$$
\begin{array}{rcll}
x & ::= & \text{ALPHANUMERIC} \mid {\star}\text{ALPHANUMERIC} & \textit{variables} \\
\overset{\star}{\ell} & ::= & \ell \mid * & \textit{general labels} \\
\ell & & & \textit{labels} \\
T & ::= & [t, \ldots] & \textit{stack} \\
t & ::= & \langle \eta, \overset{\star}{\ell}, S \rangle & \textit{stack frames} \\
S & ::= & [s, \ldots] & \textit{programs} \\
s & ::= & \ell : \overset{\star}{\ell} : d & \textit{clauses} \\
d & ::= & x = e \mid \texttt{return}\ x \mid \texttt{goto}\ \ell \mid \texttt{goto}\ \ell\ \texttt{if}\ \texttt{not}\ x & \textit{directives} \\
 & & \mid \texttt{raise}\ x \mid \texttt{catch}\ x \mid \texttt{pass} & \\
B & ::= & \{x \mapsto m, \ldots\} & \textit{bindings} \\
H & ::= & \{m \mapsto v, \ldots\} & \textit{heap} \\
v & ::= & \mathbb{Z} \mid [m, \ldots] \mid (m, \ldots) \mid B \mid F \mid M \mid * & \textit{values} \\
e & ::= & v \mid x \mid \texttt{def}\ x(x, \ldots) = \{S\} \mid x(x, \ldots) \mid x.x \mid [x, \ldots] \mid (x, \ldots) & \textit{expressions} \\
Y & ::= & [y, \ldots] & \textit{microcode stack} \\
Z & ::= & [z, \ldots] & \textit{microcode literal stack} \\
y & ::= & \textsc{Store} \mid \textsc{Wrap} \mid \textsc{Bind} \mid \textsc{LookUp} \mid \textsc{List}\ n \mid \textsc{Tuple}\ n & \textit{microcode commands} \\
 & & \mid \textsc{Advance} \mid \textsc{Pop} \mid \textsc{Push}\ S \mid \textsc{Raise} \mid \textsc{Goto}\ \ell \mid \textsc{Gotoifn}\ \ell & \\
 & & \mid \textsc{Call}\ n \mid \textsc{Retrieve} \mid \textsc{AllocNameError} \mid \textsc{AllocTypeError} \mid \textsc{AllocAttrError} & \\
z & ::= & x \mid m \mid v & \textit{microcode literals} \\
P & ::= & m \mapsto m & \textit{parental map} \\
\overset{\star}{m} & ::= & m \mid \eta \mid * & \textit{general memory locations} \\
\eta, m & ::= & \text{<ADDRESS>} & \textit{memory locations} \\
F & ::= & \langle \eta, \texttt{def}\ (x, \ldots) \to S \rangle \mid \mathfrak{F} \mid \mathfrak{M} & \textit{general functions} \\
M & ::= & \langle m, F \rangle & \textit{general methods} \\
\mathfrak{F}, \mathfrak{M} & & & \textit{magic functions} \\
n & & & \textit{integers} \\
\end{array}
$$

Figure 1: Expression Grammar

$$H_{\textsc{Init}} \qquad \textit{heap}$$

Figure 2: Initialization

$\text{STORE } v$

$$\frac{m \notin H \qquad H' = H[m \mapsto v]}{P, Z \,||[v, \text{STORE}] \,||\, Y, T, H \longrightarrow^1 P, Z \,||[m] \,||\, Y, T, H'}$$

$\text{WRAP } m$

$$\frac{v = \text{GETOBJ}(H, m)}{P, Z \,||[m, \text{WRAP}] \,||\, Y, T, H \longrightarrow^1 P, Z \,||[v] \,||\, Y, T, H}$$

$\text{BIND } m \text{ TO } x$

$$\frac{B = H[\eta] \qquad B' = B[x \mapsto m] \qquad H' = H[\eta \mapsto B']}{P, Z \,||[m, x, \text{BIND}] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\, Y, T, H'}$$

