

Semantica limbajelor de programare

Tipuri de Semantică

Traian Florin Șerbănuță

Departamentul de Informatică, FMI, UNIBUC
traian.serbanuta@unibuc.ro

23 februarie 2018

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 folosind logică a efectelor unei instrucțiuni
 - Denotațională — prin asocierea unui obiect matematic (denotație)
 - Statică — Asocierea unui sistem de tipuri care exclude programe eronate

Limbajul IMP

IMP este un limbaj **IMP**erativ foarte simplu.

Ce conține

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

```
int x = 10;  
int sum = 0;  
while (0 <= x) {  
    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

$$E ::= n \mid x \\ \mid E + E \mid E * E \mid E / E$$

$$B ::= b \\ \mid E \leq E \\ \mid ! B \mid B \&\& B$$

$$C ::= \{ C \} \mid \{ \} \\ \mid C C \\ \mid x = E ; \\ \mid \text{if } (B) C \text{ else } C \\ \mid \text{while } (B) C$$

$$P ::= \text{int } x = n ; P \mid C$$

Semantică în limbaj natural

Atribuirea: $x = \text{expr}$

Semantică în limbaj natural

Atribuirea: $x = \text{expr}$

- Expresia este evaluată în starea curentă a programului
- Variabilei i se atribuie valoarea calculată, înlocuind valoarea precedentă a acelei variabile.

Avantaje și dezavantaje

- + Ușor de prezentat
- Potențial ambiguă
- Imposibil de procesat automat

Semantica denotațională

Atribuirea: $x = \text{expr}$

- Asociem expresiilor aritmetice funcții de la starea memoriei la valori:
- Asociem instrucțiunilor funcții de la starea memoriei la starea (următoare) a memoriei.

Semantica denotațională

Atribuirea: $x = \text{expr}$

- Asociem expresiilor aritmetice funcții de la starea memoriei la valori:
 - Funcția constantă $[[1]](s) = 1$
 - Funcția care selectează valoarea unui identificador $[[x]](s) = s(x)$
 - „Morfismul de adunare” $[[e1 + e2]](s) = [[e1]](s) + [[e2]](s)$.
- Asociem instrucțiunilor funcții de la starea memoriei la starea (următoare) a memoriei.
 - $[[x = e]](s)(y) = \begin{cases} s(y), & \text{dacă } y \neq x \\ [[e]](s), & \text{dacă } y = x \end{cases}$

Avantaje și dezavantaje

- + Formală, matematică, foarte precisă
- + Compozițională (morfisme și compuneri de funcții)
- Greu de stăpânit (domeniile devin din ce în ce mai complexe)

Semantica denotațională a lui IMP

```
type Id = String
type State = Id -> Int
type DAEExp = State -> Int
type DBExp = State -> Bool
type DStmt = State -> State
```

```
denotAExp :: AExp -> DAEExp
denotAExp (Int n) s = n
denotAExp (e1 :+: e2) s
    = denotAExp e1 s
    + denotAExp e2 s
```

```
denotBExp :: BExp -> DBExp
denotBExp (Bool b) s = b
denotBExp (a1 :<=: a2) s
    = denotAExp a1 s
    <= denotAExp a2 s
```

