Vježba 5. Konteksno neovisni jezici

Konteksno neovisna gramatika (CFG)

Nejednoznačnost gramatike, jezika i niza

Nejednoznačnost CFG G definiramo:

- ako je za niz $w \in L(G)$ moguće izgraditi više različitih generativnih stabala, gramatika je nejednoznačna
- ako je niz $w \in L(G)$ moguće generirati primjenom više postupaka zamjene krajnjeg desnog ili krajnjeg lijevog znaka, onda je gramatika nejednoznačna
    - gramatika

      $$G = (\{E\}, \{a, \otimes\}, \{E \rightarrow E \otimes E | a\}, E)$$

      je nejednoznačna

Nejednoznačnost niza definiramo

- ako je za niz w moguće izgraditi više različitih generativnih stabala, on je nejednoznačan

Nejednoznačnost jezika definiramo

- ako jezik L nije moguće generirati niti jednom jednoznačnom gramatikom, onda je jezik L nejednoznačan
- primjer nejednoznačnog jezika je:

$$L_n = L_1 \cup L_2 = \left\{ a^n b^n c^m d^m \middle| n \geq 1, m \geq 1 \right\} \cup \left\{ a^n b^m c^m d^n \middle| n \geq 1, m \geq 1 \right\}$$

Nejednoznačnost razrješujemo:

1. Promjenom gramatike
    - umjesto gramatike G izgradi se nova jednoznačna gramatika G'
    - jezik L kojeg generira $G = (\{E\}, \{a, \otimes\}, \{E \rightarrow E \otimes E | a\}, E)$ moguće je generirati više različitih jednoznačnih gramatika:
        - Za lijevo asocijativni $\otimes$ uvodimo gramatiku:

          $G1 = (\{E,T\}, \{a, \otimes\}, \{E \rightarrow E \otimes T | T, T \rightarrow a\}, E)$
        - Za desno asocijativni $\otimes$ uvodimo gramatiku:

          $G2 = (\{E,T\}, \{a, \otimes\}, \{E \rightarrow T \otimes E | T, T \rightarrow a\}, E)$

    Izbor jednoznačne gramatike određuje način gradnje generativnog stabla.

2. Promjenom jezika
    - umjesto gramatike L izgradi se novi jezik L' koji je moguće generirati jednoznačnom gramatikom
    - promjenu jezika primjenjujemo:
        - kada je jezik inherentno nejednoznačan
        - kada je jednoznačna gramatika previše složena

- kada se žele sačuvati sve interpretacije nizova
- primjer promjene jezika je uvođenje zagrada, zagrade su završni znakovi gramatike i dio su niza

Razlika između ove dvije promjene je da se promjenom gramatike ne mijenja jezik i odbacuje se višestruko značenje niza dok se promjenom jezika čuva višestruko značenje niza. Isto tako se definira zaseban niz za svako značenje.

Postupci pojednostavljenja gramatike

- odbacuju se beskorisne znakove i produkcije
- generiramo gramatiku sa tri svojstva:
  (i) bilo koji znak koristi se u makar jednom nizu
  (ii) ne koriste se jedinične produkcije A → B
  (iii) ne koriste se ε-produkcije
- koristimo algoritme
  - odbacivanja beskorisnih znakova
  - odbacivanja jediničnih produkcija i ε-produkcija
  - postizanja normalnih oblika Chomskog i Greibacha

(i) bilo koji znak gramatike G koristi se za generiranje makar jednog niza jezika L

- ako se znak X gramatike G = (V, T, P, S) koristi u postupku generiranja:

$$S \overset{*}{\Rightarrow} \alpha X \beta \overset{*}{\Rightarrow} w \qquad ; \qquad \alpha, \beta \in (V \cup T)^*, \quad w \in T^*$$

onda je X koristan, u suprotnom je **beskoristan**

- dva su vida beskorisnosti:
  - znak je mrtav
  - znak je nedohvatljiv
- ako iz znaka X nije moguće generirati niz završnih znakova, tj. ako ne postoji postupak:

$$\cancel{X \overset{*}{\Rightarrow} w_X} \qquad ; \qquad w_X \in T^*$$

onda je znak X **mrtav**, u suprotnom je živ.

