

## Semântica Operacional Estrutural (small-step)

### Semântica operacional estrutural para **While**

$$[\text{skip}_{\text{sos}}] \quad \overline{\langle \text{skip}, s \rangle \Rightarrow s}$$

$$[\text{ass}_{\text{sos}}] \quad \overline{\langle x := a, s \rangle \Rightarrow s [x \mapsto \mathcal{A}[\![a]\!] s]}$$

$$[\text{comp}^1_{\text{sos}}] \quad \frac{\langle C_1, s \rangle \Rightarrow \langle C'_1, s' \rangle}{\langle C_1 ; C_2, s \rangle \Rightarrow \langle C'_1 ; C_2, s' \rangle}$$

$$[\text{comp}^2_{\text{sos}}] \quad \frac{\langle C_1, s \rangle \Rightarrow s'}{\langle C_1 ; C_2, s \rangle \Rightarrow \langle C_2, s' \rangle}$$

## Semântica operacional estrutural

- O ênfase desta semântica é nos **passos individuais** de execução de um programa.
- A relação de transição tem a forma  $\langle C, s \rangle \Rightarrow \gamma$  e representa o **primeiro passo** de execução do programa  $C$  no estado  $s$ .
- Em  $\langle C, s \rangle \Rightarrow \gamma$ , se  $\gamma$  é da forma
  - ▶  $\langle C', s' \rangle$ , então a execução ainda não está completa (é uma configuração intermédia).
  - ▶  $s'$ , então a execução terminou e o estado final é  $s'$ .
- Uma configuração  $\langle C, s \rangle$  diz-se **bloqueada** se não existir nenhuma configuração  $\gamma$  tal que  $\langle C, s \rangle \Rightarrow \gamma$ .

### Semântica operacional estrutural para **While**

$$[\text{if}^{\text{tt}}_{\text{sos}}] \quad \overline{\langle \text{if } b \text{ then } C_1 \text{ else } C_2, s \rangle \Rightarrow \langle C_1, s \rangle} \quad \text{se } \mathcal{B}[\![b]\!] s = \text{tt}$$

$$[\text{if}^{\text{ff}}_{\text{sos}}] \quad \overline{\langle \text{if } b \text{ then } C_1 \text{ else } C_2, s \rangle \Rightarrow \langle C_2, s \rangle} \quad \text{se } \mathcal{B}[\![b]\!] s = \text{ff}$$

$$[\text{while}_{\text{sos}}] \quad \overline{\langle \text{while } b \text{ do } C, s \rangle \Rightarrow \langle \text{if } b \text{ then } (C ; \text{while } b \text{ do } C) \text{ else skip}, s \rangle}$$

## Sequência de derivação

### Sequência de derivação (ou de execução)

- Uma *sequência de derivação* de um programa  $C$  num estado  $s$  pode ser uma de duas coisas:
  - ▶ uma **sequência finita** de configurações  $\gamma_0, \gamma_1, \gamma_2, \dots, \gamma_k$  tais que  $\gamma_0 = \langle C, s \rangle$ ,  $\gamma_i \Rightarrow \gamma_{i+1}$  com  $0 \leq i < k$ ,  $k \geq 0$  e  $\gamma_k$  é uma configuração terminal ou bloqueada;
  - ▶ uma **sequência infinita** de configurações  $\gamma_0, \gamma_1, \gamma_2, \dots$  tais que  $\gamma_0 = \langle C, s \rangle$ ,  $\gamma_i \Rightarrow \gamma_{i+1}$  para  $0 \leq i$ .
- $\gamma \Rightarrow^k \gamma'$  indica  $k$  passos de execução.
- $\gamma \Rightarrow^* \gamma'$  indica um número finito de passos de execução.

## Árvore de derivação (execução)

Por exemplo, no primeiro passo

$$\langle \{z := y; x := y\}; y := z, s \rangle \Rightarrow \langle x := y; y := z, s[z \mapsto 5] \rangle$$

árvore de derivação é

$$\frac{\frac{\frac{\langle z := y, s \rangle \Rightarrow s[z \mapsto 5] \text{ [ass}_{\text{sos}]\text{]}} \text{ [comp}_{\text{sos}}^2\text{]}}{\langle z := y; x := y, s \rangle \Rightarrow \langle x := y, s[z \mapsto 5] \rangle} \text{ [comp}_{\text{sos}}^1\text{]}}{\langle \{z := y; x := y\}; y := z, s \rangle \Rightarrow \langle x := y; y := z, s[z \mapsto 5] \rangle} \text{ [comp}_{\text{sos}}^1\text{]}$$

## Sequência de derivação (execução)

### Exemplo

Seja  $s$  o estado que mapeia todas as variáveis em 0, excepto  $x$  e  $y$ . Nestes casos:  $s x = 5$  e  $s y = 7$ .

