Formalni jezici i jezični procesori I

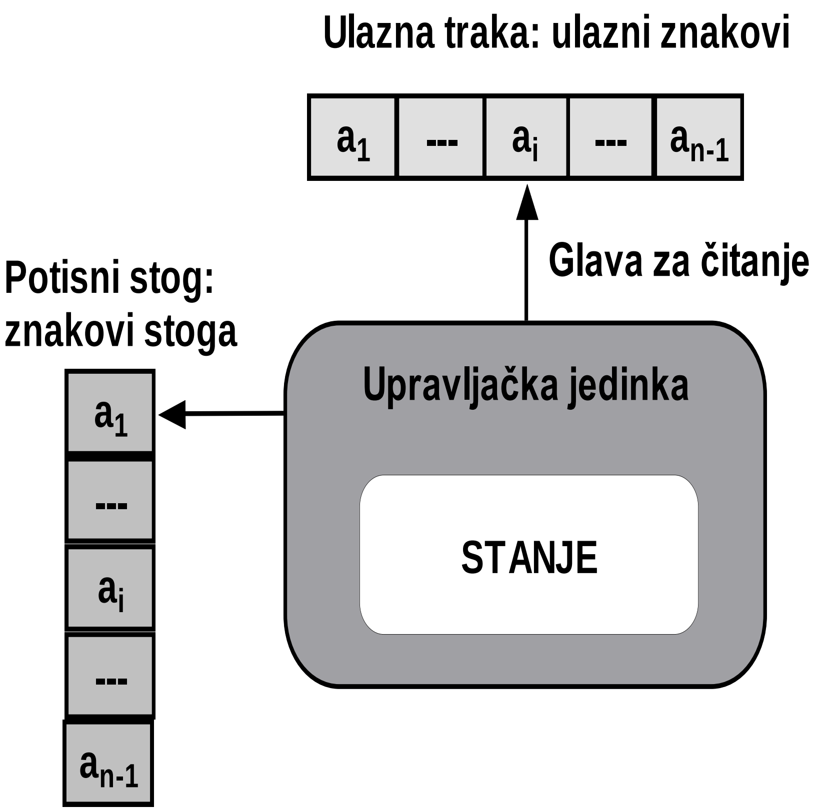
POTISNI AUTOMAT

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

Kontekstno neovisna gramatika i potisni automat

- kontekstno neovisna gramatika i potisni automat su **istovjetni**
  - prihvaćaju klasu kontekstno neovisnih jezika (KNJ)
- jezik je kontekstno neovisan ako i samo ako postoji potisni automat koji ga prihvaća

Model potisnog automata (PA)



- ulaznoj traci, upravljačkoj jedinki i glavi za čitanje dodaje se potisni stog (LIFO stog)
- glava za čitanje čita ulazni znak sa trake i znak sa vrha potisnog stoga
- upravljačka jedinka nalazi se u jednom od konačnog broja stanja:
  - prihvatljivih ili neprihvatljivih
- PA nakon čitanja znaka s vrha stoga i s ulazne trake
  - s vrha stoga uzima pročitani znak na njegovo mjesto upisuje novi znak ili niz znakova
  - te se glava pomiče za jedno mjesto u desno na ulaznoj traci
- ulazna traka je konačna

Rad PA

- upravljačka jedinka donosi odluku o promjeni sadržaja vrha stoga, pomaku glave za čitanje i promjeni stanja na osnovu:
  - stanja upravljačke jedinke
  - znaka na vrhu stoga i
  - pročitanog znaka na traci

Rad PA II

- upravljačka jedinka može donijeti odluku o promjeni na osnovu stanja, ulaznog znaka i znaka na vrhu stoga (3 od 3)
  - tada se glava za čitanje miče za jedno mjesto u **DESNO**
- upravljačka jedinka može donijeti odluku o promjeni i samo na osnovu stanja i znaka na vrhu stoga (2 od 3)
  - tada se glava za čitanje **NE** miče za jedno mjesto u **desno**

Rad PA III

- upravljačka jedinka odlučuje koji niz se stavlja na vrh stoga, a može se staviti:

1. **prazni niz ε = uzimanje znaka sa stoga** (čitanjem se znak uzima sa stoga a umjesto njega se zapiše prazni ε)
2. **niz duljine 1 znaka = zamjena znakova na vrhu** (ako se stavi isti znak koji je pročitan s vrha stoga – nema promjene, ako se stavi drugi znak – zamjena)
3. **niz duljine više znakova = primjena produkcije** (vrh stoga se zamijeni nizom znakova, nezavršni znak lijeve strane zamijeni se nizom znakova desne strane produkcije)

Rad PA IV

- odluka o prihvaćanju niza:
  - **prihvatljivim stanjem**
    - ako pročita sve znakove na ulazu i stane u prihvatljivom stanju
  - **praznim stogom**
    - ako čitanjem svih znakova na ulazu stog ostane prazan

Formalna definicija PA

- uređena sedmorka **pa=(Q, Σ, Γ, δ, q_0, Z_0, F)**
  - Q – konačni skup stanja
  - Σ - konačni skup ulaznih znakova (ulazna abeceda)
  - Γ- konačni skup znakova stoga (abeceda stoga)
  - δ : funkcija prijelaza pridružuje trojki
    - $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma$ konačan podskup skupa svih mogućih parova $Q \times \Gamma^*$
    - podskup partitivnog skupa
  - $q_0 \in Q$; početno stanje
  - $Z_0 \in \Gamma$; početni znak stoga
  - $F \subseteq Q$, skup prihvatljivih stanja

Funkcija prijelaza PA

1. **prijelaz:** na temelju trojke (q,a,Z) PA mijenja stanje u p:
   - $\delta(q,a,Z)=p$, pomiče glavu za 1 mjesto u desno
   - zamijeni znak na vrhu stoga Z nizom znakova γ
2. **ε -prijelaz:** na temelju trojke (q,ε,Z) PA mijenja stanje u p:
   - $\delta(q,\varepsilon,Z)=p$, ostavi glavu na istom mjestu
   - zamijeni znak na vrhu stoga Z nizom znakova γ – prijelaz se izvrši bez čitanja ulaznog znaka sa trake

