

Verifier Core Language BNF Grammar

Jenna Wise

October 20, 2018

x, y, z	$\in VAR$	(variables)
v	$\in VAL$	(values)
e	$\in EXPR$	(expressions)
s	$\in STMT$	(statements)
o	$\in LOC$	(object Ids)
f	$\in FIELDNAME$	(field names)
m	$\in METHODNAME$	(method names)
C	$\in CLASSNAME$	(class names)
P	$::= \overline{cls} \ s$	
cls	$::= \text{class } C \text{ extends } C \ \{\overline{field} \ \overline{A} \ \overline{method}\}$	
$field$	$::= T \ f;$	
A	$::= \text{predicate } \alpha_C(\overline{x}) := \tilde{\phi}$	
T	$::= int \mid C \mid \top$	
$method$	$::= T \ m(\overline{T} \ x) \ \text{dynamic contract static contract } \{s\}$	
$contract$	$::= \text{requires } \tilde{\phi} \ \text{ensures } \tilde{\phi}$	
\oplus	$::= + \mid - \mid * \mid \backslash$	
\odot	$::= \neq \mid = \mid < \mid >$	
s	$::= skip \mid s_1 ; s_2 \mid T \ x \mid x := e \mid \text{if } (x \odot y) \ \{s_1\} \ \text{else } \{s_2\} \mid x.f := y \mid x := \text{new } C$ $\mid y := z.m(\overline{x}) \mid y := z.m_C(\overline{x}) \mid \text{assert } \phi \mid \text{release } \phi \mid \text{hold } \phi \ \{s\}$	
e	$::= v \mid x \mid e \oplus e \mid e.f$	
x	$::= result \mid id \mid old(id) \mid this$	
v	$::= n \mid o \mid null$	
ϕ	$::= \text{true} \mid e \odot e \mid \alpha(\overline{e}) \mid acc(e.f) \mid \phi * \phi$	
$\tilde{\phi}$	$::= \phi \mid ? * \phi$	