Para o programa  $\{z := x; x := y\}; y := z$  temos a sequência de derivação

$$\begin{aligned} \langle \{z := x; x := y\}; y := z, s \rangle &\Rightarrow \langle x := y; y := z, s[z \mapsto 5] \rangle \\ &\Rightarrow \langle y := z, s[z \mapsto 5][x \mapsto 7] \rangle \\ &\Rightarrow s[z \mapsto 5][x \mapsto 7][y \mapsto 5] \end{aligned}$$

Note que para **cada passo** da sequência temos uma **árvore de derivação** que justifica esse passo.

## Sequência de derivação

A execução de um programa  $C$  num estado  $s$

- **termina** se e só se existir uma sequência de derivação finita começada em  $\langle C, s \rangle$
- **diverge** (ou **entra em ciclo**) se e só se existir uma sequência de derivação infinita começada em  $\langle C, s \rangle$
- **termina com sucesso** se e só se  $\langle C, s \rangle \Rightarrow^* s'$

Na linguagem **While** não temos configurações que bloqueiem, mas veremos extensões em que isso pode acontecer.

### Exercício

Determine um estado para o qual a sequência de derivação do programa  $z := 0; \text{while } y \leq x \text{ do } \{z := z + 1; x := x - y\}$  é finita e outro para o qual é infinita.

## Indução no comprimento das sequências

Para provar uma propriedade para todas as sequências de derivação:

- **Caso de base**

Provar que a propriedade se verifica para sequências de derivação de comprimento 0.

- **Passo indutivo**

Assumir que a propriedade se verifica para todas as sequências de derivação de comprimento até  $k$  (é a **hipótese de indução**) e provar que a propriedade também se verifica para as sequências de derivação de comprimento  $k + 1$ .

## Determinismo

A semântica operacional estrutural aqui apresentada é **determinista**.

### Teorema

Para quaisquer  $C, s, \gamma, \gamma'$ ,

se  $\langle C, s \rangle \Rightarrow \gamma$  e  $\langle C, s \rangle \Rightarrow \gamma'$  então  $\gamma = \gamma'$ .

**Prova:** por indução na estrutura da derivação de  $\langle C, s \rangle \Rightarrow \gamma$ .

## Propriedades

### Lema

Se  $\langle C_1, s \rangle \Rightarrow^k s'$  então  $\langle C_1; C_2, s \rangle \Rightarrow^k \langle C_2, s' \rangle$ .

**Prova:** Por indução no comprimento da sequência  $\langle C_1, s \rangle \Rightarrow^k s'$ , ou seja, em  $k$ .

### Lema

Se  $\langle C_1; C_2, s \rangle \Rightarrow^k s''$  então existe um estado  $s'$  e números naturais  $k_1$  e  $k_2$  tais que:  $\langle C_1, s \rangle \Rightarrow^{k_1} s'$ ,  $\langle C_2, s' \rangle \Rightarrow^{k_2} s''$  e  $k_1 + k_2 = k$ .

**Prova:** Por indução no comprimento da sequência  $\langle C_1; C_2, s \rangle \Rightarrow^k s''$ .

## Equivalência semântica

### Equivalência semântica

Dois programas,  $C_1$  e  $C_2$ , são **semanticamente equivalentes** se, para qualquer estado  $s$ ,

- $\langle C_1, s \rangle \Rightarrow^* \gamma$  sse  $\langle C_2, s \rangle \Rightarrow^* \gamma$ , caso  $\gamma$  seja uma configuração terminal ou bloqueada.
- houver uma sequência de derivação começada em  $\langle C_1, s \rangle$  divergente se e só se houver uma começada em  $\langle C_2, s \rangle$  também divergente.

### Exercício

Prove que os programas seguintes são semanticamente equivalentes.

- 1 while  $b$  do  $C$
- 2 if  $b$  then  $\{C; \text{while } b \text{ do } C\}$  else skip