Funkcija prijelaza PA II

- funkcija prijelaza δ :

$$Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \rightarrow \wp(Q \times \Gamma^*)$$

$$\delta(q, a, Z)=\{(p_1, \gamma_1), (p_2, \gamma_2),..., (p_m, \gamma_m)\}$$

- Z-znak stoga, γ – niz znakova stoga
- nedeterministički
  - nije jednoznačan prijelaz u par (p, γ)
  - već je prijelaz u skup parova

$$\delta(q, a, Z)=\{(p_1, \gamma_1), (p_2, \gamma_2),..., (p_m, \gamma_m)\}$$

- ε -prijelaz

$$\delta(q, \varepsilon, Z)=\{(p_1, \gamma_1), (p_2, \gamma_2),..., (p_m, \gamma_m)\}$$

Primjer PA I

za jezik L={ $0^n 1^n$ | n≥0} gradi se PA M=(Q, Σ, Γ, δ, q1,$, F)

Q={q1,q2,q3,q4}

Σ={0,1}

Γ={0,$}

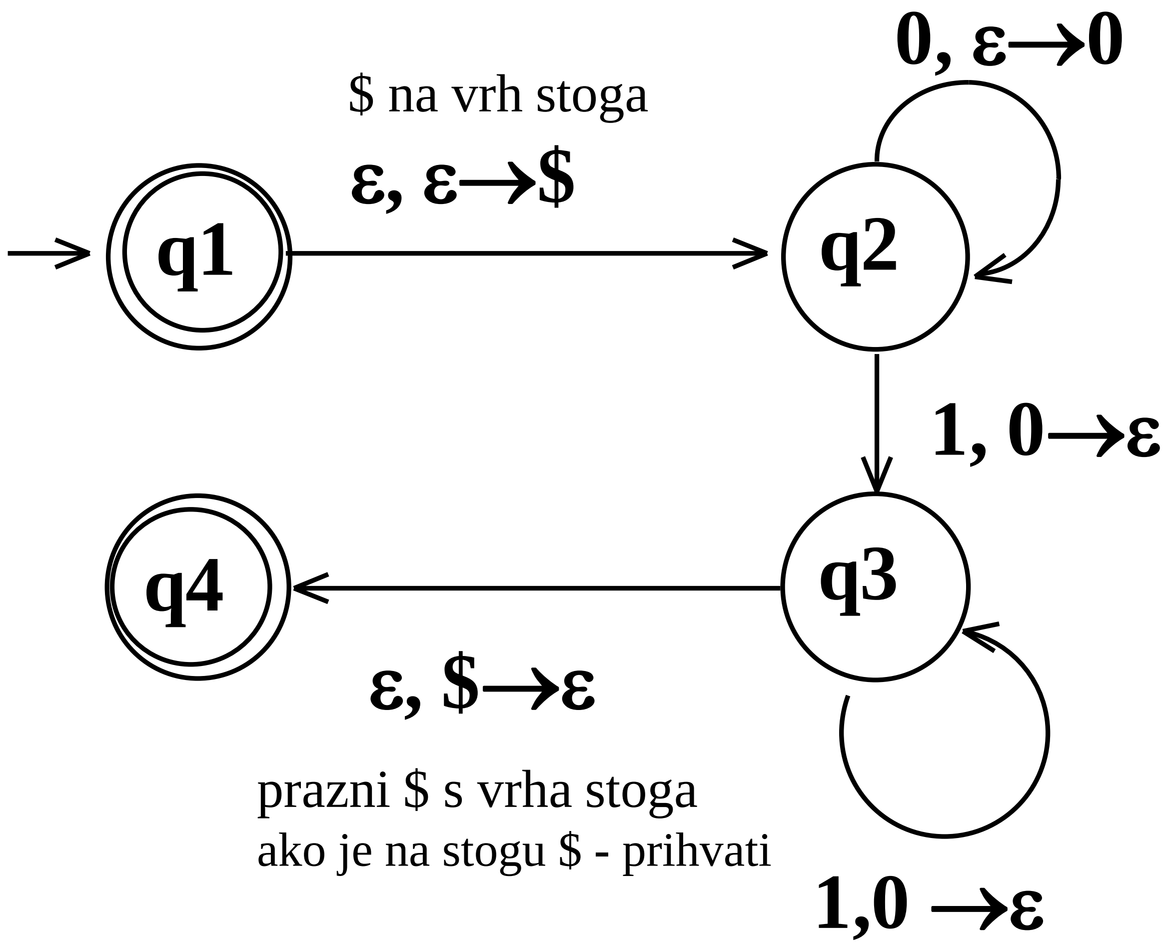
F={q1,q4}

Ulaz:	0			1			ε		
Stog:	0	$	ε	0	$	ε	0	$	ε
q1									{(q2, $)}
q2			{(q2,0)}	{(q3, ε)}					
q3				{(q3, ε)}				{(q4, ε)}	
q4									

DKA nije mogao prihvatiti jezik $0^n 1^n$ (konačan je)

PA može jer ima stog za pohranu (brojanje do n, a n može biti jako velik)

