

# Fundamentele limbajelor de programare

## Semantica execuției programelor imperative

---

prezentare de Traian Florin Șerbănuță și Andrei Sipoș

Facultatea de Matematică și Informatică, DL Info

Anul II, Semestrul II, 2025/2026

# Feluri de a da semantica

- Limbaj de programare: sintaxă și semantică
- Feluri de semantică
  - Limbaj natural — descriere textuală a efectelor
  - Operațională — asocierea unei demonstrații a execuției
  - Axiomatică — descrierea efectelor unei instrucțiuni folosind logică
  - Denotațională — prin asocierea unui obiect matematic (denotație)
  - Statică — asocierea unui sistem de tipuri care exclude programe eronate

# Limbajul IMP

IMP este un limbaj IMPerativ foarte simplu.

## Ce conține

- Expresii
  - Aritmetice
  - Booleene
- Blocuri de instrucțiuni
  - De atribuire
  - Condiționale
  - De ciclare

```
x := 10;  
sum := 0;  
while (0 <= x) do (  
    sum := sum + x;  
    x := x + -1  
)
```

## Ce nu conține

- Expresii cu efecte laterale
- Proceduri și funcții
- Schimbări abrupte de control

# Sintaxă formală

## Sintaxa BNF a limbajului IMP

$$E ::= \mathbb{Z} \mid L \\ \mid E + E \mid E - E \mid E * E$$
$$B ::= \text{true} \mid \text{false} \\ \mid E \leq E \mid E = E \\ \mid \neg B \mid B \wedge B \mid B \vee B$$
$$C ::= \text{skip} \\ \mid C ; C \\ \mid L := E \\ \mid \text{if } B \text{ then } C \text{ else } C \\ \mid \text{while } B \text{ do } C \\ \mid ( C )$$

# Semantică operațională

## Plan

- Instrumente de lucru
  - Sintaxă, memorie, configurații
  - Reguli de deducție și arbori de derivare
- Semantica evaluării
  - semantică naturală, într-un pas mare (big-step)
- Semantica tranzițională
  - Semantica operațională structurală, a pașilor mici (small-step)

# Starea execuției

Fie  $L$  mulțimea locațiilor de memorie. Starea execuției unui program IMP la un moment dat este dată de valorile deținute în acel moment de locațiile de memorie. **Matematic:** o funcție  $\sigma : L \rightarrow Int$ .

## Notatii

- Descrierea funcției prin enumerare:  $\sigma = n \mapsto 10, sum \mapsto 0$   
Convenție: toate valorile neenumerate sunt 0.
- Funcția nulă  $\mathbf{0}$ , va întoarce 0 pentru orice locație.
- Suprascrierea valorii unei variabile:

$$\sigma_{x \mapsto v}(y) = \begin{cases} \sigma(y), & \text{dacă } y \neq x \\ v, & \text{dacă } y = x \end{cases}$$

# Semantica Evaluării

- Introdusă în 1987 de Gilles Kahn sub numele de „semantică naturală”
- Denumiri alternative: „semantică relațională”, „semantica big-step”
- Relaționează fragmente de program într-o stare cu valoarea corespunzătoare evaluării lor în acea stare
  - Expresiile aritmetice se evaluează la întregi:  $\langle a, \sigma \rangle \Downarrow \langle i \rangle$
  - Expresiile Booleene se evaluează la *true/false*:  $\langle b, \sigma \rangle \Downarrow \langle t \rangle$
  - Instrucțiunile se evaluează la stări:  $\langle s, \sigma \rangle \Downarrow \langle \sigma' \rangle$
- Valoarea este obținută într-un **singur pas (mare)**
- Reguli structurale, având ca premise secvențe corespunzătoare subtermenilor

## Exemple

- $\langle 3 + x, (x \mapsto 5, y \mapsto 7) \rangle \Downarrow \langle 8 \rangle$
- $\langle x := 3 + y, (x \mapsto 5, y \mapsto 7) \rangle \Downarrow \langle x \mapsto 10, y \mapsto 7 \rangle$