```
denotStmt :: Stmt -> DStmt
denotStmt Skip = id
denotStmt (s1 ::: s2) =
    denotStmt s2 . denotStmt s1
denotStmt (If c t e) s
    | denotBExp c s = denotStmt t s
    | otherwise     = denotStmt e s
denotStmt (While c b) =
    fix (\w s ->
        if denotBExp b s
        then w (denotStmt c s)
        else s)
```

Semantica Axiomatică

Logica Floyd-Hoare

Definește o relație ternară de forma $\{Pre\}S\{Post\}$, unde:

- S este o instrucțiune (Stmt)
- Pre (precondiție), respectiv $Post$ (postcondiție) sunt aserțiuni logice asupra stării sistemului înaintea, respectiv după execuția lui S

Se asociază fiecărei construcții sintactice Stmt o regulă de deducție care definește recursiv relația ternară descrisă mai sus.

Regula Hoare pentru compunerea secvențială

$$\frac{\{P\} c1 \{Q\} \quad \{Q\} c2 \{R\}}{\{P\} c1 c2 \{R\}}$$

Logica Floyd-Hoare pentru IMP

$$(\text{SKIP}) \quad \frac{\cdot}{\{P\} \{\} \{P\}}$$

$$(\text{SEQ}) \quad \frac{\{P\} c1 \{Q\} \quad \{Q\} c2 \{R\}}{\{P\} c1; c2 \{R\}}$$

$$(\text{ASIGN}) \quad \{P[e/x]\} x = e \{P\}$$

$$(\text{IF}) \quad \frac{\{c \wedge P\} t \{Q\} \quad \{\neg c \wedge P\} e \{Q\}}{\{P\} \text{if } c \text{ then } t \text{ else } e \{Q\}}$$

$$(\text{WHILE}) \quad \frac{\{c \wedge P\} b \{P\}}{\{P\} \text{while } (c) b \{\neg c \wedge P\}}$$

Semantică Statică - Motivație

- Este sintaxa unui limbaj de programare prea expresivă?
- Sunt programe care n-aș vrea să le pot scrie, dar le pot?
- Putem detecta programe greșite înainte de rulare?

Semantică Statică - Motivație

- Este sintaxa unui limbaj de programare prea expresivă?
- Sunt programe care n-aș vrea să le pot scrie, dar le pot?
- Putem detecta programe greșite înainte de rulare?
De exemplu, în IMP, folosirea variabilelor fără a le declara
- Soluție: Sistemele de tipuri

Sisteme de tipuri

La ce folosesc?

- Descriu programele „bine formate“
- Pot preveni anumite erori
 - folosirea variabilelor nedecarate/neinițializate
 - detectarea unor bucați de cod inaccesibile
 - erori de securitate
- Ajută compilatorul
- Pot influența proiectarea limbajului

Scop (ideal)

Programele „bine formate“, i. e., cărora li se poate asocia un tip nu eșuează

Sisteme de tipuri

Intuiție

- Vom defini o relație $\Gamma \vdash e : T$
- Citim e are tipul T dacă Γ , unde
- Γ — tipuri asociate locațiilor din e

Exemple

$$\vdash \text{ if } true \text{ then } \{ \} \text{ else } \{ \} : \text{stmt}$$

$$x : \text{int} \vdash x + 13 : \text{int}$$

$$x : \text{int} \not\vdash x = y + 1 : T \quad \text{pentru orice } T$$

Tipuri în limbajul IMP

Tipurile expresiilor = tipurile gramaticale

$$T ::= \text{int} \mid \text{bool} \mid \text{stmt}$$

Γ — Mediul de tipuri

- Asociază tipuri identificatorilor

$$\Gamma : \mathbb{X} \overset{\circ}{\rightarrow} \mathbb{T}$$

- **Notăție:** o listă de perechi locație-tip

$$x_1 : \text{int}, \dots, x_n : \text{int}$$

Observații pentru limbajul IMP

- Toate locațiile din Γ au același tip: `int`
- Apariția unei locații în Γ înseamnă că locația e de fapt definită

IMP: Reguli pentru tipuri

(LOC) $\Gamma \vdash x : \text{int}$ dacă $\Gamma(x) = \text{int}$

(INT) $\Gamma \vdash n : \text{int}$ dacă $n \in \mathbb{Z}$

(OP₊)
$$\frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash e_1 + e_2 : \text{int}}$$

(BOOL) $\Gamma \vdash b : \text{bool}$ dacă $b \in \{\text{true}, \text{false}\}$

(OP_≤)
$$\frac{\Gamma \vdash e_1 : \text{int} \quad \Gamma \vdash e_2 : \text{int}}{\Gamma \vdash e_1 \leq e_2 : \text{bool}}$$

(ATTRIB)
$$\frac{\Gamma \vdash e : \text{int}}{\Gamma \vdash x = e : \text{stmt}} \quad \text{dacă } \Gamma(x) = \text{int}$$

(SECV)