Primjer PA II



dok je na ulazu 0
dodaje 0 na stog
- dodao n 0

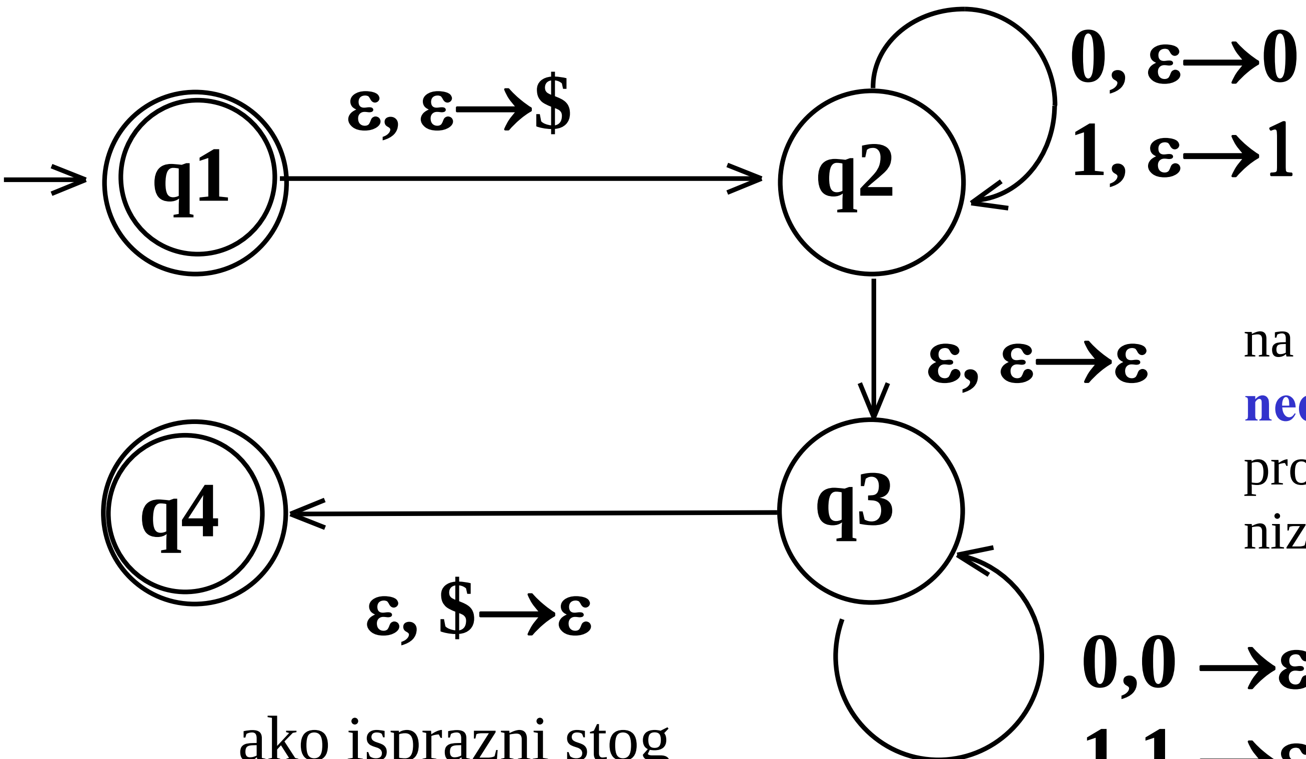
s prvom 1 prazni
0 sa stoga

dok je na ulazu 1
prazni 0 sa stoga

- 1,0→ε ako se na ulazu pročita 1, na vrhu stoga se 0 zamijeni ε

Primjer PA III

za jezik $L=\{ ww^R \mid w \in \{0,1\}^* \}$



na ovom koraku puni stog

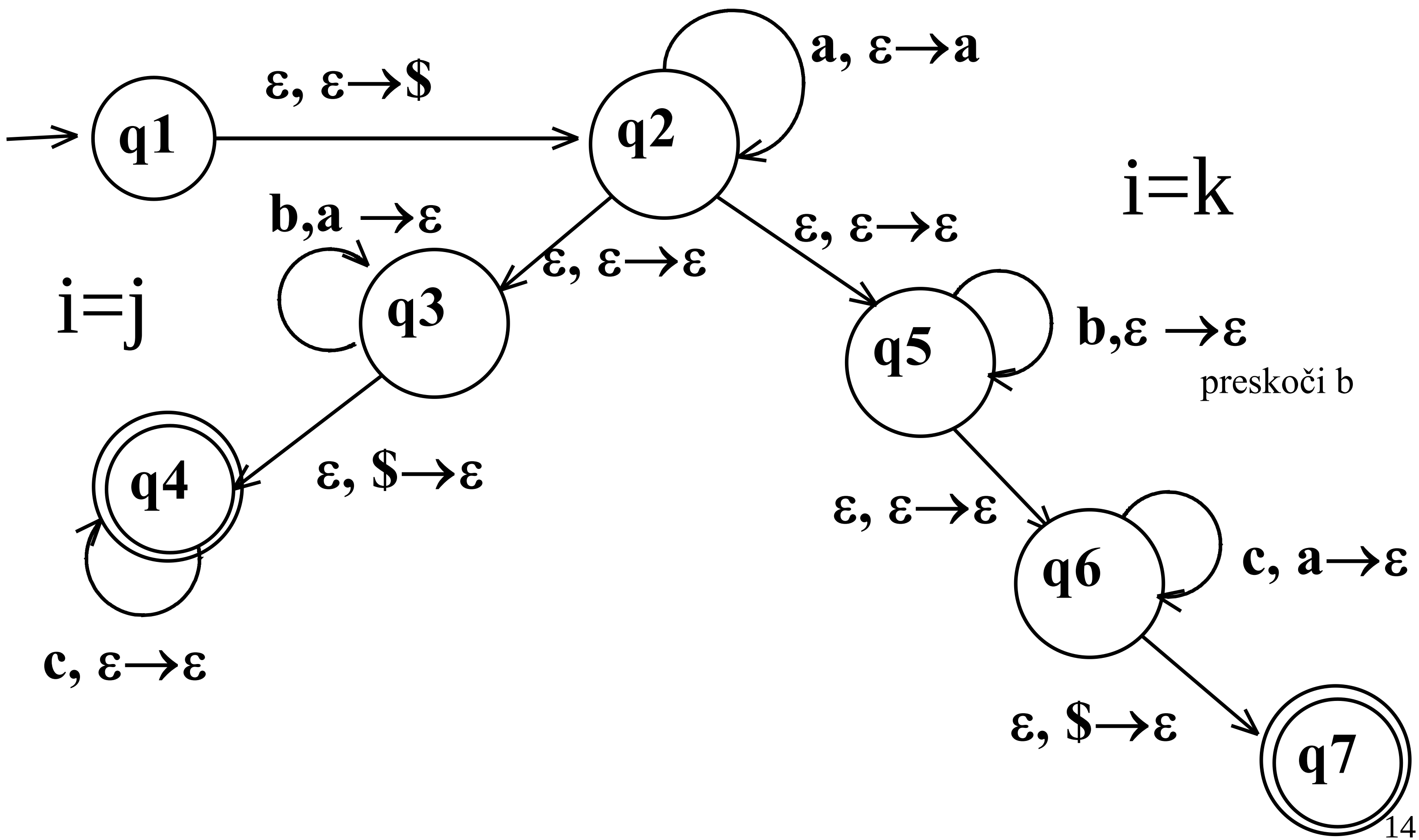
na svakom koraku **nedeterministički** provjerava je li na sredini niza

na ovom koraku prazni stog

ako isprazni stog niz se prihvaća

Primjer PA IV

za jezik $L=\{ a^i b^j c^k \mid i,j,k \geq 0 \text{ i } (i=j \text{ ili } i=k)\}$



Prihvaćanje niza PA

1. **prihvatljivo stanje**: ulaskom upravljačke jedinke PA u prihvatljivo stanje nakon čitanja svih znakova ulazne trake- jezik koji se prihvaća je **L(M)**
2. **prazan stog**: kad se čitanjem svih znakova niza isprazni stog- jezik koji se prihvaća je **N(M)**

$N(M) \neq L(M)$!!!

