Formal Methods
Fall 2017

S01 : Hoare Logic

Professor : Ultes-Nitsche Ulrich
Assistant : Christophe Stammet

Submited by Sylvain Julmy

Exercise 1 : Complete Hoare Triple

1. $\{true\}\ y = 25\ \{y = 25\}$: if y is assigned with 25, then y must be equal to 25.
2. $\{x \leq 6\}\ y = 6\ \{y \geq x\}$: if y is assigned to 6 and then y is greater or equals to x, x must be less or equals to 6 so we would have $x \leq 6 \vee y \geq x \vee y = 6$ evaluated to $true$.
3. $\{x = 2\}\ x = x - 4\ \{x = -2\}$: if the pre-condition is $x = 2$ and the post-condition is $x = -2$, we have to find a sequence of statement that transform 2 to -2.
4. $\{int\ x \wedge int\ y\}\ x = 16; y = 2; while(x > 3)\{x = \frac{x}{y}\}\ \{x' \leq 3\}$: the post-condition is equivalent to the negation of the loop-condition.
5. $\{x = 3\}\ if\ (x \equiv 0 \mod 2)\{y = 2x\}\ else\ \{x = y\}\ \{x' = y\}$: because of the pre-condition P, we have $P \implies x = 3$, so the $true$ branch of the if will never happen so we can reduce the body of S to $x = y$. Finally, we can put the post-condition Q to $x = y$.
6. $\{x = 5 \wedge int\ y\}\ x = x - 2; y = x; x = y - x;\ \{x' = 0\}$: here we assume that the pre-condition $x = 5$ implies that the x variable is an integer too. Due to $y = x$, the last statement $x = y - x$ is equivalent to $x = x - x$, because $y = x$. So we would have $x = x - x = x' = 0$.

Exercise 2 : Weird Hoare Triple

(1)

Using the following transformation $\{P\}S\{Q\} \to P \wedge \Phi_S \implies Q$, we can translate the Hoare Triple :

$$\{int\ x \wedge int\ y\}P\{true\}$$

to

$$int\ x \wedge int\ y \wedge P \implies true$$

Because $Q = true$, we can write any program P so that the Hoare Triple is valid, provided that P terminates.
So P could be, for example,

$$y = 2x; x = 2y; y = y/2;$$

.

(2)

Using the same transformation as before, we can translate the Hoare Triple :

$$\{int\ x \wedge int\ y\}P\{false\}$$

to

$$int\ x \wedge int\ y \wedge P \implies false$$

In order to obtain a **valid** Hoare Triple, we have too demonstrate that the program P we are going to write will never terminates. If P terminates, the post-condition will be evaluated to $false$ and so the Hoare Triple would not be a valid one.
For example, the following Hoare Triple is valid because P will never terminate :

$$\{int\ x \wedge int\ y\}\ while(true)\ \{skip;\}\{false\}$$

Exercise 3 : Formal Proof of Hoare Triple : if clause

$$\begin{aligned}&\{int\ a \wedge int\ b \wedge b > 0\}\\&if(a < 0)\ \{a = 2a\}\\&else\ \{a = b\}\\&\{b \geq a\}\end{aligned}$$

In order to prove the previous Hoare Triple, we have to prove the two following ones due to the If-Then-Else construction.

$$\begin{gathered}\{int\ a \wedge int\ b \wedge b > 0 \wedge a < 0\}\ a = 2a;\ \{b \geq a\}\\=\\int\ a \wedge int\ b \wedge b > 0 \wedge a < 0 \wedge a' = 2a \implies b \geq a'\end{gathered} \tag{1}$$

and

$$\begin{gathered}\{int\ a \wedge int\ b \wedge b > 0 \wedge a \geq 0\}\ a = b;\ \{b \geq a\}\\=\\int\ a \wedge int\ b \wedge b > 0 \wedge a \geq 0 \wedge a' = b \implies b \geq a'\end{gathered} \tag{2}$$

Proving (1)

Because of $a < 0$, multiplying a by 2 using $a' = 2a$ will always implies $a' < 0$, a' will never become positive due to the usage of mathematical integer, no "computer" ones which are cyclic. Therefore, $b \geq a'$ will always be true if and only if $int\ a \wedge int\ b \wedge b > 0 \wedge a < 0$ is true and after executing $a = 2a$.

