Université Paris 6
Master d'Informatique

Analyse De Programmes et Sémantique
M. Jaume &R. Rioboo
Année 2006-2007

Travaux Dirigés lambda Calcul

Étant donné un ensemble infini dénombrable de « variables », les λ-termes Λ_V ou λ-expressions sur V sont définis inductivement par

- les termes $v \in V$ sont dans Λ_V,
- si t est dans Λ_V et x est dans V alors $\lambda x.t$ est dans Λ_V. Un terme de cette forme est l'abstraction de t par la variable x.
- si t_1 et t_2 sont dans Λ_V alors $(t_1\ t_2)$ est dans Λ_V. Un terme de cette forme est l'application de t_1 sur t_2.

On peut représenter les λ-termes par

```
type var = string

type lambda =
  | Occ of var
  | App of lambda*lambda
  | Abs of int*lambda
```

Exercice 1 (Entiers de Church)
On rappelle que les enytiers de Church sont donnés par la relation

$$\begin{cases} \underline{0} & = & \lambda s\,x.x \\ \underline{n+1} & = & \lambda s\,x.s(\underline{n}\,s\,x) \end{cases}$$

Question 1.1
Donner le code caml pour les fonctions `to_church` et `from_church` permettant respectivement de passer d'un `int` (supposé positif) à un `lambda` et inversement.

Question 1.2
Donner des termes pour le test à 0, les fonction successeur, somme, produit et exponentielle.

Question 1.3
On considère le terme

$$[\text{pred}] = \lambda\,n\,f\,x.(\lambda\,g\,h.h\,(g\,f))\,(\lambda\,u.x)\,(\lambda\,u.u)$$

Montrer que $[\text{pred}]\,\underline{n+1}\,s\,z$ se réduit en $\underline{n}\,(\lambda\,g\,h.h\,(g\,s))\,(\lambda\,u.z)\,s$. Montrer que $\underline{0}\,(\lambda\,g\,h.h\,(g\,s))\,(\lambda\,u.z)\,s$ se réduit en z. En déduire par recurrence que $\underline{n}\,(\lambda\,g\,h.h\,(g\,s))\,(\lambda\,u.z)s = \underline{n}\,s\,z$ et donc que [pred] défini bien la fonction prédécesseur.

On peut aussi représenter la fonction prédécesseur par le terme

$$\lambda\,n.n.(\lambda\,g\,g.(g\,\underline{1})(\lambda\,u.\boxed{+}\,(g\,k)\,\underline{1})\,k)\,(\lambda\,v.\underline{0})\,\underline{0}$$

Question 1.4
En utilisant la conditionelle, les opérations sur les entiers de Church et un opérateur de point fixe, donner la définition de la fonction factorielle.

Exercice 2 (Réduction en λ-calcul)
On souhaite pouvoir examiner différentes approches de simplification.

Question 2.1
Étant donnés des terme t_1 et t_2 ainsi qu'une variable x, écrire la fonction `subtitute_free_occurrences` qui retourne le résultat de la substitution des occurrences libres de x dans t_1 par t_2 (notée $t_1\{t_2/x\}$)

Un redex est un λ-terme de la forme $((\lambda x.t_1)t_2)$ qui se β-réduit en $t_1\{t_2/x\}$. Un terme est dit en forme normale s'il n'a plus de redex.

Comme dans le TME nous désignerons les sous termes d'un terme par un chemin depuis la racine dans l'arbre de l'expression. Les actions possibles sont

```
type lambda_kind =
  | InFun
  | Inapp
  | Inarg
```

et un chemin est simplement une liste d'actions.

Question 2.2
Écrire la fonction qui étant donné un terme t retourne la liste des chemins de t qui sont des redex.

Question 2.3
Écrire la fonction qui étant donnés deux termes t et t' ainsi qu'un chemin c dans t remplace le sous terme désigné par c dans t par t'. En déduire une fonction qui étant donné un terme t et un chemin c désignat un redex de t opère une étape de β-réduction.

Question 2.4
Nous appellerons fonction de choix une fonction qui choisit un chemin parmi une liste de chemins. Érire la fonction qui étant donné un terme t, une fonction de choix f retourne le résultat de la réduction de t. On pourra ajouter un paramètre entier désignant un nombre maximum d'étapes pour déclencher une exception et arrêter une récursion infinie.

On veut maintenant manipuler les chemins dans un terme.

Question 2.5
Combien de chemins différents peut-il y a voir à une profondeur donnée ? Écrire les fonctions qui pour une liste non vide de chemins retournent

1. le chemin le le plus à gauche parmi les chemins les moins profonds,
2. le chemin le le plus à gauche parmi les chemins les plus profonds,
3. le chemin le le plus à droite parmi les chemins les moins profonds,
4. le chemin le le plus à droite parmi les chemins les plus profonds.

Donner des exemples de termes pour lesquels les quatre stratégies n'ont pas le même comportement.

Question 2.6
Quel est l'inconvénient de cette implémentation à l'aide d'une fonction de choix ? Écrire directement la méthode de réduction qui implante la stratégie leftmost outermost.