Université de Strasbourg
Certification du logiciel — Feuille de TP

Sémantique formelle d'un mini-langage

1. Définir en Coq le mini-langage vu en cours :

$$\begin{array}{lcl} e & ::= & n \mid x \mid e + e \\ b & ::= & e < e \\ i & ::= & \text{skip} \mid x := e \mid i; i \mid \text{while } b \text{ do } i \text{ done} \end{array}$$

   (a) Définir le type des expressions arithmétiques.
   (b) Définir l'expression $e_1 \equiv (x + 3) + y$.
   (c) Définir le type des expressions booléennes.
   (d) Définir l'expression $b_1 \equiv (x + 3) > 5$.
   (e) Définir le type des instructions.
2. Étendre la définition du langage afin d'inclure le 'if'.
3. Définir la sémantique axiomatique de ce langage :
   (a) Définir un type pour représenter les assertions. On supposera que les assertions contiennent uniquement des expression booléennes issues du programme, des négations, des conjonctions et des implications.
   (b) Définir une fonction de substitution d'une variable par une expression arithmétique sur les expressions arithmétique.
   (c) Définir une fonction de substitution d'une variable par une expression arithmétique sur les expressions booléennes.
   (d) Définir une fonction de substitution d'une variable par une expression arithmétique sur les assertions.
   (e) Définir une fonction qui, étant donnée une fonction de valuation (`g : string->Z`) qui donne l'interprétation des variables dans Z, permet d'interpréter une expression arithmétique dans Z.
   (f) Définir une fonction qui, étant donnée une fonction de valuation (`g : string->Z`) qui donne l'interprétation des variables dans Z, permet d'interpréter une expression booléenne dans Prop.
   (g) Définir une sémantique pour les assertions, c'est à dire une définition de la validité d'une assertion étant donnée une fonction de valuation (`g : string->Z`) qui donne l'interprétation des variables dans Z.
   (h) Définir un prédicat inductif pour représenter les jugements de la logique de Hoare pour notre langage.
   (i) Montrer que $\{x > 0\}\ x := x + 1\ \{x > 0\}$
   (j) Montrer que $\{x > 0\}\ x := x + 2; x := x - 1\ \{x > 0\}$
   (k) Définir le programme :

```
x:=3; if x < 4 then x:=x+1 else (x:=x+1;x:=x+1)
```

(l) Montrer que dans tout environnement définissant x, l'execution du programme précédent produit un environnement où x vaut 4.

(m) Définir le programme :

```
x:=3; while x < 5 do x:=x+1 done
```

(n) Montrer que dans tout environnement définissant x, l'execution du programme précédent produit un environnement où x vaut 5.

$$\frac{}{\{P\}\ \text{skip}\ \{P\}}\ \text{Skip}$$

$$\frac{\{P\}\ C_1\ \{R\} \qquad \{R\}\ C_2\ \{Q\}}{\{P\}\ C_1; C_2\ \{Q\}}\ \text{Seq}$$

$$\frac{}{\{P[x \leftarrow E]\}\ x := E\ \{P\}}\ \text{Affectation}$$

$$\frac{\{B \wedge P\}\ S\ \{Q\} \qquad \{\neg B \wedge P\}\ T\ \{Q\}}{\{P\}\ \text{if}\ B\ \text{then}\ S\ \text{else}\ T\ \{Q\}}\ \text{Cond}$$

$$\frac{\{P \wedge B\}\ C\ \{P\}}{\{P\}\ \text{while}\ B\ \text{do}\ C\ \text{done}\ \{P \wedge \neg B\}}\ \text{While}$$

$$\frac{P' \Rightarrow P \qquad \{P\}\ S\ \{Q\} \qquad Q \Rightarrow Q'}{\{P'\}\ S\ \{Q'\}}\ \text{Cons}$$