$$\frac{\Gamma \vdash c_1 : \text{stmt} \quad \Gamma \vdash c_2 : \text{stmt}}{\Gamma \vdash c_1 ; c_2 : \text{stmt}}$$

(IF)
$$\frac{\Gamma \vdash c : \text{bool} \quad \Gamma \vdash t : \text{stmt} \quad \Gamma \vdash e : \text{stmt}}{\Gamma \vdash \text{if } c \text{ then } t \text{ else } e : \text{stmt}}$$

(WHILE)
$$\frac{\Gamma \vdash c : \text{bool} \quad \Gamma \vdash b : \text{stmt}}{\Gamma \vdash \text{while } (c) b : \text{stmt}}$$

Semantică operațională

Plan

- Exemplu de lucru: Limbajul IMP
- 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

Starea execuției unui program IMP la un moment dat este data de valorile deținute în acel moment de variabilele declarate de program.

Matematic: o funcție **parțială** $\sigma : x \overset{\circ}{\rightarrow} Int$ de domeniu finit.

Notatii

- Descrierea funcției prin enumerare: $\sigma = n \mapsto 10, sum \mapsto 0$
- Funcția vidă \perp , nedefinită pentru nici o variabilă
- Obținerea valorii unei variabile: $\sigma(x)$
- Suprascrierea valorii unei variabile:

$$\sigma[v/x](y) = \begin{cases} \sigma(y), & \text{dacă } y \neq x \\ v, & \text{dacă } y = x \end{cases}$$

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: $\langle s, \sigma \rangle \Downarrow \langle \sigma' \rangle$
 - Blocurile se evaluează la stări: $\langle bl, \sigma \rangle \Downarrow \langle \sigma' \rangle$
 - Programul se evaluează la o stare: $\langle p \rangle \Downarrow \langle \sigma \rangle$ sau $\langle p, \sigma \rangle \Downarrow \langle \sigma' \rangle$
- Valoarea este obținută într-un **singur pas (mare)**
- Reguli structurale, având ca premise secvenți corespunzători 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 a lui IMP

Expresii aritmetice

$$(INT) \quad \langle i, \sigma \rangle \Downarrow \langle i \rangle$$

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

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

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

$$(DIV) \quad \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 / a_2, \sigma \rangle \Downarrow \langle i \rangle} \quad \text{dacă } i_2 \neq 0 \wedge i = i_1 /_{Int} i_2$$

Semantica Evaluării a lui IMP

Expresii booleene

$$(Bool) \quad \langle t, \sigma \rangle \Downarrow \langle t \rangle$$

$$(CMP) \quad \frac{\langle a_1, \sigma \rangle \Downarrow \langle i_1 \rangle \quad \langle a_2, \sigma \rangle \Downarrow \langle i_2 \rangle}{\langle a_1 \leq a_2, \sigma \rangle \Downarrow \langle t \rangle} \quad \text{dacă } t = i_1 \leq_{Int} i_2$$

$$(Not-True) \quad \frac{\langle b, \sigma \rangle \Downarrow \langle false \rangle}{\langle ! b, \sigma \rangle \Downarrow \langle true \rangle}$$

$$(Not-False) \quad \frac{\langle b, \sigma \rangle \Downarrow \langle true \rangle}{\langle ! b, \sigma \rangle \Downarrow \langle false \rangle}$$

$$(And-True) \quad \frac{\langle b_1, \sigma \rangle \Downarrow \langle true \rangle \quad \langle b_2, \sigma \rangle \Downarrow \langle t \rangle}{\langle b_1 \ \&\& \ b_2, \sigma \rangle \Downarrow \langle t \rangle}$$

$$(And-False) \quad \frac{\langle b_1, \sigma \rangle \Downarrow \langle false \rangle}{\langle b_1 \ \&\& \ b_2, \sigma \rangle \Downarrow \langle false \rangle}$$

Semantica Evaluării a lui IMP

Instrucțiuni simple

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

$$(ASGN) \quad \frac{\langle a, \sigma \rangle \Downarrow \langle i \rangle}{\langle x = a ;, \sigma \rangle \Downarrow \langle \sigma' \rangle} \quad \text{dacă } \sigma' = \sigma[i/x]$$

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

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

Semantica Evaluării a lui IMP

