. Celui que nous utiliserons est singulièrement indiqué pour la démonstration du théorème de Kleene et Turing qui suivra.

Définition des numéraux

$$\begin{aligned} \underline{0} &\equiv \langle \lambda xyz\,.\,xyzK\,,\ \lambda xF \rangle, \\ \underline{n+1} &\equiv \langle F\,,\ \underline{n} \rangle. \end{aligned}$$

Les numéraux sont des termes normaux distincts. Donc, par le corollaire au théorème de Church et Rosser, si $n \neq m$, alors $\underline{n} \neq \underline{m}$.

Lemme 2 (représentation de l'égalité)

Si n et m sont des naturels, alors

$$\underline{n}\,\underline{m} = K \text{ si } n = m \text{ et } \underline{n}\,\underline{m} = F \text{ si } n \neq m.$$

Pour la démonstration de ce lemme, on commence par vérifier, par un calcul direct, que $\underline{0}\,\underline{0} = K$, $\underline{n+1}\,\underline{0} = F$, $\underline{0}\,\underline{n+1} = F$, et $\underline{n+1}\,\underline{m+1} = \underline{m}\,\underline{n}$.

Ensuite, par récurrence sur n, on obtient

$$\underline{n}\,\underline{n} = K \text{ et } \underline{n}\,\underline{n+m} = \underline{n+m}\,\underline{n} = F, \quad \text{si } m > 0.$$

□

Définition de la calculabilité selon Cutland

Un programme URM est une suite finie d'instructions : $I_1, \ldots, I_s$, où I_i est soit $Z(n)$, soit $S(n)$, soit $J(n,m,q)$ $(1 \leq i \leq s,\ n,m,q \in \mathbf{N})$. Il y a donc trois types d'instructions dont les significations sont, respectivement, annuler le contenu du registre n, remplacer le contenu du registre n par son successeur, comparer les contenus des registres n et m et sauter (Jump) à l'instruction q, s'ils sont égaux. Pour la commodité, Cutland introduit un quatrième type d'instruction, le transfert. Celui-ci n'est pas théoriquement nécessaire. Nous ne nous en occuperons pas.

Un programme agit sur des suites de naturels $n_1, \ldots, n_r, 0, 0 \ldots$, se terminant par une infinité de zéros, placés dans les différents registres de la machine.

On applique successivement les instructions dans l'ordre sauf lorsqu'une instruction $J(n, m, q)$ enjoint de sauter à l'instruction q. Lors du déroulement du programme, il n'y a qu'un nombre fini de registres modifiables. Ce nombre est déterminé par la situation des registres au départ (l'entrée) et par les nombres apparaissant dans les instructions. On peut donc décrire le déroulement d'un programme en ne considérant que l'évolution d'une suite initiale de registres, de longueur suffisante.

Plus précisément, nous dirons que l est (un nombre) **suffisant** pour le programme P si $l \geq$ au numéro maximum des registres mentionnés dans les instructions de P.

Soit un programme $P = I_1, \ldots, I_s$ et une suite $n_1, \ldots, n_l$ de naturels (l suffisant pour P). Le **déroulement de P, appliqué à la suite $n_1, \ldots, n_l$,** est la suite d'**étapes** $e_0, e_1, \ldots$ définie ainsi :

- e_0 est $I_1\ (n_1, \ldots, n_l)$;
- si e_j est $I_i\ (r_1, \ldots, r_l)$ et $i \leq s$, alors
  - e_{j+1} est $I_{i+1}\ (r_1, \ldots, 0, \ldots, r_l)$ si I_i est $Z(n)$,
  - e_{j+1} est $I_{i+1}\ (r_1, \ldots, r_n + 1, \ldots, r_l)$ si I_i est $S(n)$,
  - e_{j+1} est $I_q\ (r_1, \ldots, r_l)$ si I_i est $J(n, m, q)$ et $r_n = r_m$,
  - e_{j+1} est $I_{i+1}\ (r_1, \ldots, r_l)$ si I_i est $J(n, m, q)$ et $r_n \neq r_m$.