- ako znak X nije ni u jednom nizu koji se generira iz S, tj. ako ne postoji postupak:

$$\cancel{S \overset{*}{\Rightarrow} \alpha X \beta}$$

onda je znak X **nedohvatljiv**.

- neka je X dohvatljiv i živ tj. neka vrijedi:

$$S \overset{*}{\Rightarrow} \alpha X \beta \qquad X \overset{*}{\Rightarrow} w_X \qquad ; \qquad w_X \in T^*$$

- moguće je da jedan od podnizova α ili β sadrži mrtvi znak
- ako u bilo kojem postupku

$$S \overset{*}{\Rightarrow} \alpha X \beta$$

barem jedan podniz sadrži mrtvi znak, X je beskoristan.

(ii) gramatika G nema jediničnih produkcija tipa A → B

- A → B je jedinična produkcija
- sve ostale produkcije uključujući A → a i A → ε nazivaju se nejedinične produkcije.

(iii) ako prazni niz ε nije element jezika L, moguće je izbjeći korištenje produkcija tipa A →ε
- produkcija A → ε naziva se ε-produkcija.
- gramatiku G preuredimo tako da sve produkcije budu oblika A → BC i A → a, te su u normalnom obliku Chomskog.
- gramatiku G preuredimo tako da sve produkcije budu oblika A → aα, α može biti prazan, te su u normalnom obliku Greibacha.

Odbacivanje mrtvih znakova

- neka CFG G = (V, T, P, S) generira neprazan jezik L(G) ≠∅ moguće je izgraditi istovjetnu CFG G' = (V', T', P', S), L(G)=L(G'), koja **nema mrtvih znakova** tako da je:
$$A \overset{*}{\Rightarrow} w \ ; \ w \in T^*$$
- npr. G = ({S, A, B}, {a,b,c,d,e,f}, P, S) P = {S → aSa, S → bAd, S → c, A → cBd, A → aAd, B → dAf}
- nezavršni znakovi A i B su mrtvi znakovi.

Algoritam traženja živih znakova

- ako su živi svi znakovi desne strane produkcije
$$A \to X_1 X_2 \cdots X_n$$
živ je i nezavršni znak s lijeve strane produkcije.
- budući da s desna nema mrtvih znakova, vrijedi:
$$X_i \to w_i ; \ w_i \in T^*$$
- stoga vrijedi:
$$A \to w_1 w_2 \cdots w_n = w$$

Algoritam traženja živih znakova

Algoritam se provodi u tri koraka:
1. U listu živih znakova stave se lijeve strane produkcija koje na desnoj nemaju nezavršnih znakova
2. Ako su s desne strane produkcije isključivo živi znakovi, onda se u listu doda znak s lijeve strane.
3. Ako se lista živih ne može proširiti, svi znakovi koji nisu na listi su mrtvi znakovi

Primjer 1. Algoritam traženja živih znakova

G = ({S, A, B, C}, {a,b,c,d,}, P, S)

1) S → aABS	4) A → cSA	7) B → cSB
2) S → bCACd	5) A → cCC	8) C → cS
3) A → bAB	6) B → bAB	9) C → c

- u listu stavljamo žive znakove: C zbog 9), A zbog 5) i S zbog 2)
- B je mrtav, odbacujemo produkcije i dobijemo:

G = ({S, A, C}, {a,b,c,d,}, P, S)

2) S $\rightarrow$ bCACd	4) A $\rightarrow$ cSA	8) C $\rightarrow$ cS
	5) A $\rightarrow$ cCC	9) C $\rightarrow$ c

Odbacivanje nedohvatljivih znakova

- neka CFG G = (V, T, P, S) generira jezik L(G) $\neq\emptyset$
- moguće je izgraditi istovjetnu CFG G' = (V', T', P', S), L(G)=L(G'), koja **nema nedohvatljivih znakova**

$$X \in V' \cup T' : \quad S \overset{*}{\Rightarrow} \alpha X \beta \;\; ; \;\; \alpha, \beta \in (V' \cup T')^*$$