Konfiguracija PA

- je uređena trojka: (q, w, γ) : q-stanje, w-nepročitani dio ulaznog niza, γ-sadržaj stoga (niz znakova)
- PA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ i konfiguracija $(q,aw,Z\alpha)$ se mijenja u $(p,w,\beta\alpha)$ ako i samo ako skup $\delta(q,a,Z)$ sadrži (p, β) pri čemu je znak a jedan od znakova ulazne abecede Σ ili ε
- promjenu konfiguracije zapisujemo znakom $>$
  – $(q,aw,Z\alpha) > (p,w,\beta\alpha)$ ako i samo ako je par (p, β) iz skupa $\delta(q,a,Z)$
- OZNAKE
  – refleksivno tranzitno okruženje relacije $>$: $>^*$
  – prijelaz iz konfiguracije J u K primjenom **m** prijelaza:

$$J >^{\mathbf{m}} K$$

Primjer: konfiguracije PA

- Zadan je PA M1=(Q, Σ, Γ, δ, q_1, K, q_3)

  Q={q_1, q_2, q_3 }

  Σ={0, 1, 2}

  Γ ={N, J, K} N- Nula, J-jedinica

  – PA prihvaća prihvatljivim stanjem q_3

  – PA prihvaća jezik

L(M1)={w2w^R| w je niz (0+1)*, a w^R je niz w zapisan obrnutim redosljedom}

Primjer: konfiguracije PA II

prijelazi PA

1. $\delta(q1, 0, K)=\{(q_1,NK)\}$
2. $\delta(q1, 1, K)=\{(q_1,JK)\}$
3. $\delta(q1, 0, N)=\{(q_1,NN)\}$
4. $\delta(q1, 1, N)=\{(q_1,JN)\}$
5. $\delta(q1, 0, J)=\{(q_1,NJ)\}$
6. $\delta(q1, 1, J)=\{(q_1,JJ)\}$
7. $\delta(q1, 2, K)=\{(q_2,K)\}$
8. $\delta(q1, 2, N)=\{(q_2,N)\}$
9. $\delta(q1, 2, J)=\{(q_2,J)\}$
10. $\delta(q2, 0, N)=\{(q_2,\varepsilon)\}$
11. $\delta(q2, 1, J)=\{(q_2,\varepsilon)\}$
12. $\delta(q2, \varepsilon, K)=\{(q_3,\varepsilon)\}$

– pokaži da PA prihvaća nizove:
  - 0012100
  - 00200
  - 0021200

Formalno prihvaćanje jezika PA

PA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

1. **prihvatljivim stanjem** prihvaća jezik **L(M)**

$L(M)=\{w \mid (q_0,w,Z_0) \succ^* (p, \varepsilon, \gamma)$ za stanje $p \in F$ i $\gamma \in \Gamma^*\}$

– stanje p mora biti prihvatljivo, a stog ne mora biti prazan

2. **praznim stogom** prihvaća jezik **N(M)**

$N(M)=\{w \mid (q_0,w,Z_0) \succ^* (p, \varepsilon, \varepsilon)$ za stanje $p \in Q\}$

– stanje p ne mora biti prihvatljivo ali se stog mora isprazniti

Deterministički PA

PA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ je deterministički ako i samo ako su ispunjena oba uvjeta

- ako je $\delta(q, \varepsilon, Z)$ **neprazni** skup onda je $\delta(q, a, Z)$ **prazni** skup za bilo koji ulazni znak a iz Σ
  - nema izbora između ε-prijelaza i prijelaza s ulaznim znakom
- u skupu $\delta(q, a, Z)$ je najviše **jedan** element i to za bilo koje stanje q iz Q, za bilo koji znak stoga Z iz Γ i za bilo koji ulazni znak a iz $\Sigma \cup \{\varepsilon\}$
  - samo jedno stanje nakon prijelaza

Istovjetnost PA

Dva PA su istovjetna ako i samo ako prihvaćaju isti jezik.

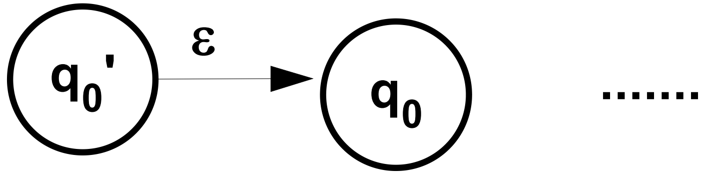
1. konstrukcija PA koji prihvaća praznim stogom iz PA koji prihvaća prihvatljivim stanjem
2. konstrukcija PA koji prihvaća prihvatljivim stanjem iz PA koji prihvaća praznim stogom
3. konstrukcija PA koji prihvaća praznim stogom jezik zadan kontekstno neovisnom gramatikom (KNG$\rightarrow$PA)
4. konstrukcija kontekstno neovisne gramatike za jezik koji se prihvaća praznim stogom zadanog PA (PA$\rightarrow$KNG)

Simulacija

- koriste se dva PA: iz M2 gradi se M1(M2→M1)
  - M1 prihvaća praznim stogom, a M2 prihvatljivim stanjem
  - pri pretvaranju se koristi simulacija (jedan simulira rad drugog)

M2=($Q, \Sigma, \Gamma, \delta, q_0, Z_0, F$) M1=($Q \cup \{q_0', q_e\}, \Sigma, \Gamma \cup X_0, \delta', q_0', X_0, \emptyset$)

