Formalni jezici i jezični procesori I

GRAMATIKE

prof. dr. sc. Sanda Martinčić - Ipšić
smarti@uniri.hr

Primjer

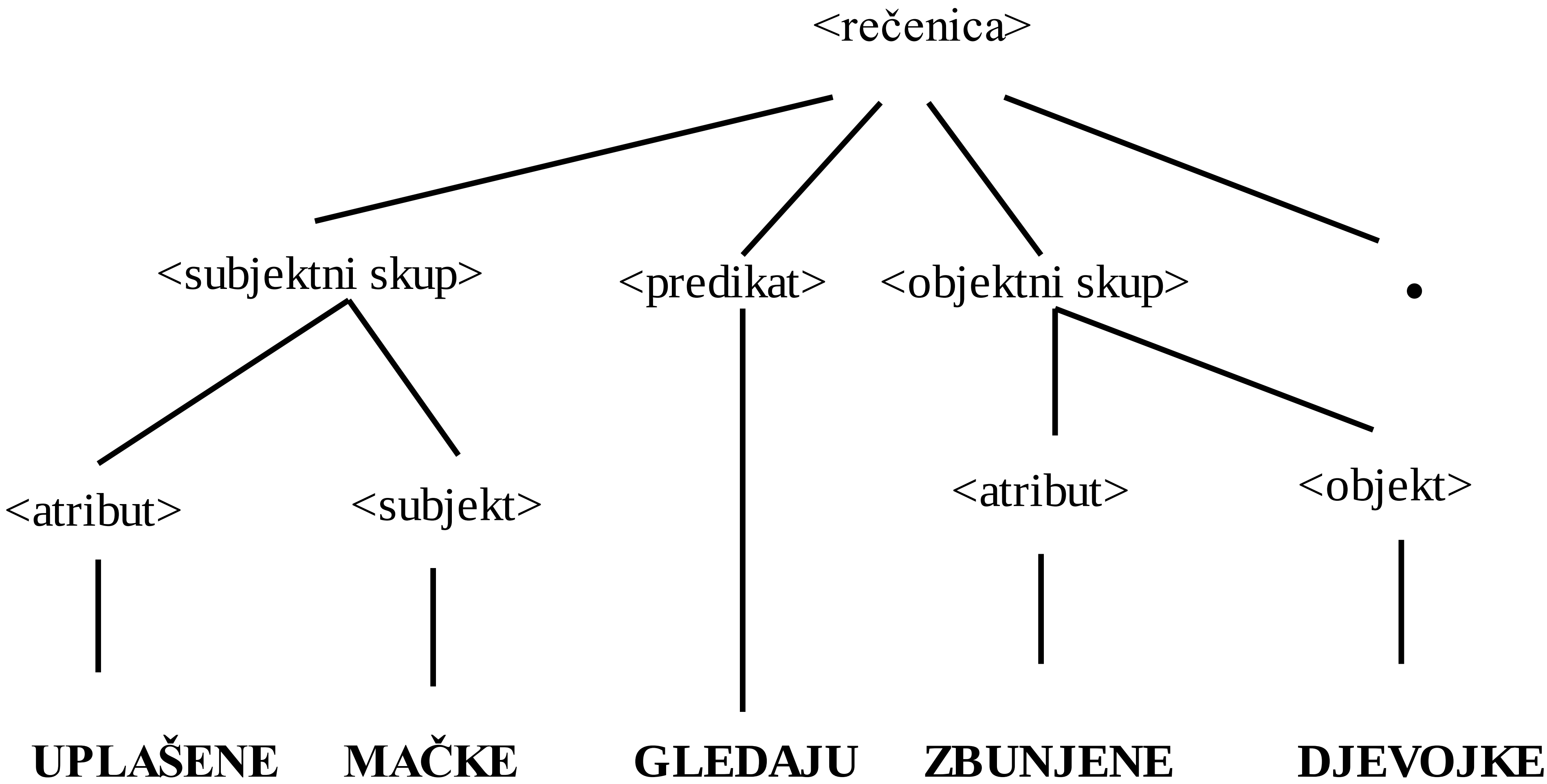
- **element rečenice** (**nezavršni** znakovi): <rečenica>, <subjektni skup>, <predikat>, <objektni skup>, <subjekt>, <objekt>, <atribut>
- **riječi** (**završni** znakovi): GLEDAJU, HRANE, MAČKE, DJEVOJKE, ZBUNJENE, UPLAŠENE, **.**
- **gramatika** (produkcije):

1. <rečenica>→<subjektni skup><predikat><objektni skup>**.**
2. <subjektni skup>→ <atribut><subjekt>
3. <objektni skup>→ <atribut><objekt>
4. < predikat>→ GLEDAJU | HRANE
5. < subjekt >→ MAČKE | DJEVOJKE
6. < atribut >→ ZBUNJENE | UPLAŠENE
7. < objekt >→ MAČKE | DJEVOJKE