# Semantica Evaluării pentru IMP

## Expresii aritmetice

$$(NUM) \langle i, \sigma \rangle \Downarrow \langle i \rangle$$

$$(ID) \langle x, \sigma \rangle \Downarrow \langle i \rangle \text{ dacă } i = \sigma(x)$$

$$(ADD) \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 + a_2, \sigma \rangle \Downarrow \langle i \rangle} \text{ dacă } i = i_1 +_{Int} i_2$$

$$(SUB) \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 - a_2, \sigma \rangle \Downarrow \langle i \rangle} \text{ dacă } i = i_1 -_{Int} i_2$$

$$(MUL) \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 * a_2, \sigma \rangle \Downarrow \langle i \rangle} \text{ dacă } i = i_1 *_{Int} i_2$$



# Semantica Evaluării pentru IMP

## Operatori de comparare

$$(\text{LEQ-TRUE}) \quad \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 \leq a_2, \sigma \rangle \Downarrow \langle \text{true} \rangle} \quad \text{dacă } i_1 \leq_{\text{Int}} i_2$$

$$(\text{LEQ-FALSE}) \quad \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 \leq a_2, \sigma \rangle \Downarrow \langle \text{false} \rangle} \quad \text{dacă } i_1 >_{\text{Int}} i_2$$

$$(\text{EQ-TRUE}) \quad \frac{\langle a_1, \sigma \rangle \Downarrow \langle i \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i \rangle}{\langle a_1 = a_2, \sigma \rangle \Downarrow \langle \text{true} \rangle}$$

$$(\text{EQ-FALSE}) \quad \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 = a_2, \sigma \rangle \Downarrow \langle \text{false} \rangle} \quad \text{dacă } i_1 \neq_{\text{Int}} i_2$$

# Semantica Evaluării pentru IMP

## Operatori logici

$$(\text{BOOL}) \quad \langle t, \sigma \rangle \Downarrow \langle t \rangle$$

$$(\text{NOT-TRUE}) \quad \frac{\langle b, \sigma \rangle \Downarrow \langle \text{false} \rangle}{\langle \neg b, \sigma \rangle \Downarrow \langle \text{true} \rangle}$$

$$(\text{NOT-FALSE}) \quad \frac{\langle b, \sigma \rangle \Downarrow \langle \text{true} \rangle}{\langle \neg b, \sigma \rangle \Downarrow \langle \text{false} \rangle}$$

$$(\text{AND-TRUE}) \quad \frac{\langle b_1, \sigma \rangle \Downarrow \langle \text{true} \rangle \quad \langle b_2, \sigma \rangle \Downarrow \langle t \rangle}{\langle b_1 \wedge b_2, \sigma \rangle \Downarrow \langle t \rangle}$$

$$(\text{AND-FALSE}) \quad \frac{\langle b_1, \sigma \rangle \Downarrow \langle \text{false} \rangle}{\langle b_1 \wedge b_2, \sigma \rangle \Downarrow \langle \text{false} \rangle}$$

$$(\text{OR-TRUE}) \quad \frac{\langle b_1, \sigma \rangle \Downarrow \langle \text{true} \rangle}{\langle b_1 \vee b_2, \sigma \rangle \Downarrow \langle \text{true} \rangle}$$

$$(\text{OR-FALSE}) \quad \frac{\langle b_1, \sigma \rangle \Downarrow \langle \text{false} \rangle \quad \langle b_2, \sigma \rangle \Downarrow \langle t \rangle}{\langle b_1 \vee b_2, \sigma \rangle \Downarrow \langle t \rangle}$$

