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

1 Semantic Domain - *Python*

$$identifier \mapsto cell$$

$$identifier \mapsto closure.$$

$$cell \mapsto value$$

$$value \Downarrow string\_literal$$

$$value \Downarrow num\_literal$$

$$value \Downarrow bool\_literal$$

$$value \Downarrow list$$

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

$$b, s \mapsto C = B \Downarrow b[C \mapsto b(B)], s$$

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

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

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

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

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

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

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

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

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

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:

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