

# Ukazni programski jezik

## Sintaksa

while <bod> do  
    <ukaz> ; <ukaz> ; <ukaz>  
done

*(ukaz)*  
*(ukaz)*

## Operacijska semantika

$n \mid b_2 \leftrightarrow$

$n \mid b_2 \leftrightarrow$

- prevajalnik

Comm

assembler

P



C

- potrebujemo opis delovanja, ki je neodvisen od prevajalnika & arhitekture
- operacijska semantika

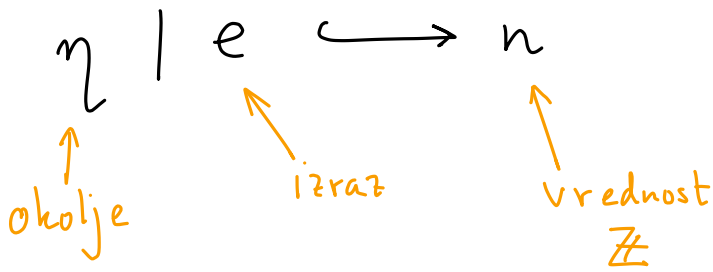
def f(n):

    return  $n * n + 3$

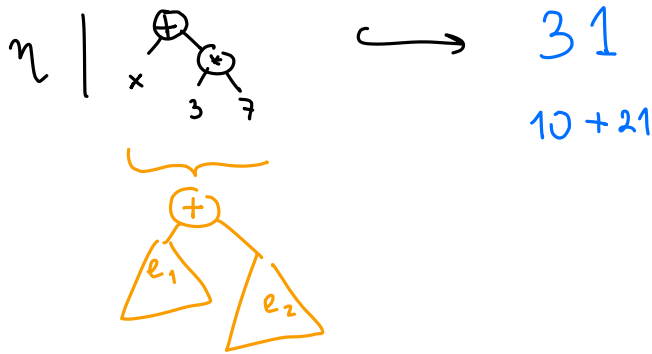
$f: \mathbb{Z} \rightarrow \mathbb{Z}$

$n \mapsto n^2 + 3$

$f(7) = 52$



$$\frac{\eta(x) = 10}{\eta \mid x \hookrightarrow 10} \quad \eta \mid \begin{matrix} \times \\ 3 \quad 7 \end{matrix} \hookrightarrow 21$$



$$\eta \mid b \hookrightarrow p$$

$$\eta \mid \text{not } b \hookrightarrow \neg p$$

$$\eta \mid e \hookrightarrow n$$

$$\eta \mid -e \hookrightarrow -n$$

$b_1$  and  $b_2$

"short-circuit"

- izračunaj  $b_1 \rightarrow$  false  $\rightarrow$  odgovor: false  
 $\rightarrow$  true  $\rightarrow$  računaj  $b_2$
- izračunaj  $b_1$  in izračunaj  $b_2$
- uporabi tabele

$$\eta \mid b_1 \hookrightarrow \text{false} \quad \eta \mid b_2 \hookrightarrow \text{false}$$

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

$$\eta \mid b_1 \hookrightarrow \text{false} \quad \eta \mid b_2 \hookrightarrow \text{true}$$

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

$b_1$	$b_2$	$b_1 \text{ and } b_2$
⊥	⊥	⊥
⊥	⊤	⊥
⊤	⊥	⊥
⊤	⊤	⊤

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

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

Ukazi:

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

$$(\eta, C) \mapsto (\eta', C') \text{ en korak}$$

V okolju  $\eta$  ukaz  $C$  naredi en korak in novo okolje je  $\eta'$  ter preostanek ukaza je ukaz  $C'$

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

V okolju  $\eta$  ukaz  $C$  v enem koraku konča izvajanje in novo okolje je  $\eta'$

$$\frac{\textcircled{1} \quad (\eta, C_1) \mapsto \eta' \quad \textcircled{2} \quad (\eta', C_2) \mapsto (\eta'', C_2')}{(\eta, C_1; C_2) \mapsto (\eta'', C_2')}$$

