Introduction au lambda calcul

Pierre Lescanne

29 novembre 2004 – 15 h 59

lambda : *adj. fam.* : moyen, quelconque. *téléspectateur lambda*.

Dictionnaire le Robert

Les fonctions comme citoyens de première classe

On peut faire que les **preuves** soient citoyens de première classe, mais pourquoi pas les **fonctions** ?

Quelques dates

autour de 1870 un Italien[1] s'oppose à Cantor sur le point de savoir quel est le concept de base des mathématiques prétendant que ça devrait être les fonctions.

1920 Schönfinkel initie la logique combinatoire,

1925 Haskell Curry crée la logique combinatoire,

1936 Alonso Church crée le λ-calcul,

1970-... Explosion du λ-calcul due à l'informatique (Barendregt, Berry, Boehm, de Bruijn, Curien, Dezani-Ciancaglini, Girard, Hindley, Klop, Krivine, Levy, Milner, Plotkin, Scott, Statmann etc.)

[1] dont j'ai oublié le nom.

Des notations différentes, un même concept

en maths	$x \mapsto x$	$f \mapsto (x \mapsto f(f(x)))$
en CAML	`fun x -> x`	`fun f -> (fun x -> (f (f x)))`
en λ-calcul	$\lambda x.x$	$\lambda f.(\lambda x.(f(fx)))$

La syntaxe

La classe Λ est la plus petite classe qui contient

1. x si x est une variable,
2. $\lambda x.M$ si $M \in \Lambda$,
3. (MN) si $M \in \Lambda$ et $N \in \Lambda$.

La syntaxe

La classe Λ est la plus petite classe qui contient

1. x si x est une variable,
2. $\lambda x.M$ si $M \in \Lambda$, **abstraction**
3. (MN) si $M \in \Lambda$ et $N \in \Lambda$. **application**

Qu'y a-t-il derrière la syntaxe ?

On peut voir les termes comme des abstractions des fonctions ou des programmes fonctionnels.
Dans $\lambda x.M$, on dit que M est le **corps** de la fonction ou du programme.
Dans (MN), on peut voir M comme une fonction que l'on **applique** au paramètre N. La **valeur** va s'obtenir par «réduction» (approche intentionnelle).
Le lambda-calcul décrit les fonctions par leur **comportement**.

L'anecdote derrière la syntaxe ?

Au début Church voulait écrire $\hat{\mathtt{x}}$.
Mais au temps des machines à écrire on ne savait écrire que $\hat{}\,\mathtt{x}$.
Ce qui a donné $\wedge\mathtt{x}$, puis λx.

Exemples de termes

Exemples

$I \equiv \lambda x.x$

$K \equiv \lambda x(\lambda y.x)$

$S \equiv \lambda x.(\lambda y.(\lambda z.((xz)(yz))))$

$B \equiv \lambda x.(\lambda y.(\lambda z.(x(yz))))$

Convention 1 / 2

1. Au lieu de $\lambda x_1(\ldots(\lambda x_n.M)\ldots)$

   on écrit $\lambda x_1 \ldots x_n.M$.

   Exemple

   $\lambda xy.x$.

2. Au lieu de $(\ldots(MN_1)\ldots N_p)$

   on écrit $MN_1 \ldots N_p$ ou $M\vec{N}$, si $\vec{N} = (N_1 \ldots N_p)$.

   Exemple

   $\lambda xyz.xz(yz)$

   à la place de $\lambda x.(\lambda y.(\lambda z.((xz)(yz))))$.

Convention 2 / 2

Exemples

$((\lambda x.x)y)y$ donne $(\lambda x.x)yy$,

«La fonction identité appliquée à y,
puis le résultat est appliqué à y».

En revanche, $\lambda x.xyy$ correspond à $\lambda x.(xy)y$.

«La fonction qui à x fait correspondre le résultat de x appliqué à y
puis à y».

