

1 Hoare triples

Triple assertion, usually written as:

$$\langle \phi \rangle P \langle \psi \rangle$$

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 .

$$\langle \top \rangle P \langle \psi \rangle$$

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

$$\langle x = x_0 \wedge x \geq 0 \rangle \text{Fac2} \langle y = x_0! \rangle$$

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

$$\frac{\langle \phi \rangle C_1 \langle \eta \rangle \quad \langle \eta \rangle C_2 \langle \psi \rangle}{\langle \phi \rangle C_1; C_2 \langle \psi \rangle} \text{Composition}$$

$$\frac{}{\langle \psi[E/x] \rangle x = E \langle \psi \rangle} \text{Assignment}$$

$$\frac{\langle \phi \wedge B \rangle C_1 \langle \psi \rangle \quad \langle \phi \wedge \neg B \rangle C_2 \langle \psi \rangle}{\langle \phi \rangle \text{if } B \{C_1\} \text{ else } \{C_2\} \langle \psi \rangle} \text{If-statement}$$

$$\frac{\langle \psi \wedge B \rangle C \langle \psi \rangle}{\langle \psi \rangle \text{while } B \{C\} \langle \psi \wedge \neg B \rangle} \text{Partial-while}$$

$$\frac{\vdash_{\text{AR}} \phi' \rightarrow \phi \quad \langle \phi \rangle C \langle \psi \rangle \quad \vdash_{\text{AR}} \psi \rightarrow \psi'}{\langle \phi' \rangle C \langle \psi' \rangle} \text{Implied}$$

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

$$\langle \phi \rangle C_1 \langle \eta \rangle \text{ and } \langle \eta \rangle C_2 \langle \psi \rangle$$

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

$$\langle \phi \rangle C_1;C_2 \langle \psi \rangle$$

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) occurrences 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 $\langle \phi \rangle C \langle \psi \rangle$ and we have formula ϕ' , which implies ϕ and another formula ψ , which implies ψ' . Then we can also prove that

$$\langle \phi' \rangle C \langle \psi' \rangle$$

Reference section

placeholder
placeholder