– First step: Define the semantic domains
– Second Step: Define Big Step semantics

1 Semantic Domain - *Python*

Relação de Transição para Expressões:

$$s, b \vdash exp \Downarrow s', b', v'$$

Relação de Transição para Comandos:

$$b, s \vdash C \Downarrow b[C \mapsto b(C)], s'$$

Bindings:

$$(identifier \mapsto cell) \cup (identifier \mapsto closure).$$

Store:

$$cell \mapsto value$$

$$value = string\_literal \cup num\_literal \cup bool\_literal \cup list$$

Regras de atribuição
Atribuição de Elemento não lista e que não exista nos bindings:

$$\frac{s, b \vdash exp \Downarrow s', b', v'}{b, s \vdash identifier = exp \Downarrow b'[identifier \mapsto v], s'} \; identifier \notin dom(b), v \notin list$$

Atribuição de Elemento não lista e que exista nos bindings:

$$\frac{s, b \vdash exp \Downarrow s', b', v'}{b, s \vdash identifier = exp \Downarrow b'[b(identifier) \mapsto v], s'} \; identifier \in dom(b), v \notin list$$

Atribuição de Elemento lista e que não exista nos bindings:

$$\frac{s, b \vdash exp \Downarrow s', b', v' \qquad k := alloc(s', v)}{b, s \vdash identifier = exp \Downarrow b'[identifier \mapsto k], s'[k \mapsto v]} \; identifier \notin dom(b), v \in list$$

Atribuição de Elemento lisãa e que exista nos bindings:

$$\frac{s, b \vdash exp \Downarrow s', b', v'}{b, s \vdash identifier = exp \Downarrow b', s'[b'(identifier \mapsto v)]} \; identifier \in dom(b), v \in list$$

Atribuição em listas lista:

$$s, b \vdash exp \Downarrow s', b', v's, b \vdash exp \Downarrow s', b', v''s, b \vdash exp \Downarrow s', b', v'''$$

(list-atrib) $v' \in list, v'' \in num_literal$

$$\overline{b, s \vdash exp1[exp2] = exp3 \Downarrow b', s'[b'(exp1(exp2)) \mapsto exp3]}$$

Regras de desvio condicional:

$$\text{\textit{(Expression evaluated as True)}}\frac{s, b \vdash exp \Downarrow s', b', v'}{b, s \vdash if(exp) : C1 else : C2 \Downarrow b'[C1], s'[C1]} v = True$$

$$\text{\textit{(Expression evaluated as False)}}\frac{s, b \vdash exp \Downarrow s', b', v'}{b, s \vdash if(exp) : C1 else : C2 \Downarrow b'[C2], s'[C2]} v = False$$

For loop

$$\text{\textit{(For loop)}}\frac{s, b \vdash exp \Downarrow s', b', v'}{b, s \vdash foridentifierinexp : command \Downarrow b'[command], s'[command]} exp \mapsto k, k \mapsto v, v \in list$$

$$\textit{(string-add)}\ \frac{E1 \Downarrow string\_literal1 \qquad E2 \Downarrow string\_literal2}{E1 + E2 \Downarrow string\_literal3}\ \textit{string\_literal3 = string\_literal1.string\_literal2}$$

$$\textit{(num-add)}\frac{E1 \Downarrow num\_literal1 \qquad E2 \Downarrow num\_literal2}{E1 + E2 \Downarrow num\_literal3}\textit{num\_literal3 = num\_literal1 + num\_literal2}$$

$$\textit{(num-sub)}\frac{E1 \Downarrow num\_literal1 \qquad E2 \Downarrow num\_literal2}{E1 - E2 \Downarrow num\_literal3}\textit{num\_literal3 = num\_literal1 - num\_literal2}$$

$$\textit{(num-mult)}\frac{E1 \Downarrow num\_literal1 \qquad E2 \Downarrow num\_literal2}{E1 * E2 \Downarrow num\_literal3}\textit{num\_literal3 = num\_literal1 * num\_literal2}$$

$$\textit{(num-div)}\frac{E1 \Downarrow num\_literal1 \qquad E2 \Downarrow num\_literal2}{E1 * E2 \Downarrow num\_literal3}\textit{num\_literal3 = num\_literal1 / num\_literal2}$$

$$\textit{(variable-set)}\frac{s}{identifier = value}\textit{(s[identifier] = cell, s[cell]=value, s', b')}$$

$$\textit{(if-clause-true)}\frac{if(exp) : command1else : command2 \qquad exp \Downarrow True}{command1}\textit{(s', b')}$$

$$\textit{(if-clause-false)}\frac{if(exp) : command1else : command2 \qquad exp \Downarrow False}{command2}\textit{(s', b')}$$

$$\textit{(access-list)}\frac{s[list\_identifier \mapsto cell] \qquad cell \mapsto value \qquad exp \Downarrow num\_literal}{list\_identifier[exp]}\textit{(value, s, b)}$$

Esquema da mquina:

$$\langle G, \rho, I, O, M, T_E, N_{MT}\rangle \ \triangleright\ \langle G', \rho', I', O', M', T'_E, N'_{MT}\rangle$$

Operation Call:

$$\frac{\begin{array}{c} indegree(w) = 0 \qquad \overline{a} = Eval_\rho(\overline{E}) \qquad A'_c = A_c - \{w \mapsto v \mid v \in V\} \qquad A'_d = A_d - \{w \mapsto v \mid v \in V\} \\ G = \langle V[w : S!\overline{E},\ w' : S?\overline{X}], A_c[w \mapsto w'], A_d\rangle \qquad G' = \langle V[w' : S?\overline{X}], A'_c, A'_d\rangle \end{array}}{\langle G, \rho, I, O, M, T_E, N_{MT}\rangle \ \triangleright\ \langle G', \rho, I, O[\langle S, w', \overline{a}\rangle], M[\langle w : S!\overline{E},\ w', now(), N_{MT}\rangle], T_E, N_{MT}\rangle}$$