# Semantica Evaluării pentru IMP

## Instrucțiuni simple

$$(\text{SKIP}) \quad \langle \text{skip}, \sigma \rangle \Downarrow \langle \sigma \rangle$$

$$(\text{ASGN}) \quad \frac{\langle a, \sigma \rangle \Downarrow \langle i \rangle}{\langle x := a, \sigma \rangle \Downarrow \langle \sigma' \rangle} \quad \text{dacă } \sigma' = \sigma_{x \mapsto i}$$

$$(\text{SEQ}) \quad \frac{\langle s_1, \sigma \rangle \Downarrow \langle \sigma' \rangle \quad \langle s_2, \sigma' \rangle \Downarrow \langle \sigma'' \rangle}{\langle s_1 ; s_2, \sigma \rangle \Downarrow \langle \sigma'' \rangle}$$

$$(\text{BLOCK}) \quad \frac{\langle s, \sigma \rangle \Downarrow \langle \sigma' \rangle}{\langle (s), \sigma \rangle \Downarrow \langle \sigma' \rangle}$$

# Semantica Evaluării pentru IMP

## Blocuri și instrucțiuni de ciclare

$$(IF-TRUE) \frac{\langle b, \sigma \rangle \Downarrow \langle true \rangle \quad \langle bl_1, \sigma \rangle \Downarrow \langle \sigma_1 \rangle}{\langle \text{if } b \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle \Downarrow \langle \sigma_1 \rangle}$$

$$(IF-FALSE) \frac{\langle b, \sigma \rangle \Downarrow \langle false \rangle \quad \langle bl_2, \sigma \rangle \Downarrow \langle \sigma_2 \rangle}{\langle \text{if } b \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle \Downarrow \langle \sigma_2 \rangle}$$

$$(WHILE-TRUE) \frac{\langle b, \sigma \rangle \Downarrow \langle true \rangle \quad \langle bl, \sigma \rangle \Downarrow \langle \sigma' \rangle \quad \langle \text{while } b \text{ do } bl, \sigma' \rangle \Downarrow \langle \sigma'' \rangle}{\langle \text{while } b \text{ do } bl, \sigma \rangle \Downarrow \langle \sigma'' \rangle}$$

$$(WHILE-FALSE) \frac{\langle b, \sigma \rangle \Downarrow \langle false \rangle}{\langle \text{while } b \text{ do } bl, \sigma \rangle \Downarrow \langle \sigma \rangle}$$

# Semantica Evaluării pentru IMP

Arbori de derivare

$$(\text{SEQ}) \frac{}{\langle a := 3; a := a + 4, \mathbf{0} \rangle \Downarrow \langle ? \rangle}$$

# Semantica Evaluării pentru IMP

## Arbori de derivare

$$\text{(SEQ)} \frac{\text{(ASGN)} \overline{\langle a := 3, \mathbf{0} \rangle \Downarrow \langle ? \rangle} \quad \text{(ASGN)} \overline{\langle a := a + 4, ? \rangle \Downarrow \langle ? \rangle}}{\langle a := 3; a := a + 4, \mathbf{0} \rangle \Downarrow \langle ? \rangle}$$

# Semantica Evaluării pentru IMP

## Arbori de derivare

$$\text{(SEQ)} \frac{\text{(ASGN)} \frac{\text{(NUM)} \langle 3, \mathbf{0} \rangle \Downarrow \langle 3 \rangle}{\langle a := 3, \mathbf{0} \rangle \Downarrow \langle a \mapsto 3 \rangle} \quad \text{(ASGN)} \frac{\dots}{\langle a := a + 4, a \mapsto 3 \rangle \Downarrow \langle a \mapsto ?? \rangle}}{\langle a := 3; a := a + 4, \mathbf{0} \rangle \Downarrow \langle a \mapsto ?? \rangle}$$

$$\text{(ASGN)} \frac{\text{(ADD)} \overline{\langle a + 4, a \mapsto 3 \rangle \Downarrow \langle ?? \rangle}}{\langle a := a + 4, a \mapsto 3 \rangle \Downarrow \langle a \mapsto ?? \rangle}$$