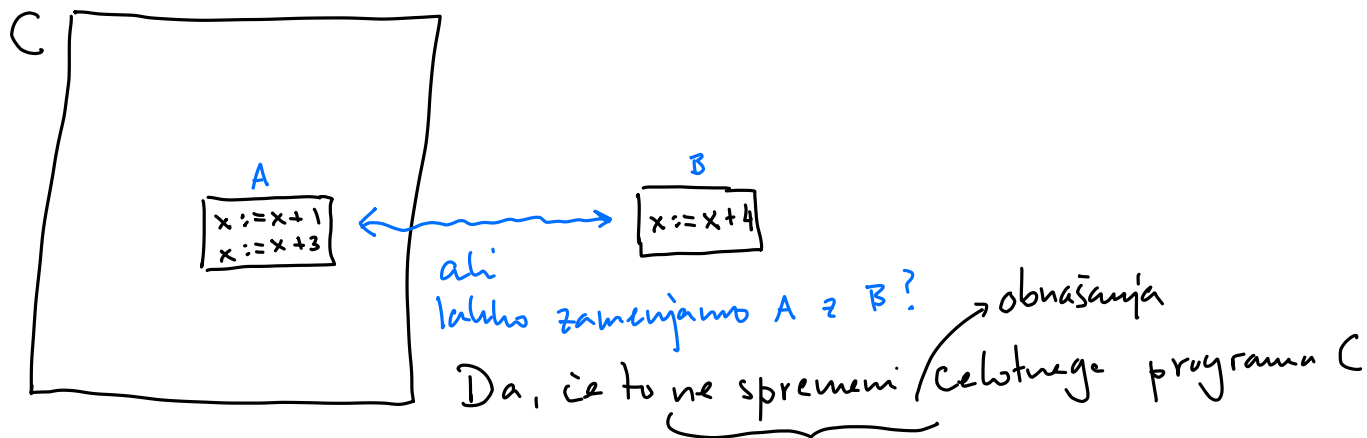
DVA KORAKA!

```
while b do
  C
done
```

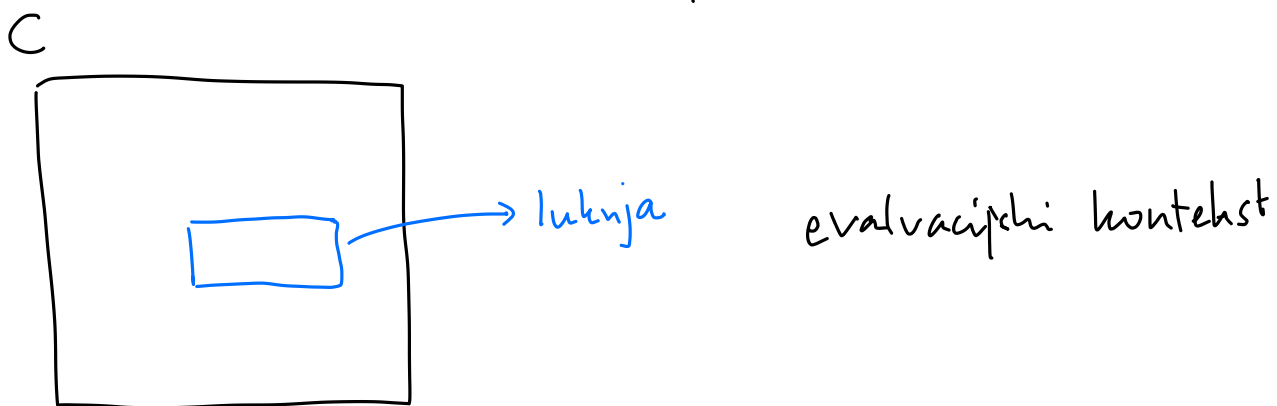
$\implies$   
 $b = \text{true}$

```
C ;
while b do
  C
done
```

# Ekvivalenca programov



C bo v obeh primerih izračunal isti rezultat  
(ali pa v obeh primerih se začelul)



```
for (A; p; B) {  
    C  
}
```

```
A;  
while (p) {  
    C;  
    B  
}
```

if p then A else B end

```
if (p) {  
  A  
}  
else {  
  B  
}
```

if p then A else skip end

```
if (p) { A }
```

---

new x := e in C

nova lokalna spremenljivka  
x z začetno vrednostjo e,  
veljavna v ukazu C

```
{  
  int x = e;  
  C  
}
```