Proof calculus for partial correctness: Proof rules

The proof calculus which we now present goes back to R. Floyd and C. A. R. Hoare. In the next subsection, we specify proof rules for each of the grammar clauses for commands. We could go on to use these proof rules directly, but it turns out to be more convenient to present them in a different form, suitable for the construction of proofs known as proof tableaux. This is what we do in the subsection following the next one.

Proof rules

They should be interpreted as rules that allow us to pass from simple assertions of the form φ P ψ to more complex ones. The rule for assignment is an axiom as it has no premises. This allows us to construct some triples out of nothing, to get the proof going.

Composition. Given specifications for the program fragments C1 and C2, say

$$(\phi)\ C_1\ (\eta) \quad \text{and} \quad (\eta)\ C_2\ (\psi),$$

where the postcondition of C1 is also the precondition of C2, the proof rule for sequential composition.

To derive a specification for C1; C2, namely

$$(\phi)\ C_1; C_2\ (\psi).$$

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

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

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

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

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

Thus, if we know that C1 takes φ-states to η-states and C2 takes η-states to ψ-states, then running C1 and C2 in that sequence will take φ-states to ψ-states.

Assignment. The rule for assignment has no premises and is therefore an axiom of our logic. It tells us that, if we wish to show that ψ holds in the state after the assignment x = E, we must show that ψ[E/x] holds before the assignment; ψ[E/x] denotes the formula obtained by taking ψ and replacing all free occurrences of x with E as defined on page 105. We read the stroke as 'in place of;' thus, ψ[E/x] is ψ with E in place of x. Several explanations may be required to understand this rule.

At first sight, it looks as if the rule has been stated in reverse; one might expect that, if ψ holds in a state in which we perform the assignment x = E, then surely ψ[E/x] holds in the resulting state, i.e. we just replace x by E. This is wrong. It is true that the assignment x = E replaces the value of x in the starting state by E, but that does not mean that we replace occurrences of x in a condition on the starting state by E.

The right way to understand the Assignment rule is to think about what you would have to prove about the initial state in order to prove that ψ holds in the resulting state. Since ψ will – in general – be saying something about the value of x, whatever it says about that value must have been true of E, since in the resulting state the value of x is E. Thus, ψ with E in place of x – which says whatever ψ says about x but applied to E – must be true in the initial state.

The right way to understand the Assignment rule is to think about what you would have to prove about the initial state in order to prove that ψ holds in the resulting state. Since ψ will – in general – be saying something about the value of x, whatever it says about that value must have been true of E, since in the resulting state the value of x is E. Thus, ψ with E in place of x – which says whatever ψ says about x but applied to E – must be true in the initial state.

- The axiom $\{\psi[E/x]\}\ x = E\ \{\psi\}$ is best applied backwards than forwards in the verification process. That is to say, if we know ψ and we wish to find ϕ such that $\{\phi\}\ x = E\ \{\psi\}$, it is easy: we simply set ϕ to be $\psi[E/x]$; but, if we know ϕ and we want to find ψ such that $\{\phi\}\ x = E\ \{\psi\}$, there is no easy way of getting a suitable ψ. This backwards characteristic of the assignment and the composition rule will be important when we look at how to construct proofs; we will work from the end of a program to its beginning.
- If we apply this axiom in this backwards fashion, then it is completely mechanical to apply. It just involves doing a substitution. That means we could get a computer to do it for us. Unfortunately, that is not true for all the rules; application of the rule for while-statements, for example, requires ingenuity. Therefore a computer can at best assist us in performing a proof by carrying out the mechanical steps, such as application of the assignment axiom, while leaving the steps that involve ingenuity to the programmer.