Rapport de Projet - Programmation Fonctionnelle

Hugo Mohamed et Thomas Grant

May 2, 2017

Ce rapport présente la Partie A de notre projet, les problèmes rencontrés et la manière dont nous les avons abordés et résolus.

Contents

1 Tri par sélection du minimum

1.1 Fonction selectionne

La fonction `selectionne` prend un comparateur et une liste en paramètre, et retourne un élément. Par exemple avec le comparateur `(<)`, la fonction renverra le plus petit élément de la liste. Pour cela, nous avons utilisé `match with` qui nous permet de parcourir la liste afin de déterminé quel élément `x` vérifie `(comparateur) x y` avec `y` étant n'importe quel autre élément de la liste.

1.2 Fonction supprime

La fonction `supprime` prend un élément et une liste en paramètre, et retourne la liste sans la première occurrence de l'élément dans la liste. Pour cela, nous parcourons aussi la liste éléments par éléments afin de voir si l'un d'entre eux est égale à l'élément passé en paramètre, si c'est le cas, l'élément est supprimé de la liste, le programme s'arrête et renvoie la nouvelle liste.

1.3 Fonction tri selection min

La fonction `tri_selection_min` prend un comparateur et une liste en paramètre, et retourne la liste triée par ordre croissant en fonction du comparateur. Pour cela nous avons utilisé les fonctions `selectionne` et `supprime` qui nous ont permis de selectionner le plus petit élément de la liste (avec `selectionne`) et de l'ajouter au début de la liste sans cet élément qui aura été supprimé de sa position initiale (avec `supprime`).
Au final on obtient notre liste initiale bien triée.

1.4 Test

Pour tester notre fonction `tri_selection_min`, nous avons utilisé le module Hasard transmit par nos professeurs. Dans ce module, nous trouvons la fonction `random_list` qui prend deux nombres en paramètre, et retourne une liste d'entiers: le premier paramètre définissant les valeurs que peuvent prendre les entiers de la liste; le second, le nombre d'éléments de la liste. Cette fonction nous permet donc de tester notre fonction `tri_selection_min` sur différentes tailles de listes. Par exemple on peut rentrer:

```
# tri_selection_min (<) (random_list 500 200) ;;
```

Ce qui nous renverra la liste triée du plus petit au plus grand entier allant de 0 à 500 et contenant 200 entiers.

2 Tri par partition-fusion

2.1 Fonction partitionne

La fonction `partitionne` prend une liste en paramètre et retourne deux listes créer à partir de la liste donnée. En fait, les deux nouvelles listes sont crées de la manière suivante: le premier élément de la liste donnée va dans la liste 1, le suivant dans la liste 2, et ainsi de suite en alternant à chaque fois.
Pour cela, nous avons rencontré quelques problèmes:

2.2 Fonction fusionne

2.3 Fonction tri partition fusion

2.4 Test

3 Choix d'une fonction de tri

3.1 Création d'un module Test efficacite

3.2 Test

4 Module List sup définitif

4.1 Les Fonctions retenues