Blocuri și instrucțiuni de ciclare

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

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

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

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

Semantica Evaluării a lui IMP

Inițializări; Semantica programului

$$(INIT) \quad \frac{\langle p, \sigma' \rangle \Downarrow \langle \sigma'' \rangle}{\langle \text{int } x = i ; p, \sigma \rangle \Downarrow \langle \sigma'' \rangle} \quad \text{dacă } \sigma' = \sigma[i/x]$$

$$(PGM) \quad \frac{\langle p, \perp \rangle \Downarrow \langle \sigma \rangle}{\langle p \rangle \Downarrow \langle \sigma \rangle}$$

Semantica Evaluării a lui IMP

Arbori de derivare

(P_{GM}) $\overline{\langle \text{int } a = 3; a = a + 4 ; \rangle} \Downarrow \langle ? \rangle$

Semantica Evaluării a lui IMP

Arbori de derivare

$$\begin{array}{c}
 (P_{GM}) \quad \frac{(I_{INIT}) \quad \overline{\langle \text{int } a = 3; a = a + 4 ;, \perp \rangle} \Downarrow \langle ? \rangle}{\langle \text{int } a = 3; a = a + 4 ; \rangle \Downarrow \langle ? \rangle}
 \end{array}$$

Semantica Evaluării a lui IMP

Arbori de derivare

$$\begin{array}{c}
 \text{(P}_{\text{GM}}\text{)} \quad \frac{\text{(INIT)} \quad \frac{\text{(ASGN)} \quad \overline{\langle a = a + 4 ;, a \mapsto 3 \rangle} \Downarrow \langle ? \rangle}{\langle \text{int } a = 3 ;, a = a + 4 ;, \perp \rangle \Downarrow \langle ? \rangle}}{\langle \text{int } a = 3 ;, a = a + 4 ; \rangle \Downarrow \langle ? \rangle}
 \end{array}$$

Semantica Evaluării a lui IMP

Arbori de derivare

$$\begin{array}{c}
 \text{(PGM)} \quad \frac{\text{(INIT)} \quad \frac{\text{(ASGN)} \quad \frac{\text{(ADD)} \quad \overline{\langle a + 4, a \mapsto 3 \rangle} \Downarrow \langle ?? \rangle}{\langle a = a + 4 ;, a \mapsto 3 \rangle \Downarrow \langle a \mapsto ?? \rangle}}{\langle \text{int } a = 3; a = a + 4 ;, \perp \rangle \Downarrow \langle a \mapsto ?? \rangle}}{\langle \text{int } a = 3; a = a + 4 ; \rangle \Downarrow \langle a \mapsto ?? \rangle}
 \end{array}$$

Semantica Evaluării a lui IMP

Arbori de derivare

$$\begin{array}{c}
 \text{(Pgm)} \quad \frac{\text{(Init)} \quad \frac{\text{(ASGN)} \quad \frac{\text{(Add)} \quad \frac{\text{(Id)} \quad \frac{\langle a, a \mapsto 3 \rangle \Downarrow \langle 3 \rangle}{\langle a + 4, a \mapsto 3 \rangle \Downarrow \langle 7 \rangle} \quad \text{(Int)} \quad \langle 4, a \mapsto 3 \rangle \Downarrow \langle 4 \rangle}{\langle a = a + 4 ;, a \mapsto 3 \rangle \Downarrow \langle a \mapsto 7 \rangle}}{\langle \text{int } a = 3 ; a = a + 4 ;, \perp \rangle \Downarrow \langle a \mapsto 7 \rangle}}{\langle \text{int } a = 3 ; a = a + 4 ; \rangle \Downarrow \langle a \mapsto 7 \rangle}
 \end{array}$$

Semantica Evaluării

Avantaje și dezavantaje

Avantaje

- Compozițională: arborii de demonstrații sunt compoziționali
- Ușor și relativ eficient de implementat și executat
- Foarte folosită pentru definirea sistemelor de tipuri

Dezavantaje

- Lipsa granularității — computația e un monolit
- Greu de capturat nedeterminismul/concurența
- Greu de capturat schimbările de control
- Nemodulară - extensiile solicită modificarea regulilor existente.

Semantica Tranzițională

- 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:

$$\langle \text{Cod}, \text{Stare} \rangle \rightarrow \langle \text{Cod}', \text{Stare}' \rangle$$

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

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

Redex. Reguli structurale. Axiome

Expresie reducibilă — redex

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