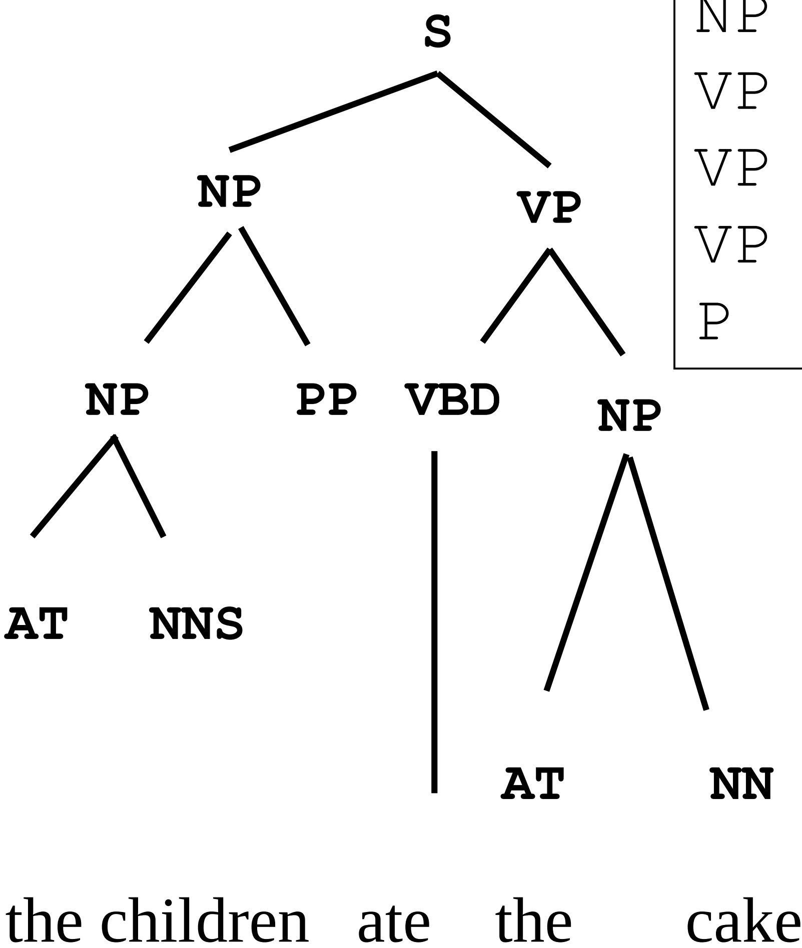
Primjer II

- UPLAŠENE DJEVOJKE GLEDAJU ZBUNJENE MAČKE.
- ZBUNJENE DJEVOJKE GLEDAJU UPLAŠENE MAČKE.
- UPLAŠENE MAČKE GLEDAJU DJEVOJKE.
- ZBUNJENE DJEVOJKE HRANE UPLAŠENE MAČKE.
- DJEVOJKE HRANE ZBUNJENE MAČKE.
- ZBUNJENE MAČKE HRANE UPLAŠENE DJEVOJKE.
- ....

Primjer III


<rečenica>
<subjektni skup>
<predikat>
<objektni skup>
.
<atribut>
<subjekt>
<atribut>
<objekt>
UPLAŠENE
MAČKE
GLEDAJU
ZBUNJENE
DJEVOJKE

Primjer IV



S	→	NP	VP
NP	→	AT	NNS
NP	→	AT	NN
NP	→	NP	PP
VP	→	VP	PP
VP	→	VBD	
VP	→	VBD	NP
P	→	IN	NP

AT	→	*the*
NNS	→	*children*
NNS	→	*students*
NNS	→	*mountains*
VBD	→	*slept*
VBD	→	*ate*
VBD	→	*saw*
IN	→	*in*
IN	→	*of*
NN	→	*cake*

```
NN    /* singular noun */
IN    /* preposition */
AT    /* article */
NP    /* proper noun */
JJ    /* adjective */
NNS   /* plural noun */
VB    /* un-inflected verb */
...
```

From Church (1991) - 79 tags

Gramatika

- formalna gramatika koristi se u generiranju i analizi nizova znakova formalnog jezika
- regularna gramatika generira regularne jezike
- kontekstno neovisna gramatika je uređena četvorka **G=(V, T, P, S)**
  - V – konačni skup nezavršnih znakova;
  - T - konačni skup završnih znakova i $V \cap T = \emptyset$
  - P – konačni skup produkcija (pravila) oblika $A \rightarrow \alpha$;
    - A je nezavršni znak, α je niz znakova skupa $(V \cup T)^*$ i α može biti prazan niz ε;
  - S – početni, nezavršni znak

Oznake

- A, B, C, D, E, S – velika slova koriste se za nezavršne znakove, S – početni nezavršni znak
- **a, b, c, d, e** i **brojke** – mala slova i brojke su završni znakovi (vizualno podebljani)
- X, Y, Z su završni ili nezavršni znakovi
- u, v, w, x, y i z – mala slova označavaju nizove završnih znakova
- α, β, γ - mala grčka slova označavaju nizove završnih i nezavršnih znakova
- | - znak za više produkcija (pravila) pridruženih istom znaku
  - $A \rightarrow$ **a** i $A \rightarrow$ **b** može se zapisati $A \rightarrow$ **a | b**

Primjer BNF V-I

- definicija sintakse programskih jezika
- BNF (Backus-Naurov format)
- opis jezika zadaje se nizom pravila koja imaju lijevu i desno stranu odvojenu znakom jednakosti
  - ***lijeva strana ::= desna strana***
- lijevu stranu čini točno jedna varijabla
- desnu stranu čini više izraza odvojenih operatorom izbora |
  - desna strana može biti prazna bez znaka (ε-produkcija)
- izraz je niz varijabli i konstanti
  - varijabla v: **<*v*>**

Primjer BNF V-II

Aritmetički izrazi u BNF formatu

$G=(\{E\}, \{a,*,+,(,)\}, \{E \to E+E \mid E \to E*E \mid E \to (E) \mid E \to a\}, E)$

- Provjeri jesu li dani izrazi aritmetički izrazi zadanog prg. jezika?
  - a
  - a*a
  - a+a
  - (a+a)*a
  - ...

Relacija $\underset{G}{\Rightarrow}$

- za gramatiku G=(V, T, P, S) definira se relacija $\underset{G}{\Rightarrow}$ nad

nizovima iz skupova $(V \cup T)^*$

- ako je $A \rightarrow \beta$ produkcija iz P i ako su α i γ iz $(V \cup T)^*$ onda vrijedi:
  - $\boldsymbol{\alpha\, A\, \gamma \underset{G}{\Rightarrow} \alpha\, \beta\, \gamma}$
- $\alpha\, \beta\, \gamma$ generira se neposredno iz $\alpha\, A\, \gamma$ prema produkciji $A \rightarrow \beta$, dok G označava kojoj gramatici pripada primijenjena produkcija

$\overset{*}{\underset{G}{\Rightarrow}}$ Refleksivno i tranzitivno okruženje $\underset{G}{\Rightarrow}$

- neka su $\alpha_1, \alpha_2, ..\alpha_m$ nizovi iz $(V \cup T)^*$ i $m \geq 1$ i

$$\alpha_1 \underset{G}{\Rightarrow} \alpha_2 , \alpha_2 \underset{G}{\Rightarrow} \alpha_3 ,.. \alpha_{m-1} \underset{G}{\Rightarrow} \alpha_m$$

- gramatika G generira niz α_m iz α_1:

$$\alpha_1 \overset{*}{\underset{G}{\Rightarrow}} \alpha_m$$

- ukoliko se zna na koju gramatiku G se relacije odnose G možemo ispustiti te pišemo: $\Rightarrow$ odnosno $*\Rightarrow$

