

| | | | |
|----------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| x | $::=$ | $\text{ALPHANUMERIC} \mid \star\text{ALPHANUMERIC}$ | <i>variables</i> |
| $\star\ell$ | $::=$ | $\ell \mid *$ | <i>general labels</i> |
| ℓ | | | <i>labels</i> |
| T | $::=$ | $[t, \dots]$ | <i>stack</i> |
| t | $::=$ | $\star\ell \times S$ | <i>stack frames</i> |
| S | $::=$ | $[s, \dots]$ | <i>programs</i> |
| s | $::=$ | $\ell : \star\ell : d$ | <i>clauses</i> |
| d | $::=$ | $x = e \mid \text{return } x \mid \text{goto } \ell \mid \text{goto } \ell \text{ if not } x$ $\mid \text{raise } x \mid \text{catch } x \mid \text{pass}$ | <i>directives</i> |
| B | $::=$ | $\{x \mapsto m, \dots\}$ | <i>bindings</i> |
| H | $::=$ | $\{m \mapsto v, \dots\}$ | <i>heap</i> |
| v | $::=$ | $\mathbb{Z} \mid [m, \dots] \mid (m, \dots) \mid B \mid F \mid M \mid \text{undefined} \mid \text{None}$ | <i>values</i> |
| e | $::=$ | $v \mid x \mid \text{def } x(x, \dots) = \{S\} \mid x(x, \dots) \mid [x, \dots] \mid (x, \dots)$ | <i>expressions</i> |
| Y | $::=$ | $[y, \dots]$ | <i>microcode stack</i> |
| Z | $::=$ | $[z, \dots]$ | <i>microcode literal stack</i> |
| y | $::=$ | $\text{STORE} \mid \text{WRAP} \mid \text{BIND} \mid \text{LOOKUP} \mid \text{LIST } n \mid \text{TUPLE } n$ $\mid \text{ADVANCE} \mid \text{POP} \mid \text{RAISE} \mid \text{ALLOCNAMEERROR}$ | <i>microcode commands</i> |
| z | $::=$ | $x \mid m \mid v$ | <i>microcode literals</i> |
| P | $::=$ | $m \mapsto m$ | <i>parental map</i> |
| $\star m$ | $::=$ | $m \mid \eta \mid *$ | <i>general memory locations</i> |
| η, m | | | <i>memory locations</i> |
| F | $::=$ | $\langle m, \text{def } (x) \rightarrow S \rangle \mid \mathfrak{F}$ | <i>general functions</i> |
| M | $::=$ | $\langle m, m, \text{def } (x) \rightarrow S \rangle \mid \langle m, \mathfrak{M} \rangle$ | <i>general methods</i> |
| \mathfrak{F} | | | <i>magic functions</i> |
| \mathfrak{M} | | | <i>magic methods</i> |

