TAS TD 2 - Lambda-Calcul Simplement Typé

1 Types Simples

Typer les termes suivants, dans l'environnement vide :

1. I
2. $I\ I$
3. K
4. S
5. $S\ K\ K$
6. Ω
7. $K\ I\ \Omega$

Donner un type pour les entiers.

2 Listes

On peut représenter la liste $[e_1, e_2, \ldots, e_k]$ par le terme $\lambda cn.c\, e_1\ (c\, e_2\ \ldots\ (c\, e_k\ n) \ldots)$

1. Donner deux termes réalisant la liste vide et la construction d'une liste (à partir d'une liste et d'un élément).
2. Donner un λ-terme réalisant la concaténation de deux listes.
3. Donner un λ-terme réalisant l'extraction de la tête.
4. Donner un λ-terme réalisant un *map* sur une liste.
5. Donner un λ-terme réalisant un *reduce/fold* sur une liste.
6. Donner un λ-terme calculant la longueur d'une liste.
7. Donner un λ-terme réalisant l'extraction de la queue.
8. Donner un λ-terme réalisant un *filter* sur une liste.
9. Expliquer comment représenter un arbre binaire en λ-calcul.
10. Expliciter un type pour les listes.

3 Indices de De Bruijn

Pour de nombreuses raisons, il peut être pratique de se débarrasser de l'α-conversion en ayant une représentation unique des λ-termes. La syntaxe du λ-calcul en indice de De Bruijn est définie par la grammaire :

$$M ::= n \mid MM \mid \lambda.M$$

où n est un entier naturel. Intuitivement n représente la variable liée par le $(n+1)$-ème λ obtenu en remontant le terme. Ainsi $\lambda\lambda.(10)(\lambda.20)$ désigne le λ-terme $\lambda xy.(xy)(\lambda z.xz)$.

1. Ecrire les règles décrivant la sémantique opérationnelle à petits pas du λ-calcul en indice de De Bruijn.
2. Ecrire la multiplication en indice de De Bruijn.
3. Donner la traduction du λ-calcul dans celui en indice de De Bruijn, et la traduction inverse.

4 Stratégies

Une stratégie de réduction pour le λ-calcul est une relation $\rightarrow_s \subseteq \rightarrow$ qui est déterministe.

1. Que signifie "$\rightarrow_s$ est déterministe" ?

4.1 Call-by-value

On définit le λ-calcul en appel par valeur en définissant (x est une variable) :

— les termes : $M ::= x \mid \lambda x.M \mid MN$

— les valeurs : $V ::= x \mid \lambda x.M$

— la sémantique opérationnelle à petits pas :

$$\frac{}{(\lambda x.M)V \rightarrow_v M[V/x]}\ \beta \qquad \frac{M \rightarrow_v M'}{MN \rightarrow_v M'N}\ \mu_l \qquad \frac{N \rightarrow_v N'}{VN \rightarrow_v VN'}\ \mu_r$$

1. Réduisez les termes $\delta(I\delta I)$ et $I(F\delta I)$ avec la relation $\rightarrow_v$.
2. Prouvez que $\rightarrow_v$ est déterministe.
3. Cette présentation est appelée "left-to-right", quelle est la présentation "right-to-left" ?

4.2 Call-by-name

La sémantique du λ-calcul en appel par nom est :

$$\frac{}{(\lambda x.M)N \rightarrow_n M[N/x]}\ \beta \qquad \frac{M \rightarrow_n M'}{MN \rightarrow_n M'N}\ \mu_l$$

1. Répondez aux questions prècèdentes pour l'appel par nom.

5 Combinateurs de point fixe

Un λ-terme M est un *point-fixe* d'un λ-terme F si $M =_\beta FM$. Un terme C est un *combinateur de point fixe* si pour tout terme F, CF est un point fixe de F.

On définit les termes suivants :

$$p = \lambda fx.f(xx) \qquad \mathbf{Y} = \lambda f.pf(pf) \qquad q = \lambda xy.y(xxy) \qquad \mathbf{Theta} = qq$$

1. Montrer que $\mathbf{Y}$ et $\mathbf{Theta}$ sont des combinateurs de point fixe.