1 Source Language

⟨*statement*⟩ ::= ⟨*compound-statement*⟩
  | ‘`if`’ (⟨*expression*⟩) ⟨*compound-statement*⟩ ‘`else`’ ⟨*compound-statement*⟩
  | ‘`while`’ (⟨*expression*⟩) ⟨*compound-statement*⟩
  | ‘`for`’ (⟨*statement*⟩ ‘`;`’ ⟨*statement*⟩ ‘`;`’ ⟨*statement*⟩) ⟨*compound-statement*⟩
  | ⟨*qualifier*⟩ ⟨*type*⟩ ⟨*name*⟩ ‘`;`’
  | ⟨*qualifier*⟩ ⟨*type*⟩ ⟨*name*⟩ ‘`=`’ ⟨*expression*⟩ ‘`;`’
  | ‘`return`’ ⟨*expression*⟩ ‘`;`’
  | ⟨*expression*⟩ ‘`;`’

⟨*compound-statement*⟩ ::= ‘`{`’ ⟨*statement*⟩* ‘`}`’

⟨*expression*⟩ ::= ⟨*name*⟩ | ⟨*literal*⟩
  | (⟨*expression*⟩)
  | ⟨*expression*⟩ ⟨*binop*⟩ ⟨*expression*⟩
  | ⟨*unop*⟩ ⟨*expression*⟩
  | ⟨*name*⟩ (⟨*parameter-list*⟩)
  | ‘`&`’ ⟨*name*⟩
  | ⟨*expression*⟩ ‘`++`’
  | ⟨*expression*⟩ ‘`--`’

⟨*parameter-list*⟩ ::= ⟨*qualifier*⟩ ⟨*type*⟩ ⟨*name*⟩ ‘`,`’ ⟨*parameter-list*⟩
  | ⟨*type-qualifier*⟩ ⟨*type*⟩ ⟨*name*⟩

⟨*binop*⟩ ::= ‘`+`’ | ‘`-`’ | ‘`*`’ | ‘`/`’ | ‘`%`’
  | ‘`&&`’ | ‘`||`’
  | ‘`&`’ | ‘`|`’ | ‘`<<`’ | ‘`>>`’
  | ‘`==`’ | ‘`<=`’ | ‘`>=`’ | ‘`<`’ | ‘`>`’

⟨*unop*⟩ ::= ‘`-`’ | ‘`!`’ | ‘`~`’ | ‘`*`’ | ‘`++`’ | ‘`--`’

⟨*type-qualifier*⟩ ::= ‘`const`’ | ‘`volatile`’

⟨*type*⟩ ::= ⟨*type-qualifier*⟩ ⟨*type-specifier*⟩

2 Symbolic Execution

2.1 Expressions

$$\frac{}{\langle S; v\rangle \Downarrow \langle S; v\rangle}\ \textsc{Literal}$$

2.2 Statements

$$\frac{\langle S; e\rangle \Downarrow \langle S'; s\rangle}{\langle S; e\texttt{;}\rangle \Downarrow \langle S'; \emptyset\rangle}\ \textsc{Expression}$$

$$\frac{\forall i \in 1..n, \langle S_i; c_i\rangle \Downarrow \langle S_{i+1}; s_{i+1}\rangle}{\langle S_1; \texttt{\{}c_1..c_n\texttt{\}}\rangle \Downarrow \langle S_{n+1}; s_{n+1}\rangle}\ \textsc{CompoundStatement}$$

$$\frac{\begin{array}{c}\langle S; e\rangle \Downarrow \langle S_1; g_1\rangle \qquad g(S) \not\Longrightarrow g_1 \qquad g(S) \not\Longrightarrow \neg g_1\\ \langle S_1[g \mapsto g(S_1) \wedge g_1]; c_1\rangle \Downarrow \langle S_2; s_2\rangle\\ \langle S_1[g \mapsto g(S_1) \wedge \neg g_1]; c_1\rangle \Downarrow \langle S_3; s_3\rangle\\ S' = \langle (g_1?g(S_2){:}g(S_3)); (g_1?\rho(S_2){:}\rho(S_3)); (g_1?\mu(S_2){:}\mu(S_3))\rangle\end{array}}{\langle S; \textbf{if } e\ c_1\ \textbf{else}\ c_2\rangle \Downarrow \langle S'; \emptyset\rangle}\ \textsc{IfElse}$$

$$\frac{\langle S; e\rangle \Downarrow \langle S_1; g_1\rangle \qquad g(S) \Longrightarrow g_1 \qquad \langle S_1; c_1\rangle \Downarrow \langle S_2; s\rangle}{\langle S; \textbf{if } e\ c_1\ \textbf{else}\ c_2\ \rangle \Downarrow \langle S_2; \emptyset\rangle}\ \textsc{IfTrue}$$

$$\frac{\langle S; e\rangle \Downarrow \langle S_1; g_1\rangle \qquad g(S) \Longrightarrow \neg g_1 \qquad \langle S_1; c_2\rangle \Downarrow \langle S_2; s\rangle}{\langle S; \textbf{if } e\ c_1\ \textbf{else}\ c_2\rangle \Downarrow \langle S_2; \emptyset\rangle}\ \textsc{IfFalse}$$

2.3 Memory

$$\frac{\rho(S)[x] = s}{\langle S; x\rangle \Downarrow \langle S; s\rangle}\ \textsc{Var}$$

$$\frac{x \notin \mathrm{dom}\ \rho(S)}{\langle S; \tau\ \ x\texttt{;}\rangle \Downarrow \langle S[\rho \mapsto (\rho(S), (x \to \emptyset)]; s\rangle}\ \textsc{DeclareLocal}$$

$$\frac{x \notin \mathrm{dom}\ \rho(S) \qquad \langle S; e\rangle \Downarrow \langle S'; s\rangle}{\langle S; \tau\ \ x\ \texttt{=}\ e\texttt{;}\rangle \Downarrow \langle S'[\rho \mapsto (\rho(S'), (x \to s)]; s\rangle}\ \textsc{DeclareAssignLocal}$$

$$\frac{\langle S; e_1\rangle \Downarrow \langle S_1; \mathrm{ptr}\ x\rangle \qquad x \in \mathrm{dom}\ \rho(S_1) \qquad \langle S_1; e_2\rangle \Downarrow \langle S_2; s\rangle}{\langle S; \texttt{*}e_1\ \texttt{=}\ e_2\rangle \Downarrow \langle S_2[\rho \mapsto (\rho(S_2), (x \to s)]; s\rangle}\ \textsc{UpdLocal}$$

$$\frac{\langle S; e_1\rangle \Downarrow \langle S_1; s_1\rangle \qquad s_1 \neq \mathrm{ptr}\ x \qquad \langle S_1; e_2\rangle \Downarrow \langle S_2; s_2\rangle}{\langle S; \texttt{*}e_1\ \texttt{=}\ e_2\rangle \Downarrow \langle S_2[\mu \mapsto (\mu(S_2), (s_1 \to s_2)]; s_2\rangle}\ \textsc{UpdGlobal}$$

$$\frac{\langle S; e\rangle \Downarrow \langle S'; \mathrm{ptr}\ x\rangle \qquad \rho(S')[x] = s}{\langle S; \texttt{*}e\rangle \Downarrow \langle S'; s\rangle}\ \textsc{SelLocal}$$

$$\frac{\langle S; e\rangle \Downarrow \langle S'; \mathrm{ptr}\ x\rangle \qquad \rho(S')[x] = s}{\langle S; \texttt{*}e\rangle \Downarrow \langle S'; s\rangle}\ \textsc{SelGlobal}$$