Devoir de Programmation Fonctionnelle

Amine Kheddaoui Gowshigan Selladurai

Avril 2022

Contents

1 Présentation du problème et de sa solution algorithmique

Notre problème ici est d'implémenter un solveur d'équation booléennes dans le langage Ocaml. La première étape consistait donc à définir le type des expressions booléennes que nous allions utiliser. Les éléments booléennes sont définies comme suit :

```
type eb = V of int
   | Vrai
   | Faux
   | AND of eb * eb
   | OR of eb * eb
   |XOR of eb * eb
   | NAND of eb * eb
   | NOT of eb;;
```

Nous avons donc du définir les différentes expressions booléennes NOT, AND, OR, XOR et NAND :

```
Mettre les implémentations ici
```

1.1 Problèmes rencontrés

2 Discussion des choix d'implémentation

2.1 Implémentation des opérateurs booléens

```
ICI
```

2.2 Implémentation des environnements

Tout d'abord, nous avons une fonction append, qui ici nous sert à ajouter un même élément à chaque élément de la liste.

```
let append x ll = List.map ( fun l -> x::l) ll;;
```

Nous avons ensuite une fonction qui génère l'environnement, en se servant de la fonction précédente.

```
let rec generateur_aux l =
 match l with
 |[] -> [[]]
 | hd::tl ->let l_aux = generateur_aux tl in
 (append (a,FALSE) l_aux)@(append (a,TRUE) l_aux);;
```

2.3 Fonctions d'évaluation

2.4 Test des fonctions auxiliaires

2.5 Test des fonctions d'évaluation