Verifier Core Language BNF Grammar

Jenna Wise, Johannes Bader, Jonathan Aldrich, Joshua Sunshine

October 29, 2018

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

$$
\begin{array}{rcl}
P & ::= & \overline{cls}\ s \\
cls & ::= & class\ C\ extends\ D\ \{\overline{field}\ \overline{A}\ \overline{method}\} \\
field & ::= & T\ f; \\
A & ::= & predicate\ \alpha_C(\overline{x}) := \widetilde{\phi} \\
T & ::= & int \mid C \mid \top \\
method & ::= & T\ m(\overline{T\ x})\ dynamic\ contract\ static\ contract\ \{s\} \\
contract & ::= & requires\ \widetilde{\phi}\ ensures\ \widetilde{\phi} \\
\oplus & ::= & + \mid - \mid * \mid \backslash \\
\odot & ::= & \neq \mid = \mid < \mid > \\
s & ::= & skip \mid s_1\ ;\ s_2 \mid T\ x \mid x := e \mid if\ (x \odot y)\ \{s_1\}\ else\ \{s_2\} \mid x.f := y \mid x := new\ C \\
 & & \mid y := z.m(\overline{x}) \mid y := z.m_C(\overline{x}) \mid assert\ \phi \mid release\ \phi \mid hold\ \phi\ \{s\} \\
e & ::= & v \mid x \mid e \oplus e \mid e.f \\
x & ::= & result \mid id \mid old(id) \mid this \\
v & ::= & n \mid o \mid null \\
\phi & ::= & \text{true} \mid e \odot e \mid \alpha(\overline{e}) \mid acc(e.f) \mid \phi * \phi \\
\widetilde{\phi} & ::= & \phi \mid ? * \phi \\
\end{array}
$$