1 Hoare Logic Example

As an example illustrating how we can use Hoare logic to verify the correctness of a program, consider a program that computes the factorial of a number n:

$$\begin{array}{l}\{x = n \wedge n > 0\} \\ \quad y := 1; \\ \quad \textsf{while } x > 0 \textsf{ do } \{ \\ \qquad y := y * x; \\ \qquad x := x - 1 \\ \quad \} \\ \{y = n!\}\end{array}$$

Because the derivation for this proof is somewhat large, we will go through the reasoning used to construct it step by step.

At the top level, the program is a sequence of an assignment and a loop. To use the Seq rule, we need to find an assertion that holds after the assignment and before the loop. Examining the rule for while loops, we see that the assertion before the loop must be an invariant for the loop. Inspecting the loop we see that it builds the factorial up in y starting with n, then multiplying it by $n-1$, then $n-2$, etc. At each iteration, x contains the next value multiplied into y, that is:

$$y = n * (n-1) * ... * (x+1)$$

If we multiply both sides of this equality by $x!$ and re-write the equality we get

$$x! * y = n!$$

, which is an invariant for the loop. However, to make the proof go through, we need a slightly stronger invariant:

$$I = x! * y = n! \wedge x \geq 0$$

Having identified a suitable loop invariant, let us take a step back and review where we are. We want to prove that our overall partial correctness specification is valid. To do this, we need to show two facts:

$$\{x = n \wedge n > 0\}\ y := 1\ \{I\} \tag{1}$$

$$\{I\}\ \textsf{while } x > 0 \textsf{ do } \{y := y * x; x := x - 1\}\ \{y = n!\} \tag{2}$$

After showing that both (1) and (2) hold, we can use the rule Seq to obtain the desired result.

To show (1), we use the Assign axiom and obtain the following:

$$\{I[1/y]\}\ y := 1\ \{I\}$$

. Expanding this out, we obtain:

$$\{x! * 1 = n! \wedge x \geq 0\}\ y := 1\ \{x! * y = n! \wedge x \geq 0\}$$

With the following implication,

$$x = n \wedge n > 0 \implies x! * 1 = n! \wedge x \geq 0,$$

(which can be shown by an easy calculation) we obtain (1) using the rule Consequence.

Now let us prove (2). To use the While rule, we need to show that I is an invariant for the loop:

$$\{I \wedge x > 0\}\ y := y * x; x := x - 1\ \{I\} \tag{3}$$

We will show this by going backwards through the sequence of assignments:

$$\{(x-1)! * y = n! \wedge (x-1) \geq 0\}\ x := x - 1\ \{I\} \tag{4}$$

$$\{(x-1)! * y * x = n! \wedge (x-1) \geq 0\}\ y := y * x\ \{(x-1)! * y = n! \wedge (x-1) \geq 0\} \tag{5}$$

Then, using the following implication:

$$I \wedge x > 0 \implies (x-1)! * y * x = n! \wedge (x-1) \geq 0$$

we obtain (3) using Consequence, (4), and (5). Thus, I is an invariant for the loop and so by While we obtain,

$$\{I\}\ \mathsf{while}\ x > 0\ \mathsf{do}\ \{y := y * x; x := x - 1\}\ \{I \wedge x \leq 0\}$$

To finish the proof, we just have to show

$$\begin{array}{rl} & I \wedge x \leq 0 \implies y = n! \\ \textit{i.e.}, & x! * y = n! \wedge x \geq 0 \wedge x \leq 0 \implies y = n! \end{array}$$

which holds as $x \geq 0$ and $x \leq 0$ implies $x = 0$ and so $x! = 1$. The result follows by Consequence.

2 A More Compact Proof

For a more compact representation of the proof, one can indicate the assertions at each program point, along with the implications necessary for the Consequence rule. Specifically, the assertion at the start of the loop must be equal to conjunction of the assertion at the end of the loop and the guard of the loop. The assertion at the start of each branch of a conditional must be equal to the conjunction of the assertion before the conditional and the guard (for the "true" branch) or its negation (for the "false" branch). In addition, both branches must have the same assertion at the end, which is also the assertion after the conditional itself. Finally, the assertion before an assignment $x := a$ must be equal to the assertion obtained by substituting a for x in the assertion after the assignment.

$\{x = n \wedge n > 0\}$
$\implies$
$\{x! * 1 = n! \wedge x \geq 0\}$
y := 1
$\{x! * y = n! \wedge x \geq 0\}$
while $(x > 0)$ do$\{$
  $\{x! * y = n! \wedge x \geq 0 \wedge x > 0\}$
  $\implies$
  $\{(x-1)! * y * x = n! \wedge (x-1) \geq 0\}$
  y := $y * x$;
  $\{(x-1)! * y = n! \wedge (x-1) \geq 0\}$
  x := $x - 1$
  $\{x! * y = n! \wedge x \geq 0\}$
$\}$
$\{x! * y = n! \wedge x \geq 0 \wedge \neg(x > 0)\}$
$\implies$
$\{= n!\}$