Expression Logique et Fonctionnelle . . . Évidemment

TP n° 5 : λ-calcul. Réduction. Pouvoir d'expression.

Objectifs du TP
- Beta-réduction des λ-termes, calcul de forme normale.
- Représentations des booléens, des couples, des entiers par des λ-termes.
- Programmation de fonctions récursives.

1 Matériel fourni

Le fichier `lambda_calcul.ml` contient

1. la déclaration du type `lambda_terme` ;
2. la déclaration de quelques combinateurs ;
3. un imprimeur de λ-termes à utiliser dans l'interpréteur ;
4. deux fonctions donnant les variables libres et liées d'un λ-terme ;
5. et une fonction de substitution.

Bref, ce module contient les déclarations que vous aviez à réaliser dans le TP précédent.
Voir ici pour la documentation complète de ce module.

Question 1 Récupérez ce fichier.
Question 2 Lisez la documentation.
Question 3 Dans un shell, compilez ce module avec la commande

```
> ocamlc lambda_calcul.ml
```

Vous devez obtenir deux fichiers `lambda_calcul.cmi` et `lambda_calcul.cmo`.
Pour utiliser ce module compilé avec un interpréteur, lancez-le avec la commande

```
> ocaml lambda_calcul.cmo
```

(on suppose dans cette commande que le fichier `lambda_calcul.cmo` est situé dans le répertoire courant.)
Puis dans l'interpréteur tapez la directive

```
# open Lambda_calcul ;;
```

qui permet de ne pas utiliser les noms pleinement qualifiés des déclarations que le module contient (i.e. on peut utiliser par exemple la fonction `substitue` en la nommant simplement `substitue` au lieu de `Lambda_calcul.substitue`).
Question 4 Effectuez quelques tests des fonctions `var_libres`, `var_liees` et `substitue`.
Question 5 Appliquez la directive

```
# #install_printer imprimer_lambdaterme ;;
```

et observez son effet sur les valeurs de quelques λ-termes imprimées par l'interpréteur.

2 Beta-réduction, Formes normales

Question 6 Récupérez le fichier `reduction.ml`.
Ce fichier contient
- la déclaration d'une exception `Normal` ;
- la déclaration d'une fonction `reduit_redex_gauche` ;
- et la déclaration (inachevée) d'une fonction `normal`.

Il nécessite l'utilisation du module `Lambda_calcul`.
Question 7 Chargez ce fichier dans un interpréteur dans lequel le module `Lambda_calcul` est défini.

```
# #use "reduction.ml" ;;
```

Question 8 Réduisez le redex le plus à gauche de quelques λ-termes

1. les termes t_i du module `Lambda_calcul` ;

2. des termes de la forme $t_i\ t_j$.

Question 9 Complétez la fonction `normal` pour qu'elle calcule la forme normale d'un λ-terme si celle-ci existe.

Question 10 À l'aide de cette fonction, calculez les formes normales

1. des termes t_i, $i = 1..8$ du module `Lambda_calcul` ;
2.
3. de $\mathbf{K\ I\ \Omega}$;
4. de $\mathbf{K}_*\ \mathbf{\Omega\ I}$.

3 Représentations de données par des λ-termes

Il s'agit dans cette partie de représenter
- les booléens,
- les couples,
- les entiers

par des λ-termes, et de définir quelques combinateurs représentant les opérateurs logiques ou arithmétiques les accompagnant.

Question 11 Récupérez le fichier `lambda_expression.ml`.

Ce fichier contient quelques déclarations complètes et d'autres inachevées. Pour être utilisé dans un interpréteur, il nécessite le module `Lambda_calcul`.

Dans ce fichier, mis à part `int_to_church` et `church_to_int`, aucune déclaration n'est une fonction au sens du langage CAML.

Question 12 Progressivement complétez les déclarations inachevées, et testez la validité de vos réalisations.

4 Fonctions récursives

Question 13 Représentez la fonction factorielle par un λ-terme.

Jusqu'à quel entier pouvez vous calculer la factorielle ?