Les mêmes termes avec conventions

$$
\begin{aligned}
I &\equiv \lambda x.x \\
K &\equiv \lambda xy.x \\
S &\equiv \lambda xyz.xz(yz) \\
B &\equiv \lambda xyz.x(yz)
\end{aligned}
$$

Les mêmes termes avec conventions

$I \equiv \lambda x.x$	I est la fonction identité
$K \equiv \lambda xy.x$	Kc est la fonction constante c
$S \equiv \lambda xyz.xz(yz)$	$Sabc$ distribue c
$B \equiv \lambda xyz.x(yz)$	B permute l'effet des parenthèses

Plan

Les variables liées

$$
\begin{array}{rcl}
BV(x) & = & \varnothing \\
BV(\lambda x.M) & = & BV(M) \cup \{x\} \\
BV(MN) & = & BV(M) \cup BV(N)
\end{array}
$$

Les variables liées

$$\begin{aligned} BV(x) &= \varnothing \\ BV(\lambda x.M) &= BV(M) \cup \{x\} \\ BV(MN) &= BV(M) \cup BV(N) \end{aligned}$$

Exemple

$$\begin{aligned} BV(\lambda x.x) &= \{x\} \\ BV(\lambda fx.f(fx)) &= \{f, x\} \\ BV(\lambda fx.f(fxy)y) &= \{f, x\} \end{aligned}$$

Les variables libres 1 / 2

$$
\begin{array}{rcl}
FV(x) & = & \{x\} \\
FV(\lambda x.M) & = & FV(M) - \{x\} \\
FV(MN) & = & FV(M) \cup FV(N)
\end{array}
$$

Les variables libres 1 / 2

$$\begin{array}{rcl} FV(x) & = & \{x\} \\ FV(\lambda x.M) & = & FV(M) - \{x\} \\ FV(MN) & = & FV(M) \cup FV(N) \end{array}$$

Exemple

$$\begin{array}{rcl} FV(\lambda x.x) & = & \emptyset \\ FV(\lambda fx.f(fx)) & = & \emptyset \\ FV(\lambda fx.f(fxy)y) & = & \{y\} \\ FV(\lambda x.f(fx)) & = & \{f\} \end{array}$$

Les variables libres 2 / 2

Un terme qui n’a pas de variable libre est dit **clos** ou **fermé** ou est appelé un **combinateur**.

Les variables libres 2 / 2

Un terme qui n’a pas de variable libre est dit **clos** ou **fermé** ou est appelé un **combinateur**.

ATTENTION ! Une variable peut être *à la fois libre et liée* dans un terme.

Exemple

$x(\lambda x.x)$.

Le produit cartésien et la curryfication

Il n'y a pas de produit cartésien dans le λ-calcul simple.
Si on veut écrire :

$$\lambda(x,y).f(x,y) \quad \text{ou} \quad (x,y) \mapsto f(x,y)$$

on le remplace par

$$\lambda xy.fxy$$

C'est la **curryfication** (nommée après Haskell Curry).

Substitution 1 / 2

Substituer une variable par un terme ne consiste pas simplement à remplacer toutes les occurrences de la variable par ce terme, à cause du phénomène de **capture**.

Quand on écrit $M[x := P]$ on ne remplace pas simplement les occurrences de x dans M par P.

Ainsi

$$\begin{array}{rcl} x(\lambda x.x)[x := y] & \neq & y(\lambda x.y) \\ (\lambda y.x)[x := y] & \neq & \lambda y.y \end{array}$$

donc il faut être prudent.

Substitution avec renommage

1. $x[x := P] = P$
2. $y[x := P] = y$
3. $(\lambda x.M)[x := P] = \lambda x.M$
4. $(\lambda y.M)[x := P] = \lambda y.(M[x := P])$
   si $x \notin FV(M)$ ou $y \notin FV(P)$
5. $(\lambda y.M)[x := P] = \lambda z.(M[y := z][x := P])$
   si $x \in FV(M)$ et $y \in FV(P)$
   et z est une *nouvelle* variable
6. $(M_1 M_2)[x := P] = M_1[x := P] M_2[x := P]$

La convention de Barendregt

C’est une *convention sur les variables libres* d’un terme *dans un énoncé mathématique*.

