x	::=	AlphaNumeric | $\star$AlphaNumeric	*variables*
$\overset{\star}{\ell}$	::=	$\ell \mid *$	*general labels*
ℓ			*labels*
T	::=	$[t, \ldots]$	*stack*
t	::=	$\overset{\star}{\ell} \times S$	*stack frames*
S	::=	$[s, \ldots]$	*programs*
s	::=	$\ell : \overset{\star}{\ell} : d$	*clauses*
d	::=	$x = e \mid$ `return` $x \mid$ `goto` $\ell \mid$ `goto` ℓ `if not` x	*directives*
		| `raise` $x \mid$ `catch` $x \mid$ `pass`	
B	::=	$\{x \mapsto m, \ldots\}$	*bindings*
H	::=	$\{m \mapsto v, \ldots\}$	*heap*
v	::=	$\mathbb{Z} \mid [m, \ldots] \mid (m, \ldots) \mid B \mid F \mid M \mid$ `undefined` | `None`	*values*
e	::=	$v \mid x \mid$ `def` $x(x, \ldots) = \{S\} \mid x(x, \ldots) \mid x.x \mid [x, \ldots] \mid (x, \ldots)$	*expressions*
Y	::=	$[y, \ldots]$	*microcode stack*
Z	::=	$[z, \ldots]$	*microcode literal stack*
y	::=	Store | Wrap | Bind | LookUp | List n | Tuple n	*microcode commands*
		| Advance | Pop | Raise | Goto ℓ | Gotoifn ℓ | AllocNameError	
z	::=	$x \mid m \mid v$	*microcode literals*
P	::=	$m \mapsto m$	*parental map*
$\overset{\star}{m}$	::=	$m \mid \eta \mid *$	*general memory locations*
η, m			*memory locations*
F	::=	$\langle m,$ `def` $(x) \to S \rangle \mid \mathfrak{F}$	*general functions*
M	::=	$\langle m, m,$ `def` $(x) \to S \rangle \mid \langle m, \mathfrak{M} \rangle$	*general methods*
$\mathfrak{F}$			*magic functions*
$\mathfrak{M}$			*magic methods*
n			*integers*

Figure 1: Expression Grammar

STORE v

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

WRAP m

$$\frac{v = \textsc{GetObj}(H, m)}{P, Z \,||[m, \textsc{Wrap}]\,||\, Y, T, H, \eta \longrightarrow^1 P, Z \,||[v]\,||\, Y, T, H, \eta}$$

BIND m TO x

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

ADVANCE

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

POP

$$\frac{\eta' = P[\eta]}{P, Z \,||[\textsc{Pop}]\,||\, Y, t \,||\, T, H, \eta \longrightarrow^1 P, Z \,||\, Y, T, H, \eta'}$$

LOOK UP x (BOUND)

$$\frac{\textsc{Lookup}(P, H, \eta, x) = m}{P, Z \,||[x, \textsc{LookUp}]\,||\, Y, T, H, \eta \longrightarrow^1 P, Z \,||[m]\,||\, Y, T, H, \eta}$$

LOOK UP x (NAMEERROR)

$$\frac{\textsc{Lookup}(P, H, \eta, x) = *}{P, Z \,||[x, \textsc{LookUp}]\,||\, Y, T, H, \eta \longrightarrow^1 P, [\textsc{AllocNameError}, \textsc{Raise}], T, H, \eta}$$

MAKE LIST

$$\frac{v = [m_1, \ldots, m_n]}{P, Z \,||[m_1, \ldots, m_n, \textsc{List}\ n]\,||\, Y, T, H, \eta \longrightarrow^1 P, Z \,||[v]\,||\, Y, T, H, \eta}$$

MAKE TUPLE

$$\frac{v = (m_1, \ldots, m_n)}{P, Z \,||[m_1, \ldots, m_n, \textsc{Tuple}\ n]\,||\, Y, T, H, \eta \longrightarrow^1 P, Z \,||[v]\,||\, Y, T, H, \eta}$$

RAISE (NO EXCEPTION LABEL)

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

Figure 2: Microcommands

RAISE (CAUGHT)

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

GOTO ℓ

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

GOTOIFN ℓ (SUCCESS)

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

GOTOIFN ℓ (FAILURE)

$$\frac{H[m] = \textsc{True}}{P, Z \,||[m, \textsc{Gotoifn}\ \ell]\,||\, Y, T, H, \eta \longrightarrow^1 P, Z\,||[\textsc{Advance}]\,||\, Y, T, H, \eta}$$

Figure 3: Microcommands (cont.)

Literal Assignment

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

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

Name Assignment

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

List Assignment

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

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

Tuple Assignment

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

Figure 4: Operational Semantics: Assignment

Pass

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

Return

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

Goto

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

GotoIfNot

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

Figure 5: Operational Semantics: Flow