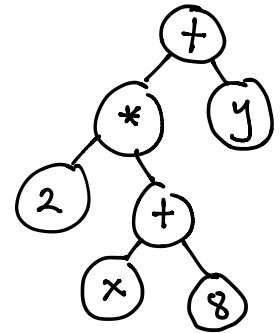


Zadnjič:

Aritmetični izraz  $e$

npr.

konkretna  
 $2 * (x + 8) + y$



abstraktna sintaksa

Evalvacija:

$$\eta \mid e \mapsto n$$

veliki koraki

"v okolju  $\eta$  je vrednost izraza  $e$  število  $n$ "

↑  
okolje

$[x \mapsto 10, y \mapsto 5, z \mapsto 8]$

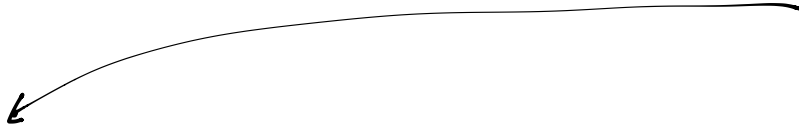
$$\eta \mid e \mapsto e'$$

"v okolju  $\eta$  izraz  $e$  naredi računski korake v izraz  $e'$ "

$$e \mapsto e' \mapsto e'' \mapsto e''' \mapsto \dots$$

1) postopek se konča

$$\dots \mapsto e''''''$$



$e_1 \mapsto e_2 \mapsto \dots \mapsto e_n$  ni naslednjega koraka

2) postopek se ne konča  
 (program "se začne")

Primer:

$$[] \mid 2 + 3 * 5 \mapsto 2 + 15 \mapsto 17$$

✓ dosegli smo končni rezultat  
 (17 je vrednost)

$$[] \mid x + 3 * 5 \mapsto x + 15$$

✗ izvajanje je blidirano ("runtime error")  
 "crash"  
 npr. segment violation

# Ukazni programski jezik

- cela števila
- boolove vrednosti (true, false)
- spremenljivke
- pogojni stavki & zanka

- Koraki:
1. Sintaksa
  2. Tipi (jih ne potrebujemo)
  3. Operacijska semantika

PRESUČIMO

## ① Sintaksa

nehaј, kar uporablja profesor, da opiše slovnična pravila  
::=

:= } del sintakse  
= } prog. jezika

Aritmetični izrazi (konkretna sintaksa, opisuje zaporedja leksemov):

```
{aritmetični-izraz} ::= {aditivni-izraz}
{aditivni-izraz} ::= {multiplikativni-izraz} | {aditivni-izraz} + {multiplikativni-izraz}
{multiplikativni-izraz} ::= {osnovni-izraz} | {multiplikativni-izraz} * {osnovni-izraz}
{osnovni-izraz} ::= {spremenljivka} | {številka} | ( {aritmetični-izraz} )
{spremenljivka} ::= [a-zA-z]+
{številka} ::= -? [0-9]+
```

Boolovi izrazi (abstraktna sintaksa, opisuje drevesa):

```
{boolov-izraz} ::= true | false |
                 {aritmetični-izraz} = {aritmetični-izraz} |
                 {aritmetični-izraz} < {aritmetični-izraz} |
                 {boolov-izraz} and {boolov-izraz} |
                 {boolov-izraz} or {boolov-izraz} |
                 not {boolov-izraz}
```

Ukazi:

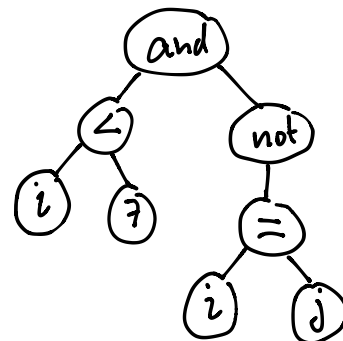
```
{ukaz} ::= skip |
          {spremenljivka} := {aritmetični-izraz} |
          {ukaz} ; {ukaz} |
          while {boolov-izraz} do {ukaz} done |
          if {boolov-izraz} then {ukaz} else {ukaz} end
```

oklepaji?  
prioriteta & asociativnost?

while i < 10 do  
i := i + 2 ;  
s := s \* i  
done

$C_1 ; C_2 ; C_3$   
( ) ( ) ( )

$i < 7$  and not  $i = j$



Naštejmo jih od nižje do višje prioritete:

