$$
\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,\dots] & \textit{stack} \\
t & ::= & \langle \eta, \overset{\star}{\ell}, S \rangle & \textit{stack frames} \\
S & ::= & [s,\dots] & \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 not } x & \textit{directives} \\
& & \mid \texttt{raise } x \mid \texttt{catch } x \mid \texttt{pass} & \\
B & ::= & \{x \mapsto m, \dots\} & \textit{bindings} \\
H & ::= & \{m \mapsto v, \dots\} & \textit{heap} \\
v & ::= & \mathbb{Z} \mid \mathbb{S} \mid [m,\dots] \mid (m,\dots) \mid B \mid F \mid M \mid * & \textit{values} \\
e & ::= & \mathbb{Z} \mid \mathbb{S} \mid \texttt{None} \mid x \mid \texttt{def } x(x,\dots) = \{S\} \mid x(x,\dots) \mid x.x \mid [x,\dots] \mid (x,\dots) & \textit{expressions} \\
Y & ::= & [y,\dots] & \textit{microcode stack} \\
Z & ::= & [z,\dots] & \textit{microcode literal stack} \\
y & ::= & \text{STORE} \mid \text{WRAP} \mid \text{BIND} \mid \text{LOOKUP} \mid \text{LIST } n \mid \text{TUPLE } n & \textit{microcode instructions} \\
& & \mid \text{ADVANCE} \mid \text{POP} \mid \text{PUSH } S \mid \text{RAISE} \mid \text{GOTO } \ell \mid \text{GOTOIFN } \ell & \\
& & \mid \text{CALL } n \mid \text{GETCALL } n \mid \text{CONVERT } n \mid \text{RETRIEVE} & \\
& & \mid \text{ALLOCNAMEERROR} \mid \text{ALLOCTYPEERROR} \mid \text{ALLOCATTRERROR} & \\
z & ::= & x \mid m \mid v & \textit{microcode literals} \\
\overset{\star}{m} & ::= & m \mid \eta \mid * & \textit{general memory locations} \\
\eta, m & ::= & \text{<ADDRESS>} & \textit{memory locations} \\
F & ::= & \langle \eta, \texttt{def } (x,\dots) \to S \rangle \mid \mathfrak{F} \mid \mathfrak{M} & \textit{general functions} \\
M & ::= & \langle m, F \rangle & \textit{general methods} \\
\mathfrak{F}, \mathfrak{M} & ::= & \mathfrak{CallFunc} \mid \mathfrak{Type} & \textit{magic functions} \\
n & & & \textit{integers}
\end{array}
$$

Figure 1: Expression Grammar

Definition 0.1. *Initialization*

$$
\begin{aligned}
H_{\text{INIT}} &= \{\eta_{\text{INIT}} \mapsto B_{\text{INIT}}, m_{None} \mapsto *, m_{AttrError} \mapsto \{\ \}, m_{FunType} \mapsto \{\ \}\} \\
B_{\text{INIT}} &= \{\text{ATTRIBUTEERROR} \mapsto m_{AttrError}, \text{FUNCTIONTYPE} \mapsto m_{FunType}\} \\
t_{\text{INIT}} &= \langle \ell_{\text{INIT}}, S_{\text{INIT}} \rangle \\
T_{\text{INIT}} &= [t_{\text{INIT}}] \\
\text{GETATTRIBUTE} &= \langle \eta, \textit{def } (o, a) \to \$1 = o.a; \rangle \\
\text{GETCALL} &= \langle \eta, \textit{def } (f) \to \rangle
\end{aligned}
$$

("if \$1 == None then return getattribute(o.__class__,a) else return \$1" – TC)
(todo: add builtin mappings $m \mapsto F$ – TC)

1

STORE v

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

WRAP m

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

BIND m TO x

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

ADVANCE

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

POP

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

PUSH S

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

LOOK UP x (BOUND)

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

LOOK UP x (NAMEERROR)

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

MAKE LIST

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

MAKE TUPLE

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

Figure 2: Microcommands

RAISE (NO EXCEPTION LABEL)

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

RAISE (CAUGHT)

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

GOTO ℓ

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

GOTOIFN ℓ (SUCCESS)

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

GOTOIFN ℓ (FAILURE)

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

CALL FUNCTION m

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

CALL FUNCTION (WRONG ARGS)

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

GET CALL m

$$\frac{v = H[m_0][\star x_{\text{value}}], \ v \text{ is of form } M, \mathfrak{M} \text{ or } \mathfrak{F} \qquad Y' = [v, m_1, \ldots, m_n]}{Z \,||\,[m_0, \ldots, m_n, \text{GETCALL } n] \,||\, Y, T, H \longrightarrow^1 Z \,||\, Y' \,||\, Y, T, H}$$

GET CALL (TYPEERROR)

$$\frac{v = H[m_0][\star x_{\text{value}}], \ v \text{ is not of form } M, \mathfrak{M} \text{ or } \mathfrak{F}}{Z \,||\,[m_0, \ldots, m_n, \text{GETCALL } n] \,||\, Y, T, H \longrightarrow^1 [\text{ALLOCTYPEERROR}, \text{RAISE}], T, H}$$

Figure 3: Microcommands (cont.)

