

Principi programskih jezikov

- Sintaksa
- aritmetični izrazi
- ukazni prog. j.
- ...
-

Anatomija programskega jezika

- sintaksa: pravila, kako se piše koda
- statična semantika:
preverjanje, ali je program smiselen (spremenljivka i ni definirana)
- dinamična semantika: kako se program izvaja
- analiza: ali deluje pravilno,
koliko pomnilnika bo program abcinat

Aritmetični izrazi

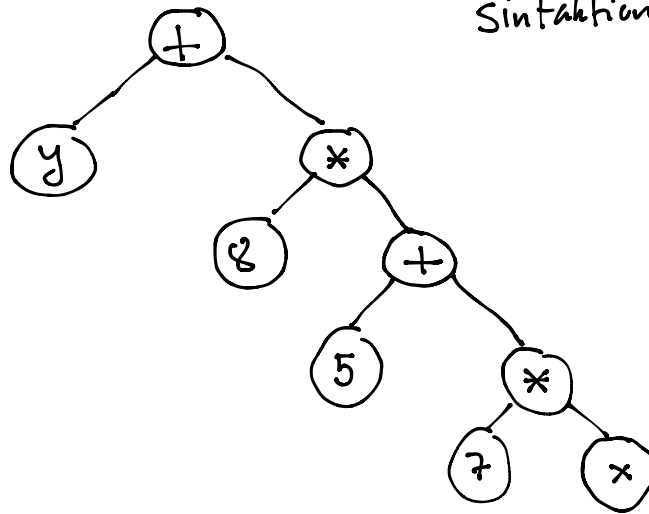
Sintaksa

"y + 8 * (5 + 7 * x)" konkretna sintaksa

Abstraktna sintaksa

izraz / koda je drevo

Sintaktično drevo



Slovnicihna pravila ali gramatika

- pravila, ki opisujejo, kako tvorimo izraze ali drevesa.
- uporabimo stil podajanja pravil BNF

$\langle \text{izraz} \rangle ::= \langle \text{aditivni-izraz} \rangle \text{ EOF}$
↓ definiramo "end of file" (koniec vhoda)

$\langle \text{aditivni-izraz} \rangle ::= \langle \text{multiplikativni-izraz} \rangle \mid \langle \text{aditivni-izraz} \rangle + \langle \text{multiplikativni-izraz} \rangle$
↖ ali ↗ toksem (token)

$\langle \text{multiplikativni-izraz} \rangle ::= \langle \text{osnovni-izraz} \rangle \mid \langle \text{multiplikativni-izraz} \rangle * \langle \text{osnovni-izraz} \rangle$

$\langle \text{osnovni-izraz} \rangle ::= \langle \text{spremenljivka} \rangle \mid \langle \text{številka} \rangle \mid (\langle \text{aditivni-izraz} \rangle)$

$\langle \text{spremenljivka} \rangle ::= [a-zA-Z]^+$ "neprazen niz črk"
+ ena ali več ponovitev

$\langle \text{številka} \rangle ::= -? [0-9]^+$ ↑ "neprazen niz števk"
↑ "neprazen niz števk"
nič ali ena ponovitev znaka -
* nič ali več ponovitev
? nič ali ena

Primer: "x * (5 + 8)"
 (izraz)
 (aditivni izraz)
 (multiplikativni izraz)

x * (5 + 8)
 <multiplikativni-izraz> * <osnovni-izraz>
 (osnovni-izraz)
 (spremenljivka)
 ✓
 (5 + 8)
 (aditivni)
 5 + 8
 (aditivni) + (multiplikativni)
 (multiplikativni) (osnovni)
 (osnovni) (številka)
 (številka) ✓

Ali je to
 $(x+y)+z$
 ali
 $x+(y+z)$

Operacija \star

$$X \star y \star z =$$

① $(X \star y) \star z$ levo asociativna $+, -$

② $X \star (y \star z)$ desno asociativna
 $a^{b^c} = a^{(b^c)}$

Prioriteta:

množenje ima prednost pred seštevanjem

Primer: $X + y + z$
 aditivni

① $X + \underbrace{y + z}_{\text{multiplikativni}}$ NE DELUJE
 aditivni + multiplikativni
 tukaj se zatakne

② $\underbrace{X + y}_{\text{aditivni}} + z$ multiplikativni ✓

Kaj bi se zgodilo, če bi imeli

$$\langle \text{aditivni-izraz} \rangle ::= \langle \text{aditivni-izraz} \rangle + \langle \text{aditivni-izraz} \rangle$$

Potem delujeta obe možnosti ① in ② zgoraj.

