Guião para aula laboratorial de Verificação Formal (2018/19)

Frama-C, plugin WP (1)

Esta aula é dedicada à familiarização com o sistema Frama-C, em particular à utilização do seu plugin WP de verificação dedutiva de programas.

Deverá ter instalado o Frama-C, o plugin WP, assim como alguns SMT solvers.

O website do Frama-C disponibiliza toda a documentação, assim como alguns tutoriais.

1 Introdução

A ubiquidade da linguagem C é devida a razões históricas e ao facto de C estar bem adaptado para um número significativo de aplicações (por exemplo, código embebido). No entanto, a linguagem C é demasiado premisiva e permite muitos acções notoriamente perigosas, que podem colocar em risco a correcção e a segurança dos programas.

É, por isso, muito importante ter ferramentas que permitem provar formalmente propriedades de software crítico escrito em C.

O Frama-C é uma plataforma contituída por um conjunto de ferramentas dedicadas à análise do código C. Esta plataforma reúne várias técnicas de análise estática numa única estrutura colaborativa. A abordagem colaborativa do Frama-C permite que os analisadores estáticos se baseiem nos resultados já calculados por outros analisadores da plataforma.

O Frama-C tem uma arquitetura de plugins. Um kernel comum centraliza informações e conduz a análise. Os plugins interagem entre si por meio de interfaces definidas pelo kernel. Isso contribui para a robustez no desenvolvimento do Frama-C, permitindo um amplo espectro de funcionalidades.

O pluing WP para verificação dedutiva de programas, que vamos estudar, é baseado na *lógica de Hoare* e num *weakest precondition calculus*, parametrizado por um modelo de memória para representar apontadores e valores na heap.

O WP permite provar que as funções C satisfazem o seu contracto expresso na linguagem de especiifcação ACSL (ANSI C Specification Language). Essas provas são modulares: as especificações das funções que são evocados são usadas para estabelecer a prova sem olhar para o código. As especificações podem ser parciais, concentrando-se num aspecto do programa analisado de cada vez.

O Frama-C pode correr a partir da linha de comando, em modo bash, ou através de um interface gráfico. Nestas aulas vamos utilizar o modo gráfico. Para o invocar faça:

```
$ frama-c-gui &
```

Pode ser necessário solicitar a detecção de solvers recém-instalados. Para isso, use o botão "Provers" e, em seguida, "Detect Provers" na janela que deve aparecer.

2 Exemplos para explorar

Os slides sobre *Verificação Dedutiva em Frama-C*, apresentados na aula, contêm uma serie de exemplos e vários desafios para resolver. Esses exemplos são acompanhados de ficheiros C disponíveis no wiki da disciplina.

Siga as instruções indicadas nos slides, analise estes ficheiros com o plugin WP, e acrescente/altere as anotações ACSL de forma a completar as provas.

3 Exercícios

1. Escreva um contrato, e prove a correcção face a esse contrato, da seguinte função:

```
void change(int *a, int *b) {
  int tmp = *a;
  *a = *b;
  *b = tmp;
}
```

2. Para cada uma das funções abaixo indicadas:

   - Escreva um contrato ACSL.
   - Codifique a função em C e prove a sua correcção funcional e propriedades de *safety*.
   - Escreva uma função que invoque a função que definiu e teste-a.

   (a) Testa se um array tem valores negativos.

   ```
   int negs(int A[], int N);
   ```

   (b) Devolve o índice onde está o menor elemento do array.

   ```
   int minarray(int A[], int N);
   ```

   (c) Testa se o segmento [a..b] de dois arrays, A e B, de tamanho N, são iguais.

   ```
   int equal_seg(int A[],int B[],int a,int b,int N);
   ```

   (d) Devolve um índice onde está o valor x, caso x exista no array A; senão devolve -1.

   ```
   int where(int A[],int N, int x);
   ```