1 Hoare triples

Triple assertion, usually writen as:

$$(\!|\,\phi\,|\!)\ P\ (\!|\,\psi\,|\!)$$

Which (roughly) means:
If the program P is run in a stat that satisfies ϕ, then the state resulting from P's execution will satisfy ψ.

ϕ - Is called *precondition* of P and ψ is called *postcondition*

Often, we do not want to put any constraints on the initial state; we simply wish to say that, no matter what state we start the program in, the resulting state should satisfy ψ. In that case the precondition can be set to $\top$.

$$(\!|\,\top\,|\!)\ P\ (\!|\,\psi\,|\!)$$

We need a way of remembering the initial value of x, to cope with the fact that it is modified by the program. Logical variables achieve just that: in the specification

$$(\!|\,x = x_0 \land x \geq 0\,|\!)\ Fac2\ (\!|\,y = x_0!\,|\!)$$

The x_0 is a logical variable and we read it as being universally quantified in the precondition. Therefore, this specification reads: for all integers x_0, if x equals x_0, $x \geq 0$ and we run the program such that it terminates, then the resulting state will satisfy y equals x_0!.

2 Proof rules

$$\frac{(\!|\phi|\!)\ C_1\ (\!|\eta|\!) \qquad (\!|\eta|\!)\ C_2\ (\!|\psi|\!)}{(\!|\phi|\!)\ C_1;C_2\ (\!|\psi|\!)}\ \text{Composition}$$

$$\frac{}{(\!|\psi[E/x]|\!)\ x = E\ (\!|\psi|\!)}\ \text{Assignment}$$

$$\frac{(\!|\phi \land B|\!)\ C_1\ (\!|\psi|\!) \qquad (\!|\phi \land \neg B|\!)\ C_2\ (\!|\psi|\!)}{(\!|\phi|\!)\ \texttt{if}\ B\ \{C_1\}\ \texttt{else}\ \{C_2\}\ (\!|\psi|\!)}\ \text{If-statement}$$

$$\frac{(\!|\psi \land B|\!)\ C\ (\!|\psi|\!)}{(\!|\psi|\!)\ \texttt{while}\ B\ \{C\}\ (\!|\psi \land \neg B|\!)}\ \text{Partial-while}$$

$$\frac{\vdash_{\mathrm{AR}} \phi' \to \phi \qquad (\!|\phi|\!)\ C\ (\!|\psi|\!) \qquad \vdash_{\mathrm{AR}} \psi \to \psi'}{(\!|\phi'|\!)\ C\ (\!|\psi'|\!)}\ \text{Implied}$$

Figure 4.1. Proof rules for partial correctness of Hoare triples.

2.1 Composition

Composition given specifications for the program fragments C_1 and C_2 say

$$(\!|\,\phi\,|\!)\ C_1\ (\!|\,\eta\,|\!) \text{ and } (\!|\,\eta\,|\!)\ C_1\ (\!|\,\psi\,|\!)$$

where the postcondition of C_1 is also precondition of C_2, the proof rule allows us to derive a specification for $C_1;C_2$

$$(\!|\,\phi\,|\!)\ C_1;C_2\ (\!|\,\psi\,|\!)$$

2.2 Assignment

Assignment rule has no premises and so is an axiom of logic. It states that we want to show that ψ holds in the state following the assignment $x = E$, we must show that $\psi[E/x]$ holds before the assignment.
We obtain $\psi[E/x]$ by taking ψ and replacing all (free) occurences of x in ψ with E

2.3 If then else

If then else proof rule allows us to prove a triple by decomposing it into two triples on in which B evaluates to true, and one where B evaluates to false.

2.4 While

The key idea of While rule is the *invariant* ψ. In general, the body of the loop C changes the values of the variables. The *invariant* expresses a relationship between the values of these variables that is preserved by executing C.
It states that (provided B is true) if ψ is true before C is executed, and C terminates, then ψ will be true in the resulting state.

2.5 Implied

If we have proved $(\!|\, \phi \,|\!)\ C\ (\!|\, \psi \,|\!)$ and we have formula ϕ', which implies ϕ and another formula ψ, which implies ψ'. Then we can also prove that

$$(\!|\, \phi' \,|\!)\ C\ (\!|\, \psi' \,|\!)$$

Reference section

placeholder
placeholder