Gramatika i jezik

- gramatika G=(V, T, P, S) generira jezik

$$L(G) = \{ w \mid w \in T^* \text{ za koju vrijedi } S \overset{*}{\underset{G}{\Rightarrow}} w \}$$

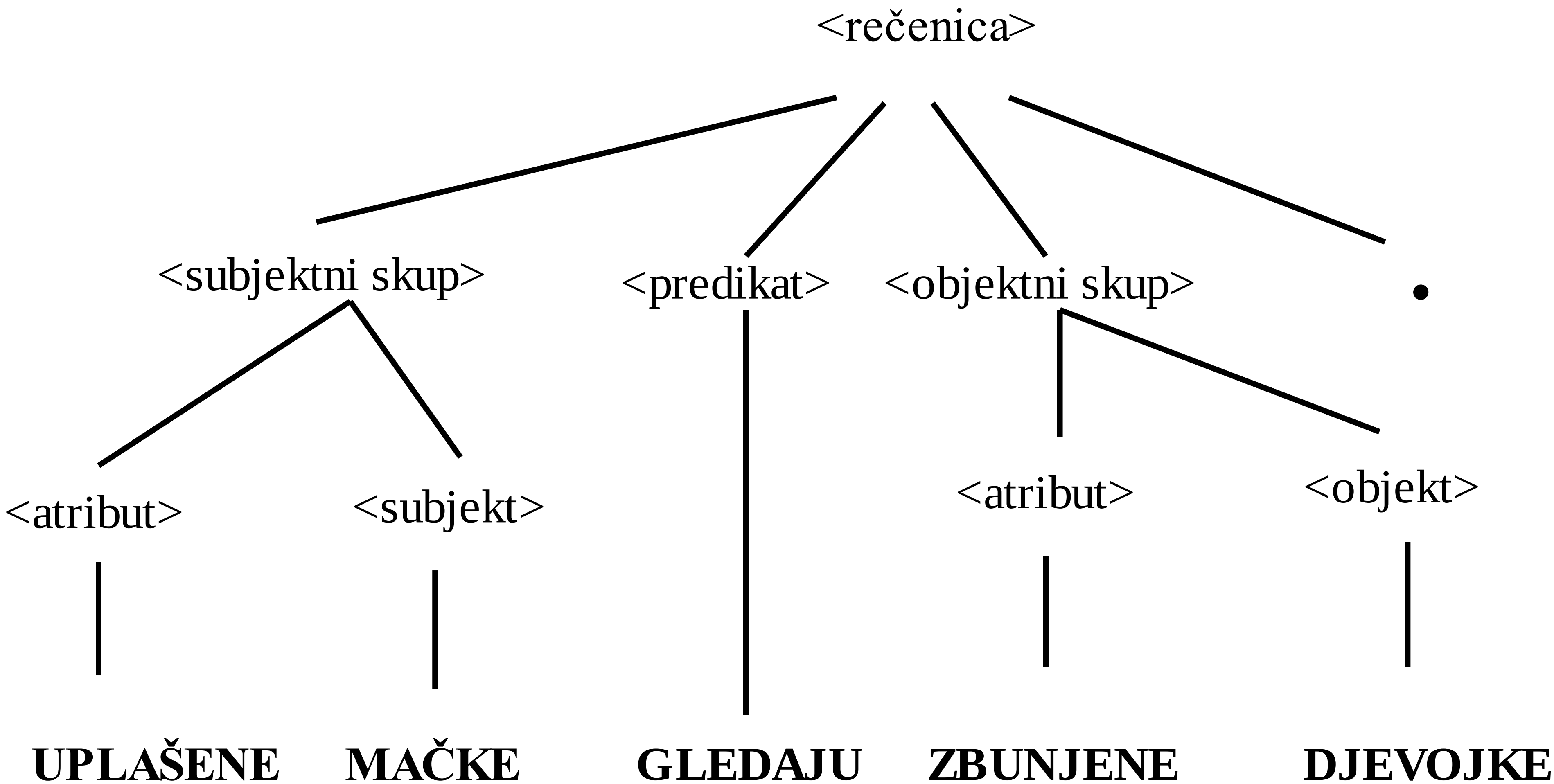
- niz w je u jeziku L(G) ako za niz w vrijedi:
  - u nizu w su isključivo završni znakovi gramatike G
  - niz w je moguće generirati iz početnog nezavršnog znaka S gramatike G
- gramatike G_1 i G_2 su istovjetne ukoliko generiraju iste jezike $L(G_1) = L(G_2)$

