TD 4 : Logique de Hoare

Matthieu Sozeau (matthieu.sozeau@inria.fr)

October 20, 2014

Ce TD porte sur la preuve de programmes impératifs en utilisant la logique de Hoare. Les notes de cours et corrigés des TDs précédents sont disponibles sur la page du cours:

www.pps.univ-paris-diderot.fr/~sozeau/teaching/MVF-2014.fr.html

On rappelle que les règles d'inférence de la logique de Hoare sont données par :

$$\{P\}\ \texttt{skip}\ \{P\} \qquad \{P[exp/x]\}\ x := exp\ \{P\}$$

$$\frac{\{P\}\ C\ \{Q\} \qquad \{Q\}\ D\ \{R\}}{\{P\}\ C; D\ \{R\}}$$

$$\frac{\{E = \texttt{true} \wedge P\}\ C\ \{Q\} \qquad \{E = \texttt{false} \wedge P\}\ D\ \{Q\}}{\{P\}\ \texttt{if}\ E\ \texttt{then}\ C\ \texttt{else}\ D\ \{Q\}}$$

$$\frac{\{E = \texttt{true} \wedge I\}\ C\ \{I\}}{\{I\}\ \texttt{while}\ E\ \texttt{do}\ C\ \texttt{done}\ \{E = \texttt{false} \wedge I\}}$$

$$\frac{P^{\emptyset} \Rightarrow P \qquad \{P\}\ C\ \{Q\} \qquad Q \Rightarrow Q^{\emptyset}}{\{P^{\emptyset}\}\ C\ \{Q^{\emptyset}\}}$$

(où C, D désignent des commandes, E, V des expressions sans effets de bord).

Triplets

Exercice 1. *Dire pour chacun des triplets suivants s'ils sont dérivables ou non. S'ils sont dérivables, donner les règles d'inférence nécessaires et les raisonnement logiques utiles.*

- $\{x = 2\}\ x := 3\ \{x = 3\}$
- $\{x = 2\}\ x := x + 1\ \{x = 3\}$
- $\{y = 2\}\ x := y\ \{x = 2\}$

- $\{y > 0\}\ x := y; y = \quad 1\ \{x > 0\}$
- $\{\}\ \texttt{if}\quad \texttt{then}\ y\ \texttt{else}\ \ := 2\quad x; x := y\quad 1\ \{x > 0\}$
- $\{y > 0\}\ \texttt{if}\ \ y > 0\ \ \texttt{then}\ x := y\ \ \texttt{else}\ x := \quad y\ \{x > 0\}$
- $\{\}\ \texttt{if}\ \ y > 0\ \ \texttt{then}\ x := y\ \ \texttt{else}\ x := \quad y\ \{x > 0\}$

Exercice 2 (Renversement itératif)**.** *Sans vous aider du cours, retrouvez l'invariant de boucle de la version itérative du renversement d'une liste. Vérifiez la correction de la fonction en donnant et justifiant les triplets de Hoare pour chaque commande.*

$\rho : List[\star]$;

irev $(\ell : List[\star]) =$
  assume$(true)$;
  $\ell^\theta : List[\star]$;
  $\rho := []$; % ρ *is the reverse of the treated prefix of* ℓ
  $\ell^\theta := \ell$; % ℓ^θ *is the non-treated suffix of* ℓ
  while $\ell^\theta \neq []$ do
    invariant?
    $\rho := head(\ell^\theta) \quad \rho$;
    $\ell^\theta := tail(\ell^\theta)$;
  assert$(\rho = Rev(\ell))$

Exercice 3. (Tableaux) *En logique de Hoare, les tableaux sont manipulés à l'aide des deux fonctions suivantes :*

- *La fonction* `access(t, i)` *qui retourne le* i*-ème élément du tableau* t*;*
- *La fonction* `store(t, i, v)` *qui retourne un nouveau tableau ayant les mêmes éléments que* t*, sauf le* i*-ème qui est remplacé par* v*.*

Ces deux fonctions permettent de définir la lecture et l'écriture dans un tableau:[1]

$$t[i] \qquad \texttt{access}(t, i) \qquad \text{et} \qquad t[i] := E \qquad t := \texttt{store}(t, i, E)$$

1. *Quels axiomes est-il raisonnable de supposer sur les fonctions access et store ?*
2. *Montrer la correction du programme suivant (où* x *et* y *sont des variables fraîches) :*

$$\{t[i] = x \wedge t[j] = y\}\ \ v := t[i];\ \ t[i] := t[j];\ \ t[j] := v\ \ \{t[i] = y \wedge t[j] = x\}$$

[1] On notera que l'opération qui consiste à écrire dans une seule case du tableau est traduite en logique de Hoare par le remplacement du tableau complet.

Exercice 4 (Calcul de la racine par addition)**.** *On cherche un programme calculant la racine carré entière d'un entier.*

1. *Donner la spécification du problème.*

2. *Implémenter une solution naïve du problème qui repose sur une méthode par incrémentation.*

3. *Annoter et prouver votre implémentation en utilisant la logique de Hoare.*

4. *On considère l'amélioration du programme où l'on remplace une multiplication coûteuse par une addition.*

```
r := 0; y := 1; z := 1;
while (y<=n) do
  z := z+2; y := y+z; r := r+1;
done;
```

À l'aide des axiomes d'affectation, trouver les expressions E_i et S_i afin que les formules suivantes soient valides :

$$\begin{array}{lll}
\{E_1\} & \texttt{z:=1} & \{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\} \\
\{E_2\} & \texttt{y:=1} & \{y = (r+1)^2 \wedge 1 = 2r+1 \wedge r^2 \quad n\} \\
\{E_3\} & \texttt{r:=0} & \{1 = (r+1)^2 \wedge 1 = 2r+1 \wedge r^2 \quad n\} \\
\{E_4\} & \texttt{r:=r+1} & \{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\} \\
\{E_5\} & \texttt{y:=y+z} & \{y = (r+2)^2 \wedge z = 2r+3 \wedge (r+1)^2 \quad n\} \\
\{E_6\} & \texttt{z:=z+2} & \{y+z = (r+2)^2 \wedge z = 2r+3 \wedge (r+1)^2 \quad n\}
\end{array}$$

$$\begin{array}{llll}
\{y = (r+1)^2 \wedge z = 2r+3 \wedge r^2 \quad n\} & \texttt{y:=y+z} & \{S_1\} \\
\{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\} & \texttt{z:=z+2} & \{S_2\}
\end{array}$$

5. *À l'aide de la règle de la séquence et de l'affaiblissement, démontrer les théorèmes suivants: Init:*

$$\begin{array}{l}
\{n \quad 0\} \\
r := 0;\ y := 1;\ z := 1 \\
\{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\}
\end{array}$$

Loop:

$$\begin{array}{l}
\{y <= n \wedge y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\} \\
z := z+2;\ y := y+z;\ r := r+1 \\
\{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\}
\end{array}$$

6. *En appliquant la règle d'itération, montrer la validité de la formule:*

$$\begin{array}{l} \{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n\} \\ \texttt{while } y \quad n \texttt{ do } z := z+2;\ y := y+z;\ r := r+1 \texttt{ done} \\ \{y = (r+1)^2 \wedge z = 2r+1 \wedge r^2 \quad n \wedge y > n\} \end{array}$$

7. *En vous aidant des questions précédentes, annoter le programme et prouver sa correction.*