Dans cette définition, nous supposons que $I_t\ (r_1, \ldots, r_l)$ désigne r_1 si $t > s$. Donc, si le déroulement est fini, la dernière étape est un nombre naturel (le résultat du calcul).

Définition de la calculabilité

Une fonction de $\mathbf{N}^k \to \mathbf{N}$ est **calculable** s'il existe un programme $P = I_1, \ldots, I_s$ tel que pour tout l suffisant pour P et $\geq k$:

- si $h\ (n_1, \ldots, n_k) = m$ alors le déroulement appliqué à la suite (de longueur l) $n_1, \ldots, n_k, 0, \ldots, 0$ est fini et sa dernière étape est m;
- si $h\ (n_1, \ldots, n_k) \uparrow$ alors le déroulement de P appliqué à la suite (de longueur l) $n_1, \ldots, n_k, 0, \ldots, 0$ est infini.

Exemples

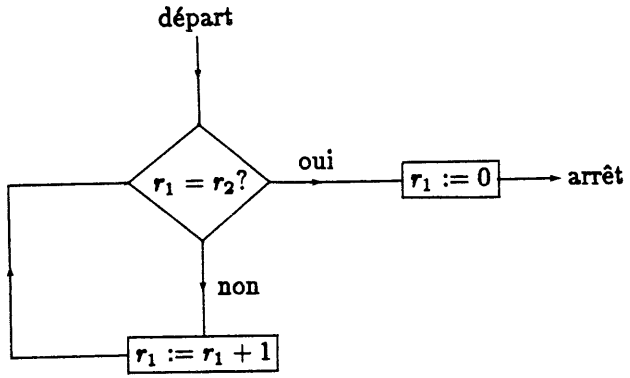
Le programme : $J(1, 2, 4)$, $S(1)$, $J(1, 1, 1)$, $Z(1)$ calcule la fonction $f(x, y)$

définie par :

$$f(x,y) = 0, \quad \text{si } x \leq y,$$

$$f(x,y) \uparrow, \quad \text{sinon.}$$

Ce programme est schématisé par l'organigramme :



Appliqué à la suite 5, 6, le déroulement est :

$$\begin{aligned} e_0 &: I_1(5,6) \\ e_1 &: I_2(5,6) \\ e_2 &: I_3(6,6) \\ e_3 &: I_1(6,6) \\ e_4 &: I_4(6,6) \\ e_5 &: 0. \end{aligned}$$

Appliqué à la suite 6, 5, le déroulement est infini.

Le programme : $J(2,3,5)$, $S(1)$, $S((3)$, $J(1,1,1)$ calcule l'addition. Tous ses déroulements sont finis.

Nous nous proposons de mimer dans le calcul λ les déroulements des programmes URM. Pour cela, nous définissons ici un concept de représentation intensionnelle.

Soit un programme $(P = I_1, \ldots, I_s)$ et X un terme du calcul λ. A chaque déroulement de P : $e_0, e_1, \ldots$ nous associons la suite de termes $E_0, E_1, \ldots$, où E_j est $Xp_i^s\underline{r_1}\ldots\underline{r_l}$ lorsque e_j est $I_i(r_1, \ldots, r_l)$ (il est entendu que $Xp_i^s\underline{r_1}\ldots\underline{r_l}$ n'est autre que $\underline{r_1}$ si $t > s$).

Nous dirons que X représente P jusqu'à l (l suffisant pour P) si pour tout déroulement de P (appliqué à une suite de longueur l) : $e_o, e_1, \ldots$, on a :

1. $E_0 \longrightarrow E_1 \longrightarrow \cdots$
2. E_0 n'a pas forme normale si le déroulement est infini.

La première condition aurait suffi si nous nous étions limités aux fonctions totales.