- ; (levo)  $\longrightarrow c_1; c_2; c_3 = (c_1; c_2); c_3$
- or (levo)  $\} p \text{ or } q \text{ and } s = p \text{ or } (q \text{ and } s)$
- and (levo)
- not  $\} \text{not } p \text{ and } q = (\text{not } p) \text{ and } q$
- =, <
- + (levo)
- \* (levo)

Kakšno je s prioriteto  $\neq$   $<$  in  $=$ ?  
 $2 + 3 = 5 \text{ and } i < 7$

Na primer, or je levo asociativen in ima prednost pred ;.

```
s = 0 ;
i = 0 ;
while (i < 100) {
  s = s + i ;
  i = i + 1 ;
}
```

Java

```
s := 0;
i := 0;
while i < 100 do
  s := s + i;
  i := i + 1
done
```

naš jezik

??  $2 + (3 = 5 \text{ and } i) < 7$   
 $3 < 7 < 5$   
 ??  $(3 < 7) < 5$   
 ??  $3 < (7 < 5)$

# Operacijska semantika

Aritmetični izrazi:  $\eta \mid e \hookrightarrow n$

Boolovi izrazi:  $\eta \mid e \hookrightarrow b$   
 $\hookrightarrow \text{true ali false}$

```
(boolov-izraz) ::= true | false |
  (aritmetični-izraz) = (aritmetični-izraz) |
  (aritmetični-izraz) < (aritmetični-izraz) |
  (boolov-izraz) and (boolov-izraz) |
  (boolov-izraz) or (boolov-izraz) |
  not (boolov-izraz)
```

$\eta \mid \text{true} \hookrightarrow \text{true}$

$\eta \mid \text{false} \hookrightarrow \text{false}$

$\eta \mid b \hookrightarrow \text{false}$

$\eta \mid \text{not } b \hookrightarrow \text{true}$

$\eta \mid b \hookrightarrow \text{true}$

$\eta \mid \text{not } b \hookrightarrow \text{false}$

} alternativa

$\eta \mid e \hookrightarrow b \quad b' = \neg b$

$\eta \mid \text{not } e \hookrightarrow b'$

$\eta \mid b_1 \leftrightarrow \text{false}$   
 $\eta \mid b_1 \text{ and } b_2 \leftrightarrow \text{false}$

$\eta \mid b_1 \leftrightarrow \text{true} \quad \eta \mid b_2 \leftrightarrow v_2$   
 $\eta \mid b_1 \text{ and } b_2 \leftrightarrow v_2$

$\eta \mid b_1 \leftrightarrow \text{true}$   
 $\eta \mid b_1 \text{ or } b_2 \leftrightarrow \text{true}$

$\eta \mid b_1 \leftrightarrow \text{false} \quad \eta \mid b_2 \leftrightarrow v_2$   
 $\eta \mid b_1 \text{ or } b_2 \leftrightarrow v_2$

$\eta \mid e_1 \leftrightarrow n_1 \quad \eta \mid e_2 \leftrightarrow n_2 \quad n_1 < n_2$   
 $\eta \mid e_1 < e_2 \leftrightarrow \text{true}$

$\eta \mid e_1 \leftrightarrow n_1 \quad \eta \mid e_2 \leftrightarrow n_2 \quad n_1 \geq n_2$   
 $\eta \mid e_1 < e_2 \leftrightarrow \text{false}$

"short-circuit and"  
 ne evaluiramo  $b_2$ ,  
 če ni potrebno

"Andalso"  
 $\eta \mid b_1 \hookrightarrow v_1 \quad \eta \mid b_2 \hookrightarrow v_2 \quad v = v_1 \wedge v_2$

$\eta \mid b_1 \text{ and } b_2 \hookrightarrow v$

vedno evaluiramo  $b_1$  in  $b_2$

tabela  $a[100]$

if ( $i < 100 \ \&\& \ a[i] == 42$ ) { ... }

se zanašamo na short-circuit  $\&\&$

→ uporabljamo  
 evaluacijo  
 celih števil

Vaja: pravilo za  $e_1 = e_2$

Ukazi:

$[x \mapsto 7, y \mapsto 8] \mid (x := 15) \mapsto ? \quad [x \mapsto 15, y \mapsto 8]$