Generativno stablo gramatike G

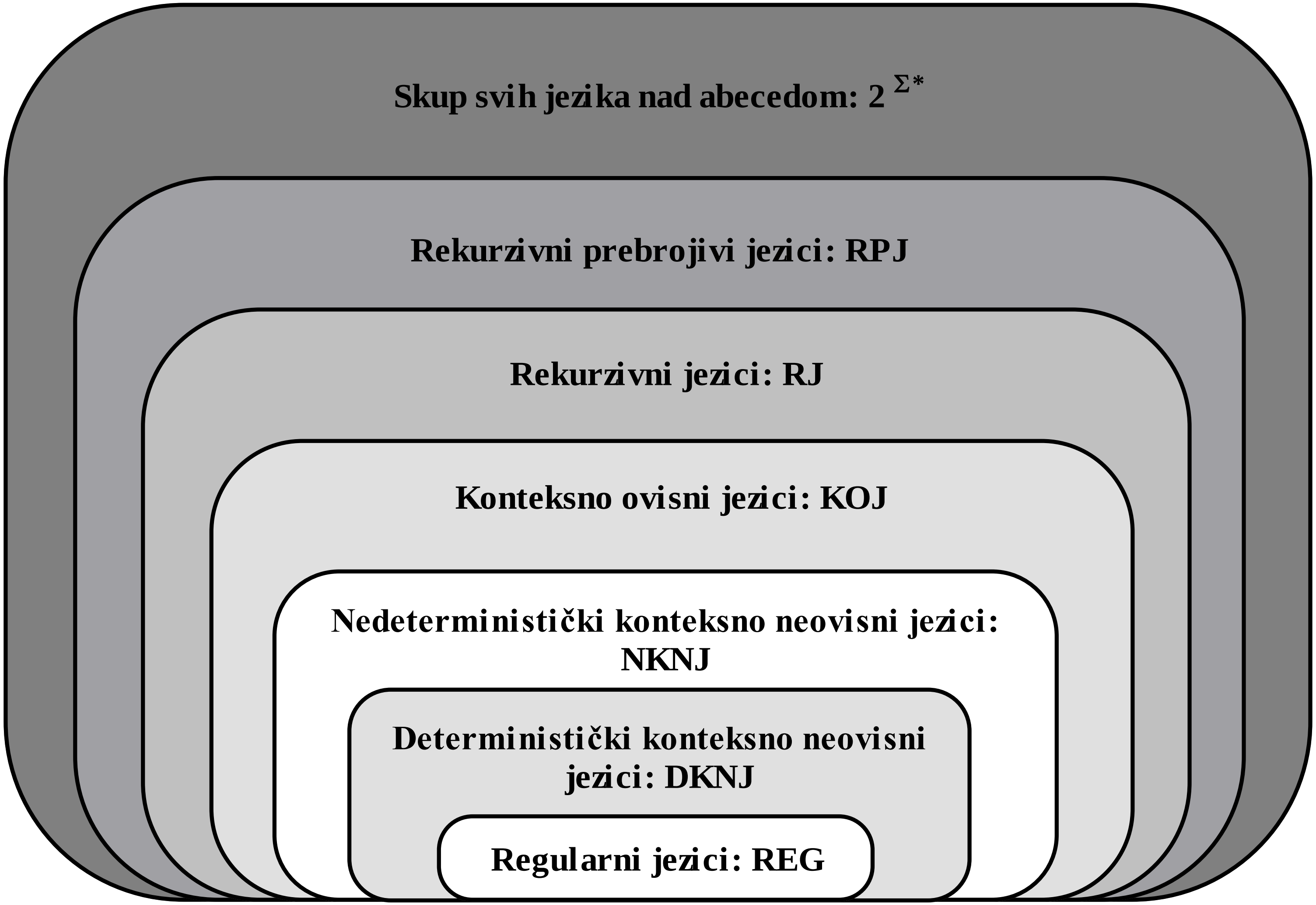
- gramatika G generira niz završnih znakova w i za gramatiku G je moguće izgraditi generativno stablo čiji su listovi isključivo označeni znakovima iz niza w i znakom ε
- čvorovi stabla označeni su znakovima iz $V \cup T \cup \{\varepsilon\}$
- korijen je označen početnim nezavršnim znakom S
- unutrašnji čvorovi su označeni nezavršnim znakovima $A \in V$
- neka su čvorovi $n_1, n_2, .. n_k$ djeca čvora n; ako je n označen znakom A i ako su čvorovi $n_1, n_2, .. n_k$ označeni $X_1, X_2, .. X_k$ onda je

  $A \rightarrow X_1, X_2 .. X_k$ produkcija iz skupa P
- znakom ε moguće je označiti isključivo list stabla; takav list je jedino dijete svojih roditelja odnosno dijete jednog od unutarnjih čvorova
- listovi su označeni znakovima iz skupa $T \cup \{\varepsilon\}$ i čitani s lijeva na desno čine generirani niz jezika L(G)