Nous dirons que P est **représentable** si pour tout l, suffisant pour P, il y a un terme X qui représnte P jusqu'à l.

Le théorème du point fixe suivant aura pour fonction essentiellement de régler le cas des programmes ayant des déroulements infinis.

Théorème du point fixe (bis)

Si $M[x]$ est un terme **normal**, il existe un terme N tel que $\varrho(N) \equiv M[N]$.

Démonstration

$M[x]$ étant normal, $M[xx]$ l'est également. Dès lors, $\varrho(M[xx]) \equiv M[xx]$. Le terme $N \equiv (\lambda x M[xx])(\lambda x M[xx])$ a donc bien la propriété requise.

□

Théorème

Tout programme URM est représentable.

Démonstration

Considérons un programme $P = I_1, \ldots, I_s$ et un nombre l, suffisant pour P. A chaque instruction I_i $(1 \leq i \leq s)$ nous associons un terme $P_i[x]$ ayant au plus x comme variable libre :

- $P_i[x] \equiv \lambda x_1 \ldots x_n \ldots x_l \,.\, x p^s_{i+1} x_1 \ldots \underline{0} \ldots x_l$, si I_i est $Z(n)$;
- $P_i[x] \equiv \lambda x_1 \ldots x_n \ldots x_l \,.\, x p^s_{i+1} x_1 \ldots \langle F, x_n \rangle \ldots x_l$, si I_i et $S(n)$;
- $P_i[x] \equiv \lambda x_1 \ldots x_l \,.\, (x p^s_q x_1 \ldots x_l$ si $x_n x_m$ et $x p^s_{i+1} x_1 \ldots x_l$ sinon), si I_i est $J(n, m, q)$.

(L'expression $x p^s_t x_1 \ldots x_l$ désigne x_1 si $t > s$.)

Les termes $P_i[x]$ étant normaux, le terme $\langle P_1[x], \ldots, P_s[x] \rangle$ est également normal.

Par le théorème du point fixe (bis) il existe un terme *Ins* tel que :

$$\varrho(Ins) \equiv \langle P_1[Ins], \ldots, P_s[Ins] \rangle.$$

Nous montrons que *Ins* représente P jusqu'à l. Soit $e_0, e_1, \ldots$ le déroulement de P appliqué à une suite de longueur l. Par construction, le lemme 2 et la proposition 1.1, on constate que :

$$\begin{aligned}
E_j &\equiv Ins\, p_i^s \underline{r_1} \ldots \underline{r_l} \\
&\to \varrho(E_j) \\
&\equiv \varrho(Ins) p_i^s \underline{r_1} \ldots \underline{r_l} \\
&\equiv \langle P_1[Ins], \ldots, P_s[Ins] \rangle\, p_i^s \underline{r_1} \ldots \underline{r_l} \\
&\to P_i[Ins]\, \underline{r_1} \ldots \underline{r_l} \\
&\to E_{j+1}, \quad \text{si } 1 \leq i \leq s.
\end{aligned}$$

Il ne reste donc qu'à montrer que E_0 n'a pas de forme normale si le déroulement est infini.

Remarquons d'abord que nous venons de montrer que $\varrho(E_j) \longrightarrow E_{j+1}$ (si e_j et e_{j+1} sont dans le déroulement de P). Donc, par récurrence, $\varrho^j(E_0) \longrightarrow E_j$ (si e_j est une étape du déroulement de P). En effet, $\varrho^0(E_0) \equiv E_0 \longrightarrow E_0$ et si $\varrho^j(E_0) \longrightarrow E_j$ alors $\varrho^{j+1}(E_0) \longrightarrow \varrho(E_j) \longrightarrow E_{j+1}$ (proposition 1.3).

