

- 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:

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

Atribuição de Elemento não lista e que exista nos bindings:

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

Atribuição de Elemento lista e que não exista nos bindings:

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

Atribuição de Elemento lista e que exista nos bindings:

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

---

Atribuição em listas:

$$\frac{\begin{array}{l} s, b \vdash exp \Downarrow s', b', v' \quad s, b \vdash exp \Downarrow s', b', v'' \quad s, b \vdash exp \Downarrow s', b', v''' \\ (list-atrib) \end{array} \quad v' \in list, v'' \in num\_literal}{b, s \vdash exp1[exp2] = exp3 \Downarrow b', s'[b'(exp1[exp2]) \mapsto exp3]}$$

Regras de desvio condicional:

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

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

For loop

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

$$(string-add) \frac{s, b \vdash exp \Downarrow s', b', v' \quad s, b \vdash exp \Downarrow s', b', v''}{s, b \vdash exp + exp \Downarrow s', b', v'''} (v''' = v'.v''), v' \in string\_literal, v'' \in string\_literal$$

$$(num-add) \frac{s, b \vdash exp \Downarrow s', b', v' \quad s, b \vdash exp \Downarrow s', b', v''}{s, b \vdash exp + exp \Downarrow s', b', v'''} (v''' = v' + v''), v' \in num\_literal, v'' \in num\_literal$$

$$(num-sub) \frac{s, b \vdash exp \Downarrow s', b', v' \quad s, b \vdash exp \Downarrow s', b', v''}{s, b \vdash exp - exp \Downarrow s', b', v'''} (v''' = v' - v''), v' \in num\_literal, v'' \in num\_literal$$

$$(num-mult) \frac{s, b \vdash exp \Downarrow s', b', v' \quad s, b \vdash exp \Downarrow s', b', v''}{s, b \vdash exp * exp \Downarrow s', b', v'''} (v''' = v' * v''), v' \in num\_literal, v'' \in num\_literal$$

$$(num-div) \frac{s, b \vdash exp \Downarrow s', b', v' \quad s, b \vdash exp \Downarrow s', b', v''}{s, b \vdash exp/exp \Downarrow s', b', v'''} (v''' = v'/v''), v' \in num\_literal, v'' \in num\_literal$$