Il n’existe aucun sous-terme dans lequel une variable apparaît à la fois libre et liée.

L'α-conversion (règles structurelles)

$$\lambda x.N \equiv_\alpha \lambda y.(N[x := y]) \qquad \text{si } y \notin FV(N) \qquad \textit{base}$$

$$\frac{M_1 \equiv_\alpha N_1 \qquad M_2 \equiv_\alpha N_2}{M_1 M_2 \equiv_\alpha N_1 N_2}\ \alpha APP$$

$$\frac{M \equiv_\alpha N}{\lambda z.M \equiv_\alpha \lambda z.N}\ \alpha ABS$$

$$x \equiv_\alpha x \qquad \alpha var$$

Réflexivité de l'α-conversion

Lemme

Pour tout $M \in \Lambda$, on a $M \equiv_\alpha M$.

Démonstration.

Par induction structurelle sur M.

- M est la variable x. Dans ce cas on applique l'axiome αVar.
- M est une application $M_1 M_2$. Alors par induction on a $M_1 \equiv_\alpha M_1$ et $M_2 \equiv_\alpha M_2$ donc on peut appliquer la règle αAPP pour obtenir $M_1 M_2 \equiv_\alpha M_1 M_2$.
- M est une abstraction $\lambda x.P$. Par induction $P \equiv_\alpha P$. Donc par la règle αABS on a $\lambda x.P \equiv_\alpha \lambda x.P$.

$\square$

L'α-conversion (règles de congruence)

$$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}\ \alpha\text{symétrie}$$

$$\frac{M \equiv_\alpha N \qquad N \equiv_\alpha P}{M \equiv_\alpha P}\ \alpha\text{transitivité}$$

L'α-conversion (règles de congruence)

$$\frac{M \equiv_\alpha N}{N \equiv_\alpha M}\ \alpha symétrie$$

$$\frac{M \equiv_\alpha N \qquad N \equiv_\alpha P}{M \equiv_\alpha P}\ \alpha transitivité$$

L'α-conversion est une **relation d'équivalence**, stable par passage au contexte, on dit que c'est une **congruence**.

α-*conversion et convention de Barendregt*

L'α-conversion ne change pas la «signification» des termes.

- On suppose que dans tout théorème que l'on énonce, on suit la convention de Barendregt.
- Si l'on a un terme qui ne satisfait pas la convention de Barendregt, on s'y ramène par α-conversion

Substitution et convention de Barendregt

Avec la convention de Barendregt, la définition des substitutions devient beaucoup plus simple.

- $x[x := P] = P$
- $y[x := P] = y$
- $(\lambda y.M)[x := P] = \lambda y.M[x := P]$
- $(M_1 M_2)[x := P] = M_1[x := P] M_2[x := P]$

Plan

La β-contraction

Les fonctions sont faites pour calculer !
Les réductions d'un terme représentent son calcul.
La β-**contraction** en est l'étape élémentaire.

$$(\lambda x.M)P \xrightarrow[\beta]{} M[x := P]$$

bêta : *Subst. masc.* : Personne peu intelligente.

Le Trésor de la Langue française informatisé

R-contraction 1 / 2

On se donne un ensemble R de règles, c-à-d de paires de termes, par exemple β.

$M \xrightarrow[R]{} N$ signifie que

- M se contracte en N par R,
- ou M se réduit à N par R en une étape.

R-contraction 2 / 2

$$\frac{(M,N) \in R}{M \xrightarrow[R]{} N} \text{ (contraction)} \qquad \frac{M \xrightarrow[R]{} N}{\lambda x M \xrightarrow[R]{} \lambda x N} \ (\xi)$$

$$\frac{M \xrightarrow[R]{} N}{MP \xrightarrow[R]{} NP} \text{ (congruence gauche)}$$

$$\frac{M \xrightarrow[R]{} N}{PM \xrightarrow[R]{} PN} \text{ (congruence droite)}$$

$\xrightarrow[R]{}$ est alors appelée une **contraction**.