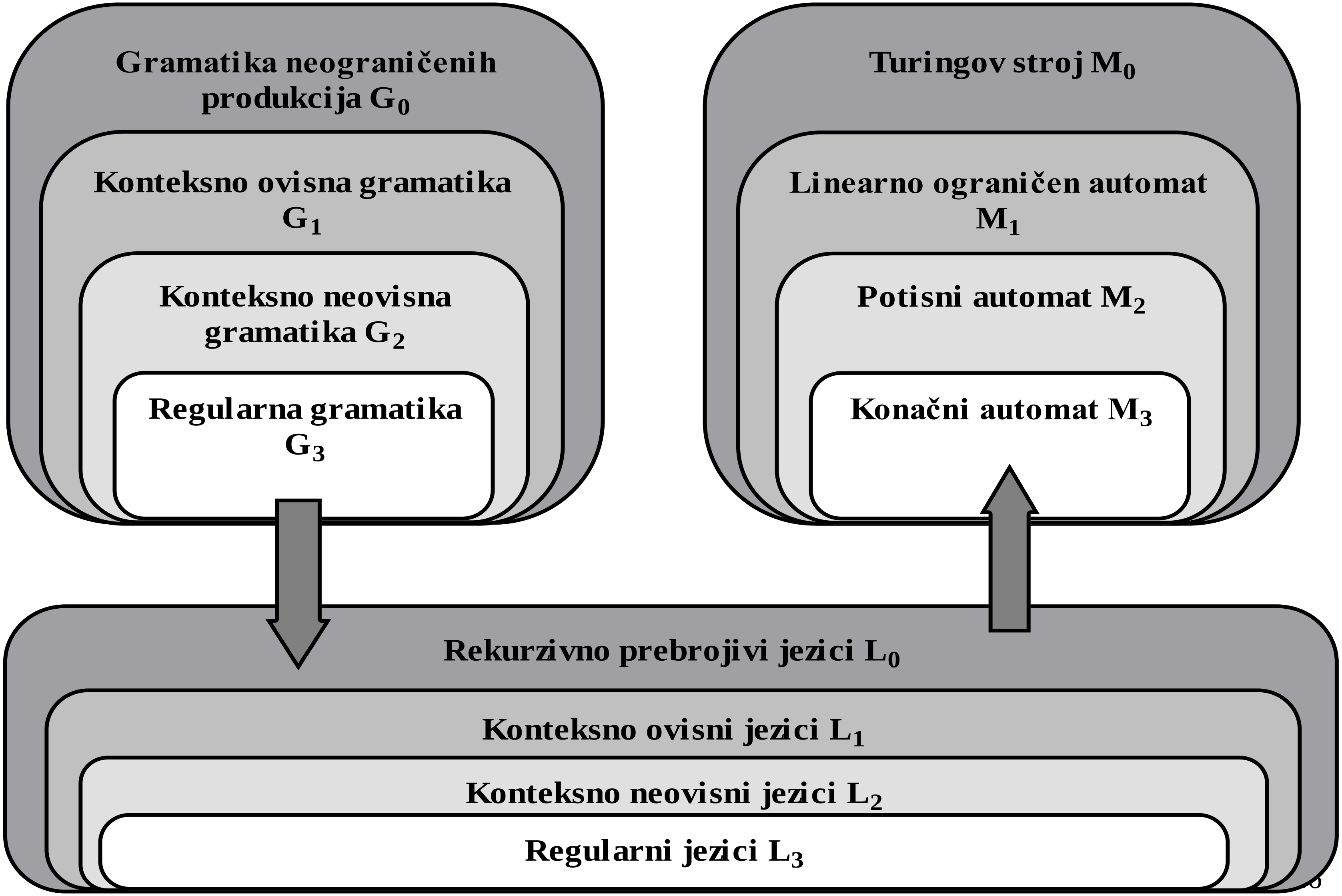
Primjer III


<rečenica>
<subjektni skup>
<predikat>
<objektni skup>
.
<atribut>
<subjekt>
<atribut>
<objekt>
UPLAŠENE
MAČKE
GLEDAJU
ZBUNJENE
DJEVOJKE

Chomskyjeva hijerarhija jezika



Hijerarhija automata i gramatika



Istovjetnost

Gramatika	**Automat**	**Jezik**
Gramatika neograničenih produkcija $G_0=(V,T,P,S)$	Turingov stroj $M_0=(Q,\Sigma,\Gamma,\delta,q_0,B,F)$	Rekurzivno-prebrojiv jezik $L_0= L(G_0)=L(M_0)$
Kontekstno ovisna gramatika $G_1=(V,T,P,S)$	Linearno ograničen stroj $M_1=(Q,\Sigma,\Gamma,\delta,q_0, ¢,\$,F)$	Kontekstno ovisan jezik $L_1= L(G_1)=L(M_1)$
Kontekstno neovisna gramatika $G_2=(V,T,P,S)$	Potisni automat $M_2=(Q,\Sigma,\Gamma,\delta,q_0,Z_0,F)$	Kontekstno neovisan jezik $L_2= L(G_2)=L(M_2)$
Regularna gramatika $G_3=(V,T,P,S)$	Konačni automat $M_3=(Q,\Sigma,\delta,q_0,F)$	Regularan jezik $L_3= L(G_3)=L(M_3)$

DKA => Gramatika

- za regularni jezik zadan s DKA=(Q,Σ,δ,q_0,F) gradi se kontekstno neovisna gramatika G=(V,T,P,S) za koju vrijedi L(G)=L(M):
  - skup završnih znakova T=Σ skupu ulaznih znakova
  - skup nezavršnih znakova V=Q skupu stanja
  - početni nezavršni znak S= q_0 početnom stanju
  - iz prijelaza $\delta(A,a)=B$ iz stanja A u B za ulazni znak a gradi se produkcija $A\rightarrow aB$, A i B su nezavršni a **a** je završni znak
  - za sva prihvatljiva stanja $A \in F$ grade se produkcije $A\rightarrow\varepsilon$, A je nezavršni znak a ε je prazni niz

Regularna gramatika => NKA

- neka su produkcije gramatike G=(V,T,P,S) tipa **A→aB** ili **A→ε**, A i B su nezavršni znakovi, **a** je završni znak za gramatiku G je moguće izgraditi NKA $M=(Q,\Sigma,\delta,q_0,F)$ za kojeg vrijedi L(M)=L(G):
  - skup ulaznih znakova Σ=T skup završnih znakova
  - skup stanja Q=V skup nezavršnih znakova
  - početnom stanju q_0=S početni nezavršni znak
  - iz produkcije A→aB gradi se prijelaz $\delta(A,a)=\delta(A,a)\cup B$
    - npr. početno je $\delta(A,a)=\emptyset$; ako imamo produkcije A→aB i A→aC onda je $\delta(A,a)=\{B, C\}$ (nedeterminizan je dozvoljen)
  - ako je u gramatici produkcija A→ε, A je završno stanje; $A \in F$

Desno-linearna (DL) gramatika

- ukoliko gramatika ima najviše jedan nezavršni znak na desnoj strani i to na krajnje **desnom** mjestu:

  $$\mathbf{A \rightarrow wB} \text{ ili } \mathbf{A \rightarrow w}$$

  – A i B su nezavršni znakovi $\in$ V; w je niz završnih znakova $\in$ T* proizvoljne duljine uključujući i prazan niz

- desno linearna gramatika je regularna gramatika

Lijevo-linearna (LL) gramatika

- ukoliko gramatika ima najviše jedan nezavršni znak na desnoj strani i to na krajnje **lijevom** mjestu:

$$\mathbf{A \rightarrow Bw} \text{ ili } \mathbf{A \rightarrow w}$$

  - A i B su nezavršni znakovi $\in$ V; w je niz završnih znakova $\in$ T* proizvoljne duljine uključujući i prazan niz

- lijevo linearna gramatika je regularna gramatika

DL - gramatika => NKA
LL- gramatika => NKA

- jezik L je regularan ako i samo ako postoji desno linearna gramatika G_D koja generira jezik

  $L=L(G_D)$

  – isto vrijedi i za lijevo linearnu gramatiku $L=L(G_L)$
- DL gramatika ili LL gramatika se preurede tako da su sve produkcije oblika **$A \rightarrow aB$** ili **$A \rightarrow \varepsilon$** i zatim se gradi NKA

Pravila za preuređivanje produkcija

1. $A \rightarrow w$ - w je neprazni niz doda se novi nezavršni znak $[\varepsilon]$ i gradi se produkcije $[\varepsilon] \rightarrow \varepsilon$; $A \rightarrow w$ zamijeni se s $A \rightarrow w[\varepsilon]$
2. $A \rightarrow a_1 a_2 .. a_n B$; $n \geq 1$ zamijene se: ($[a_1 a_2 .. a_n B]$ su novi nezavršni znakovi)

$$A \rightarrow a_1[a_2..a_nB], \quad ...... \quad [a_i..a_nB] \rightarrow a_i[a_{i+1}..a_nB],\ 1<i<n$$

$$[a_2..a_nB] \rightarrow a_2[a_3..a_nB], \quad ...... \quad [a_{n-1}..a_nB] \rightarrow a_{n-1}[a_nB],$$

$$[a_3..a_nB] \rightarrow a_3[a_4..a_nB], \quad [a_nB] \rightarrow a_nB;$$

3. ako je nezavršni znak B jedini znak desne strane produkcije $A \rightarrow B$; onda se izuzmu sve produkcije s jednakom lijevom i desnom stanom $B \rightarrow B$
   – ostanu li produkcije koje imaju različite strane $A \rightarrow B$ one se zamjene s $A \rightarrow y$ za sve kombinacije nezavršnih znakova A i desnih strana produkcija y za koje vrijedi : $A \rightarrow B$ i $B \rightarrow y$

