Mini-SymEx – Weakest-Precondition Engine

Alexander Weigl

June 19, 2021

$$
\begin{aligned}
vc(v = e\,;P) &= vc(P)[v/e] \\
vc(\texttt{if } c \texttt{ then } b_1 \texttt{ else } b_2 \;;P) &= (c \to vc(b_1\,;P)) \wedge (\neg c \to vc(b_2\,;P)) \\
vc(\texttt{choose } v : e\,;P) &= \exists v.\, e \wedge vc(P) \\
vc(\texttt{havoc } v\,;P) &= \forall v.\, vc(P) \\
vc(\texttt{assume } e\,;P) &= e \to vc(P) \\
vc(\texttt{assert } e\,;P) &= e \wedge vc(P) \\
vc(\texttt{assert } e\,;P) &= e \wedge (e \to vc(P)) \\
vc(\epsilon) &= \mathit{true}
\end{aligned}
$$

Definition 1. *A program P is valid w.r.t. to its specification iff $vc(P)$ is valid.*

Example 1. *Let us consider the following program P_0:*

```
int x = 0;
choose x : x > 0;
assert x == 2;
```

$$
\begin{aligned}
vc(P_0) &= vc(x = 0\,;(\texttt{choose } x : x > 0\,;(\texttt{assert } x == 2\,;\epsilon))) \\
&= vc((\texttt{choose } x : x > 0\,;(\texttt{assert } x == 2\,;\epsilon)))[x/0] \\
&= (\exists x.\, x > 0 \wedge vc(\texttt{assert } x == 2\,;\epsilon))[x/0] \\
&= (\exists x.\, x > 0 \wedge x = 2 \wedge vc(\epsilon))[x/0] \\
&= (\exists x.\, x > 0 \wedge x = 2 \wedge \mathit{true})[x/0] \\
&= (\exists x.\, x > 0 \wedge x = 2 \wedge \mathit{true})
\end{aligned}
$$