Exercice

Réduire

- $\lambda y.(\lambda x.x)z$
- $(\lambda fx.f(fx))(\lambda x.x)$
- $(\lambda fx.f(fx))(\lambda fx.fx)$

D'autres exemples de contraction 1 / 2

La contraction β_v ou **appel par valeur**

Pour toute **abstraction** ou toute **variable** $P \in \Lambda$

$$(\lambda x.M)P \xrightarrow[\beta_v]{} M[x := P]$$

La **contraction** η

Pour tout $M \in \Lambda$ et $x \notin FV(M)$,

$$\lambda x.Mx \xrightarrow[\eta]{} M.$$

L'**expansion** η

Pour tout $M \in \Lambda$ et $x \notin FV(M)$,

$$M \xrightarrow[\eta_{exp}]{} \lambda x.Mx.$$

D'autres exemples de contraction 2 / 2

La **contraction par β et η**

$$\xrightarrow[\beta\eta]{} \quad = \quad \xrightarrow[\beta]{} \cup \xrightarrow[\eta]{} .$$

On s'intéresse à la **$\beta\eta$-réduction** $\twoheadrightarrow_{\beta\eta}$ qui est la fermeture transitive et réflexive de $\xrightarrow[\beta\eta]{}$.

Fermeture transitive et réflexive

$$\frac{M \xrightarrow[R]{} N}{M \xrightarrow[R]{}\!\!\!\!\to N} \text{(cas de base)} \qquad \frac{}{M \xrightarrow[R]{}\!\!\!\!\to M} \text{(réflexivité)}$$

$$\frac{M \xrightarrow[R]{}\!\!\!\!\to N \qquad N \xrightarrow[R]{}\!\!\!\!\to L}{M \xrightarrow[R]{}\!\!\!\!\to L} \text{(transitivité)}$$

Proposition

Préservation de $\underset{\beta}{\twoheadrightarrow}$ *par abstraction et application.*

$$\frac{M \underset{\beta}{\twoheadrightarrow} N}{\lambda x.M \underset{\beta}{\twoheadrightarrow} \lambda x.N}$$

$$\frac{M \underset{\beta}{\twoheadrightarrow} N \qquad P \underset{\beta}{\twoheadrightarrow} Q}{MP \underset{\beta}{\twoheadrightarrow} NQ}$$

Preuve 1 / 2

Cas $\dfrac{M \xrightarrow[\beta]{} \!\!\!\!\!\!\to N}{\lambda x.M \xrightarrow[\beta]{} \!\!\!\!\!\!\to \lambda x.N}$

On doit montrer que

- sous l'**hypothèse** $M \twoheadrightarrow_{\beta} N$
- on a la **conclusion** $\lambda x.M \twoheadrightarrow_{\beta} \lambda x.N$.

La démonstration est par **induction** sur la taille de l'arbre de preuve de $M \twoheadrightarrow_{\beta} N$.

Elle utilise les règles de la définition de $\twoheadrightarrow_{\beta}$ et celle de $\xrightarrow[\beta]{}$.

Preuve 2 / 2

Trois cas se présentent :

1. $M \xrightarrow[\beta]{} N$, on a utilisé le «cas de base», alors par (ξ),
   $\lambda x.M \xrightarrow[\beta]{} \lambda x.N$ et on conclut par le «cas de base».
2. $M \equiv N$, alors $\lambda x.M \equiv \lambda x.N$ et conclut par «réflexivité».
3. Il existe P tel que $M \twoheadrightarrow_{\beta} P$ et $P \twoheadrightarrow_{\beta} N$. Par induction, on tire,
   - $\lambda x.M \twoheadrightarrow_{\beta} \lambda x.P$
   - et $\lambda x.P \twoheadrightarrow_{\beta} \lambda x.N$,

   et par «transitivité» $\lambda x.M \twoheadrightarrow_{\beta} \lambda x.N$.