LL Gramatika => ε-NKA

- LL Gramatika G=(V, T, P, S): ε-NKA koji prihvaća L(M')=L(G) gradi se :
  - izgradi se DL gramatika G'=(V', T', P', S'): da se produkcije P LL gramatike G napišu obrnutim redoslijedom $P'=\{A\to\alpha^R|$ $A\to\alpha$ je u skupu P$\}$; gramatika G' generira nizove završnih znakova koji su obrnuto napisani od nizova iz G: $L(G')= L(G)^R$
  - iz DL gramatike G' konstruira se NKA M: $L(M)= L(G')= L(G)^R$
  - iz NKA M izgradi se ε-NKA M': $L(M')=L(M)^R=L(G')^R =L(G)$
    - NKA M ima samo jedno prihvatljivo stanje
    - ako ih ima više doda se novo stanje, koje je jedino prihvatljivo i ε-prijelazi iz svih prihvatljivih stanja
    - početno stanje ε-NKA M' = prihvatljivo stanje NKA M
    - prihvatljivo stanje ε-NKA M' = početno stanje NKA M
    - funkcijama prijelaza se okrene smjer

ε-NKA => LL Gramatika

- ε-NKA M koji prihvaća $L(M)=L^R$
- iz ε-NKA M konstruira se DL gramatika G za koju vrijedi $L(G) = L(M) = L^R$
- desne strane produkcija zapišu se obrnutim redoslijedom
- G' je LL gramatika: $L(G') = L(G)^R = L(G)^R = L$

Nejednoznačnost gramatike

- za isti niz završnih znakova moguće je generirati dva generativna stabla
- primjer gramatika

$G=(\{S,B\},\{if,then,else\},$

$\{S \rightarrow$ if B then S else S | if B then S,

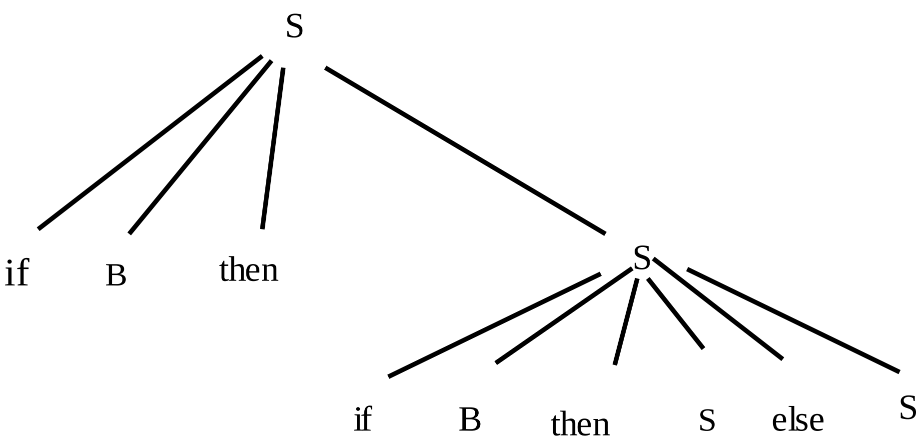
$B \rightarrow$ true | else$\}, S)$

za niz **if B then if B then S else S** moguće je izgraditi dva generativna stabla