CONVERT FUNCTION v

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

CONVERT METHOD v

$$\frac{v = \langle m_0, F \rangle \qquad v' = F}{Z \,||\,[v, m_1, \ldots, m_n, \text{CONVERT } n] \,||\, Y, T, H \longrightarrow^1 Z \,||\,[v', m_0, m_1, \ldots, m_n, \text{CALL } n+1] \,||\, Y, T, H}$$

RETRIEVE x

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

RETRIEVE x (ATTRIBUTEERROR)

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

Figure 4: Microcommands (cont.)

Figure 5: Magic Functions

NONE ASSIGNMENT

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : x = \texttt{None} \qquad Y = [m_{\texttt{None}}, x, \text{BIND}, \text{ADVANCE}]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

LITERAL ASSIGNMENT

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : x = e,\ e \text{ is of form } \mathbb{Z}, \mathbb{S} \qquad Y = [v, \text{STORE}, \text{WRAP}, \text{STORE}, x, \text{BIND}, \text{ADVANCE}]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

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

NAME ASSIGNMENT

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

LIST ASSIGNMENT

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : x = [x_1, \ldots, x_n] \qquad Y = [(x_1, \text{LOOKUP}), \ldots, (x_n, \text{LOOKUP}), \text{LIST } n, \text{STORE}, \text{WRAP}, \text{STORE}, x, \text{BIND}, \text{ADVANCE}]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

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

TUPLE ASSIGNMENT

$$\frac{S(\ell) = \ell : \overset{\star'}{\ell} : x = [x_1, \ldots, x_n] \qquad Y = [(x_1, \text{LOOKUP}), \ldots, (x_n, \text{LOOKUP}), \text{TUPLE } n, \text{STORE}, \text{WRAP}, \text{STORE}, x, \text{BIND}, \text{ADVANCE}]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

FUNCTIONDEF ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell' : x = \texttt{def } (x_1, \ldots, x_n) = \{S'\} \qquad v = \langle \eta, \texttt{def } (x_1, \ldots, x_n) \to S' \rangle \qquad Y = [v, \text{STORE}, \text{WRAP}, \text{STORE}, x, \text{BIND}, \text{ADVANCE}]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

ATTRIBUTE ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell' : x = x_1.x_2 \qquad Y = [x_1, \text{LOOKUP}, x_2, \text{RETRIEVE}, \text{WRAP}, \text{STORE}, x, \text{BIND}, \text{ADVANCE}]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

CALL ASSIGNMENT

$$\frac{S(\ell) = \ell : \ell' : x = x_0(x_1, \ldots, x_n) \qquad Y = [x_0, \text{LOOKUP}, \ldots, x_n, \text{LOOKUP}, \text{GETCALL } n, \text{CONVERT } n]}{[\,], [\langle \eta, \ell, S \rangle] \,||\, T, H \longrightarrow^1 Y, [\langle \eta, \ell, S \rangle] \,||\, T, H}$$

Figure 6: Operational Semantics: Assignment

$$\text{Pass}$$

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

$$\text{Return}$$

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

$$\text{Goto}$$

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

$$\text{GotoIfNot}$$

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

$$\text{End of Function}$$

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

*(m$_{\texttt{None}}$ is a memory location reserved for **None**. – TC)*

$$\text{End of Program}$$

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

Figure 7: Operational Semantics: Flow

Definition 0.2.

$$\textsc{Lookup}(H, \eta, x) = \textit{(todo – TC)}$$

Definition 0.3.

$$H[m] = v, B_{obj} = \{\star x_{value} \mapsto v, \texttt{\_\_getattribute\_\_} \mapsto \textsc{Getattribute}\}$$

$$\textsc{GetObj}(H, m) = \begin{cases} B, & \textit{if } v = B \\ B = B_{obj}[\texttt{\_\_class\_\_} \mapsto \star\textsc{funType}], & \textit{if } v = F \end{cases} \qquad (1)$$

(getobj takes (H,m,memory location of getattribute) – TC)

Figure 8: Helper Functions