```
if (0 <= 5 + 7 * x) { 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 reducibilă — redex

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

```
if (0 <= 5 + 7 * x) { 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 reducibilă — redex

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

`if (0 <= 5 + 7 * x) { r = 1 ; } else { r = 0 ; }`

Reguli structurale — Folosesc la identificarea următorului redex

- Definite recursiv pe structura termenilor

$$\frac{\langle b, \sigma \rangle \rightarrow \langle b', \sigma \rangle}{\langle \text{if } (\textcolor{red}{b}) \text{ } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle \text{if } (\textcolor{blue}{b}') \text{ } bl_1 \text{ else } bl_2, \sigma \rangle}$$

Axiome — Realizează pasul computațional

Redex. Reguli structurale. Axiome

Expresie reducibilă — redex

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

`if (0 <= 5 + 7 * x) { r = 1 ; } else { r = 0 ; }`

Reguli structurale — Folosesc la identificarea următorului redex

- Definite recursiv pe structura termenilor

$$\frac{\langle b, \sigma \rangle \rightarrow \langle b', \sigma \rangle}{\langle \text{if } (\textcolor{red}{b}) \text{ } bl_1 \text{ else } bl_2, \sigma \rangle \rightarrow \langle \text{if } (\textcolor{blue}{b}') \text{ } bl_1 \text{ else } bl_2, \sigma \rangle}$$

Axiome — Realizează pasul computațional

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

Semantica SOS a lui IMP

Expresii aritmetice

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

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

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

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

- Ordine nespecificată de evaluare a argumentelor

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

- Regula pentru înmulțire este la fel

Semantica SOS a lui IMP

Ordine de evaluare. Împărțire

Semantica împărțirii

Evaluăm al doilea argument, și dacă e diferit de 0, atunci evaluăm și primul argument și apoi împărțirea.

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

$$\frac{\langle a_1, \sigma \rangle \rightarrow \langle a'_1, \sigma \rangle}{\langle a_1 / i_2, \sigma \rangle \rightarrow \langle a'_1 / i_2, \sigma \rangle} \quad \text{dacă } i_2 \neq 0$$

- Evaluarea argumentelor de la dreapta la stânga

$$(Div) \quad \langle i_1 / i_2, \sigma \rangle \rightarrow \langle i, \sigma \rangle \quad \text{dacă } i_2 \neq 0, i = i_1 / i_2$$

Semantica SOS a lui IMP

Expresii Booleene. Constante și operatorul de comparație.

- Constantele Booleene sunt valori — nu pot fi redex

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

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

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

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

Semantica SOS a lui IMP

Expresii Booleene. Negația logică

$$\frac{\langle a, \sigma \rangle \rightarrow \langle a', \sigma \rangle}{\langle ! a, \sigma \rangle \rightarrow \langle ! a', \sigma \rangle}$$

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

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

Semantica SOS a lui IMP

Expresii Booleene. Și-ul logic

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

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

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

Semantica SOS a lui IMP

Blocuri

- Blocul vid $\{\}$ este „valoarea” blocurilor și instrucțiunilor
- O instrucțiune poate modifica starea curentă

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

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

Semantica SOS a lui IMP

Compunerea secvențială

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

$$(\text{NEXT-STMT}) \quad \langle \{\} \ s_2, \sigma \rangle \rightarrow \langle s_2, \sigma \rangle$$

Semantica SOS a lui IMP

Atribuirea

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

$$(\text{ASGN}) \quad \langle x = i ;, \sigma \rangle \rightarrow \langle \{\}, \sigma' \rangle \quad \text{dacă } \sigma' = \sigma[i/x]$$

Semantica SOS a lui IMP

Condițional

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

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

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

Semantica SOS a lui IMP

Instrucțiunea de ciclare

$$(\text{WHILE}) \quad \langle \text{while } (b) \text{ } bl, \sigma \rangle \rightarrow \langle \text{if } (b) \text{ } \{ bl \text{ while } (b) \text{ } bl \} \text{ else } \{\}, \sigma \rangle$$

Semantica SOS a lui IMP

Inițializări

$$(\text{INIT}) \quad \langle \text{int } x = i ; p, \sigma \rangle \rightarrow \langle p, \sigma' \rangle \quad \text{dacă } \sigma' = \sigma[i/x]$$

Semantica SOS a lui 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 a lui IMP

Execuție pas cu pas

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

INIT
 \longrightarrow

Semantica SOS a lui IMP

Execuție pas cu pas

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

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

INIT



WHILE



Semantica SOS a lui IMP

Execuție pas cu pas

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

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

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

INIT
→

WHILE
→

ID
→

Semantica SOS a lui IMP

