Programmation rigoureuse (NFP209) – Examen

février 2009

2h30 – Documents autorisés

I Sémantique

Exercice 1

(2pts) Construire un arbre de dérivation pour la configurations suivante :

$$\langle \mathtt{z} := 0; \mathtt{while}\ (\mathtt{y} \leq \mathtt{x})\ \mathtt{do}\ (\mathtt{z} := \mathtt{z}+1; \mathtt{x} := \mathtt{x}-\mathtt{y}), \sigma \rangle$$

où $\sigma(x) = 5$ et $\sigma(y) = 2$.

Exercice 2

On souhaite étendre le langage **While** avec la construction :

$$\mathtt{repeat}\ I\ \mathtt{until}\ c$$

dont la sémantique informelle est "repéter I jusqu'à ce que c prenne la valeur vrai". On se donne les règles suivantes pour rendre formellement compte de cette sémantique :

$$\frac{< I, s > \rightarrow s_1 \qquad < b, s_1 > \rightarrow \mathtt{true}}{< \mathtt{repeat}\ I\ \mathtt{until}\ b, s > \rightarrow s_1} \qquad \frac{< I, s > \rightarrow s_1 \qquad < b, s_1 > \rightarrow \mathtt{false} \qquad < \mathtt{repeat}\ I\ \mathtt{until}\ b, s_1 > \rightarrow s_2}{< \mathtt{repeat}\ I\ \mathtt{until}\ b, s > \rightarrow s_2}$$

1. (2pts) Soit s l'état tel que $s(z) = 3$ et $s(x) = 2$. Donner un arbre de dérivation permettant d'évaluer l'instruction `repeat z:=z*x; x:=x-1 until x=0`.
2. (3pts) Montrer l'équivalence : `repeat` I `until` $b \sim I$; `if` b `then skip else repeat` I `until` b

II Logique de Hoare

Exercice 3

Démontrez la correction *totale* des triplets suivants (si vous prouvez la correction partielle, une partie des points vous sera accordée) :

1. (2pts) $\{\}$ `a:=abs(x); if a <> x then y:=x else y:=-x` $\{y \leq 0\}$. En supposant que le triplet suivant est déjà démontré : $\{\}$ `a:=abs(b)` $\{a \geq 0 \wedge (a=-b \vee a=b)\}$.

Solution:

$$\text{HYP}\ \frac{}{\{\}\texttt{a:=abs(x)}\{a \geq 0 \wedge (a=-b \vee a=b)\}} \quad \frac{\overset{\dots}{\{a \geq 0 \wedge (a=-b \vee a=b)\}\texttt{if a <> x then y:=x else y:=-x}\{y \leq 0\}}\ \dots}{\text{SEQ}\ \{\}\texttt{a:=abs(x); if a <> x then y:=x else y:=-x}\{y \leq 0\}}$$

2. (2pts)

$\{y \geq x\}$
```
while x <> y do
  x:=x+2;
  y:=y+1;
done
```
$\{x=y\}$

Solution:

Invariant true. *Variant :* $y-x$.

Exercice 4

On veut maintenant démontrer le triplet (I) suivant :

$\{y \geq x \wedge a=x \wedge b=y\}$
```
while x <> y do
  x:=x+2;
  y:=y+1;
done
```
$\{ x = 2b-a \}$

1. (1pt) Donnez un variant permettant de prouver que cette boucle termine.

Solution:

C'est le même : $y-x$.

On s'intéresse maintenant à la correction partielle du triplet. Il est difficile de la prouver directement. C'est nettement plus facile en ajoutant une *variable fantôme* dans le programme. Une variable fantôme est une variable qui peut prendre sa valeur en fonction des autres variables du programme, mais sa valeur à elle n'est pas utilisée par les autres. On ajoute ici la variable `i` qui va servir à compter le nombre d'itérations de la boucle. On doit donc maintenant prouver ce triplet (II) :

$\{y \geq x \wedge a=x \wedge b=y \wedge i=0\}$
```
while x <> y do
  i:=i+1;
  x:=x+2;
```

```
  y:=y+1;
done
{ x = 2b-a }
```

2. (3pt) Donnez un invariant permettant de relier `i` aux autres variables.

Solution:

Invariant : $x = a+2i \wedge y = b+i$

3. (3pts) Donnez une idée de l'arbre de déduction permettant de prouver (correction partielle) le triplet (II). Détaillez de la racine jusqu'à la règle WHILE. Si vous utilisez la règle CONSEQ, détaillez les déductions logiques en dessous de l'arbre.

