Verifier Core Language BNF Grammar

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

June 5, 2019

$$
\begin{array}{rcll}
x, y, z & \in & VAR & \textit{(variables)} \\
v & \in & VAL & \textit{(values)} \\
e & \in & EXPR & \textit{(expressions)} \\
s & \in & STMT & \textit{(statements)} \\
o & \in & LOC & \textit{(object Ids)} \\
f & \in & FIELDNAME & \textit{(field names)} \\
m & \in & METHODNAME & \textit{(method names)} \\
C, D & \in & CLASSNAME & \textit{(class names)} \\
\alpha & \in & PREDNAME & \textit{(predicate names)} \\
\end{array}
$$

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