# Semantica Evaluării pentru IMP

## Arbori de derivare

$$\text{(SEQ)} \frac{\text{(ASGN)} \frac{\text{(NUM)} \langle 3, \mathbf{0} \rangle \Downarrow \langle 3 \rangle}{\langle a := 3, \mathbf{0} \rangle \Downarrow \langle a \mapsto 3 \rangle} \quad \text{(ASGN)} \frac{\dots}{\langle a := a + 4, a \mapsto 3 \rangle \Downarrow \langle a \mapsto 7 \rangle}}{\langle a := 3; a := a + 4, \mathbf{0} \rangle \Downarrow \langle a \mapsto 7 \rangle}$$

$$\text{(ASGN)} \frac{\text{(ADD)} \frac{\text{(ID)} \langle a, a \mapsto 3 \rangle \Downarrow \langle 3 \rangle \quad \text{(NUM)} \langle 4, a \mapsto 3 \rangle \Downarrow \langle 4 \rangle}{\langle a + 4, a \mapsto 3 \rangle \Downarrow \langle 7 \rangle}}{\langle a := a + 4, a \mapsto 3 \rangle \Downarrow \langle a \mapsto 7 \rangle}$$



# Semantica Tranzitională

- Introdusă de Gordon Plotkin (1981) ca Semantică Operațională Structurală
- Denumiri alternative: „semantică prin tranziții”, „semantică prin reducere”
- Definește cel mai mic pas de execuție ca o relație „de tranziție” între configurații:

$$\langle Cod, Stare \rangle \rightarrow \langle Cod', Stare' \rangle$$

- Fiecare pas de execuție este concluzia unei demonstrații
- Execuția se obține ca o succesiune de astfel de tranziții:

$$\begin{aligned} \langle x := 0 ; x := x + 1, 0 \rangle &\rightarrow \langle \text{skip}; x := x + 1, x \mapsto 0 \rangle \rightarrow \\ \langle x := x + 1, x \mapsto 0 \rangle &\rightarrow \langle x := 0 + 1, x \mapsto 0 \rangle \rightarrow \\ \langle x := 1, x \mapsto 0 \rangle &\rightarrow \langle \text{skip}, x \mapsto 1 \rangle \end{aligned}$$

# Redex. Reguli structurale. Axiome

Expresie reductibilă — redex

Reprezintă fragmentul de sintaxă care va fi modificat la următorul pas.

```
if  $0 \leq 5 + 7 * x$  then  $r := 1$  else  $r := 0$ 
```

Reguli structurale — Folosesc la identificarea următorului redex

- Definite recursiv pe structura termenilor

Axiome — Realizează pasul computațional

# Redex. Reguli structurale. Axiome

Expresie reductibilă — redex

Reprezintă fragmentul de sintaxă care va fi modificat la următorul pas.

`if  $0 \leq 5 + 7 * x$  then  $r := 1$  else  $r := 0$`

Reguli structurale — Folosesc la identificarea următorului redex

- Definite recursiv pe structura termenilor

Axiome — Realizează pasul computațional

# Redex. Reguli structurale. Axiome

Expresie reductibilă — redex

Reprezintă fragmentul de sintaxă care va fi modificat la următorul pas.

`if 0 ≤ 5 + 7 * x then r := 1 else r := 0`

Reguli structurale — Folosesc la identificarea următorului redex

- Definite recursiv pe structura termenilor

$$\frac{\langle b, \sigma \rangle \rightarrow \langle b', \sigma \rangle}{\langle \text{if } b \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle \text{if } b' \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle}$$

Axiome — Realizează pasul computațional

# Redex. Reguli structurale. Axiome

## Expresie reductibilă — redex

Reprezintă fragmentul de sintaxă care va fi modificat la următorul pas.