- npr. G = ({S, A}, {a,b,c}, P, S) P = {S $\rightarrow$ aSb, S $\rightarrow$ c,A $\rightarrow$ bS, A $\rightarrow$ a}
- produkcije sa S na lijevoj strani nemaju A na desnoj, pa je A nedohvatljiv, ostaju samo produkcije iz S

Traženje dohvatljivih znakova

- ako je dohvatljiv nezavršni znak s lijeve strane produkcije

$$A \rightarrow \alpha_1 | \alpha_2 | \alpha_3 \cdots | \alpha_n$$

- onda su dohvatljivi svi završni i nezavršni znakovi s desne strane produkcije
- neka je S početni nezavršni znak, i neka je A dohvatljiv. Onda vrijedi:

$$S \overset{*}{\Rightarrow} \beta A \gamma \Rightarrow \beta \alpha_i \gamma ; \;\; i = 1 \cdots n$$

pa su svi znakovi α_i dohvatljivi.

Algoritam se provodi u tri koraka:

1. U listu dohvatljivih znakova stavi se početni nezavršni znak gramatike.
2. Ako je znak s lijeve strane produkcije dohvatljiv, u listu se dodaju svi znakovi s desne strane produkcije.
3. Ako se lista dohvatljivih ne može proširiti, svi znakovi koji nisu na listi su nedohvatljivi znakovi.

Primjer 2. Algoritam traženja nedohvatljivih znakova

G = ({S, A, B, C, D, E}, {a,b,c,d,e,f,g}, P, S)

1) S $\rightarrow$ aAB	5) B $\rightarrow$ bE	9) C $\rightarrow$ a
2) S $\rightarrow$ E	6) B $\rightarrow$ f	10) D $\rightarrow$ eA
3) A $\rightarrow$ dDA	7) C $\rightarrow$ cAB	11) E $\rightarrow$ fA
4) A $\rightarrow$ e	8) C $\rightarrow$ dSD	12) E $\rightarrow$ g

- u listu stavljamo dohvatljive znakove: S; A, B i a zbog 1), E zbog 2) d i D zbog 3), e zbog 4), b zbog 5), f zbog 6), e zbog 10) i g zbog 12)
- C i c su nedohvatljivi, odbacujemo produkcije pa je: G = ({S, A, B, D, E}, {a,b,d,e,f}, P, S). odbacimo produkcije 7, 8 i 9

Odbacivanje beskorisnih znakova

- primjenom algoritma
  - odbacivanja mrtvih znakova
  - odbacivanja nedohvatljivih znakova
- iz gramatike se izbace svi beskorisni znakovi
- nužno je ići tim redoslijedom (odbaciti prvo mrtve)
- primjer: G = ({S, A, B}, {a}, P, S), P: $S \rightarrow AB|a$ $A \rightarrow a$
  – B je mrtav, ostaje: $S \rightarrow a$ $A \rightarrow a$
  – A je nedohvatljiv, ostaje $S \rightarrow a$
  – obrnuto: A bi bio dohvatljiv, a samo B bi bio mrtav

Neka CFG G = (V, T, P, S) generira jezik L(G) $\neq\emptyset$. Moguće je izgraditi istovjetnu CFG G' = (V', T', P', S), L(G)=L(G'), koja **nema beskorisnih znakova.**

- neka je G_1 nastala odbacivanjem mrtvih znakova iz G
- neka je G_2 nastala odbacivanjem nedohvatljivih iz G_1
- G_2 nema nedohvatljivih znakova i vrijedi: $S \overset{*}{\underset{G_2}{\Rightarrow}} \alpha X \beta$
- kako G_1 i G_2 imaju iste znakove, a G_1 nema mrtvih, onda ni G_2 nema mrtvih znakova pa su svi znakovi u nizu $\alpha X \beta$ živi

$$S \overset{*}{\underset{G_2}{\Rightarrow}} \alpha X \beta \overset{*}{\underset{G_2}{\Rightarrow}} w, \qquad w \in T^*$$

