Proof calculus for total correctness

In the preceding section, we developed a calculus for proving *partial* correctness of triples $(\!|\phi|\!)\ P\ (\!|\psi|\!)$. In that setting, proofs come with a disclaimer: *only if* the program P terminates an execution does a proof of $\vdash_{\mathsf{par}} (\!|\phi|\!)\ P\ (\!|\psi|\!)$ tell us anything about that execution. Partial correctness does not tell us anything if P 'loops' indefinitely. In this section, we extend our proof calculus for partial correctness so that it also proves that programs terminate. In the previous section, we already pointed out that only the syntactic construct `while` $B\ \{C\}$ could be responsible for non-termination.

A proof of total correctness for a while-statement will consist of two parts: the proof of partial correctness and a proof that the given while-statement terminates. Usually, it is a good idea to prove partial correctness first since this often provides helpful insights for a termination proof.

The proof of termination usually has the following form. We identify an integer expression whose value can be shown to decrease every time we execute the body of the while-statement in question, but which is always non-negative. If we can find an expression with these properties, it follows that the while-statement must terminate; because the expression can only be decremented a finite number of times before it becomes 0. That is because there is only a finite number of integer values between 0 and the initial value of the expression.

We can codify this intuition in the following rule for total correctness which replaces the rule for the while statement:

$$\frac{(\!|\eta \wedge B \wedge 0 \leq E = E_0|\!)\ C\ (\!|\eta \wedge 0 \leq E < E_0|\!)}{(\!|\eta \wedge 0 \leq E|\!)\ \texttt{while}\ B\ \{C\}\ (\!|\eta \wedge \neg B|\!)}\ \text{Total-while.}$$

In this rule, E is the expression whose value decreases with each execution of the body C. This is coded by saying that, if its value equals that of the logical variable E0 before the execution of C, then it is strictly less than E0 after it – yet still it remains non-negative. As before, η is the invariant. We use the rule Total-while in tableaux similarly to how we use Partial while, but note that the body of the rule C must now be shown to satisfy

$$(\!|\eta \wedge B \wedge 0 \leq E = E_0|\!)\ C\ (\!|\eta \wedge 0 \leq E < E_0|\!).$$

When we push $\eta \wedge 0 \leq E < E_0$ upwards through the body, we have to prove that what emerges from the top is implied by $\eta \wedge B \wedge 0 \leq E = E_0$; and the weakest precondition for the entire while-statement, which gets written above that while-statement, is $\eta \wedge 0 \leq E$.