Figure 1: Expression Grammar

$$\begin{array}{c}
\text{STORE } v \\
\frac{m \notin H \quad H' = H[m \mapsto v]}{P, Z \parallel [v, \text{STORE}] \parallel Y, T, H, \eta \longrightarrow^1 P, Z \parallel [m] \parallel Y, T, H', \eta} \\
\\
\text{WRAP } m \\
\frac{B = H[\eta] \quad B' = B[x \mapsto m] \quad H' = H[\eta \mapsto B']}{P, Z \parallel [m, x, \text{WRAP}] \parallel Y, T, H, \eta \longrightarrow^1 P, Z \parallel [v] \parallel Y, T, H, \eta} \\
\\
\text{BIND } m \text{ TO } x \\
\frac{v = \text{GETOBJ}(H, m)}{P, Z \parallel [m, x, \text{BIND}] \parallel Y, T, H, \eta \longrightarrow^1 P, Z \parallel Y, T, H', \eta} \\
\\
\text{ADVANCE} \\
\frac{S(\ell) = \ell : \ell^{\star'} : d \quad \ell \blacktriangleleft^S \ell^{\star''}}{P, Z \parallel [\text{ADVANCE}] \parallel Y, [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, [\], [\langle \ell^{\star''}, S \rangle] \parallel T, H, \eta} \\
\\
\text{POP} \\
\frac{\eta' = P[\eta] \quad T = [t, \langle \ell'', S' \rangle] \parallel T' \quad \ell'' \blacktriangleleft^{S'} \ell^{\star''''}}{P, Z \parallel [\text{POP}] \parallel Y, T, H, \eta \longrightarrow^1 P, [\], [\langle \ell^{\star''''}, S \rangle] \parallel T', H, \eta'} \\
\\
\text{LOOK UP } x \text{ (BOUND)} \\
\frac{B = H[\eta] \quad m = B[x]}{P, Z \parallel [x, \text{LOOKUP}] \parallel Y, T, H, \eta \longrightarrow^1 P, Z \parallel [m] \parallel Y, T, H, \eta} \\
\\
\text{LOOK UP } x \text{ (NAMEERROR)} \\
\frac{\text{LOOKUP}(P, H, \eta, x) = *}{P, Z \parallel [x, \text{LOOKUP}] \parallel Y, T, H, \eta \longrightarrow^1 P, [\text{ALLOCNAMEERROR}, \text{RAISE}], T, H, \eta} \\
\\
\text{MAKE LIST} \\
\frac{v = [m_1, \dots, m_n]}{P, Z \parallel [m_1, \dots, m_n, \text{LIST } n] \parallel Y, T, H, \eta \longrightarrow^1 P, Z \parallel [v] \parallel Y, T, H, \eta} \\
\\
\text{MAKE TUPLE} \\
\frac{v = (m_1, \dots, m_n)}{P, Z \parallel [m_1, \dots, m_n, \text{TUPLE } n] \parallel Y, T, H, \eta \longrightarrow^1 P, Z \parallel [v] \parallel Y, T, H, \eta} \\
\\
\text{RAISE (NO EXCEPTION LABEL)} \\
\frac{S(\ell) = \ell : * : d \quad \eta' = P(\eta)}{P, Z \parallel [\text{RAISE}] \parallel Y, [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Z \parallel [\text{RAISE}] \parallel Y, T, H, \eta'} \\
\\
\text{RAISE (CAUGHT)} \\
\frac{S(\ell) = \ell : \ell_0 : d \quad S(\ell_0) = \ell_0 : \ell_1 : \text{CATCH } x \quad Y' = [x, \text{BIND}, \text{ADVANCE}]}{P, Z \parallel [\text{RAISE}] \parallel Y, [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Z \parallel Y' \parallel Y, [\langle \ell_0, S \rangle] \parallel T, H, \eta'}
\end{array}$$

Figure 2: Mi2rocommands

LITERAL ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell^{\star'} : x = e \quad \text{VALUEOFLITERAL}(e) = v \quad Y = [v, \text{STORE}, \text{WRAP}, \text{STORE}, x, \text{BIND}, \text{ADVANCE}]}{P, [], [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Y, [\langle \ell, S \rangle] \parallel T, H, \eta}$$

NAME ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell^{\star'} : x_1 = x_2 \quad Y = [x_2, \text{LOOKUP}, x_1, \text{BIND}, \text{ADVANCE}]}{P, [], [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Y, [\langle \ell, S \rangle] \parallel T, H, \eta}$$

LIST ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell^{\star'} : x = [x_1, \dots, x_n] \quad Y = [(x_1, \text{LOOKUP}, \text{STORE}), \dots, (x_n, \text{LOOKUP}, \text{STORE}), \text{LIST } n, x, \text{BIND}, \text{ADVANCE}]}{P, [], [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Y, [\langle \ell, S \rangle] \parallel T, H, \eta}$$

TUPLE ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell^{\star'} : x = [x_1, \dots, x_n] \quad Y = [(x_1, \text{LOOKUP}, \text{STORE}), \dots, (x_n, \text{LOOKUP}, \text{STORE}), \text{TUPLE } n, x, \text{BIND}, \text{ADVANCE}]}{P, [], [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Y, [\langle \ell, S \rangle] \parallel T, H, \eta}$$

Figure 3: Operational Semantics: Assignment

PASS

$$\frac{S(\ell) = \ell : \ell^{\star'} : \text{pass} \quad Y = [\text{ADVANCE}]}{P, [], [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Y, [\langle \ell, S \rangle] \parallel T, H, \eta}$$

RETURN

$$\frac{S(\ell) = \ell : \ell^{\star'} : \text{return } x \quad T = [\langle \ell'', S' \rangle] \parallel T' \quad S(\ell'') = \ell'' : \ell^{\star'''} : x' = e \quad Y = [x, \text{LOOKUP}, \text{POP}, x', \text{BIND}, \text{ADVANCE}]}{P, [], [\langle \ell, S \rangle] \parallel T, H, \eta \longrightarrow^1 P, Y, [\langle \ell, S \rangle] \parallel T, H, \eta}$$

Figure 4: Operational Semantics: Flow