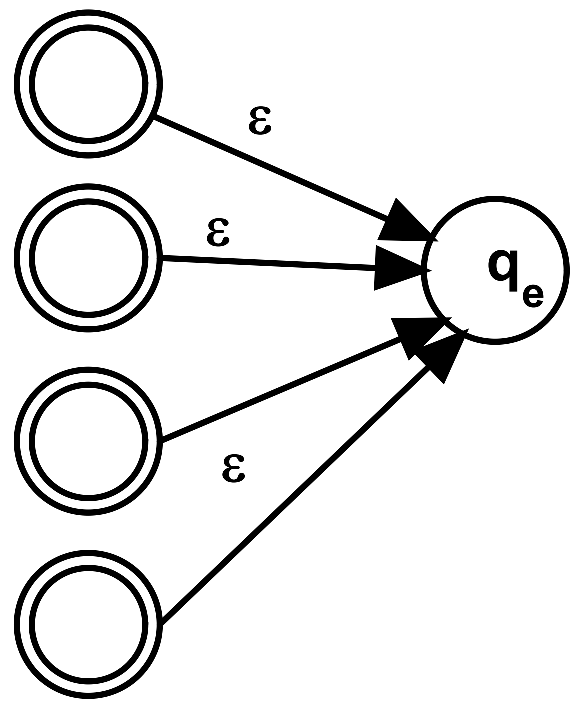
iz početne konfiguracije M2
u početnu konfiguraciju M1



na stog Z_0X_0

uvodi se X0 dodatni znak stoga
osiguranje da M2 ne može prihvatiti
praznim stogom

sva završna stanja M2



Konstrukcija PA koji prihvaća praznim stogom iz PA koji prihvaća prihvatljivim stanjem

iz PA $M2=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ koji jezik L(M2) prihvaća **prihvatljivim stanjem** konstruira se PA M1 koji prihvaća **praznim stogom**:

$$M1=(Q\cup\{q_0',q_e\}, \Sigma, \Gamma\cup X_0, \delta', q_0', X_0, \emptyset)$$

- X0 je dodatni znak stoga kojeg M1 na početku rada stavi na dno stoga
- M2 svojim prijelazima ne može X_0 uzeti sa stoga
- ako M2 isprazni stog a ne uđe u prihvatljivo stanje X_0 na dnu stoga sprječava M1 da prihvati niz

Konstrukcija funkcije prijelaza δ' (4 koraka) **(M2→M1)**

1. $\delta'(q_0', \varepsilon, X_0)=\{(q_0, Z_0X_0)\}$
   - M1 na početku rada prelazi iz svoje početne konfiguracije u početnu konfiguraciju M2 i ostavlja Z_0 na dnu stoga (Z_0X_0)
2. u $\delta'(q, a, Z)$ se stave svi elementi iz $\delta(q, a, Z)$
   – funkcije prijelaza su iste
3. u $\delta'(q, \varepsilon, Z)$ dodaje se ε-prijelaz (q_e, ε), $\forall\ q \in F$
   - **svim prihvatljivim stanjima iz F** dodaju se ε-prijelazi u nezavršno stanje q_e
   - ako M2 uđe u prihvatljivo stanje, skup prijelaza se proširi ε-prijelazom u stanje q_e i s vrha stoga se uzme **jedan znak**
4. u skup $\delta'(q_e, \varepsilon, Z)$ dodaje se ε-prijelaz (q_e, ε)
   - svim znakovima stoga Z iz $\Gamma\cup\{X_0\}$ se dodaju se ε-prijelazi u stanje q_e (znači i za X_0)
   - ε-prijelazi u stanju q_e **prazne čitav stog uključujući i X_0**

Dokaz istovjetnosti M1 i M2

- u 2 dijela:

1. dokaz da PA M1 prihvaća niz x ako ga prihvaća i PA M2
2. dokaz da PA M2 prihvaća niz x ako ga prihvaća i PA M1

pretpostavka: PA M2 prihvaća niz x prihvatljivim stanjem ili $(q_0,x,Z_0) \succ^* (q, \varepsilon, A\gamma)$

gdje je $q \in F$, $A \in \Gamma$ i $\gamma \in \Gamma^*$

Dokaz da PA M1 prihvaća niz x ako ga prihvaća i PA M2

- za početnu konfiguraciju M1 vrijedi (1.korak)
  - $(q_0', x, X_0) \succ (q_0, x, Z_0X_0)$
- na temelju 2.koraka su svi prijelazi iz M1 i prijelazi M2
  - $(q_0, x, Z_0X_0) \succ (q, \varepsilon, A\gamma X_0)$ (pročita niz x, i na stog stavi $A\gamma$)
- na temelju 3. koraka omogućen je prijelaz u stanje q_e (uzima se jedan znak s vrha stoga - A)
  - $(q, \varepsilon, A\gamma X_0) \succ (q_e, \varepsilon, \gamma X_0)$
- 4. korak omogućava pražnjenje stoga (svih znakova γ i X_0)
  - $(q_e, \varepsilon, \gamma X_0) \succ (q_e, \varepsilon, \varepsilon)$ (stog je prazan)

Dokaz da PA M2 prihvaća niz x ako ga prihvaća i PA M1

- M2 prihvaća prihvatljivim stanjem ako M1 prihvaća praznim stogom
- najprije se napravi 1. korak iz početne konfig. M1 u početnu konfig. M2
- zatim se izvrši sljed prijelaza M2 za cijeli ulazni niz x –prijelazi su zajednički s M1
- zatim sljed prijelaza koji prazni stog i to ako i samo ako je u konfiguraciji $(q, \varepsilon, \gamma X_0)$ stanje q prihvatljivo stanje

Konstrukcija PA koji prihvaća prihvatljivim stanjem iz PA koji prihvaća praznim stogom

iz PA $M1=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$ koji prihvaća jezik **praznim stogom** konstruira se PA

$M2=(Q \cup \{q_0', q_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta', q_0', X_0, \{q_f\})$ koji jezik prihvaća **prihvatljivim stanjem**