Fermeture transitive, réflexive et symétrique 1 / 2

Avec les règles

$$\frac{M \xrightarrow[R]{} N}{M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} N}\ \text{(cas de base)} \qquad \frac{}{M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} M}\ \text{(réflexivité)}$$

$$\frac{M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} N \qquad N \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} L}{M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} L}\ \text{(transitivité)} \qquad \frac{M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} N}{N \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} M}\ \text{(symétrie)}$$

on obtient la fermeture transitive, réflexive et symétrique de $\xrightarrow[R]{}$

dite aussi **R-conversion**.

Fermeture transitive, réflexive et symétrique 2 / 2

- La R-conversion s'écrit $\overset{}{\underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow}}$ ou $=_R$ ou $\overset{*}{\underset{R}{\longleftrightarrow}}$
- $M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} P$ se dit
  - M est R-**égal** à P
  - ou M est R-**convertible** à P.

Plan

Contexte 1 / 2

Un **contexte** $C[\ ]$ est défini ainsi

1. $[\ ]$ est un contexte,
2. si $M \in \Lambda$ et si $C[\ ]$ est un contexte alors $MC[\ ]$ et $C[\ ]M$ sont des contextes,
3. si $C[\ ]$ est un contexte alors $\lambda x.C[\ ]$ est un contexte.

Définition

Si $C[\,]$ est un contexte et $A \in \Lambda$ alors $C[A]$ est défini par induction sur $C[\,]$.

- $[A] = A$,
- si $C[\,] = \lambda x.D[\,]$ alors $C[A] = \lambda x.D[A]$,
- si $C[\,] = MD[\,]$ alors $C[A] = MD[A]$,
- si $C[\,] = D[\,]M$ alors $C[A] = D[A]M$,

Stabilité 1 / 2

Un relation $\xrightarrow[R]{}$ est **stable par contexte** si

$M \xrightarrow[R]{} N$ alors $C[M] \xrightarrow[R]{} C[N]$.

Un relation $\xrightarrow[R]{}$ est **stable par substitution** si

$M \xrightarrow[R]{} N$ alors $P[x := M] \xrightarrow[R]{} P[x := N]$.

Stabilité 2 / 2

Proposition

Soit $\xrightarrow[R]{}$ *une contraction.*

$\xrightarrow[R]{}$, $\xrightarrow[R]{*}$ *et* $\xleftrightarrow[R]{*}$ *sont stables par contexte.*

$\xrightarrow[R]{*}$ *et* $\xleftrightarrow[R]{*}$ *sont stables par substitutions.*

Proposition

$\xrightarrow[R]{}$ *est stable par contexte.*

Démonstration.

Si $M \xrightarrow[R]{} N$ alors $C[M] \xrightarrow[R]{} C[N]$

Par induction sur la structure de $C[\,]$, sachant que $M \xrightarrow[R]{} N$.

1. $C[\,] = [\,]$ alors $C[M] = M$ et $C[N] = N$, évident.
2. $C[\,] = AD[\,]$,
   - par induction $D[M] \xrightarrow[R]{} D[N]$,
   - d'autre part, $C[M] = AD[M]$ et $C[N] = AD[N]$,
   - donc par congruence à droite $C[M] \xrightarrow[R]{} C[N]$.
3. $C[\,] = D[\,]A$, comme 2 en changeant «droite» en «gauche».
4. $C[\,] = \lambda x.D[\,]$, par induction $D[M] \xrightarrow[R]{} D[N]$, d'où la conclusion par (ξ). □

Stabilité par substitution

Proposition

$M \xrightarrow[R]{} \!\!\!\!\rightarrow N$ *implique* $A[x := M] \xrightarrow[R]{} \!\!\!\!\rightarrow A[x := N]$.

Stabilité par substitution

Proposition

$M \xrightarrow[R]{} \!\!\!\!\!\!\!\!\rightarrow N$ *implique* $A[x := M] \twoheadrightarrow_R A[x := N]$.

Démonstration.

L'hypothèse est $M \twoheadrightarrow_R N$.

La démonstration se fait par induction sur A.

- $A \equiv x$, alors $A[x := M] \equiv M \twoheadrightarrow_R N \equiv A[x := N]$.
- $A \equiv y$, alors $A[x := M] \equiv y \twoheadrightarrow_R y \equiv A[x := N]$ par réflexivité.

