Introduction au λ-calcul

Khelifa SABER
LAIC
&
LAMA
(Université de Savoie)

14 novembre 2006

Plan

1. **Quelques définitions**
   - Syntaxe
   - Calcul et réduction
   - Codage des entiers

2. **Principaux théorèmes**
   - Confluence
   - Réductions parallèles
   - Standardisation

3. **Typage du λ-calcul**
   - Le λ-calcul simplement typé
   - Forte normalisation
   - Extension au $\lambda\mu$-calcul

Un peu d'histoire

1. Inventé par le logicien américain **Alonzo Church** dans les années 30. Pour fournir un fondement au mathématiques plus simple que la théorie des ensembles. Basé sur la notion de fonction.
2. Cette initiative a échoué, le λ-calcul a un pouvoir expressif plus faible que la théorie des ensembles.
3. Par contre, il a le même pouvoir d'expression que les machines de Turing.
4. Ce qui fournit un bon outil pour fonder la notion de fonction calculable (traitement de la récursivité).
5. Explosion du λ-calcul dans les années 70 due à l'informatique, et grâce aussi à la découverte de **L'isomorphisme de Curry-Howard**.

La syntaxe...

Définition

1. Les variables : x, y, z,...
2. L'abstraction : si x est une variable et t un terme, alors $\lambda x.t$ est un terme.
3. L'application : si t et u sont deux termes, alors, $t\,u$ est un terme.

$$\mathcal{T} := \mathcal{X} \;\mid \lambda \mathcal{X}.\mathcal{T} \;\mid \mathcal{T}\,\mathcal{T}$$

...La syntaxe

Exemples

- $I = \lambda x.x$
- $\Delta = \lambda x.x\,x$
- $\Omega = (\lambda x.x\,x)\,\lambda x.x\,x = \Delta\,\Delta$
- $V = \lambda y.\lambda z.y$, $F = \lambda y.\lambda z.z$, $N = \lambda x.(x\,F)\,V$

La α-équivalence (α-renommage)

- $\lambda x.x$ et $\lambda y.y$ sont identifiés.
- $\lambda f.\lambda x.x\,f$, $\lambda x.\lambda f.f\,x$, $\lambda x.\lambda y.y\,x$ et $\lambda y.\lambda z.z\,y$ sont identifiés.

La β-réduction...

$$(\lambda x.u)\, v \;\longrightarrow_\beta\; u[x := v]$$

Définition

1. $(\lambda x.u)\, v \;\longrightarrow_\beta\; u[x := v]$
2. Si $u \;\longrightarrow_\beta\; u'$, alors $\lambda x.u \;\longrightarrow_\beta\; \lambda x.u'$
3. Si $u \;\longrightarrow_\beta\; u'$, alors $u\, v \;\longrightarrow_\beta\; u'\, v$
   et $v\, u \;\longrightarrow_\beta\; v\, u'$

Exemples

- $(I)\, t = (\lambda x.x)\, t \;\longrightarrow_\beta\; x[x := t] = t$
- $(N)\, V = (\lambda x.(x\, F)\, V)\, V \;\longrightarrow_\beta\; (V\, F)\, V \;\longrightarrow_\beta\; F$
- $(N)\, F = (\lambda x.(x\, F)\, V)\, F \;\longrightarrow_\beta\; (F\, F)\, V \;\longrightarrow_\beta\; V$
- $\Omega = (\lambda x.x\, x)\, \lambda x.x\, x \;\longrightarrow_\beta (\lambda x.x\, x)\, \lambda x.x\, x \;\longrightarrow_\beta \ldots$

... La β-réduction

Définition
$u[x := v]$ est obtenu de u en remplaçant les occurences libres de x par v.

Attention !

$$\begin{aligned}(\lambda x.\lambda y.x)\, y\, z &\longrightarrow_\beta ((\lambda y.x)[x := y])z \\ &= (\lambda y.y)\, z \\ &\longrightarrow_\beta z \qquad !\,!\,!\end{aligned}$$

Solution Le α-renommage :
On renomme d'abord les variables liées de la fonction qui sont libres dans l'argument.
D'où

$$\begin{aligned}(\lambda x.\lambda y.x)\, y\, z &= (\lambda x.\lambda a.x)\, y\, z \\ &\longrightarrow_\beta ((\lambda a.x)[x := y])\, z \\ &= (\lambda a.y)\, z \\ &\longrightarrow_\beta y\end{aligned}$$

Codage des entiers...

