Verifier Core Language BNF Grammar

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

June 4, 2019

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

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