Le λ-calcul pur.

Le λ-calcul a trois domaines proches :

- la *calculabilité*, avec l'équivalence entre machines de Turing et λ-expression (vue en FDI) ;
- la *programmation fonctionnelle* (vue en **Théorie de la Programmation** [Chapitre 6] avec le petit langage FUN) ;
- la *théorie de la démonstration* (vue dans la suite de ce cours).

On se donne un ensemble infini $\mathscr{V}$ de variables notées $x, y, z, \ldots$ Les *termes* (du λ-calcul) ou λ-termes sont définis par la grammaire

$$M, N, \ldots ::= \lambda x.\ M \mid M\ N \mid x.$$

La construction $\lambda x.\ M$ s'appelle l' *abstraction* ou λ-*abstraction*. Elle était notée `fun` $x \to M$ en cours de théorie de la programmation.

Notation.

- On notera $M\ N\ P$ pour $(M\ N)\ P$.
- On notera $\lambda xyz.M$ pour $\lambda x.\lambda y.\lambda z M$ (il n'y a pas lieu de mettre des parenthèses ici, vu qu'il n'y a pas d'ambigüités).
- On notera $\lambda x.\ M\ N$ pour $\lambda x.\ (M\ N)$. ***Attention***, c'est différent de $(\lambda x.\ M)\ N$.

1 Liaison et α-conversion.

Remarque 1 (Liaison). Le « λ » est un lieur. Dans $\lambda y.\ x\ y$, la variable y est *liée* mais pas x (la variable x libre). On note $\mathscr{V}\ell(M)$ l'ensemble des variables libres de M, définie par induction sur M (il y a *3* cas).

Remarque 2 (α-conversion).
On note $=_\alpha$ la relation d' α-*conversion*. C'est une relation binaire sur les λ-termes fondée sur l'idée de renommage des abstractions ***en évitant la capture de variables libres*** :

$$\lambda x.\, x\, y =_\alpha \lambda t.\, x\, t \neq_\alpha \lambda x. x\, x.$$

Ainsi $\lambda x.\, M =_\alpha \lambda z.\, M'$ où M' est obtenu en remplaçant x par z ***là où il apparaît libre*** et ***à condition que*** $z \notin \mathscr{V}\ell(M)$. Ceci, on peut le faire partout.

Lemme 1. La relation $=_\alpha$ est une relation d'équivalence. Si $M =_\alpha N$ alors $\mathscr{V}\ell(M) = \mathscr{V}\ell(N)$.

Par convention, on peut identifier les termes modulo $=_\alpha$. On pourra donc toujours dire

« considérons $\lambda x.\, M$ où $x \notin E$ [...] »

avec E un ensemble ***fini*** de variables.

Ceci veut dire qu'on notera

$$M = N \text{ pour signifier que } M =_\alpha N.$$

2 La β-réduction.

Définition 1 (β-réduction). On définit la relation de β-réduction sur les λ-termes, notée $\to_\beta$ ou $\to$, définie par les règles d'inférences :

$$\frac{}{(\lambda x.\, M)\, N \to_\beta M[N/x]} \qquad \frac{M \to_\beta M'}{\lambda x.\, M \to_\beta \lambda x.\, M'}$$

$$\frac{M \to_\beta M'}{M\, N \to_\beta M'\, N} \qquad \frac{N \to_\beta N'}{M\, N \to_\beta M\, N'}$$

où $M[{}^N\!/\!_x]$ est la substitution de x par N dans M (on le défini ci-après).

Définition 2. Un terme de la forme $(\lambda x.\ M)\ N$ est appelé un *redex* (pour *reducible expression*) ou *β-redex*. Un terme M est une *forme normale* s'il n'existe pas de N tel que $M \to_\beta N$.

