TD2 19 février 2013

λ-Calcul et Logique Informatique

Jean Goubault David Baelde
{goubault,baelde}@lsv.ens-cachan.fr

Exercice 1 — Réduction parallèle
Fin du TD précédent : montrer que la réduction parallèle conflue fortement, conclure que la β-réduction conflue.

Exercice 2 — Encodages
La dernière fois, nous avons vu comment encoder les booléens :

$$[\top] = \lambda x.\lambda y.\ x \qquad [\bot] = \lambda x.\lambda y.\ y$$

L'idée derrière cet encodage particulier est que $[b]\ x\ y$ est **if** b **then** x **else** y. Plus généralement, notre slogan doit être le suivant : *puisqu'on n'a pas de types de données, représentons une donnée par les opérations qu'on peut faire dessus.*

1. Définir un codage des paires, c'est à dire des termes **pair** et π_i ($i \in \{1, 2\}$) tels que pour tout x_1 et x_2 on a :

   $$\pi_i\ (\mathbf{pair}\ x_1\ x_2) \to^* x_i$$

   A-t-on **pair** $(\pi_1\ M)\ (\pi_2\ M)$ pour tout M ? comment interpréter cela ?
2. Pour les entiers, on utilise l'encodage de Church :

   $$\overline{n} = \lambda s.\lambda x.\ s^n\ x = \lambda s.\lambda x.\ s\ (s\ (\dots\ x))$$

   Coder zéro, le successeur, l'addition, la multiplication, le test à zéro pour cet encodage.
3. Définir le prédecesseur pour les entiers de Church, avec **pred** $\overline{0} = \overline{0}$.
   Bonus : Peut-on définir toutes les fonctions récursives primitives sur les entiers de Church ?
4. Donner un codage des listes, et définir la liste vide, l'ajout d'un élement en tête d'une liste, la concaténation et le renversement.

Exercice 3 — Stratégie interne faible
La stratégie interne faible, aussi appelée *appel par valeur*, est aussi proche que possible des langages de programmation traditionnels. Elle ne réduit jamais sous les abstractions, et ne réduit un β-redex que lorsque son argument est déja complètement réduit — on dit alors que c'est une *valeur*.

On définit l'ensemble des valeurs par la grammaire suivante, où $\mathcal{V}$ est l'ensemble des variables et Λ l'ensemble de tous les termes :

$$V := \mathcal{V}V \dots V \mid \lambda x.\Lambda$$

On définit formellement notre stratégie comme suit :

$$\frac{v \in V}{(\lambda x.u)v \rhd u[x := v]} \qquad \frac{u \rhd u'}{uv \rhd u'v} \qquad \frac{u \in V \quad v \rhd v'}{uv \rhd uv'}$$

1. Utiliser le combinateur Y pour définir la fonction factorielle, de sorte que **fact** $\overline{n} \to_\beta^* \overline{n!}$. On rappelle la définition :

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

2. Quelles sont les réductions de **fact** (sans argument) par $\rhd^*$? Cela pose-t-il un problème ? Si oui, proposer une solution.
3. Quelles sont les réductions de **fact** $\overline{0}$ par $\rhd^*$? Cela pose-t-il un problème ? Si oui, proposer une solution.

Exercice 4 — Chemins sur Λ^*

On définit l'ensemble des chemins d'un terme $u \in \Lambda$, noté $\mathcal{C}(u)$, comme l'ensemble des objets p tels que $p \prec u$ est dérivable dans le système suivant :

$$\frac{}{x \prec x} \qquad \frac{P \prec M}{\mathbf{L}\ P \prec M\ N} \qquad \frac{P \prec N}{\mathbf{R}\ P \prec M\ N} \qquad \frac{P \prec M}{\lambda x.P \prec \lambda x.M}$$

On travaille modulo α, en renommant implicitement les variables liées pour pouvoir appliquer la règle de passage sous l'abstraction. Par conséquent, deux termes α-équivalents ont les mêmes chemins, et l'ensemble des chemins d'un terme est clos par α-équivalence.

1. Donner les chemins des termes $\lambda x.\ x\ y$, $\lambda x.\ x\ x$, et $\lambda x.\ (\lambda y.\ y)\ x$.
2. Que peut-on dire de deux termes qui ont les mêmes chemins ?
3. On étend notre définition de chemins à Λ^* en rajoutant la règle suivante :

$$\frac{P \prec M \quad Q \prec N}{P[x := Q] \prec (\lambda^* x.\ M)\ N}$$

   L'intuition graphique est qu'un chemin du redex est soit un chemin de M ne contenant pas x, soit un chemin de M se terminant sur x au bout duquel on a recollé un chemin de N.

   Montrer que $M \to N$ implique $\mathcal{C}(M) = \mathcal{C}(N)$, où $\to$ est la réduction de Λ^* qui ne touche qu'aux rédexes annotés d'une étoile.
4. Montrer la réciproque : $\mathcal{C}(M) = \mathcal{C}(N)$ implique $M \leftrightarrow^* N$, c'est à dire la convertibilité dans Λ^*.