`if 0 ≤ 5 + 7 * x then r := 1 else r := 0`

## Reguli structurale — Folosesc la identificarea următorului redex

- Definite recursiv pe structura termenilor

$$\frac{\langle b, \sigma \rangle \rightarrow \langle b', \sigma \rangle}{\langle \text{if } b \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle \text{if } b' \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle}$$

## Axiome — Realizează pasul computațional

$$\langle \text{if true then } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle bl_1, \sigma \rangle$$

# Semantica SOS pentru IMP

## Expresii aritmetice

- Un întreg este o valoare — nu poate fi redex, deci nu avem regulă

$$(I_D) \langle x, \sigma \rangle \rightarrow \langle i, \sigma \rangle \text{ dacă } i = \sigma(x)$$

- Ordine nespecificată de evaluare a argumentelor la adunare:

$$\frac{\langle a_1, \sigma \rangle \rightarrow \langle a'_1, \sigma \rangle}{\langle a_1 + a_2, \sigma \rangle \rightarrow \langle a'_1 + a_2, \sigma \rangle}$$

$$\frac{\langle a_2, \sigma \rangle \rightarrow \langle a'_2, \sigma \rangle}{\langle a_1 + a_2, \sigma \rangle \rightarrow \langle a_1 + a'_2, \sigma \rangle}$$

$$(A_{DD}) \langle i_1 + i_2, \sigma \rangle \rightarrow \langle i, \sigma \rangle \text{ dacă } i = i_1 + i_2$$

- Regulile pentru înmulțire (Mul) sunt la fel

# Semantica SOS pentru IMP

Expresii Booleene. Constante și operatorii de comparație.

- Constantele Booleene sunt valori — nu pot fi redex

$$\frac{\langle a_1, \sigma \rangle \rightarrow \langle a'_1, \sigma \rangle}{\langle a_1 \leq a_2, \sigma \rangle \rightarrow \langle a'_1 \leq a_2, \sigma \rangle}$$

$$\frac{\langle a_2, \sigma \rangle \rightarrow \langle a'_2, \sigma \rangle}{\langle a_1 \leq a_2, \sigma \rangle \rightarrow \langle a_1 \leq a'_2, \sigma \rangle}$$

$$(\text{LEQ-FALSE}) \quad \langle i_1 \leq i_2, \sigma \rangle \rightarrow \langle \text{false}, \sigma \rangle \text{ dacă } i_1 > i_2$$

$$(\text{LEQ-TRUE}) \quad \langle i_1 \leq i_2, \sigma \rangle \rightarrow \langle \text{true}, \sigma \rangle \text{ dacă } i_1 \leq i_2$$

- Regulile pentru egalitate (Eq) sunt la fel

# Semantica SOS pentru IMP

## Expresii Booleene. Negația logică

$$\frac{\langle b, \sigma \rangle \rightarrow \langle b', \sigma \rangle}{\langle \neg b, \sigma \rangle \rightarrow \langle \neg b', \sigma \rangle}$$

$$(\text{NEG-TRUE}) \quad \langle \neg \text{true}, \sigma \rangle \rightarrow \langle \text{false}, \sigma \rangle$$

$$(\text{NEG-FALSE}) \quad \langle \neg \text{false}, \sigma \rangle \rightarrow \langle \text{true}, \sigma \rangle$$



# Semantica SOS pentru IMP

## Expresii Booleene. Și-ul logic

- Observați că evaluarea e scurtcircuitată

$$\frac{\langle b_1, \sigma \rangle \rightarrow \langle b'_1, \sigma \rangle}{\langle b_1 \wedge b_2, \sigma \rangle \rightarrow \langle b'_1 \wedge b_2, \sigma \rangle}$$

$$(\text{AND-FALSE}) \quad \langle \text{false} \wedge b_2, \sigma \rangle \rightarrow \langle \text{false}, \sigma \rangle$$

$$(\text{AND-TRUE}) \quad \langle \text{true} \wedge b_2, \sigma \rangle \rightarrow \langle b_2, \sigma \rangle$$

- Regulile pentru disjuncție (OR) sunt asemănătoare

# Semantica SOS pentru IMP

## Blocuri

- Instrucțiunea `skip` este „valoarea” blocurilor și instrucțiunilor
- Blocurile evaluează pas cu pas instrucțiunile ce le conțin

$$\frac{\langle s, \sigma \rangle \rightarrow \langle s', \sigma' \rangle}{\langle (s), \sigma \rangle \rightarrow \langle (s'), \sigma' \rangle}$$

$$(\text{BLOCK}) \quad \langle (\text{skip}), \sigma \rangle \rightarrow \langle \text{skip}, \sigma \rangle$$