Stabilité par substitution (suite)

- $A \equiv \lambda y.B$,
  par induction $B[x := M] \xrightarrow[R]{} \!\!\!\!\!\rightarrow B[x := N]$,
  donc $A[x := M] \equiv \lambda y.B[x := M]$ et $A[x := N] \equiv \lambda y.B[x := N]$,
  par (ξ) pour $\xrightarrow[R]{} \!\!\!\!\!\rightarrow$, on a $\lambda y.B[x := M] \xrightarrow[R]{} \!\!\!\!\!\rightarrow \lambda y.B[x := N]$.

Stabilité par substitution (suite)

- $A \equiv BB'$

  par induction

$$\begin{array}{ccc} B[x := M] & \xrightarrow[R]{} \!\!\!\!\! \twoheadrightarrow & B[x := N] \\ B'[x := M] & \xrightarrow[R]{} \!\!\!\!\! \twoheadrightarrow & B'[x := N] \end{array}$$

  par définition de la substitution

$$\begin{array}{ccc} A[x := M] & \equiv & B[x := M]\, B'[x := M] \\ A[x := N] & \equiv & B[x := N]\, B'[x := N] \end{array}$$

Stabilité par substitution (fin)

par congruence et transitivité de $\underset{R}{\twoheadrightarrow}$ on a

$$\dfrac{\dfrac{B[x := M] \twoheadrightarrow B[x := N]}{\begin{array}{c} B[x := M]B'[x := M] \\ \twoheadrightarrow \\ B[x := N]B'[x := M] \end{array}} \qquad \dfrac{B'[x := M] \twoheadrightarrow B'[x := N]}{\begin{array}{c} B[x := N]B'[x := M] \\ \twoheadrightarrow \\ B[x := N]B'[x := N] \end{array}}}{B[x := M]B'[x := M] \twoheadrightarrow B[x := N]B'[x := N]}$$

$\square$

Exercice

Montrez que ça ne peut pas marcher pour $\xrightarrow[R]{}$,

c'est-à-dire qu'*on n'a pas* :

$M \xrightarrow[R]{} N$ implique $A[x := M] \xrightarrow[R]{} A[x := N]$.

Plan

Quelques définitions

- Un R-**redex** est un terme M tel que $(M, N) \in R$.
- N et le R-contracté de M.
- Un terme M est R-**irréductible** si M ne contient aucun R-redex.

Quelques exemples de redex

Les β-redex sont de la forme $(\lambda x.M)N$.
Les η-redex sont de la forme $\lambda x.(Mx)$ si x *n'est pas libre dans* M.
Tout terme est un redex pour l'expansion η.

Formes normales

Un terme N est une **forme normale** de M,

si N est R-irréductible et si $M \underset{R}{\twoheadleftarrow\!\!\twoheadrightarrow} N$.

On n'affirme

- *ni l'existence* cf le terme $(\lambda x.xx)\ (\lambda x.xx)$,
- *ni l'unicité*, il y a unicité pour β, mais il faut le prouver.

  La forme β-normale de M si elle existe (et si on a prouvé l'unicité) est la valeur intentionnelle de M.

Exercice

Lesquels de ces termes sont des formes normales ?

$(\lambda x.x)$

$((\lambda xy.x)v)w$

$(\lambda xy.xv)w$

$\lambda xy.xvw$

$(\lambda x.xx)\ (\lambda x.xx)$

$(\lambda xy.y)((\lambda x.xx)\ (\lambda x.xx))$

Plan

Ω *et les autres*

$$\omega \equiv \lambda x.xx$$
$$\Omega \equiv (\lambda x.xx)(\lambda x.xx)$$
$$\mathsf{Y} \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$
$$\mathsf{W}_F \equiv (\lambda x.F(xx))$$

Exercices 1 / 2

Montrez que Ω se récrit vers un unique terme. Lequel ?
Plus précisément, montrez qu'il existe un terme unique $M \in \Lambda$ tel que $\Omega \xrightarrow[\beta]{} M$.