Proving (2)

Because of $b > 0$, applying $a' = b$ will always implies $a' > 0 \wedge a' = b$. Because $a' = b \implies b \geq a'$, $b \geq a'$ will always be true if and only if $int\ a \wedge int\ b \wedge b > 0 \wedge a \geq 0$ is true and after executing $a = b$.

Exercise 4 : Formal Proof of Hoare Triple : while loop

$$
\begin{aligned}
&\{int\ n \wedge n > 0 \wedge int\ x \wedge x > 0\}\\
&i = 0;\\
&power = 1;\\
&while(i < n)\{\\
&\quad power = power * x\\
&\quad i = i + 1\\
&\}\\
&\{power = x^n\}
\end{aligned}
\tag{3}
$$

In order to prove the previous Hoare Triple, we have to prove the *total correctness* = *partial correctness*+ *termination*.

Proving *Termination*

In order to prove the termination of (3), we need to find a variant *var* which is a non-negative integer expression that is decreased by 1 in each execution of the loop body and cannot go below 0.
We transform (3) into

$$\{int\ var \wedge var > 0\}\ power = power * x;\ \ i = i + 1;\ \ \{var > var' \geq 0\}$$

where

$$var = n - i$$

We know that $int\ n \wedge n > 0 \wedge i = 0 \wedge int\ i\ (i = 0 \implies int\ i$, then $int\ (n-i) \wedge (n-i) > 0$ is true, so the pre-condition is fullfiled.
Now we transform the previous Hoare Triple into

$$int\ (n-i) \wedge (n-i) > 0 \wedge power' = power * x \wedge i' = i + 1 \implies n - i > n - i' \geq 0$$

Due to $n - i > 0$ and $i' = i + 1$, we would have $n - i' = n - (i + 1) \geq 0$ since the lowest value greater than 0 is 1, $n - i + 1$ could,at the lowest, be 1, therefore $n - i' \geq 0$ is true.
Due to $n > 0 \wedge int\ n \wedge int\ i \wedge n - i > 0$, we know that $n - i > n - i + 1$. So we have proved that (3) is terminating.

Proving *Partial Correctness*

We have to find a loop invariant *inv* that is true at the following points :

- before the loop
- before each execution of the loop body
- after each execution of the loop body
- after the loop

Then, we could transform the equation 3 to

$$\{int\ n \land n > 0 \land int\ x \land x > 0\}\ i = 0;\ power = 1;\ \{inv\} \tag{4}$$
$$\{inv \land i < n\}\ power = power * x;\ i = i + 1;\ \{inv\} \tag{5}$$
$$\{inv \land \neg(i < n)\}\ skip;\ \{power = x^n\} \tag{6}$$

where

$$inv = (power = x^i)$$

and we will prove the Hoare Triple (4), (5) and (6) in order to demonstrate the partial correctness of 3.

Proving (4)

To prove that (4) is true, we transform it into

$$int\ n \land n > 0 \land int\ x \land x > 0 \land i = 0 \land power = 1 \implies power = x^i$$

Due to $i = 0$, $power = 1$ and $x > 0$, $x^i = x^0 = 1 = power$, so (4) is true.

Proving (5)

To prove that (5) is true, we transform it into

$$power = x^i \land i > n \land power' = power * x \land i' = i + 1 \implies power' = x^{i'}$$

Due to $power = x^i$, executing $power' = power * x$ is the same as $power' = x^i * x = x^{i+1}$ and due to $i' = i + 1$, the invariant $power' = x^{i'} = x^{i+1}$ is true.

Proving (6)

To prove that (6) is true, we transform it into

$$power = x^i \land i \geq n \implies power = x^n$$

We know that the variable i would be equals to n after the loop : $i = 0 \implies int\ i$, i is incremented by 1 only and the loop end when $i \geq n$.
Finally, we have $(i = n \implies i \geq n) \land (x^i = x^n = power)$, so we have demonstrate the partial correctness of (3).