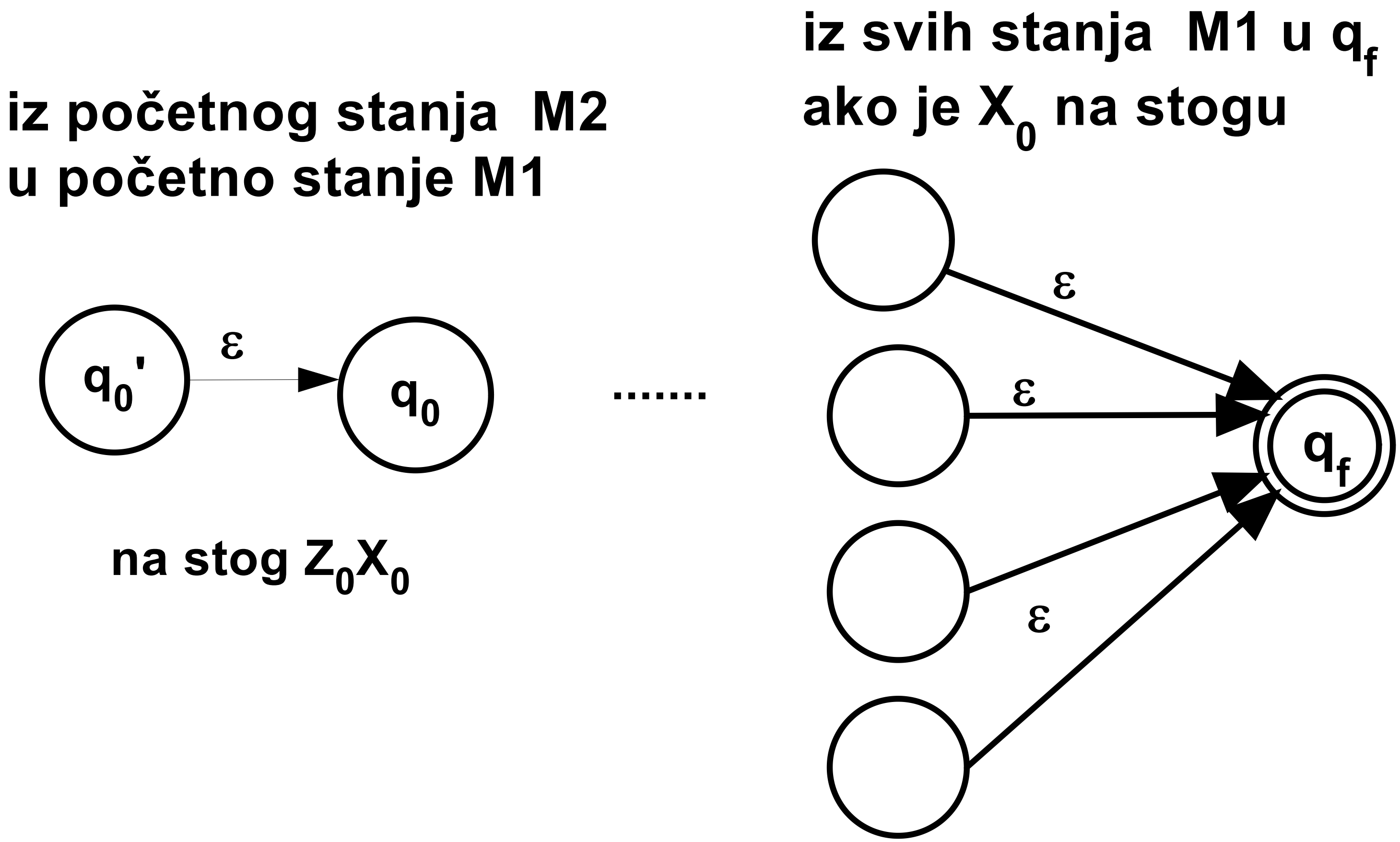
- q_f novo prihvatljivo stanje

- M2 prelazi u prihvatljivo stanje samo ako M1 isprazni stog
- M2 simulira rad M1

Simulacija

- koriste se dva PA: iz M1 se gradi M2 (M1$\rightarrow$M2)
  - pri pretvaranju se koristi simulacija (jedan simulira rad drugog)

  $M1=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, \emptyset)$ $M2=(Q\cup\{q_0',q_f\}, \Sigma, \Gamma\cup X_0, \delta',q_0', X_0, \{q_f\})$



Konstrukcija funkcije prijelaza δ' ($M1 \rightarrow M2$)

1. $\delta'(q_0', \varepsilon, X_0)=\{(q_0, Z_0X_0)\}$
   - M2 na početku rada prelazi iz svoje početne konfiguracije u početnu konfiguraciju M1 i ostavlja Z_0X_0 na dnu stoga
   - ako M1 pročita X_0 znači da je stog prazan
2. u $\delta'(q, a, Z)$ se stave svi elementi iz $\delta(q, a, Z)$
   - za sva stanja $q \in Q$, i sve ulazne znakove $a \in \Sigma \cup \{\varepsilon\}$ i sve znakove stoga $Z \in \Gamma$
   - funkcije prijelaza su iste (tu M2 radi isto kao i M1)
3. u $\delta'(q, \varepsilon, X_0)$ dodaje se ε-prijelaz (q_f, ε)
   - **iz svih stanja q** dodaju se ε-prijelazi u prihvatljivo stanje q_f
   - ako M1 pročita X_0 znači da je ispraznio stog pa M2 uđe u prihvatljivo stanje q_f
   - stanje q_f omogućuje prihvaćanje niza **prihvatljivim stanjem**

Dokaz da PA M2 prihvaća niz x ako ga prihvaća i PA M1

- za početnu konfiguraciju M2 vrijedi prijelaz u M1(1.korak)
  - $(q_0', x, X_0) \succ (q_0, x, Z_0X_0)$
- na temelju 2.koraka su svi prijelazi iz M1 i prijelazi M2
  - $(q_0, x, Z_0X_0) \succ (q, \varepsilon, X_0)$ (pročita niz x)
  - X_0 ostaje na dnu stoga
- na temelju 3. koraka omogućen je prijelaz u stanje q_f
  - $(q, \varepsilon, X_0) \succ (q_f, \varepsilon, \varepsilon)$

Konstrukcija PA koji prihvaća praznim stogom jezik zadan kontekstno neovisnom gramatikom (KNG)

- za kontekstno neovisni jezik zadan gramatikom $G=(V, T, P, S)$ u GNO
  - zapisanom u Graibachovom normalnom obliku (sve produkcije oblika $A \to a\alpha$)
- konstruira se PA

  $M=(\{q\}, \Sigma, \Gamma, \delta, q, S, \emptyset)$ postupkom:

Greibachov normalni oblik produkcija (GNO)

