Verifier Core Language BNF Grammar

Jenna Wise, Johannes Bader, Henry Blanchette, Jonathan Aldrich, Éric Tanter

May 31, 2019

$$
\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)} \\
\alpha & \in & \mathit{PREDNAME} & \textit{(predicate names)} \\
P & ::= & \overline{\mathit{cls}}\ s & \\
\mathit{cls} & ::= & \texttt{class}\ C\ \texttt{extends}\ D\ \{\overline{\mathit{field}}\ \overline{\mathit{pred}}\ \overline{\mathit{method}}\} & \\
\mathit{field} & ::= & T\ f; & \\
\mathit{pred} & ::= & \texttt{predicate}\ \alpha_C(\overline{T\ x}) = \widetilde{\phi} & \\
T & ::= & \texttt{int} \mid C \mid \top & \\
\mathit{method} & ::= & T\ m(\overline{T\ x})\ \texttt{dynamic}\ \texttt{static}\ \mathit{contract}\ \{s\} & \\
\mathit{contract} & ::= & \texttt{requires}\ \widetilde{\phi}\ \texttt{ensures}\ \widetilde{\phi} & \\
\oplus & ::= & + \mid - \mid * \mid \backslash & \\
\odot & ::= & \neq \mid = \mid < \mid > \mid \leq \mid \geq & \\
s & ::= & \texttt{skip} \mid s_1\ ;\ s_2 \mid T\ x \mid x := e \mid \texttt{if}\ (e)\ \{s_1\}\ \texttt{else}\ \{s_2\} \mid \texttt{while}\ (e)\ \texttt{inv}\ \widetilde{\phi}\ \{s\} & \\
 & & \mid x.f := y \mid x := \texttt{new}\ C \mid y := z.m(\overline{x}) \mid y := z.m_C(\overline{x}) \mid \texttt{assert}\ \phi \mid \texttt{release}\ \phi & \\
 & & \mid \texttt{hold}\ \phi\ \{s\} \mid \texttt{fold}\ \alpha(\overline{e}) \mid \texttt{unfold}\ \alpha(\overline{e}) & \\
e & ::= & v \mid x \mid e \oplus e \mid e \odot e \mid e.f & \\
x & ::= & \texttt{result} \mid \mathit{id} \mid \texttt{old}(\mathit{id}) \mid \texttt{this} & \\
v & ::= & n \mid o \mid \texttt{null} \mid \texttt{true} \mid \texttt{false} & \\
\phi & ::= & e \mid \alpha(\overline{e}) \mid \texttt{acc}(e.f) \mid \phi \wedge \phi \mid \phi * \phi \mid (\texttt{if}\ e\ \texttt{then}\ \phi\ \texttt{else}\ \phi) \mid (\texttt{unfolding}\ \alpha(\overline{e})\ \texttt{in}\ \phi) & \\
\widetilde{\phi} & ::= & \phi \mid ? * \phi &
\end{array}
$$