

Devoir de Programmation Fonctionnelle

I. Présentation du sujet

On demande de programmer en `Ocaml` deux évaluateurs de lambda-termes :

- `eval_nom`, qui implémente la stratégie correspondant à l'appel par nom (dans le cas de deux redex emboîtés, on réduit le plus extérieur)
- `eval_val`, qui implémente la stratégie d'appel par valeur (dans le cas de deux redex emboîtés, on réduit le plus intérieur)

Concrètement, lors de l'appel à l'un de ces deux évaluateurs sur un lambda-terme fourni par l'utilisateur, l'évaluateur en question applique la stratégie qui lui est propre jusqu'à l'obtention de la forme normale, si elle existe.

Pour représenter les lambda-termes, on devra utiliser le type `terme` suivant :

```
type terme = X of int | L of int * terme | A of terme * terme
```

On pourra penser à définir (et utiliser) des fonctions permettant de :

- déterminer les variables libres d'un lambda-terme
- déterminer les variables liantes d'un lambda-terme
- renommer les variables d'un lambda-terme
- substituer une variable par un lambda-terme dans un lambda-terme

II. Document attendu

Le devoir se fera en binôme. Il donnera lieu à un rapport écrit qui sera rendu, sous forme de document papier, le 25 mai à 14h.

Ce rapport comprendra :

- une présentation du problème et de sa solution algorithmique
- une discussion des choix d'implémentation et des problèmes rencontrés
- un listing commenté du code comprenant les types et fonctions utilisés
- des jeux d'essais nombreux et pertinents