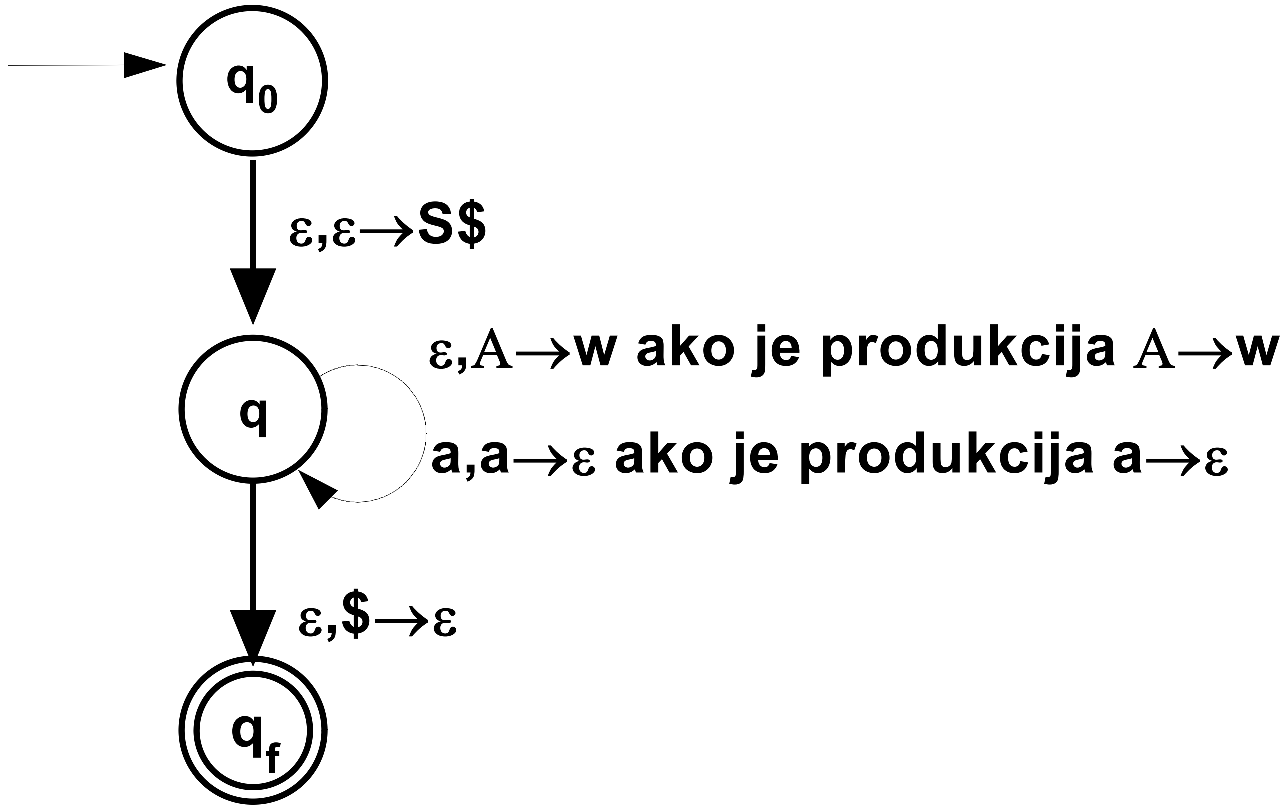
- neka gramatika G=(V, T, P, S) generira kontekstno neodvisni jezik L(G)\{ε}. Moguće je izgraditi istovjetnu gramatiku G'=(V', T', P', S') koja ima sve produkcije oblika **$A \rightarrow a\alpha$**; **a** je završni znak a **α niz nezavršnih znakova koji može biti i prazan**
- najprije se gramatika pretvori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

Konstrukcija PA koji prihvaća praznim stogom jezik zadan KNG II (KNG$\rightarrow$PA)

1. PA ima samo jedno stanje q koje je i početno q_0=q
2. skup ulaznih znakova = skupu završnih znakova Σ=T
3. skup znakova stoga = skupu nezavršnih znakova Γ=V
4. početni znak stoga = početni nezavršni znak gramatike S
5. skup prihvatljivih stanja je prazan F= $\emptyset$
6. PA prihvaća praznim stogom

- **Definicija funkcije prijelaza δ:**
  - iz $A \rightarrow a\gamma$ $\delta(q, a, A)=\{(q,\gamma)\}$
  - iz $A \rightarrow a$ $\delta(q, a, A)=\{(q,\varepsilon)\}$

Primjer: iz produkcija gramatike u automat



Konstrukcija KNG za jezik koji se prihvaća praznim stogom zadanog PA (PA$\rightarrow$KNG)

- za PA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ konstruira se kontekstno neovisna gramatika $G=(V, T, P, S)$ postupkom:
  - nezavršni znakovi gramatike označeni su zagradama $[q,A,p] \in V$
    - gdje su q i p stanja iz Q, a A znak stoga iz skupa Γ

      (sve trojke stanja i svih znakova)
    - u skup nezavršnih znakova dodaje se početni nezavršni znak S
  - a skup produkcija se gradi postupkom:

Konstrukcija KNG za jezik koji se prihvaća praznim stogom zadanog PA II

- **skup produkcija se gradi postupkom:**

1. za početno stanje q_0, početni znak stoga Z_0 i sva stanja $q \in Q$ grade se produkcije: $S \rightarrow [q_0, Z_0, q]$
   - ako je kardinalni broj Q = n onda n produkcija
2. ako skup $\delta(q, a, A)$ sadrži $(q_1, B_1 B_2 .. B_m)$ onda se grade

$$[q,A,q_{m+1}] \rightarrow a[q_1,B,q_2]\ [q_2,B,q_3]..[q_{m-1},B,q_m][q_m,B,q_{m+1}]$$

   - gdje su q, A, a, q1, B1,B2,..,Bm dani funkcijom prijelaza PA
   - za znakove q2,q3,q_m,q_{m+1} uzimaju se sve moguće kombinacije svih stanja iz Q, ima ih n^{m-1}, n kardinalni broj od Q
3. ako $\delta(q, a, A)$ sadrži (q_1, ε) onda se gradi produkcija

