Verifier Core Language BNF Grammar

Jenna Wise, Johannes Bader, Jonathan Aldrich, Éric Tanter

November 12, 2018

$$
\begin{array}{rcll}
x, y, z & \in & \mathit{VAR} & \textit{(variables)} \\
v & \in & \mathit{VAL} & \textit{(values)} \\
e & \in & \mathit{EXPR} & \textit{(expressions)} \\
s & \in & \mathit{STMT} & \textit{(statements)} \\
o & \in & \mathit{LOC} & \textit{(object Ids)} \\
f & \in & \mathit{FIELDNAME} & \textit{(field names)} \\
m & \in & \mathit{METHODNAME} & \textit{(method names)} \\
C, D & \in & \mathit{CLASSNAME} & \textit{(class names)} \\
P & ::= & \overline{\mathit{cls}}\ s & \\
\mathit{cls} & ::= & \mathit{class}\ C\ \mathit{extends}\ D\ \{\overline{\mathit{field}}\ \overline{A}\ \overline{\mathit{method}}\} & \\
\mathit{field} & ::= & T\ f; & \\
A & ::= & \mathit{predicate}\ \alpha_C(\overline{x}) := \widetilde{\phi} & \\
T & ::= & \mathit{int} \mid C \mid \top & \\
\mathit{method} & ::= & T\ m(\overline{T\ x})\ \mathit{dynamic}\ \mathit{contract}\ \mathit{static}\ \mathit{contract}\ \{s\} & \\
\mathit{contract} & ::= & \mathit{requires}\ \widetilde{\phi}\ \mathit{ensures}\ \widetilde{\phi} & \\
\oplus & ::= & + \mid - \mid * \mid \backslash & \\
\odot & ::= & \neq \mid = \mid < \mid > \mid \leq \mid \geq & \\
s & ::= & \mathit{skip} \mid s_1\ ;\ s_2 \mid T\ x := e \mid \mathit{if}\ (x \odot y)\ \{s_1\}\ \mathit{else}\ \{s_2\} \mid x.f := y \mid x := \mathit{new}\ C & \\
 & & \quad \mid y := z.m(\overline{x}) \mid y := z.m_C(\overline{x}) \mid \mathit{assert}\ \phi \mid \mathit{release}\ \phi \mid \mathit{hold}\ \phi\ \{s\} & \\
e & ::= & v \mid x \mid e \oplus e \mid e.f & \\
x & ::= & \mathit{result} \mid \mathit{id} \mid \mathit{old}(\mathit{id}) \mid \mathit{this} & \\
v & ::= & n \mid o \mid \mathit{null} & \\
\phi & ::= & \mathrm{true} \mid e \odot e \mid \alpha(\overline{e}) \mid \mathit{acc}(e.f) \mid \phi * \phi & \\
\widetilde{\phi} & ::= & \phi \mid ? * \phi &
\end{array}
$$