# Semantica SOS pentru IMP

## Compunerea secvențială

$$\frac{\langle s_1, \sigma \rangle \rightarrow \langle s'_1, \sigma' \rangle}{\langle s_1 ; s_2, \sigma \rangle \rightarrow \langle s'_1 ; s_2, \sigma' \rangle}$$

$$(\text{SEQ}) \quad \langle \text{skip}; s_2, \sigma \rangle \rightarrow \langle s_2, \sigma \rangle$$

# Semantica SOS pentru IMP

## Atribuirea

$$\frac{\langle a, \sigma \rangle \rightarrow \langle a', \sigma \rangle}{\langle x := a, \sigma \rangle \rightarrow \langle x := a', \sigma \rangle}$$

$$(\text{ASGN}) \quad \langle x := i, \sigma \rangle \rightarrow \langle \text{skip}, \sigma' \rangle \text{ dacă } \sigma' = \sigma_{x \mapsto i}$$

# Semantica SOS pentru IMP

## Condițional

$$\frac{\langle b, \sigma \rangle \rightarrow \langle b', \sigma \rangle}{\langle \text{if } b \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle \text{if } b' \text{ then } bl_1 \text{ else } bl_2, \sigma \rangle}$$

$$(\text{IF-TRUE}) \quad \langle \text{if true then } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle bl_1, \sigma \rangle$$

$$(\text{IF-FALSE}) \quad \langle \text{if false then } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle bl_2, \sigma \rangle$$

# Semantica SOS pentru IMP

## Instrucțiunea de ciclare

(WHILE)  $\langle \text{while } b \text{ do } bl, \sigma \rangle \rightarrow$   
 $\langle \text{if } b \text{ then}( bl ; \text{while } b \text{ do } bl )\text{else skip}, \sigma \rangle$

# Semantica SOS pentru IMP

Demonstrarea unui pas. Execuție.

- Fiecare pas de deducție este o demonstrație liniară alcătuită din mai multe reguli structurale și având la vârf o axiomă
- Execuția este o succesiune de astfel de stări

# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle \text{while } 0 \leq i \text{ do } i := i + -4, i \mapsto 3 \rangle$

$\xrightarrow{\text{WHILE}}$



# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle \text{while } 0 \leq i \text{ do } i := i + -4, i \mapsto 3 \rangle$

$\langle \text{if } 0 \leq i \text{ then } ( i := i + -4 ;$   
                     $\text{while } 0 \leq i \text{ do } i := i + -4$   
                     $) \text{ else skip}$   
                     $, i \mapsto 3 \rangle$

$\xrightarrow{\text{WHILE}}$   
 $\xrightarrow{\text{ID}}$

# Semantica SOS pentru IMP

## Execuție pas cu pas

$\langle \text{while } 0 \leq i \text{ do } i := i + -4, i \mapsto 3 \rangle$

$\xrightarrow{\text{WHILE}}$

$\langle \text{if } 0 \leq i \text{ then } ( i := i + -4 ;$   
                     $\text{while } 0 \leq i \text{ do } i := i + -4$   
                     $) \text{ else skip}$   
                     $, i \mapsto 3 \rangle$

$\xrightarrow{\text{ID}}$

$\langle \text{if } 0 \leq 3 \text{ then } ( i := i + -4 ;$   
                     $\text{while } 0 \leq i \text{ do } i := i + -4$   
                     $) \text{ else skip}$   
                     $, i \mapsto 3 \rangle$

$\xrightarrow{\text{LEQ-TRUE}}$

# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle \text{while } 0 \leq i \text{ do } i := i + -4, i \mapsto 3 \rangle$