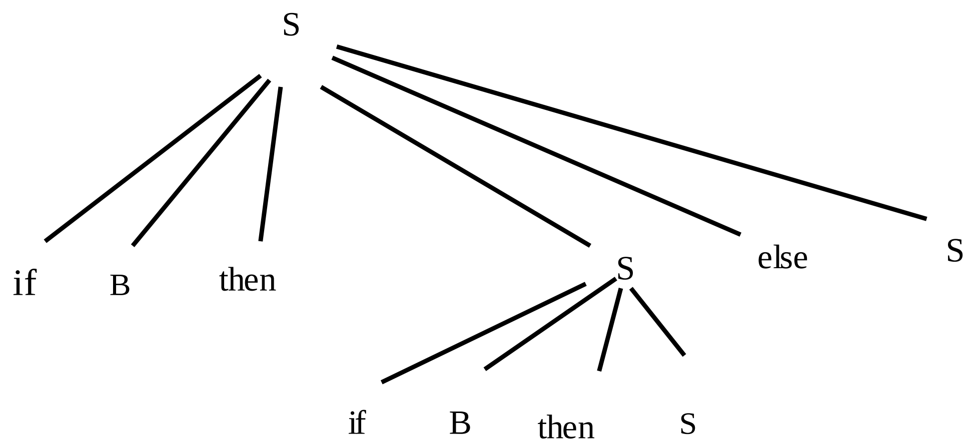
Generativna stabla za isti niz

if B then if B then S else S

- S=>if B then S=>if B then if B then S else S



- S=>if B then S else S => if B then if B then S else S



S→if B then S else S | if B then S

B → true | else

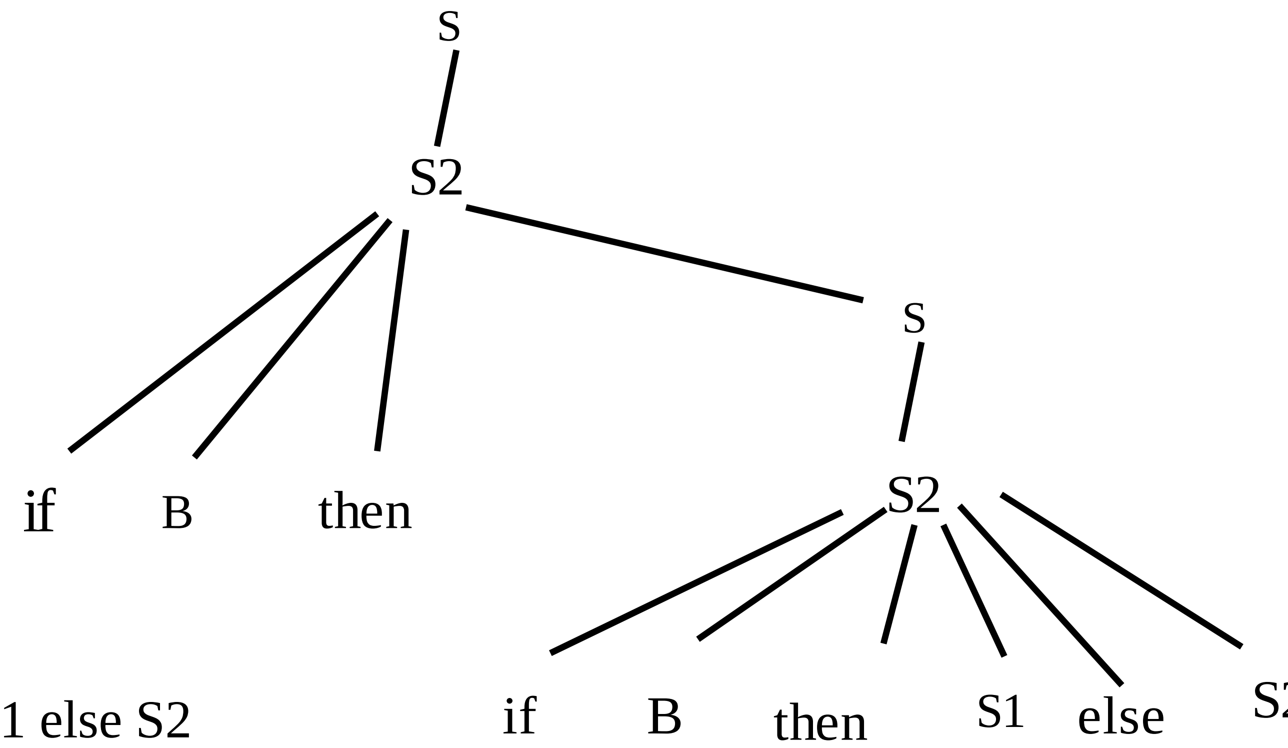
Jednoznačna gramatika

- else se pridružuje bližem if
- G=({S,S1,S2,B},{if,then,else}, P,S}
- P={
  - $S \rightarrow S1|S2$
  - $S1 \rightarrow$ if B then S1 else S2
  - $S2 \rightarrow$if B then S | if B then S1 else S2
  - $B \rightarrow$ true | else

  }

Generativno stablo

- niz **if B then if B then S1 else S2**
- S=>S2=>if B then <u>S</u>=>if B then <u>S2</u> =>if B then if B then S1 else S2



S→S1|S2
S1→ if B then S1 else S2
S2→if B then S | if B then S1 else S2
B → true | else

Pojednostavljenje gramatike

- **odbacivanje beskorisnih znakova**
  - odbacivanje mrtvih znakova
  - odbacivanje nedohvatljivih znakova
- **odbacivanje jediničnih produkcija**
- **odbacivanje ε-produkcija**

- Chomskyev normalni oblik produkcija (CNF)
- Greibachov normalni oblik produkcija (GNF)

KNJ => KNG

- Za neprazni kontekstno neovisni jezik L moguće je izgraditi kontekstno neovisnu gramatiku sa svojstvima:
  - bilo koji znak gramatike G, koristi se u postupku generiranja barem jednog niza jezika L
    - **koristan** i beskoristan: **živ** i mrtav te **dohvatljiv** i nedohvatljiv znak
  - gramatika G **nema** jediničnih produkcija $A \rightarrow B$, A i B su nezavršni znakovi
    - $A \rightarrow B$ su jedinične produkcije; sve ostale su nejedinične
  - ako prazni niz ε nije element jezika L, onda je moguće **izbjeći** korištenje ε-produkcija: $A \rightarrow \varepsilon$

Odbacivanje mrtvih znakova: Algoritam traženja živih znakova

- ako su živi svi znakovi $X_1,X_2..X_k$ desne strane produkcije $A \rightarrow X_1,X_2..X_k$ onda je živ i znak A s lijeve stane

- u listu živih znakova stave se lijeve stane produkcija koje na desnoj stani nemaju nezavršne znakove
- ako su na desnoj stani isključivo živi znakovi, onda se nezavršni znak lijeve strane doda u listu živih
- ako listu živih nije moguće proširiti stani
  - svi znakovi koji nisu u listi su mrtvi i možemo ih odbaciti tako da odbacimo sve produkcije u kojima nastupaju

Odbacivanje nedohvatljivih znakova: Algoritam traženja dohvatljivih znakova

- ako je A dohvatljiv nezavršni znak lijeve strane produkcije $A \rightarrow \alpha_1 | \alpha_2 | .. | \alpha_m$ onda su dohvatljivi svi završni i nezavršni znakovi u nizovima $\alpha_1 | \alpha_2 | .. | \alpha_m$ s desne stane produkcije

- u listu dohvatljivih znakova stavi se početni nezavršni znak gramatike
- ako je znak lijeve stane produkcije u listi dohvatljivih znakova onda se svi znakovi desne strane produkcije dodaju u listu dohvatljivih znakova
- ako listu nije moguće proširiti algoritam se zaustavlja
  - svi znakovi koji nisu u listi su nedohvatljivi i možemo ih odbaciti tako da odbacimo sve produkcije u kojima nastupaju

Odbacivanje beskorisnih znakova

- **prvo** se odbace mrtvi znakovi
- **zatim** se odbace svi nedohvatljivi znakovi
- **redoslijed je bitan!**
  - ako se redoslijed zamjeni: nije nužno da će se odbaciti svi beskorisni znakovi

Odbacivanje ε-produkcija

- neka gramatika G=(V, T, P, S) generira kontekstno neodvisni jezik L(G)-{ε}. Moguće je izgraditi istovjetnu gramatiku G'=(V', T', P', S') koji nema ε-produkcija

