

λ -calcul et logique informatique

`guillaume.scerri@lmf.cnrs.fr`

Exercice 1 — Encodages de listes

Définir (et justifier le cas échéant)

1. un codage des listes. Quel est la liste vide ?
2. l'ajout d'un élément en tête d'une liste.
3. le test du vide.
4. le calcul du premier élément d'une liste.
5. la concaténation.
6. la longueur d'une liste, le renversement.

Exercice 2 — 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 :

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

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 :

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

Montrer que $\mathcal{C}(A[x := B]) = \mathcal{C}(A)[x := \mathcal{C}(B)]$ pour tous termes A et B dans Λ^* .

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

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 déjà complètement réduit — on dit alors que c'est une *valeur*.

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

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

On définit formellement notre stratégie comme suit :

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

1. Utiliser le combinateur Y pour définir la fonction factorielle, de sorte que **fact** $\bar{n} \rightarrow_{\beta}^* \overline{n!}$. 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 $\triangleright^*$? Cela pose-t-il problème ? Si oui, proposer une solution.

3. Quelles sont les réductions de **fact** $\bar{0}$ par $\triangleright^*$? Cela pose-t-il problème?
Si oui, proposer une solution.