Primer: $5 * 8 + 7$ multiplikativni
 $5 * (8 + 7)$
 multiplikativni $*$ osnovni $+$ ni osnovni
 ✓

$(5 * 8) + 7$
 aditivni $+$ aditivni
 multiplikativni $*$ multiplikativni
 ✓

Množenje ima prednost pred seštevanjem

iz konkretne sintakse v abstraktno

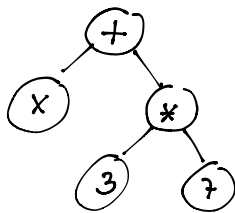
"x + 3 * 7"

leksikalna
analiza

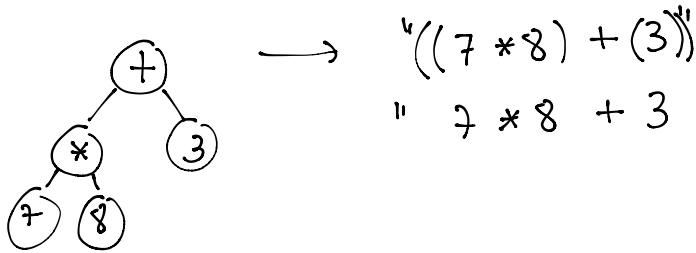
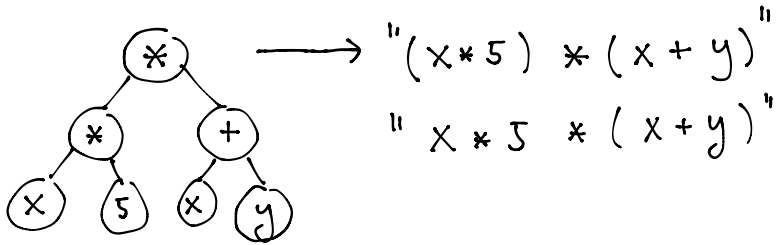
Seznam osnovnih enot (leksemi)

SPREMENLJIVKA(x), PLUS, ŠTEVILKA(3), KRAT, ŠTEVILKA(7), EOF

razčlenjevanje
(parsing)

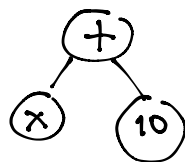
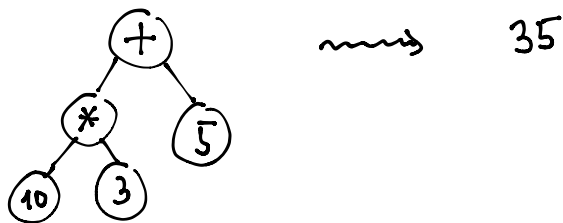


iz abstraktne sintakse v konkretno



Operacijska semantika aritmetičnih izrazov

Kako izračunamo vrednost izraza?



okolje
 $[x \mapsto 5, y \mapsto 7]$

$\rightsquigarrow 15$

$$\eta = [x_1 \mapsto n_1, \dots, x_i \mapsto n_i]$$

preslikava, ki spremenljivke slika v njihove vrednosti

- α
- β
- γ
- δ
- π
- η eta
- μ mi
- θ theta
- ζ zeta
- ξ ksi

Semantika velikih korakov

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

"V okoljih η se izraz e evalvira v vrednost n "
evalvira
izračuna

$$[x \mapsto 10, y \mapsto 5] \mid y+7 \mapsto 20 \quad \text{to ni res}$$

$$[x \mapsto 10, y \mapsto 13] \mid y+7 \mapsto 20 \quad \text{to je res}$$

Podati bomo pravila sklepanja:

$$\frac{P_1 \quad P_2 \quad \dots \quad P_i}{S} \quad \begin{array}{l} \leftarrow \text{predpostavke} \\ \leftarrow \text{sklep} \end{array}$$

Če veljajo P_1, P_2, \dots, P_i , potem velja S