ETHzürich

Alexander J. Summers

Program Verification

Exercise Sheet 7: Verification Condition Generation

Assignment 1 (Avoiding Duplication)

On slide 161, an alternative *wlp* definition for non-deterministic choice is given, which avoids the duplication of the current postcondition. The idea is to introduce a fresh propositional atom p, and use a formula $p \Leftrightarrow A$ to define p to be equivalent to A. This formula is then placed on the left-hand-side of an implication: why is this the correct thing to do, rather than conjoining it?

Assignment 2 (Multiple Verification Conditions)

Let s be the example program from slide 165:

$$\begin{aligned}
&\textit{if}\,(x > 0)\,\{ \\
&\quad \textit{assert } x = 2 \\
&\}\,\textit{else}\,\{ \\
&\quad \textit{assert } x < 0; \\
&\quad \textit{assert } x \neq 0 \\
&\}
\end{aligned}$$

1. Show what verification conditions the *wlp** operator from slide 167 would compute for the program s and postcondition $\{x = 2\}$. Remember to rewrite assert statements as assert-assume pairs (cf. slide 166) before applying the operator.

2. Which of the generated verification conditions (expressed as entailments) are true? What errors should be reported for the program?

3. The verification condition corresponding to the postcondition $x = 2$ will end up being duplicated. This illustrates a problem with the proposed definition of *wlp** for programs with many branching statements: what is it?

4. How could you improve the definition of *wlp** to avoid this problem?