Définition

1. $\underline{0} = \lambda f.\lambda x.x$
2. $\underline{1} = \lambda f.\lambda x.f\,x$
3. $\underline{n} = \lambda f.\lambda x.\underbrace{(f)\ldots(f)}_{n\,fois}\,x$
4. $\underline{s} = \lambda n.\lambda f.\lambda x.(f)\,((n)\,f)\,x$
5. $\underline{+} = \lambda n.\lambda m.\lambda f.\lambda x.((n)\,f)\,((m)\,f)\,x$
6. $\underline{\times} = \lambda n.\lambda m.\lambda f.(n)\,(m)\,f$

Exemples

- $(\underline{s})\,\underline{2} \;= (\lambda n.\lambda f.\lambda x.(f)\,((n)\,f)\,x)\,\underline{2}$

$$\longrightarrow_\beta \lambda f.\lambda x.(f)\,((\underline{2})\,f)\,x$$

$$= \lambda f.\lambda x.(f)\,((\lambda f.\lambda x.(f)\,(f)\,x)\,f)\,x$$

$$\longrightarrow_\beta^* \lambda f.\lambda x.(f)\,(f)\,(f)\,x$$

$$= \underline{3}$$

...Codage des entiers

- $(\underline{+})\,\underline{2}\,\underline{3} = (\lambda n.\lambda m.\lambda f.\lambda x.((n)\,f)\,((m)\,f))\,\underline{2}\,\underline{3}$

$$\longrightarrow_\beta^* \lambda f.\lambda x.((\underline{2})\,f)\,((\underline{3})\,f)\,x$$

$$\longrightarrow_\beta^* \lambda f.\lambda x.(\lambda x.(f)\,(f)\,x)\,(f)\,(f)\,(f)\,x$$

$$\longrightarrow_\beta \lambda f.\lambda x.(f)\,(f)\,(f)\,(f)\,(f)\,x = \underline{5}$$

Exercice

Trouver un autre terme pour coder la fonction successeur $\underline{s}$.

Trouver un terme pour coder l'exponentielle.

Combinateur de point fixe

Théorème

Tout λ- terme M a un point fixe N, $M\,N =_\beta N$.

Mieux encore

Théorème

Il existe un terme Y appelé combinateur de point fixe qui trouve le point fixe de chaque terme M, $Y\,M$ est le point fixe de M

Preuve

1. $A = \lambda z.\lambda x.(x)((z)\,z)\,x$, et $Y = A\,A$
   On a $Y\,M \longrightarrow^* M\,(Y\,M)$.
2. $Y = \lambda f.(\lambda x.f\,(x\,x))\,(\lambda x.f\,(x\,x))$

Confluence

Théorème (Confluence locale)
Si $u \longrightarrow u_1$ et $u \longrightarrow u_2$, alors il existe u_3 tel que : $u_1 \longrightarrow^* u_3$ et $u_2 \longrightarrow^* u_3$.

Théorème (Confluence)
Si $u \longrightarrow^* u_1$ et $u \longrightarrow^* u_2$, alors il existe u_3 tel que : $u_1 \longrightarrow^* u_3$ et $u_2 \longrightarrow^* u_3$.

Corollaire (Unicité de la forme normale) Si v et w sont deux formes normales de u, alors $v = w$.

Réductions parallèles

Définition

1. $u \succ u$
2. $u \succ u'$, alors $(\lambda x.u) \succ \lambda x.u'$
3. $u \succ u'$ et $v \succ v'$, alors $u\,v \succ u'\,v'$
4. $u \succ u'$ et $v \succ v'$, alors $(\lambda x.u)\,v \succ u'[x := v']$

Lemme $\longrightarrow_\beta \subseteq \succ \subseteq \longrightarrow_\beta^*$

Théorème (Confluence forte)
Si $u \succ u_1$ et $u \succ u_2$, alors il existe u_3 tel que : $u_1 \succ u_3$ et $u_2 \succ u_3$.