Primjer 3. Odbacivanje beskorisnih znakova

G = ({S, A, B, C}, {a,b,c,d}, P, S), 1) $S \rightarrow ac$ 3) $A \rightarrow cBC$ 5) $C \rightarrow bC$
2) $S \rightarrow bA$ 4) $B \rightarrow aSA$ 6) $C \rightarrow d$

- u listu stavljamo žive znakove: C zbog 6) i S zbog 1)
- A i B su mrtvi, dobijemo: G_1 = ({S, C}, {a,b,c,d,}, P_1, S)
  1) $S \rightarrow ac$ 5) $C \rightarrow bC$ 6) $C \rightarrow d$
- u listu stavljamo dohvatljive znakove: S
- C, b i d su nedohvatljivi, dobijemo: G_2 = ({S}, {a,c,}, P_2, S) 1) $S \rightarrow ac$

Odbacivanje ε-produkcija

- neka CFG G = (V, T, P, S) generira jezik L(G)\{ε}
- moguće je izgraditi istovjetnu CFG G' = (V', T', P', S)
  L(G)=L(G'), koja **nema ε-produkcija** $A \rightarrow \varepsilon$

Primjer 4. Odbacivanje ε-produkcija

G = ({S, A}, {a,b,c}, P, S), 1) $S \rightarrow aASA$ 2) $S \rightarrow b$ 3) $A \rightarrow c$ 4) $A \rightarrow \varepsilon$

- umjesto nezavršnog znaka A definiramo dva: A_{DA} i A_{NE}
- A_{DA} koristimo u produkciji 4, A_{NE} u produkciji 3, zamijenimo:
  1a) $S \rightarrow a\,A_{NE}\,S\,A_{NE}$ 1b) $S \rightarrow a\,A_{NE}\,S\,A_{DA}$ 1c) $S \rightarrow a\,A_{DA}\,S\,A_{NE}$
  1d) $S \rightarrow a\,A_{DA}\,S\,A_{DA}$ 2) $S \rightarrow b$ 3) $A_{NE} \rightarrow cS$ 4) $A_{DA} \rightarrow \varepsilon$

- zamijenimo A_{DA} s ε, odbacimo 4), A_{NE} zamijenimo s A:
  1a) S → aASA 1b) S → aAS 1c) S → aSA 1d) S → aS
  2) S → b 3) A → cS

Algoritam se izvodi u dva osnovna koraka:

1. pronađu se svi nezavršni znakovi koji generiraju prazni niz:

$$A \stackrel{*}{\Rightarrow} \varepsilon$$

   - u listu praznih znakova stave se sve lijeve strane ε-produkcija
   - ako su svi znakovi desne strane u listi, lista se nadopuni lijevom
   - algoritam se nastavlja sve dok se lista može širiti

2. gradi se novi skup produkcija gramatike G'
   - za produkciju iz G: $A \rightarrow X_1\ X_2 \ldots X_n$ dodaju se u G' produkcije $A \rightarrow \xi_1 \xi_2 \ldots \xi_n$
   - oznake ξ i poprimaju vrijednosti:
     - X_i ako je X_i neprazan,
     - X_i ili ε ako je X_i prazan
   - kada svi ξ i poprime vrijednost ε nastaje ε-produkcija koja se NE dodaje u listu produkcija G'
   - ako produkcija na desnoj strani ima k praznih znakova,
     - potrebno je izgraditi 2^k novih produkcija
     - ako je s desne strane neprazni znak, svih 2^k produkcija ostaje
     - ako s desne strane nije neprazni znak, ostaje 2^k - 1 produkcija

Odbacivanje jediničnih produkcija

- neka CFG G = (V, T, P, S) generira jezik L(G)\{ε}
- moguće je izgraditi istovjetnu CFG G' = (V', T', P', S), L(G)=L(G'), koja **nema jediničnih produkcija** A → B
- algoritam se provodi u dva koraka:
  1. u P' stave se sve produkcije iz P koje nisu jedinične
  2. za sve postupke generiranja B iz A

$$A \stackrel{*}{\Rightarrow} B$$

  na osnovu B → α stvore se nove produkcije A → α

Chomskyjev normalni oblik (Chomsky Normal Form, CNF)

