Verifier Core Language BNF Grammar

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

May 31, 2019

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

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