Remarque 3. La relation $\to_\beta$ n'est pas terminante :

$$\Omega := (\lambda x.\ x\ x)\ (\lambda y.\ y\ y) \to_\beta (\lambda y.\ y\ y)\ (\lambda y.\ y\ y) =_\alpha \Omega.$$

Exemple 1.

$$\begin{tikzcd}
(\lambda x.\ x\ x)\ ((\lambda z.\ z)\ t) \arrow[r] \arrow[dd, "(\star)"] & (\lambda x\ x\ x)\ t \arrow[dr] & \\
& ((\lambda z.\ z)\ t)\ t \arrow[r] & tt\ . \\
((\lambda z.\ z)\ t)\ ((\lambda z.\ z)\ t) \arrow[ur] \arrow[r, "(\star\star)"] & t\ ((\lambda z.\ z)\ t) \arrow[ur] &
\end{tikzcd}$$

Un pas de β-réduction peut :

- dupliquer un terme (*c.f.* $(\star)$) ;
- laisser un redex inchangé (*c.f.* $(\star\star)$) ;
- faire disparaitre un redex (qui n'est pas celui que l'on contracte) :

$$(\lambda x.\ u)((\lambda z.\ z)\ t)\ \to_\beta u\ ;$$

- créer de nouveaux redex :

$$(\lambda x.\ x\ y)\ (\lambda z.\ z) \to_\beta (\lambda z.\ z)\ y.$$

3 Substitutions.

Exemple 2. Le terme $\lambda xy.\, x$ c'est une « fonction fabriquant des fonctions constantes » au sens où

$$(\lambda xy.\, x)M \to_\beta \lambda y.\, M,$$

à condition que $y \notin \mathscr{V}\ell(M)$. On doit cependant α-renommer pour éviter la capture :

$$\begin{array}{c} (\lambda xy.x)\,(\lambda t.\, y) \not\to_\beta \lambda y.\,(\lambda t.\, y) \\ \|_\alpha \\ (\lambda xy'.x)\,(\lambda t.\, y) \to_\beta \lambda y'.\,(\lambda t.\, y). \end{array}$$

Définition 3. On procède par induction, il y a trois cas :

- $y[{}^N\!/_x] := \begin{cases} N & \text{si } y = x \\ y & \text{si } y \neq x \end{cases}$
- $(M_1\ M_2)[{}^N\!/_x] := (M_1[{}^N\!/_x])\ (M_2[{}^N\!/_x])$
- $(\lambda y.\, M)[{}^N\!/_x] := \lambda y.\,(M[{}^N\!/_x])$ ***à condition que*** $y \notin \mathscr{V}\ell(N)$ ***et*** $y \neq x$.

Lemme 2 (Gymnastique des substitutions)**.** Pour $y \notin \mathscr{V}\ell(R)$,

$$(P[{}^Q\!/_y])[{}^R\!/_x] = (P[{}^R\!/_x])[{}^{Q[{}^R\!/_x]}\!/_y].$$

Lemme 3. Si $M \to_\beta M'$ alors $\mathscr{V}\ell(M') \subseteq \mathscr{V}\ell(M)$.

4 Comparaison λ-calcul et FUN.

En λ-calcul, on a une règle

$$\frac{M \to_\beta M'}{\lambda x.\, M \to_\beta \lambda x.\, M'}.$$

Cette règle n'existe pas en FUN (ni en fouine) car on traite les fonctions comme des valeurs. Et, en FUN, les trois règles suivantes sont mutuellement exclusives :

$$\dfrac{}{(\lambda x.\ M)\ N \to_\beta M[{}^{N}/_{x}]} \qquad \dfrac{M \to_\beta M'}{M\ N \to_\beta M'\ N} \qquad \dfrac{N \to_\beta N'}{M\ N \to_\beta M\ N'}$$

car on attend que N soit une ***valeur*** avant de substituer.

En FUN (comme en fouine), pour l'exemple 1, on se limite à n'utiliser que les flèches rouges.

La relation $\to_\beta$ est donc « plus riche » que $\to_{\mathsf{FUN}}$. En FUN, on a une *stratégie de réduction* : on a au plus un redex qui peut être contracté. On n'a pas de notion de valeur en λ-calcul pur. Le « *résultat d'un calcul* » est une forme normale.