OKOLJE SE LAHKO SPREMENI

$(\eta, c) \mapsto (\eta', c')$

v okolju  $\eta$  ukaz  $c$  naredi  
 en korak v okolje  $\eta'$  in ukaz  $c'$

$(\eta, c) \mapsto \eta'$

v okolju  $\eta$  ukaz  $c$  v enem  
 koraku konča v okolju  $\eta'$

$(\eta, c) \mapsto (\eta', c') \mapsto (\eta'', c'') \mapsto \dots \mapsto (\eta''', c''') \mapsto \eta''''$

$(\eta, \text{skip}) \mapsto \eta$

$\eta \mid e \hookrightarrow n$

$(\eta, (x := e)) \mapsto \eta[x \mapsto n]$

$(\eta, c_1) \mapsto (\eta', c_1')$

$(\eta, (c_1 ; c_2)) \mapsto (\eta', (c_1' ; c_2'))$

$(\eta, c_1) \mapsto \eta'$

$(\eta, (c_1 ; c_2)) \mapsto (\eta', c_2)$

$\eta \mid b \hookrightarrow \text{true}$

$(\eta, (\text{if } b \text{ then } c_1 \text{ else } c_2 \text{ end})) \mapsto (\eta, c_1)$

$\eta \mid b \hookrightarrow \text{false}$

$(\eta, (\text{if } b \text{ then } c_1 \text{ else } c_2 \text{ end})) \mapsto (\eta, c_2)$

$\eta \mid b \hookrightarrow \text{false}$

$(\eta, (\text{while } b \text{ do } c \text{ done})) \mapsto \eta$

$\eta \mid b \hookrightarrow \text{true}$

$(\eta, (\text{while } b \text{ do } c \text{ done})) \mapsto (\eta, (c ; \text{while } b \text{ do } c \text{ done}))$

$\eta[x \mapsto n]$  oboje  $\eta$ , v katerem vrednost  $x$  nastavimo na  $n$

$[x \mapsto 5, y \mapsto 8, z \mapsto 10] [y \mapsto 30] = [x \mapsto 5, y \mapsto 30, z \mapsto 10]$

NE DELAJ  
TEGA:

$(\eta, c_1) \mapsto \eta' \quad (\eta', c_2) \mapsto (\eta'', c_2')$

$(\eta, c_1 ; c_2) \mapsto (\eta'', c_2')$  TO STA  
DVA  
KORAKA

Opomba:

if (b) {c1} isto kot

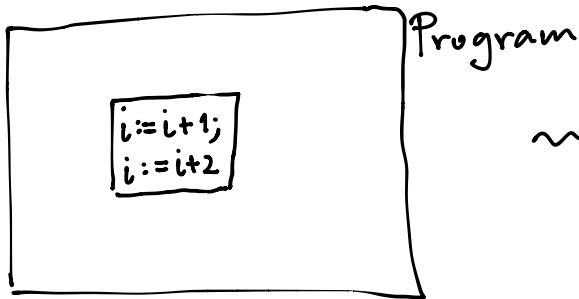
if (b) {c1} else {}

if b then c1 else skip end

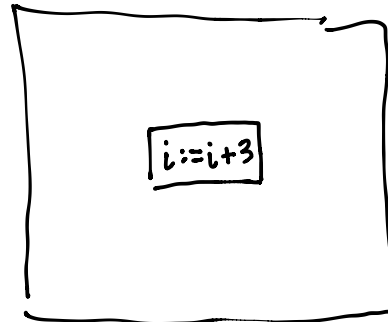
$(\text{while } b \text{ do } c \text{ done}) \equiv$

if b then (while b do c done) else skip end

## Ekvivalenca programov

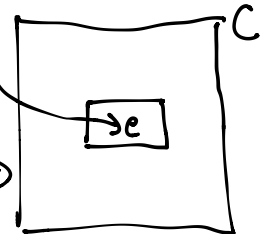


$\rightsquigarrow$



Evaluacijski kontekst: program z luknjo

$C[-]$  vstavimo  $e$ :  $C[e]$



Primer:

A  $x := x + 1;$   
 $x := x + 2$

B  $y := y + 3$

C

$x := 0;$   
 $y := 0;$



C[A]

$[x \mapsto 5, y \mapsto 7]$   
 $x := 0;$   
 $y := 0;$   
 $x := x + 1;$   
 $x := x + 2$   
 $[x \mapsto 3, y \mapsto 0]$

C[B]

$[x \mapsto 5, y \mapsto 7]$   
 $x := 0;$   
 $y := 0;$   
 $y := y + 3$   
 $[x \mapsto 0, y \mapsto 3]$