Algoritam za odbacivanje ε-produkcija

- pronađu se svi nezavršni znakovi koji generiraju prazan niz (prazni znakovi):
  - u listu praznih znakova (PZ) se stave lijeve stane svih ε-produkcija
  - ako su svi znakovi desne stane produkcija u listi PZ onda se i lijeve strane stave u listu PZ i to se ponavlja sve dok je moguće proširiti listu praznih znakova novim nezavršnim znakom
- produkcijama oblika $A \rightarrow X_1, X_2 .. X_n$ dodaju se produkcije oblika $A \rightarrow \xi_1, \xi_2 .. \xi_n$: (za svaki znak se raširi: 1. je ε i 2. nije ε)
  - X_i nije prazan znak onda je $\xi_i = X_i$
  - ako je X_i prazan znak onda je $\xi_i = \varepsilon$ ili X_i
  - produkcije se grade na osnovu svih mogućih kombinacija vrijednosti ξ_i ako ξ_i poprime sve vrijednosti ε onda se te produkcije odbace

Odbacivanje jediničnih produkcija $A \rightarrow B$

- u skup produkcija stave se sve nejedinične produkcije
- postupkom generiranja iz nezavršnog znaka A dobijemo nezavršni znak B
  - za sve produkcije $B \rightarrow \alpha$ koje nisu jedinične grade se nove produkcije $A \rightarrow \alpha$
  - $A \rightarrow B$ i $B \rightarrow \alpha \Rightarrow A \rightarrow \alpha$

Chomskyjev normalni oblik produkcija (CNO)

- neka gramatika G=(V, T, P, S) generira kontekstno neodvisni jezik L(G)\{ε}. Moguće je izgraditi istovjetnu gramatiku G'=(V', T', P', S') koja ima sve produkcija oblika **A→BC** ili **A→a**
- znakovi A, B, C su nezavršni znakovi gramatike a znak **a** je završni
- gramatika G nema:
  - beskorisnih znakova,
  - ε-produkcija i
  - jediničnih produkcija

Algoritam pretvorbe u CNO

- u skup produkcija P' uvrste se sve produkcije u CNO: $A \rightarrow BC$ ili $A \rightarrow a$ i u skup V' se uvrste svi nezavršni znakovi
- ako je $\mathbf{X_i}$ završni znak **a** u produkciji $A \rightarrow X_1,X_2..X_m$ onda se skup produkcija P' proširi produkcijom $\mathbf{C_a \rightarrow a}$ i dobije se $A \rightarrow C_a,X_2..X_m$
  - postupak se ponovi za sve završne znakove na desnoj strani produkcije
  - postupak se ponovi za sve produkcije oblika $A \rightarrow X_1,X_2..X_m$
- sve nove produkcije koje nisu u CNO odnosno koje su oblika $A \rightarrow B_1,B_2..B_m$ $m>=3$ zamijene se novim produkcijama:
  - $A \rightarrow B_1D_1$; $D_1 \rightarrow B_2D_2$; $D_2 \rightarrow B_3D_3$;... $D_{m-2} \rightarrow B_{m-1}B_m$.

Greibachov normalni oblik produkcija (GNO)

- neka gramatika G=(V, T, P, S) generira kontekstno neodvisni jezik L(G)\{ε}. Moguće je izgraditi istovjetnu gramatiku G'=(V', T', P', S') koja ima sve produkcija oblika **A→aα;** **a** je završni znak a **α niz nezavršnih znakova koji može biti i prazan**
- najprije se gramatika pretvori u CNO
- zatim se izvede algoritam zamjene krajnjeg lijevog nezavršnog znaka i algoritam razrješenja lijeve rekurzije
- izvede se pretvorba u GNO

Algoritam zamjene krajnje lijevog nezavršnog znaka

- neka je u gramatici G r produkcija:
    - $D_j \to \alpha_1$;
    - $D_j \to \alpha_2$; ....
    - $D_j \to \alpha_r$; i produkcija $D_i \to \mathbf{D_j} \gamma$;
  - α i γ su nizovi završnih i nezavršnih znakova
- tih r+1 produkcija se zamijeni sa r produkcija:
  - iz produkcije $D_i \to D_j \gamma$ nezavršni znak D_j se zamijeni sa desnim stranama svih r produkcija $D_j \to \alpha_i$
    - $D_i \to \alpha_1 \gamma$;
    - $D_i \to \alpha_2 \gamma$; ....
    - $D_i \to \alpha_r \gamma$;
    - a produkciju $D_i \to D_j \gamma$ se ispusti

Algoritam razrješenja lijeve rekurzije

- produkcija je rekurzivna ukoliko je isti nezavršni znak na lijevom i krajnje lijevom mjestu desne strane produkcije
- r lijevo rekurzivnih produkcija: $D_i \rightarrow D_i\, \alpha_k$; $1 \leq k \leq r$ i s ne lijevo rekurzivnih produkcija: $D_i \rightarrow \beta_l$ $1 \leq l \leq s$ zamjene se slijedećim produkcijama:
  - $D_i \rightarrow \beta_l$; $1 \leq l \leq s$
  - $D_i \rightarrow \beta_l\, C_i$; $1 \leq l \leq s$
  - $C_i \rightarrow \alpha_k$; $1 \leq k \leq r$
  - $C_i \rightarrow \alpha_k\, C_i$; $1 \leq k \leq r$

Algoritam pretvorbe u GNO

- produkcije se preurede u CNO; sve produkcije oblika $\mathbf{D_i \to a}$ su u GNO i dalje ih ne preuređujemo
- produkcije oblika $D_i \to D_j D_k$ preurede se u $D_i \to D_j\, \beta$ za sve produkcije gdje vrijedi j>i gdje je β niz nezavršnih znakova:
  - postupak započnemo pri D_1 i onda nastavljamo za $D_2\, D_3 \ldots D_m$
- produkcije oblika $D_i \to D_j\, \beta$ preurede se u $D_i \to a\, \alpha\, \beta$
  - postupak započinje znakom $D_{m-1}\, D_{m-2} \,..\, D_1$
- preurede se produkcije koje imaju C_i na lijevoj stani (nastale prilikom razrješenja lijeve rekurzije)

LITERATURA

- S. Srbljić: *Jezični procesori I + II*, Element, Zagreb, 2002.
- J.E. Hopcroft, J.D. Ullman: *Introduction to Automata Theory*, *Languages and Computation*, Addison-Wesley, USA,1979.
- A.V. Aho, R. Sethi, J.D. Ullman: *Compilers Principles*, *Techniques and Tools*, 1987.
- Michael Sipser, *Introduction to the Theory of Computation, second edition*, Course Technology, MIT, 2005.