

## Semântica Natural *versus* Semântica Operacional Estrutural

72

### Equivalência entre semânticas

**Teorema:** Para qualquer programa  $S$  da linguagem **While**,

$$\mathcal{S}_{\text{ns}}[S] = \mathcal{S}_{\text{sos}}[S].$$

**Prova:** Consequência directa dos lemas anteriores.

74

### Equivalência entre semânticas

**Lema:** Para qualquer programa  $S$  e estados  $s$  e  $s'$ ,

se  $\langle S, s \rangle \rightarrow s'$  então  $\langle S, s \rangle \Rightarrow^* s'$ .

**Prova:** Por indução na estrutura da derivação  $\langle S, s \rangle \rightarrow s'$ .

**Lema:** Para qualquer programa  $S$ , estados  $s$  e  $s'$ , e número natural  $k$ ,

se  $\langle S, s \rangle \Rightarrow^k s'$  então  $\langle S, s \rangle \rightarrow s'$

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

73

### Extensões à linguagem **While**

- Para ilustrar o poder e as fraquezas das duas abordagens à semântica operacional vamos considerar as seguintes pequenas extensões à linguagem **While**:
  - um comando **abort**
  - execução não determinista de dois comandos
  - execução paralela de dois comandos

75

## While + abort

```
S ::= x := a | skip | S1 ; S2 | if b then S1 else S2
    | while b do S | abort
```

O comando **abort** pára a execução do programa.

Note que **abort**

- é diferente de **while true** do **skip**
- é diferente de **skip**

76

## While + abort

- A semântica natural é definida pelo mesmo conjunto de regras que define a semântica natural da linguagem **While**.
- A semântica operacional estrutural é definida pelo mesmo conjunto de regras que define a semântica operacional estrutural da linguagem **While**.
- As configurações  $\langle \text{abort}, s \rangle$  são bloqueadas.

77

## While + abort

### Semântica operacional estrutural

- **abort** e **skip** não são semanticamente equivalentes
- **abort** e **while true** do **skip** não são semanticamente equivalentes

### Semântica natural

- **abort** e **skip** não são semanticamente equivalentes
- **abort** e **while true** do **skip** são semanticamente equivalentes

**Porquê?**

78

## While + abort

- Numa **semântica natural** não conseguimos distinguir entre o programa *entrar em ciclo* ou *terminar abruptamente*.
- Numa **semântica operacional estrutural**
  - o programa *entrar em ciclo* corresponde a uma sequência de derivação infinita, enquanto
  - uma *terminação abrupta* corresponde a uma sequência de derivação finita com uma configuração bloqueada no fim.
- Se considerarmos um **estado de erro** e modelarmos a terminação abrupta por uma transição para esse estado de erro (incluído nas configurações terminais), então conseguimos distinguir entre os três comandos em ambas as semânticas.

79

## While + or

$S ::= x := a \mid \text{skip} \mid S_1 ; S_2 \mid \text{if } b \text{ then } S_1 \text{ else } S_2$   
|  $\text{while } b \text{ do } S \mid S_1 \text{ or } S_2$

Em  $S_1 \text{ or } S_2$  podemos escolher entre executar  $S_1$  ou  $S_2$  **não deterministicamente**.

No estado após executar

$x := 1 \text{ or } (x := 2; x := x + 2)$

o valor de  $x$  tanto pode ser 1 como 4.

80

## While + or

**Semântica natural:** acrescentar as seguinte regras

$$[\text{or}_{\text{ns}}^1] \quad \frac{\langle S_1, s \rangle \rightarrow s'}{\langle S_1 \text{ or } S_2, s \rangle \rightarrow s'}$$

$$[\text{or}_{\text{ns}}^2] \quad \frac{\langle S_2, s \rangle \rightarrow s'}{\langle S_1 \text{ or } S_2, s \rangle \rightarrow s'}$$