ADVANCE

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : d \qquad \ell \overset{s}{\blacktriangleleft} \overset{\star''}{\ell}}{P, Z \,||[\text{ADVANCE}] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Z \,||\, Y, [\langle \eta, \overset{\star''}{\ell}, S \rangle] \,||\, T, H}$$

POP

$$\frac{}{P, Z \,||[\text{POP}] \,||\, Y, t \,||\, T, H \longrightarrow^1 P, Z \,||\, Y, T, H}$$

$\text{PUSH } S$

$$\frac{P' = P[\eta' \mapsto \eta], \eta' \notin P \qquad S = [\ell : \overset{\star'}{\ell} : d, \dots]}{P, Z \,||[\eta, \text{PUSH } S] \,||\, Y, T, H \longrightarrow^1 P', Z \,||\, Y, [\langle \eta', \ell, S \rangle] \,||\, T, H}$$

$\text{LOOK UP } x \text{ (BOUND)}$

$$\frac{\text{LOOKUP}(P, H, \eta, x) = m}{P, Z \,||[x, \text{LOOKUP}] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Z \,||[m] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

$\text{LOOK UP } x \text{ (NAMEERROR)}$

$$\frac{\text{LOOKUP}(P, H, \eta, x) = *}{P, Z \,||[x, \text{LOOKUP}] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, [\text{ALLOCNAMEERROR}, \text{RAISE}], [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

MAKE LIST

$$\frac{v = [m_1, \dots, m_n]}{P, Z \,||[m_1, \dots, m_n, \text{LIST } n] \,||\, Y, T, H \longrightarrow^1 P, Z \,||[v] \,||\, Y, T, H}$$

MAKE TUPLE

$$\frac{v = (m_1, \dots, m_n)}{P, Z \,||[m_1, \dots, m_n, \text{TUPLE } n] \,||\, Y, T, H \longrightarrow^1 P, Z \,||[v] \,||\, Y, T, H}$$

Figure 3: Microcommands

RAISE (NO EXCEPTION LABEL)

$$\frac{S(\ell) = \ell : * : d}{P, Z \,||\,[\text{RAISE}] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Z \,||\,[\text{POP}, \text{RAISE}] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

RAISE (CAUGHT)

$$\frac{S(\ell) = \ell : \ell_0 : d \qquad S(\ell_0) = \ell_0 : \ell_1 : \texttt{catch}\ x \qquad Y' = [x, \text{BIND}, \text{ADVANCE}]}{P, Z \,||\,[\text{RAISE}] \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Z \,||\, Y' \,||\, Y, [\langle \eta, \ell_0, S \rangle] \,||\, T, H}$$

GOTO ℓ

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : d}{P, Z \,||\,[\text{GOTO}\ \ell] \,||\, Y, [\langle \eta, \ell', S \rangle] \,||\, T, H \longrightarrow^1 P, Z \,||\, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

GOTOIFN ℓ (SUCCESS)

$$\frac{H[m] = \text{FALSE} \qquad S(\ell) = \ell : \overset{\star'}{\ell} : d}{P, Z \,||\,[m, \text{GOTOIFN}\ \ell] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\,[\text{GOTO}] \,||\, Y, T, H}$$

GOTOIFN ℓ (FAILURE)

$$\frac{H[m] = \text{TRUE}}{P, Z \,||\,[m, \text{GOTOIFN}\ \ell] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\,[\text{ADVANCE}] \,||\, Y, T, H}$$

CALL FUNCTION m

$$\frac{\begin{array}{c} v = \langle \eta, \ \texttt{def}\ (x_1, \ldots, x_n) \to S \rangle \\ Y' = [\eta, \text{PUSH}\ S, m_1, x_1, \text{BIND}, \ldots, m_n, x_n, \text{BIND}] \end{array}}{P, Z \,||\,[v, m_1, \ldots, m_n, \text{CALL}\ n] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\, Y' \,||\, Y, T, H}$$

