Rapport de projet

Implémenter (en C) et prouver (avec Frama-C) les algorithmes décrits dans les sections 3 et 4 de l'article donné en référence

Prudence Deteix, Paul Passeron, Jean Rousseau

Table des matières

Dépot du projet : `https://github.com/Paul-Passeron/PRRD.git`

1 Introduction

L'article "When Separation Arithmetic is Enough" de Filliâtre, Paskevich et Danvy présente une méthode pour la vérification formelle de programmes impératifs manipulant des structures de données récursives basées sur des pointeurs (ex : listes chaînées ou arbres binaires). Les auteurs proposent une solution qui réconcilie deux objectifs souvent contradictoires : maintenir des spécifications claires et compréhensibles tout en permettant une vérification largement automatisée par les solveurs SMT.

L'idée centrale de l'article consiste à projeter les structures récursives sur des séquences plates indexées par des entiers, transformant ainsi les propriétés de séparation (est-ce que plusieurs objets pointent vers le même endroit en mémoire) en contraintes d'arithmétique linéaires.

2 Implémentation en C

2.1 Partie 3

On commence par remplir le fichir `lst.h` dans lequel on inclut les bibliothèques suivantes :

```
#include "common.h"
#include <stddef.h>
#include <stdbool.h>
#include <stdio.h>
```

Puis, on va ..... :

```
lst_t aux(lst_t r, lst_t l) {
    if (!l) {
        return r;
    }
    else {
        lst_t n = l->cdr;
        l->cdr = r;
        return aux(l, n);
    }
}
```

```
lst_t list_reversal(lst_t l) {
    return aux(NULL, l);
}
```

2.2 Partie 4

3 Preuve avec Frama-C

3.1 Partie 3

3.2 Partie 4

4 Problèmes

5 Pistes d'amélioration

6 Conclusion