Hoare triples

Program fragments generated commence running in a 'state' of the machine. After doing some computation, they might terminate. If they do, then the result is another, usually different, state. Since our programming language does not have any procedures or local variables, the 'state' of the machine can be represented simply as a vector of values of all the variables used in the program. What syntax should we use for φR, the formal specifications of requirements for such programs? Because we are interested in the output of the program, the language should allow us to talk about the variables in the state after the program has executed, using operators like = to express equality and < for less than. You should be aware of the overloading of =. In code, it represents an assignment instruction; in logical formulas, it stands for equality, which we write == within program code. For example, if the informal requirement R says that we should

Compute a number y whose square is less than the input x.

This means we need to be able to talk not just about the state after the program executes, but also about the state before it executes. The assertions we make will therefore be triples, typically looking like φ P ψ.

which (roughly) means:

If the program P is run in a state that satisfies φ, then the state resulting from P's execution will satisfy ψ.

The specification of the program P, to calculate a number whose square is less than x, now looks like this:

$$(\!|x > 0|\!)\ P\ (\!|y \cdot y < x|\!).$$

It means that, if we run P in a state such that x > 0, then the resulting state will be such that y · y 0.

Definition 4.3 1. The form $(\!|\phi|\!)\ P\ (\!|\psi|\!)$ of our specification is called a Hoare triple, after the computer scientist C. A. R. Hoare.

2. In (4.5), the formula ϕ is called the precondition of P and ψ is called the postcondition.
3. A store or state of core programs is a function l that assigns to each variable x an integer $l(x)$.
4. For a formula ϕ of predicate logic with function symbols $-$ (unary), $+$, $-$, and $*$ (binary); and a binary predicate symbols $<$ and $=$, we say that a state l satisfies ϕ or l is a ϕ-state – written $l \vDash \phi$ – iff $\mathcal{M} \vDash_l \phi$ from page 128 holds, where l is viewed as a look-up table and the model $\mathcal{M}$ has as set A all integers and interprets the function and predicate symbols in their standard manner.
5. For Hoare triples in (4.5), we demand that quantifiers in ϕ and ψ only bind variables that do not occur in the program P.