5 Exercice : les booléens.

On définit

$$\mathbf{T} := \lambda xy.\ x \qquad\qquad \mathbf{F} := \lambda xy.\ y.$$

Ainsi, pour tout M (si $y \notin \mathscr{V}\ell(M)$),

$$\mathbf{T}\ M \to \lambda y.\ M \qquad \mathbf{F}\ M \to \lambda y.\ y =: \mathbf{I}.$$

La construction `if` b `then` M `else` N se traduit en $b\ M\ N$.

Le « non » booléen peut se définir par :

- $\mathbf{not} := \lambda b.\ b\ \mathbf{F}\ \mathbf{T} = \lambda b.\ b\ (\lambda xy.\ y)\ (\lambda tu.\ t)$;
- $\mathbf{not}' := \lambda b.\ \lambda xy.\ byx$.

La première version est plus abstraite, la seconde est « plus électricien ». On a deux formes normales ***différentes***.

De même, on peut définir le « et » booléen :

- $\mathbf{and} := \lambda b_1.\ \lambda b_2.\ b_1\ (b_2\ \mathbf{T}\ \mathbf{F})\ \mathbf{F}$;
- $\mathbf{and}' := \lambda b_1.\ \lambda b_2.\ \lambda xy.\ b_1\ (b_2\ x\ y)\ y$.

6 Confluence de la β-réduction.

Définition 4 (Rappel, *c.f.* **Théorie de la Programmation** [Chapitre 10])**.** On dit que $\to$ est *confluente* en $t \in A$ si, dès que $t \to^\star u_1$ et $t \to^\star u_2$ il existe t' tel que $u_1 \to^\star t'$ et $u_2 \to^\star t'$.

$$\begin{tikzcd} & t \arrow[dl, "\star"'] \arrow[dr, "\star"] & \\ u_1 \arrow[dr, dashed, "\star"'] & & u_2 \arrow[dl, dashed, "\star"] \\ & t' & \end{tikzcd} .$$

Les flèches en pointillés représentent l'existence.

On dit que $\to$ est *confluente* si $\to$ est confluente en tout $a \in A$.

La propriété du diamant correspond au diagramme ci-dessous :

$$\begin{tikzcd} & t \arrow[dl] \arrow[dr] & \\ u_1 \arrow[dr, dashed] & & u_2 \arrow[dl, dashed] \\ & t' & \end{tikzcd} ,$$

c'est-à-dire si $t \to u_1$ et $t \to u_2$ alors il existe t' tel que $u_1 \to t'$ et $u_2 \to t'$.

La confluence pour $\to$, c'est la propriété du diamant pour $\to^\star$. On sait déjà que la β-réduction n'a pas la propriété du diamant (certains chemins de l'exécution sont plus longs), mais on va montrer qu'elle est confluente.

Définition 5. On définit la relation de *réduction parallèle*, notée $\rightrightarrows$, par les règles d'inférences suivantes :

$$\frac{}{x \rightrightarrows x} \qquad \frac{M \rightrightarrows M'}{\lambda x.\, M \rightrightarrows \lambda x.\, M'}$$

$$\frac{M \rightrightarrows M' \qquad N \rightrightarrows N'}{M\ N \rightrightarrows M'\ N'} \qquad \frac{M \rightrightarrows M' \qquad N \rightrightarrows N'}{(\lambda x.\, M)\ N \rightrightarrows M'[{}^{N'}/_{x}]}$$

Lemme 4. La relation $\rightrightarrows$ est réflexive.

Lemme 5. Si $\mathscr{R} \subseteq \mathscr{S}$ alors $\mathscr{R}^\star \subseteq \mathscr{S}^\star$. De plus, $(\mathscr{R}^\star)^\star = \mathscr{R}^\star$.

Lemme 6. Les relations $\to^\star$ et $\rightrightarrows^\star$ coïncident.

Preuve.