Réduction standard

Définition ($u \longrightarrow_{st}^{*} u'$)

1. si $u = \lambda x.t$, alors $u' = \lambda x.t'$ où $t \longrightarrow_{st}^{*} t'$

2. si $u = (x)\, t_1 ... t_n$, alors $u' = (x)\, t'_1 ... t'_n$ où $t_i \longrightarrow_{st}^{*} t'_i$

3. si $u = (\lambda x.t)\, v\, v_1 ... v_n$, alors ou bien :

   - $u' = (\lambda x.t')\, v'\, v'_1 ... v'_n$ où $t \longrightarrow_{st}^{*} t'$, $v \longrightarrow_{st}^{*} v'$ et $v_i \longrightarrow_{st}^{*} v'_i$

     Ou bien
   - $u \longrightarrow (t[x := v])\, v_1 ... v_n \longrightarrow_{st}^{*} u'$

Standardisation et stratégies de réduction

Théorème
Si $u \longrightarrow^* u'$, alors $u \longrightarrow^*_{st} u'$.

Définition
La réduction gauche d'un terme consiste à réduire à chaque étape le redex le plus à gauche.

Corollaire
si u est faiblement normalisable alors la réduction gauche de u aboutit à sa forme normale.

Exemple
$(\lambda z.y)\ \underbrace{(\lambda x.x\,x)\,\lambda x.x\,x}_{\Omega} \longrightarrow y$

Le λ-calcul simplement typé

Définition

Les types : les formules de la logique propositionnelle implicative.

Un jugement $\Gamma \vdash t : A$
t un terme, A un type
et $\Gamma = \{x_i : A_i \,/\, 1 \leq i \leq n\}$ un contexte.

$$\frac{}{\Gamma, x : A \ \vdash x : A}ax$$

$$\frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : A \to B}\to_i \qquad \frac{\Gamma \vdash u : A \to B \qquad \Gamma \vdash v : A}{\Gamma \vdash u\ v : B}\to_e$$

La forte normalisation

Théorème
Le λ-calcul typé est fortement normalisable.

Preuve(Les candidats de réductibilités)

1. Associer à chaque type A un ensemble de termes fortement normalisables $|A|$ appelé interprétation de A.
2. Montrer que chaque terme est dans l'interprétation de sont type.

Le $\lambda\mu$-calcul...

La syntaxe

$\mathcal{X} = \{x, y, z, ...\}$ variables intuitionnistes.
$\mathcal{A} = \{a, b, c, ...\}$ variables classiques.

$$\mathcal{T} := x \,|\, \lambda x.t \,|\, t\,u \,|\, \mu a.t \,|\, (a\,t)$$

Le typage

$$\frac{}{\Gamma, x : A \ \vdash x : A;\ \Delta}ax$$

$$\frac{\Gamma, x : A \vdash u : B;\ \Delta}{\Gamma \vdash \lambda x.u : A \to B;\ \Delta}\to_i$$

$$\frac{\Gamma \vdash u : A \to B;\ \Delta \qquad \Gamma \vdash v : A;\ \Delta}{\Gamma \vdash u\,v : B;\ \Delta}\to_e$$

$$\frac{\Gamma \vdash u : \bot;\ \Delta, a : A}{\Gamma \vdash \mu a.u : A;\ \Delta}\bot_e$$

$$\frac{\Gamma \vdash u : A;\ \Delta, a : A}{\Gamma \vdash (a\,u) : \bot;\ \Delta, a : A}\bot_i$$

Les réductions

1. $(\lambda x.u)\, v \longrightarrow_\beta u[x := v]$

2. $(\mu a.u)\, v \longrightarrow_\mu \mu a.u[a :=^* v]$

   Où $u[a :=^* v]$ est obtenu de u en remplaçant chaque sous terme de la forme $(a\, t)$ par $(a\, (t\, v))$.

Théorème
Le $\lambda\mu$-calcul est confluent.

Théorème
Le $\lambda\mu$-calcul typé est fortement normalisable.

Références

1. **H. Barendregt.** *The Lambda Calculus : its Syntax and Semantics.* North-Holland, Amsterdam, Holland, 1984.

2. **R. David.** *A simple proof of basic results in λ calculus.* Infomatique théorique/Computer Science, p.1401-1406, 1995.

3. **R. David et K.Nour.** *Notes de DEA 2002/2003* Université de Savoie.

4. **J.-L. Krivine.** *Lambda calcul, types et modèles.* Masson, Paris, 1990.

5. **M. Parigot.** *$\lambda\mu$-calculus : An algorithm interpretation of classical natural deduction.* Lecture Notes in Artificial Intelligence (624), pp. 190-201. Springer Verlag 1992.

6. **M. Parigot.** *Proofs of strong normalization for second order classical natural deduction.* Journal of Symbolic Logic, 62 (4), pp. 1461-1479, 1997.