$\xrightarrow{\text{WHILE}}$

$\langle \text{if } 0 \leq i \text{ then } ( i := i + -4 ; \quad , i \mapsto 3 )$   
     $\text{while } 0 \leq i \text{ do } i := i + -4$   
     $) \text{ else skip}$

$\xrightarrow{\text{ID}}$

$\langle \text{if } 0 \leq 3 \text{ then } ( i := i + -4 ; \quad , i \mapsto 3 )$   
     $\text{while } 0 \leq i \text{ do } i := i + -4$   
     $) \text{ else skip}$

$\xrightarrow{\text{LEQ-TRUE}}$

$\langle \text{if } (\text{true}) ( i := i + -4 \quad , i \mapsto 3 )$   
     $\text{while } 0 \leq i \text{ do } i := i + -4$   
     $) \text{ else skip}$

$\xrightarrow{\text{IF-TRUE}}$



# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle (i := 3 + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

Add  
 $\longrightarrow$

# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle (i := 3 + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{Add}}$

$\langle (i := -1 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ASGN}}$

# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle (i := 3 + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ADD}}$

$\langle (i := -1 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ASGN}}$

$\langle (\text{skip}; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

$\xrightarrow{\text{SEQ}}$

# Semantica SOS pentru IMP

## Execuție pas cu pas

$\langle (i := 3 + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

ADD  
→

$\langle (i := -1 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

ASGN  
→

$\langle (\text{skip}; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

SEQ  
→

$\langle (\text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

WHILE  
→



# Semantica SOS pentru IMP

## Execuție pas cu pas

$\langle (i := 3 + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ADD}}$

$\langle (i := -1 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ASGN}}$

$\langle (\text{skip} ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

$\xrightarrow{\text{SEQ}}$

$\langle (\text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

$\xrightarrow{\text{WHILE}}$

$\langle (\text{if } 0 \leq i \text{ then } (i := i + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4) \text{ else skip}), i \mapsto -1 \rangle$

$\xrightarrow{\text{ID}}$

# Semantica SOS pentru IMP

## Execuție pas cu pas

$\langle (i := 3 + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ADD}}$

$\langle (i := -1 ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto 3 \rangle$

$\xrightarrow{\text{ASGN}}$

$\langle (\text{skip} ; \text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

$\xrightarrow{\text{SEQ}}$

$\langle (\text{while } 0 \leq i \text{ do } i := i + -4), i \mapsto -1 \rangle$

$\xrightarrow{\text{WHILE}}$

$\langle (\text{if } 0 \leq i \text{ then } (i := i + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4) \text{ else skip}), i \mapsto -1 \rangle$

$\xrightarrow{\text{ID}}$

$\langle (\text{if } 0 \leq -1 \text{ then } (i := i + -4 ; \text{while } 0 \leq i \text{ do } i := i + -4) \text{ else skip}), i \mapsto -1 \rangle \xrightarrow{\text{LEQ-FALSE}}$

# Semantica SOS pentru IMP

Execuție pas cu pas

$\langle (\text{if false then } (i := i + -4 ;$   
                     $\text{while } 0 \leq i \text{ do } i := i + -4$   
                     $) \text{ else skip } )$   
                     $, i \mapsto -1 \rangle \xrightarrow{\text{IF-FALSE}}$

## Execuție pas cu pas

$$\begin{array}{l} \langle (\text{if false then } (i := i + -4 ; \quad , i \mapsto -1) \\ \quad \text{while } 0 \leq i \text{ do } i := i + -4 \\ \quad ) \text{ else skip} ) \\ \langle (\text{skip}), i \mapsto -1 \rangle \end{array} \xrightarrow{\text{IF-FALSE}}$$

$$\xrightarrow{\text{Block}}$$

28 / 29

- G. D. Plotkin, "The origins of structural operational semantics". The Journal of Logic and Algebraic Programming, vol. 60, pp. 3–15, 2004.
- T. K. Astarte, "Formalising Meaning: a History of Programming Language Semantics". PhD Thesis, Newcastle University, 2019.