CALL FUNCTION (WRONG ARGS)

$$\frac{v = \langle \eta, \ \texttt{def}\ (x_1, \ldots, x_q) \to S \rangle q \neq n}{P, Z \,||\,[v, m_1, \ldots, m_n, \text{CALL}\ n] \,||\, Y, T, H \longrightarrow^1 P, [\text{ALLOCTYPEERROR}, \text{RAISE}], T, H}$$

GET CALL

$$\frac{v = \text{GETCALL}(H, m_0) Y' = [v, m_1, \ldots, m_n, \text{CONVERT}]}{P, Z \,||\,[m_0, \ldots, m_n, \text{GETCALL}\ n] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\, Y' \,||\, Y, T, H}$$

GET CALL (TYPEERROR)

$$\frac{* = \text{GETCALL}(H, m_0)}{P, Z \,||\,[m_0, \ldots, m_n, \text{GETCALL}\ n] \,||\, Y, T, H \longrightarrow^1 P, [\text{ALLOCTYPEERROR}, \text{RAISE}], T, H}$$

CONVERT FUNCTION

$$\frac{}{P, Z \,||\,[v, m_1, \ldots, m_n, \text{CONVERT}] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\,[v, m_1, \ldots, m_n, \text{CALL}\ n] \,||\, Y, T, H}$$

RETRIEVE x

$$\frac{\text{LOOKUPOBJ}(P, H, m, x) = m'}{P, Z \,||\,[m, x, \text{RETRIEVE}] \,||\, Y, T, H \longrightarrow^1 P, Z \,||\,[m'] \,||\, Y, T, H}$$

RETRIEVE x (ATTRIBUTEERROR)

$$\frac{\text{LOOKUPOBJ}(P, H, m, x) = *}{P, Z \,||\,[m, x, \text{RETRIEVE}] \,||\, Y, T, H \longrightarrow^1 P, [\text{ALLOCATTRERROR}, \text{RAISE}], T, H}$$

Figure 4: Microcommands (cont.)

LITERAL ASSIGNMENT

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : x = \mathbb{Z} \qquad Y = [v, \textsc{Store}, \textsc{Wrap}, \textsc{Store}, x, \textsc{Bind}, \textsc{Advance}]}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

(TODO: make literal category (ints, str, bool, None) – TC)

NAME ASSIGNMENT

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : x_1 = x_2 \qquad Y = [x_2, \textsc{Lookup}, x_1, \textsc{Bind}, \textsc{Advance}]}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

LIST ASSIGNMENT

$$\frac{\begin{array}{c} S(\ell) = \ell : \overset{\star'}{\ell} : x = [x_1, \ldots, x_n] \\ Y = [(x_1, \textsc{LookUp}), \ldots, (x_n, \textsc{LookUp}), \textsc{List } n, \textsc{Store}, \textsc{Wrap}, \textsc{Store}, x, \textsc{Bind}, \textsc{Advance}] \end{array}}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

(Parentheses in Y group instructions together for convenience of reading. – TC)

TUPLE ASSIGNMENT

$$\frac{\begin{array}{c} S(\ell) = \ell : \overset{\star'}{\ell} : x = [x_1, \ldots, x_n] \\ Y = [(x_1, \textsc{LookUp}), \ldots, (x_n, \textsc{LookUp}), \textsc{Tuple } n, \textsc{Store}, \textsc{Wrap}, \textsc{Store}, x, \textsc{Bind}, \textsc{Advance}] \end{array}}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

FUNCTIONDEF ASSIGNMENT

$$\frac{\begin{array}{c} S(\ell) = \ell : \ell' : x = \texttt{def } (x_1, \ldots, x_n) = \{S'\} \qquad v = \langle \eta, \texttt{def } (x_1, \ldots, x_n) \rightarrow S' \rangle \\ Y = [v, \textsc{Store}, \textsc{Wrap}, \textsc{Store}, x, \textsc{Bind}, \textsc{Advance}] \end{array}}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