Ω n'a pas de forme normale.

Exercices 2 / 2

1. Montrez que $\mathsf{Y}F \xrightarrow[\beta]{} F(\mathsf{W}_F\mathsf{W}_F)$.
2. Montrez que $F(\mathsf{Y}F) \xrightarrow[\beta]{} F(\mathsf{W}_F\mathsf{W}_F)$.
3. Conclure que $\mathsf{Y}F \xleftrightarrow[\beta]{} F(\mathsf{Y}F)$.

Y est appelé le combinateur de **point de fixe**.
Y et $\mathsf{Y}F$ n'ont pas de formes normales.

Plan

Entiers de Church

- $(\lambda fx.f(fx)) \equiv \mathbf{2}$ correspond au nombre entier *deux*.
- $(\lambda fx.x) \equiv T$ correspond à *zéro*.
- $(\lambda fx.fx) \equiv \mathbf{1}$ correspond à *un*.
- Plus généralement l'entier **n** est le terme $(\lambda fx.f^n x)$.

Exercice

1. Montrez que

$$\mathbf{1} \xrightarrow[\eta]{} I \equiv \lambda x.x.$$

2. Écrivez l'opération «successeur».
3. Écrivez les opérations d'«addition» et de «multiplication».
4. Calculez
   - $\mathbf{1}\mathbf{2}$,
   - $\mathbf{2}\mathbf{1}$,
   - $\mathbf{2}\mathbf{2}$,
5. A quoi correspond $\mathbf{m}\mathbf{n}$?

En effet,

$$\begin{array}{lcl} \mathbf{1} & \equiv & \lambda fx.fx \\ & \xrightarrow[\eta]{} & \lambda f.f \quad \equiv \quad I. \end{array}$$

Corrigé 2 / 6

Le **successeur** est **succ** $\equiv \lambda nfx.n\ f\ (f\ x)$,
l'**addition** est **add** $\equiv \lambda mnfx.\ m\ f\ (n\ f\ x)$,
tandis que la **multiplication** est **mult** $\equiv \lambda mnf.\ m\ (n\ f)$.

Corrigé 3 / 6

$$
\begin{array}{lcl}
\mathbf{22} & \equiv & (\lambda fx.f(f\ x))\ (\lambda gy.g(g\ y)) \\
 & \longrightarrow & \lambda x.(\lambda gy.g(g\ y))\ ((\lambda hz.h(h\ z))\ x) \\
 & \longrightarrow & \lambda x.(\lambda gy.g(g\ y))\ (\lambda z.x(x\ z)) \\
 & \longrightarrow & \lambda xy.(\lambda z.x(x\ z))\ ((\lambda w.x(x\ w))y) \\
 & \longrightarrow & \lambda xy.(\lambda z.x(x\ z))\ (x\ (x\ y)) \\
 & \longrightarrow & \lambda xy.x(x\ (x\ (x\ y))) \\
 & \equiv & \lambda fx.f(f\ (f\ (f\ x))) \quad \equiv \quad \mathbf{4}
\end{array}
$$

Appelons $\mathbf{p} \equiv \lambda mn.m\ n$ cette opération.
Autrement dit $\mathbf{p\ m\ n} = \mathbf{m\ n}$.
On remarque que

$$\begin{array}{lcl} \mathbf{p\ 0\ n} & \xrightarrow[\beta]{} & (\lambda fx.x)\ \mathbf{n} \\ & \xrightarrow[\beta]{} & \lambda x.x \\ & \xleftarrow[\eta]{} & \mathbf{1} \end{array}$$