- ▷ On a $\to^\star \subseteq \rightrightarrows^\star$ car cela découle de $\to \subseteq \rightrightarrows$ par induction sur $\to$ en utilisant la réflexivité de $\rightrightarrows$.
- ▷ On a $\rightrightarrows^\star \subseteq \to^\star$ car cela découle de $\rightrightarrows \subseteq \to^\star$. En effet, on montre que pour tout M, M' si $M \rightrightarrows M'$ alors $M \to^\star M'$, par induction sur $\rightrightarrows$. Il y a *4* cas.
  - Pour $x \rightrightarrows x$, c'est immédiat.
  - Pour l'abstraction, on suppose $M \rightrightarrows M'$ alors par induction $M \to^\star M'$, et donc $\lambda x.\, M \to^\star \lambda x.\, M'$ par induction sur $M \to^\star M'$.
  - Pour l'application, c'est plus simple que pour la précédente.
  - Pour la substitution, supposons $M \rightrightarrows M'$ et $N \rightrightarrows N$. On déduit par hypothèse d'induction $M \to^\star M'$ et $N \to^\star N'$. Et, par induction sur $M \to^\star M'$, on peut montrer que $(\lambda x.\, M)\ N N \to^\star (\lambda x.\, M')\ N$. Puis, par induction sur $N \to^\star N'$, on montre $(\lambda x.\, M')\ N \to^\star (\lambda x.\, M')\ N'$. Enfin, par la règle de β-réduction, on a $(\lambda x.\, M')N' \to M'[{}^{N'}/_{x}]$. On rassemble tout pour

obtenir :
$$(\lambda x.\, M)\, N \to^\star M'[N'/x].$$
□

On est donc ramené à montrer que $\rightrightarrows^\star$ a la propriété du diamant. Or $\rightrightarrows$ a la propriété du diamant, ce que l'on va montrer en TD.

Lemme 7. Si $M \rightrightarrows M'$ alors $N \rightrightarrows N'$ implique $M[N/x] \rightrightarrows M'[N'/x]$.

Preuve. Par induction sur $M \rightrightarrows M'$, il y a *4* cas. On ne traite que le cas de la 4ème règle. On suppose donc $M = (\lambda y.\, P)\, Q$ avec $y \notin \mathscr{V}\ell(N)$ et $y \neq x$. On suppose aussi $P \rightrightarrows P'$, $Q \rightrightarrows Q'$ et $M' = P'[Q'/y]$. On suppose de plus $N \rightrightarrows N'$. Par hypothèse d'induction, on a $P[N/x] \rightrightarrows P'[N'/x]$ et $Q[N/x] \rightrightarrows Q'[N'/x]$. On applique la 4ème règle d'inférence définissant $\rightrightarrows$ pour déduire

$$\underbrace{(\lambda y.\, (P[N/x]))}_{\substack{\| \\ (\lambda y.\, P)[N/x] \\ \text{car } x \neq y}}(Q[N/x]) \rightrightarrows (P'[N'/x])[Q'[N'/x]/y] = (P'[Q'/y])[N'/x]$$

par le lemme de gymnastique des substitutions et car $y \notin \mathscr{V}\ell(N') \subseteq \mathscr{V}\ell(N)$ et car $N \to^\star N'$. □

Proposition 1. La relation $\rightrightarrows$ a la propriété du diamant.

Preuve. Vu en TD. □

Corollaire 1. On a la confluence de $\to_\beta$.

Définition 6. La *β-équivalence*, ou *β-convertibilité* est la plus petite relation d'équivalence contenant $\to_\beta$. On la note $=_\beta$.

Si l'on a

$$\begin{tikzcd}
 & {} \arrow[dl, "\star"] \arrow[dr, "\star"'] & & {} \arrow[dl, "\star"] \arrow[dr, "\star"'] & & N \arrow[dl, "\star"] \\
M & & {} & & {} &
\end{tikzcd},$$

alors $M =_\beta N$.

Proposition 2. Tout λ-terme est β-équivalent à au plus une forme normale.

Preuve. Si $M =_\beta N$ et M, N sont des formes normales, alors par confluence il existe P tel que $M \to^\star P$ et $N \to^\star P$. On a donc que $M = N = P$. □

Remarque 4 (Conséquences).
- ▷ Deux normales distinctes (au sens de $=_\alpha$) ne sont pas β-convertibles.
- ▷ Si on a un λ-terme qui diverge et qui a une forme normale, par exemple $(\lambda x.\, y)\, \Omega$, alors on peut toujours « revenir » sur la forme normale.