Si le déroulement est infini, la réduction $E_0 \xrightarrow{1} \cdots \xrightarrow{1} E_1 \xrightarrow{1} \cdots \xrightarrow{1} E_j \xrightarrow{1} \cdots$ est infinie. Dès lors, si E_0 avait une forme normale : $\varrho^j(E_0)$ —par le corollaire du théorème de Church et Rosser— on aurait que $\varrho^j(E_0) \longrightarrow E_j$ et donc que E_j serait normal, *quod absurdum est.*

□

Corollaire (Kleene et Turing)

Si h est une fonction calculable de $\mathbf{N}^k \to \mathbf{N}$, il existe un terme H du calcul λ tel que, pour toute suite $n_1, \ldots, n_k$:

- si $h(n_1, \ldots, n_k) = m$, alors $H\underline{n_1} \ldots \underline{n_k} = \underline{m}$;
- si $h(n_1, \ldots, n_k) \uparrow$, alors $H\underline{n_1} \ldots \underline{n_k}$ n'a pas de forme normale.

Autrement dit les fonctions récursives partielles sont représentables dans le calcul λ.

Démonstration

Soit un programme URM pour h et l un nombre $\geq k$ et suffisant pour P. Par le théorème, il y a un terme Ins représentant P jusqu'à l. H sera donc le terme $\lambda x_1 \ldots x_k . Ins\, p_1^l x_1 \ldots x_k \underbrace{\underline{0} \ldots \underline{0}}_{l-k \text{ fois}}$.

□

Annexe : les numéraux de Church

La codification des naturels que nous avons utilisée est peu naturelle. Son avantage est de mener rapidement à une représentation de l'égalité, nécessaire pour reproduire les instructions de saut dans le calcul λ. La première codification des naturels a été proposée par Church. Elle est basée sur la notion intuitive d'itération d'une fonction. Au naturel n correspond l'opération qui consiste à itérer n fois une fonction. La définition des numéraux de Church est donc la suivante :

$$\ulcorner n \urcorner \equiv \lambda xy \,.\, x^n y,$$

$x^n y$ étant défini, par récurrence, comme suit :

$$x^0 y \equiv y \text{ et } x^{k+1} y \equiv x(x^k y).$$

Donc,

- $\ulcorner 0 \urcorner \equiv \lambda xy.y \equiv F$,
- $\ulcorner 1 \urcorner \equiv \lambda xy.xy$,
- $\ulcorner 2 \urcorner \equiv \lambda xy.x(xy)$,
- $\ulcorner 3 \urcorner \equiv \lambda xy.x(x(xy))$,

etc.

L'intérêt de ces numéraux, outre leur caractère naturel, est qu'ils permettent de donner une représentation fidèle des fonctions récursives primitives sans employer de points fixes. On aurait pu également les utiliser dans la démonstration du théorème de Kleene et Turing que nous avons donnée. Pour le montrer, il suffira de représenter la fonction successeur et l'égalité dans ce système par des termes normalisables.

En ce qui concerne le successeur, on remarque que $\ulcorner k \urcorner xy = x^k y$ —quel que soit k— et donc que $\ulcorner n+1 \urcorner xy = x^{n+1} y = x(x^n y) = x(\ulcorner n \urcorner xy)$. Il s'ensuit que si nous définissons *Succ* comme étant le terme $\lambda zxy.x(zxy)$, on a bien $Succ \ulcorner n \urcorner = \ulcorner n+1 \urcorner$.

Par ailleurs, si Z est le terme $(\lambda x(F,x),\underline{0})$, alors $Z\ulcorner n \urcorner = \ulcorner n \urcorner(\lambda x(F,x))\underline{0} = (\lambda x(F,x))^n \underline{0} = \underline{n}$.

Par conséquent, en posant $Egal \equiv \lambda xy.Zx(Zy)$, on vérifie que $Egal\ulcorner n \urcorner\ulcorner m \urcorner = \underline{n}\,\underline{m} = K$, si $n = m$ et F si $n \neq m$.

Université Catholique de Louvain
Département de Philosophie
Chemin d'Aristote 1
B-1348 Louvain-la-Neuve