- neka CFG G = (V, T, P, S) generira jezik L(G)\{ε}
- moguće je izgraditi istovjetnu CFG G' = (V', T', P', S), L(G)=L(G'), koja **ima sve produkcije** oblika:

  A → BC ili A → a
- pretpostavi se da G nema beskorisnih znakova, ε-produkcija niti jediničnih produkcija
- algoritam pretvorbe u CNF se provodi u tri koraka:

1. u skup P' stave se sve produkcije koje su u CNF, tj.

   A → BC ili A → a

   a u skup V' upišu se svi nezavršni znakovi

2. neka je produkcija gramatike G oblika:

   $A \to X_1\ X_2\ \ldots Xn\ ;\quad A \in V,\ X_i \in V \cup T$

   ako je X_i završni znak $a \in T$,
   - skup nezavršnih znakova proširi se sa $C_a \in V'$
   - skup produkcija proširi se sa $C_a \to a$ koja je u CNF
   - svi završni znakovi a zamijene se sa C_a
   - postupak se nastavlja dok se ne zamijene svi završni znakovi
   - postupak se nastavlja za sve produkcije
3. nakon koraka 2
   - sve su produkcije u P' oblika $A \to a$ ili $A \to B_1 B_2 B_3 \ldots B_m$,
   - a one oblika $A \to BC$ ili $A \to a$ su u CNF produkcije koje s desna imaju 3 ili više znakova
   - mijenjaju se n ovim produkcijama
   - definiraju se novi znakovi $D_1 D_2 D_3 \ldots D_{m-2}$ pa se $A \to B_1 B_2 B_3 \ldots B_m$ zamijeni skupom produkcija:

     $\{A \to B_1\ D_1,\ D_1 \to B_2\ D_2,\ D_2 \to B_3\ D3,\ \ldots\ D_{m-2} \to B_{m-1} B_m\}$

Primjer 5. Konstrukcija normalnog oblika Chomskog

$G = (\{S, A, B\}, \{a,b\}, P, S)$

1) $S \to bA$	3) $A \to bAA$	6) $B \to aBB$
2) $S \to aB$	4) $A \to aS$	7) $B \to bS$
	5) $A \to a$	8) $B \to b$

- produkcije 5 i 8 su u CNF
- definira se C_a i C_b, te dodaju produkcije $C_a \to a$ i $C_b \to b$:

1) $S \to C_b A$	3) $A \to C_b AA$	6) $B \to C_a BB$	9) $C_a \to a$
2) $S \to C_a B$	4) $A \to C_a S$	7) $B \to C_b S$	10) $C_b \to b$
	5) $A \to a$	8) $B \to b$	

- sada su 1, 2, 4, 5, 7, 8, 9 i 10 u CNF, treba razriješiti produkcije 3 i 6
- definira se D_1 i E_1, te dodaju produkcije $D_1 \to AA$ i $E_1 \to BB$:

1) $S \to C_b A$	3a) $A \to C_b D_1$	6a) $B \to C_a\ E_1$	9) $C_a \to a$
2) $S \to C_a B$	3b) $D_1 \to AA$	6b) $E_1 \to BB$	10) $C_b \to b$
	4) $A \to C_a S$	7) $B \to C_b S$	
	5) $A \to a$	8) $B \to b$	

- sada su sve produkcije u CNF

Greibachov normalni oblik (Greibach Normal Form, GNF)

- neka CFG G = (V, T, P, S) generira jezik $L(G)\backslash\{\varepsilon\}$
- moguće je izgraditi istovjetnu CFG G' = (V', T', P', S), L(G)=L(G'), koja **ima sve produkcije** oblika:

  $A \to a\alpha;\qquad \in \alpha\ V^*$
- Koriste se tri postupka:
  - algoritam pretvorbe gramatike u normalni oblik Chomskog
  - algoritam zamjene krajnjeg lijevog nezavršnog znaka
  - algoritam razrješenja lijeve rekurzije