Corrigé 5 / 6

et que

$$
\begin{array}{llll}
\mathbf{p}\ (\mathbf{succ}\ \mathbf{m})\ \mathbf{n}\ x & \triangleq & \mathbf{succ}\ \mathbf{m}\ \mathbf{n}\ x & \text{Définition de } \mathbf{p} \\
 & \xrightarrow[\beta]{} & (\mathbf{m}\ \mathbf{n})\ (\mathbf{n}\ x) & \text{Définition de } \mathbf{succ} \\
 & \xleftarrow[\beta]{} & \mathbf{mult}\ (\mathbf{m}\ \mathbf{n})\ \mathbf{n}\ x & \text{Définition de } \mathbf{mult} \\
 & \triangleq & \mathbf{mult}\ (\mathbf{p}\ \mathbf{m}\ \mathbf{n})\ \mathbf{n}\ x & \text{Définition de } \mathbf{p}.
\end{array}
$$

Corrigé 6 / 6

On a donc

$$\begin{aligned} \mathbf{p\ 0\ n} &=_{\beta\eta} \mathbf{1} \\ \mathbf{p}\ (\mathbf{succ\ m})\ \mathbf{n} &=_{\beta\eta} \mathbf{mult}\ (\mathbf{p\ m\ n})\ \mathbf{n} \end{aligned}$$

Corrigé 6 / 6

On a donc
$$\begin{aligned}\mathbf{p\ 0\ n} &=_{\beta\eta} \mathbf{1}\\ \mathbf{p}\ (\mathbf{succ\ m})\ \mathbf{n} &=_{\beta\eta} \mathbf{mult}\ (\mathbf{p\ m\ n})\ \mathbf{n}\end{aligned}$$

Or on a
$$\begin{aligned}n^0 &= 1\\ n^{m+1} &= n^m \cdot n.\end{aligned}$$

p est un bon candidat pour représenter l'**exponentielle**. Il y a simplement un problème : *on «applique» un entier à un entier*.

Plan

Le lambda calcul est maximalement cohérent

Un théorie $\mathcal{T}$ est **maximalement cohérente**

(on dit aussi qu'elle est **complète au sens de Hilbert**)

si pour tout φ

- ou bien $\vdash \varphi$,
- $\mathcal{T} + \varphi$ est incohérente.

Un théorie $\mathcal{T}$ est **maximalement cohérente**
(on dit aussi qu'elle est **complète au sens de Hilbert**)
si pour tout φ

- ou bien $\vdash \varphi$,
- $\mathcal{T} + \varphi$ est incohérente.

La $\beta\eta$-calcul est maximalement cohérent,
autrement dit pour tout M et N,

- ou bien $\vdash M \underset{\beta\eta}{\twoheadleftarrow\!\!\twoheadrightarrow} N$,
- $\beta\eta + (M \twoheadleftarrow\!\!\twoheadrightarrow N)$ est incohérent, autrement dit
pour tout $P, Q \in \Lambda$, on a
$\beta\eta + (M \twoheadleftarrow\!\!\twoheadrightarrow N) \quad \vdash \quad P \twoheadleftarrow\!\!\twoheadrightarrow Q$.

On va considérer un cas assez spécifique

Supposons que l'on ajoute au lambda calcul l'identité $\mathbf{S} = \mathbf{K}$.

Notons que d'une part

$$\begin{array}{lll} \mathbf{S}\ \mathbf{I}\ (\mathbf{K}\ P)\ \mathbf{I} & \xrightarrow[\beta]{} & \mathbf{I}\ (\mathbf{K}\ P)\ \mathbf{I} \\ & \xrightarrow[\beta]{} & \mathbf{K}\ P\ \mathbf{I} \\ & \xrightarrow[\beta]{} & P \end{array}$$

$$\mathbf{K}\,\mathbf{I}\,(\mathbf{K}\,P)\,\mathbf{I} \;\xrightarrow[\beta]{}\; \mathbf{I}\,\mathbf{I}$$

et d'autre part

$$\xrightarrow[\beta]{}\; \mathbf{I}$$

Donc de $\mathbf{S} = \mathbf{K}$ on déduit que pour tout terme P on a $P = \mathbf{I}$.

Le lambda calcul est maximalement cohérent 4 / 4

Autrement dit, si on ajoute au lambda calcul l'unique égalité $\mathbf{S} = \mathbf{K}$, on le rend incohérent en autorisant tous les termes à être égaux.