Execuție pas cu pas

$\langle \text{int } i = 3 ; \text{while } (0 \leq i) \{ i = i + -4 ; \}, \perp \rangle$	$\xrightarrow{\text{INIT}}$
$\langle \text{while } (0 \leq i) \{ i = i + -4 ; \}, i \mapsto 3 \rangle$	$\xrightarrow{\text{WHILE}}$
$\langle \text{if } (0 \leq i) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{ID}}$
$\langle \text{if } (0 \leq 3) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{LEQ-TRUE}}$

Semantica SOS a lui IMP

Execuție pas cu pas

$\langle \text{int } i = 3 ; \text{while } (0 \leq i) \{ i = i + -4 ; \}, \perp \rangle$	$\xrightarrow{\text{INIT}}$
$\langle \text{while } (0 \leq i) \{ i = i + -4 ; \}, i \mapsto 3 \rangle$	$\xrightarrow{\text{WHILE}}$
$\langle \text{if } (0 \leq i) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{ID}}$
$\langle \text{if } (0 \leq 3) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{LEQ-TRUE}}$
$\langle \text{if } (\text{true}) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{IF-TRUE}}$

Semantica SOS a lui IMP

Execuție pas cu pas

$\langle \text{int } i = 3 ; \text{while } (0 \leq i) \{ i = i + -4 ; \}, \perp \rangle$	$\xrightarrow{\text{INIT}}$
$\langle \text{while } (0 \leq i) \{ i = i + -4 ; \}, i \mapsto 3 \rangle$	$\xrightarrow{\text{WHILE}}$
$\langle \text{if } (0 \leq i) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{ID}}$
$\langle \text{if } (0 \leq 3) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{LEQ-TRUE}}$
$\langle \text{if } (\text{true}) \{ \{ i = i + -4 ; \} \quad , i \mapsto 3 \}$ $\quad \text{while } (0 \leq i) \{ i = i + -4 ; \}$ $\quad \} \text{ else } \{ \}$	$\xrightarrow{\text{IF-TRUE}}$
$\langle \{ \{ i = i + -4 ; \} \text{while } (0 \leq i) \{ i = i + -4 ; \} \}, i \mapsto 3 \rangle$	$\xrightarrow{\text{ID}}$

Semantica SOS a lui IMP

Execuție pas cu pas

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

Add
→

Semantica SOS a lui IMP

Execuție pas cu pas

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

ADD
→

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

ASGN
→

Semantica SOS a lui IMP

Execuție pas cu pas

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

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

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

ADD
→

ASGN
→

BLOCK-END
→

Semantica SOS a lui IMP

Execuție pas cu pas

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

ADD
→

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

ASGN
→

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

BLOCK-END
→

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

NEXT-STMT
→

Semantica SOS a lui IMP

Execuție pas cu pas

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

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

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

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

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

ADD
→

ASGN
→

BLOCK-END
→

NEXT-STMT
→

WHILE
→

Semantica SOS a lui IMP

Execuție pas cu pas

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

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

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

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

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

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

ADD
→

ASGN
→

BLOCK-END
→

NEXT-STMT
→

WHILE
→

ld
→

Semantica SOS a lui IMP

Execuție pas cu pas

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

ADD
→

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

ASGN
→

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

BLOCK-END
→

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

NEXT-STMT
→

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

WHILE
→

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

Id
→

while (0 ≤ i) { i = i + -4 ; }
} else { }

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

LEQ-FALSE
→

while (0 ≤ i) { i = i + -4 ; }
} else { }

Semantica SOS a lui IMP

Execuție pas cu pas

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

Semantica SOS a lui IMP

Execuție pas cu pas

$$\begin{array}{lcl}
 \langle \{ \text{if (false) } \{ \{ i = i + -4 ; \} \\
 \quad \text{while } (0 \leq i) \{ i = i + -4 ; \} \\
 \quad \} \text{ else } \{ \} \} \rangle, i \mapsto -1 \rangle & \xrightarrow{\text{IF-FALSE}} & \\
 \langle \{ \{ \} \}, i \mapsto -1 \rangle & \xrightarrow{\text{BLOCK-END}} &
 \end{array}$$

Semantica SOS a lui IMP

Execuție pas cu pas

$$\begin{array}{lcl}
 \langle \{ \text{if (false) } \{ \{ i = i + -4 ; \} & , i \mapsto -1 \rangle & \xrightarrow{\text{IF-FALSE}} \\
 \quad \text{while } (0 \leq i) \{ i = i + -4 ; \} & & \\
 \quad \} \text{ else } \{ \} \} & & \\
 \langle \{ \{ \} \} , i \mapsto -1 \rangle & & \xrightarrow{\text{BLOCK-END}} \\
 \langle \{ \} , i \mapsto -1 \rangle & &
 \end{array}$$

Semantica SOS

Avantaje și dezavantaje

Avantaje

- Definește precis noțiunea de pas computațional
- Semnalează erorile, oprind execuția
- Execuția devine ușor de urmărit și depanat
- Nedeterminismul și concurența pot fi definite și analizate

Dezavantaje

- Regulile structurale sunt evidente și deci plictisitor de scris
- Schimbarea abruptă a controlului rămâne o sarcină dificilă
- Nemodular: adăugarea unei trăsături noi poate solicita schimbarea întregii definiții