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

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

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!.

1.1 Proof rules

Composition gicen 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

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

Reference section

placeholder
placeholder