ATTRIBUTE ASSIGNMENT

$$\frac{\begin{array}{c} S(\ell) = \ell : \ell' : x = x_1.x_2 \\ Y = [x_1, \textsc{LookUp}, x_2, \textsc{Retrieve}, \textsc{Wrap}, \textsc{Store}, x, \textsc{Bind}] \end{array}}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

CALL ASSIGNMENT

$$\frac{\begin{array}{c} S(\ell) = \ell : \ell' : x = x_0(x_1, \ldots, x_n) \\ Y = [x_0, \textsc{LookUp}, \ldots, x_n, \textsc{LookUp}, \textsc{GetCall } n] \end{array}}{P, [\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

Figure 5: Operational Semantics: Assignment

$\textsc{Pass}$

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : \texttt{pass} \qquad Y = [\textsc{Advance}]}{P, [\,], [\langle \eta, \ell, S\rangle] \,\|\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S\rangle] \,\|\, T, H}$$

$\textsc{Return}$

$$\frac{\begin{array}{c} S(\ell) = \ell : \overset{\star'}{\ell} : \texttt{return } x \qquad T = [\langle \eta', \ell'', S'\rangle] \,\|\, T' \\ S(\ell'') = \ell'' : \overset{\star'''}{\ell} : x' = e \qquad Y = [x, \textsc{LookUp}, \textsc{Pop}, x', \textsc{Bind}, \textsc{Advance}] \end{array}}{P, [\,], [\langle \eta, \ell, S\rangle] \,\|\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S\rangle] \,\|\, T, H}$$

$\textsc{Goto}$

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : \texttt{goto } \ell'' \qquad Y = [\textsc{Goto } \ell'']}{P, [\,], [\langle \eta, \ell, S\rangle] \,\|\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S\rangle] \,\|\, T, H}$$

$\textsc{GotoIfNot}$

$$\frac{S(\ell) = \ell : \ell' : \texttt{goto } \ell'' \texttt{ if not } x \qquad Y = [x, \textsc{Lookup}, \textsc{Gotoifn } \ell'']}{P, [\,], [\langle \eta, \ell, S\rangle] \,\|\, T, H \longrightarrow^1 P, Y, [\langle \eta, \ell, S\rangle] \,\|\, T, H}$$

$\textsc{End of Function}$

$$\frac{t = \langle \eta', \ell, S'\rangle \qquad S(\ell) = \ell : \overset{\star'}{\ell} : x = e \qquad Y = [\textsc{Pop}, m_{\texttt{None}}, x, \textsc{Bind}, \textsc{Advance}]}{P, [\,], [\langle \eta, *, S\rangle, t] \,\|\, T, H \longrightarrow^1 P, Y, [\langle \eta, *, S\rangle, t] \,\|\, T, H}$$

$(m_{\texttt{None}}$ is a memory location reserved for $\texttt{None}. - TC)$

$\textsc{End of Program}$

$$\frac{T = [\langle \eta, *, S\rangle] Y = [\textsc{Pop}]}{P, [\,], T, H \longrightarrow^1 P, Y, T, H}$$

Figure 6: Operational Semantics: Flow

Definition 0.1.
$$\textsc{Lookup}(P, H, \eta, x) = \textit{(todo -- TC)}$$

Definition 0.2.
$$\textsc{LookupObj}(P, H, m, x) = \textit{(todo -- TC)}$$

Definition 0.3.
$$\textsc{GetObj}(H, m) = \begin{cases} B, & \textit{if } v = B \\ B = \star x_{value} \mapsto v, & \textit{otherwise} \end{cases}, H[m] = v \tag{1}$$

Definition 0.4. $H[m] = B, B[\star x_{value}] = v$
$$\textsc{GetCall}(H, m) = \begin{cases} v, & \textit{if } v = F \mid M \\ *, & \textit{otherwise} \end{cases} \tag{2}$$

Figure 7: Helper Functions