Solution:

$$\textsc{While}\ \dfrac{\textsc{Conseq}\ \dfrac{\textsc{Seq}\ \dfrac{\textsc{Conseq}\ \dfrac{\textsc{Aff}\ \dfrac{}{\{x=a+2i\wedge y=b+i\wedge x\neq y\}\texttt{i:=i+1;}\{x=a+2i_0\wedge y=b+i_0\wedge x\neq y\wedge i:=i_0+1\}}}{\{x=a+2i\wedge y=b+i\wedge x\neq y\}\texttt{i:=i+1;}\{x=a+2(i-1)\wedge y=b+i-1\wedge x\neq y\}} \quad \dfrac{\ldots}{\texttt{x:=x+2; y:=y+1;}\{??\}}\ \ldots}{\{x=a+2i\wedge y=b+i\wedge x\neq y\}\texttt{i:=i+1; x:=x+2; y:=y+1;}\{??\}}}{\{x=a+2i\wedge y=b+i\wedge x\neq y\}\texttt{i:=i+1; x:=x+2; y:=y+1;}\{x=a+2i\wedge y=b+i\}}}{\{x=a+2i\wedge y=b+i\wedge x=y\}\texttt{while x <> y do i:=i+1; x:=x+2; y:=y+1; done}\{x=a+2i\wedge y=b+i\wedge x=y\}}$$
$$\textsc{Conseq}\ \dfrac{}{\{y\geq x\wedge a=x\wedge b=y\wedge i=0\}\texttt{while x <> y do i:=i+1; x:=x+2; y:=y+1; done}\{x=2b-a\}}$$

La déduction $(y \geq x \wedge a = x \wedge b = y \wedge i = 0) \rightarrow x = a+2i \wedge y = b+i \wedge x = y$ *est triviale.*
La déduction $(x = a+2i \wedge y = b+i \wedge x = y) \rightarrow x = 2b-a$ *se fait de la façon suivante :*

$$x = 2y - x = 2(b+i) - (a+2i) = 2b+2i-a-2i = 2b-a$$

4. (2pt) Que faudrait-il encore démontrer pour prouver le triplet (I) ? Comment faire ?

Solution:

Il faudrait démontrer que le programme du triplet (II) se comporte comme le programme du triplet (I) vis-à-vis des variables `x`, `y`, `a` *et* `b`.

Pour cela il faut calculer la sémantique de ces deux programmes pour une configuration σ *telle que la précondition de (II) et de (I) sont respectées, et démontrer que les deux configurations finales* σ' *(pour (I)) et* σ'' *(pour (II)) sont telles que* $\sigma'(x) = \sigma''(x)$, $\sigma'(y) = \sigma''(y)$, *etc. Il est probable qu'une induction sur* $x-y$, *c'est-à-dire le nombre d'itérations du* **`while`** *sera nécessaire.*

A Récapitulatif des règles de Hoare

A.1 Correction partielle :

$$\textsc{Aff1}\ \frac{}{\{P[x \leftarrow E]\}\ \mathtt{x:=}E\ \{P\}} \qquad \textsc{Aff}\ \frac{}{\{P\}\quad \mathtt{x:=}E\quad \{P[x \leftarrow x_0] \wedge x{=}E[x \leftarrow x_0]\}}$$

$$\textsc{Cond}\ \frac{\{\{P \wedge B\}\quad I_1\quad \{Q\} \qquad \{P \wedge \neg B\}\quad I_2\quad \{Q\}}{\{P\}\quad \mathtt{if}\ B\ \mathtt{then}\ I_1\ \mathtt{else}\ I_2\quad \{Q\}}$$

$$\textsc{Seq}\ \frac{\{P\}\,\mathtt{C}_1\,\{Q\} \qquad \{Q\}\,\mathtt{C}_2\,\{R\}}{\{P\}\,\mathtt{C}_1\ \mathtt{;}\ \mathtt{C}_2\,\{R\}} \qquad \textsc{While}\ \frac{}{\{P\}\quad \mathtt{while}\ B\ \mathtt{do}\ C\ \mathtt{done}\quad \{P \wedge \neg B\}}$$

$$\textsc{Conseq}\ \frac{P \Rightarrow P' \qquad \{P'\}C\{Q'\} \qquad Q' \Rightarrow Q}{\{P\}C\{Q\}}$$

A.2 Correction totale :

$$\textsc{WhileT}\ \frac{\langle P \wedge B \wedge E = n \wedge E \geq 0\rangle\quad C\quad \langle\ P \wedge E < n \wedge E \geq 0\ \rangle}{\langle P\rangle\quad \mathtt{while}\ B\ \mathtt{do}\ C\ \mathtt{done}\quad \langle P \wedge \neg B\rangle}$$