**Exercício:** Indique as árvores de derivação correspondentes às configurações

e  $\langle x := 1 \text{ or } (x := 2; x := x+2), s \rangle$

$\langle (\text{while true do skip}) \text{ or } (x := 2; x := x+2), s \rangle$

81

## While + or

**Semântica operacional estrutural:** acrescentar as seguinte regras

$$[\text{or}_{\text{sos}}^1] \quad \langle S_1 \text{ or } S_2, s \rangle \Rightarrow \langle S_1, s \rangle$$

$$[\text{or}_{\text{sos}}^2] \quad \langle S_1 \text{ or } S_2, s \rangle \Rightarrow \langle S_2, s \rangle$$

**Exercício:** Indique as sequências de derivação correspondentes às configurações

e  $\langle x := 1 \text{ or } (x := 2; x := x+2), s \rangle$

$\langle (\text{while true do skip}) \text{ or } (x := 2; x := x+2), s \rangle$

82

## While + or

- Numa **semântica natural** o não determinismo suprime a divergência.
- Numa **semântica operacional estrutural** a divergência não é suprimida pelo não determinismo.

83

## While + par

$S ::= x := a \mid \text{skip} \mid S_1 ; S_2 \mid \text{if } b \text{ then } S_1 \text{ else } S_2$   
|  $\text{while } b \text{ do } S \mid S_1 \text{ par } S_2$

Ambos os comando de  $S_1 \text{ par } S_2$  executam *paralelamente de forma intercalada* (*interleaving*).

A execução de  $x := 1 \text{ par } (x := 2; x := x+2)$   
pode dar três resultados diferentes para  $x$ : 4, 1 e 3.

84

## While + par

**Semântica operacional estrutural:** acrescentar as seguinte regras

$$\begin{array}{c} [\text{par}_{\text{sos}}^1] \frac{\langle S_1, s \rangle \Rightarrow \langle S'_1, s' \rangle}{\langle S_1 \text{ par } S_2, s \rangle \Rightarrow \langle S'_1 \text{ par } S_2, s' \rangle} \quad [\text{par}_{\text{sos}}^3] \frac{\langle S_2, s \rangle \Rightarrow \langle S'_2, s' \rangle}{\langle S_1 \text{ par } S_2, s \rangle \Rightarrow \langle S_1 \text{ par } S'_2, s' \rangle} \\ [\text{par}_{\text{sos}}^2] \frac{\langle S_1, s \rangle \Rightarrow s'}{\langle S_1 \text{ par } S_2, s \rangle \Rightarrow \langle S_2, s' \rangle} \quad [\text{par}_{\text{sos}}^4] \frac{\langle S_2, s \rangle \Rightarrow s'}{\langle S_1 \text{ par } S_2, s \rangle \Rightarrow \langle S_1, s' \rangle} \end{array}$$

**Exercício:** Indique as sequências de derivação correspondentes à configuração

$\langle x := 1 \text{ par } (x := 2; x := x+2), s \rangle$

85

## While + par

**Semântica natural:** a tentativa de acrescentar as seguinte regras

$$\frac{\langle S_1, s \rangle \rightarrow s', \langle S_2, s' \rangle \rightarrow s''}{\langle S_1 \text{ par } S_2, s \rangle \rightarrow s''}$$
$$\frac{\langle S_2, s \rangle \rightarrow s', \langle S_1, s' \rangle \rightarrow s''}{\langle S_1 \text{ par } S_2, s \rangle \rightarrow s''}$$

**não funciona!**

Não conseguimos modelar o intercalar da execução de  $S_1$  e  $S_2$ .

Os passos de execução são demasiado grandes (*big-step!*)

86

## While + par

- Numa **semântica natural** a execução dos constituintes imediatos dos comandos é uma *entidade atómica*, o que não permite modelar o entrelaçamento das computações.
- Numa **semântica operacional estrutural** o foco é nos passos individuais de execução, o que permite facilmente modelar o entrelaçamento das computações.

87

## While + par

**Exercício:** Considere a extensão **While+par+protect** que para além da construção **par** acrescenta o comando

**protect S end**

A ideia é que o comando **S** tem que ser executado como uma entidade atómica (i.e. não pode ser intercalado com outras computações).

Estenda a semântica operacional estrutural de forma a contemplar esta construção.

Será possível usar uma semântica natural para exprimir a nova linguagem **While+par+protect** ?