$$[q,A,q_1] \rightarrow a$$

Formalni jezici i jezični procesori I

POTISNI AUTOMAT

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

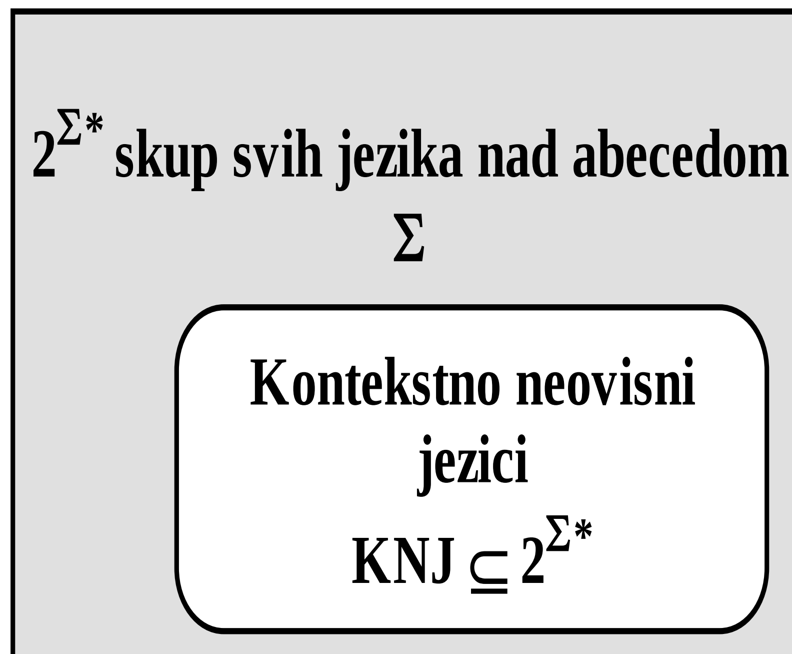
Formalni jezici i jezični procesori I

KONTEKSTNO NEOVISNI JEZICI

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

Istovjetnost: KNG, KNJ i PA

- jezik je kontekstno neovisan ako i samo ako postoji potisni automat koji ga prihvaća
- kontekstno neovisna gramatika (KNG), kontekstno neovisan jezik (KNJ) i potisni automat (PA) su **istovjetni**
  - prihvaćaju klasu kontekstno neovisnih jezika
  - klasa KNJ je pravi podskup klase svih jezika

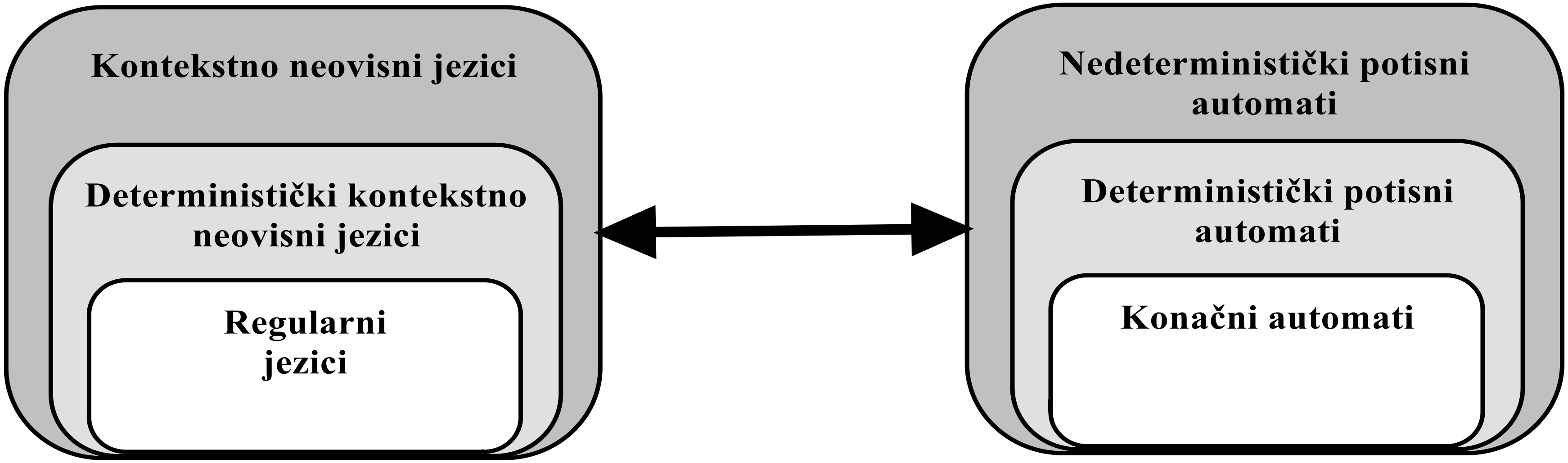


Primjer I

- Je li $L=\{a^i b^i c^i \mid i \geq 1\}$ kontekstno neovisan jezik?
- **NIJE – nema PA**

  - na stog stavimo znakove a
  - skidamo a dok na ulazu čitamo b (usporedimo broj)
  - možemo provjerili je li broj a-jeva jednak broju b-jeva
  - ali ne možemo provjeriti za broj c-jeva

Hijerarhija automata i gramatika



- DKA je posebni slučaj determinističkog PA koji ne koristi stog

Primjer II

- jezik $L=\{ww^R | \text{ w je niz } (0+1)^* \text{ a } w^R \text{ niz napisan u obrnutom redoslijedu}\}$

- je kontekstno neovisan jezik ali nije deterministički
- možemo napraviti PA koji na svakom koraku nedeterministički provjerava je li niz zapisan u obrnutom redosljedu

Svojstva zatvorenosti jezika

- *Def*: Klasa jezika je zatvorena s obzirom na neku operaciju ako primjenom te operacije na bilo koji jezik iz te klase dobijemo jezik koji je u istoj klasi
  - unija
  - nadovezivanje (konkatenacija)
  - Kleenov operator
  - supstitucija
  - presjek i
  - komplement

Svojstva zatvorenosti kontekstno neovisnih jezika KNJ

- KNJ **su zatvoreni** s obzirom na operacije:
  - unije
  - nadovezivanja (konkatenacije)
  - Kleenovog operatora
  - supstitucije
- KNJ **nije zatvoren na** presjek i komplement
  - presjek i komplement KNJ **nisu** nužno KNJ
  - presjek KNJ i regularnog jezika je KNJ (pomoću DKA i PA)

Zatvorenost s obzirom na uniju

- unija KNJ je KNJ
- gramatika $G1=(V1, T1, P1, S1)$ generira jezik $L(G1)$ i $G2=(V2, T2, P2, S2)$ generira jezik $L(G2)$ gdje je $V1 \cap V2 = \emptyset$ (gramatike nemaju istih nezavršnih znakova)
- gramatika $G3=(V3, T3, P3, S3)$ koja generira jezik $L(G3)=L(G1) \cup L(G2)$ se konstruira:
  - $V3=V1 \cup V2 \cup \{S3\}$, $V1 \cap V2 = \emptyset$ i $S3 \notin V1$ i $S3 \notin V2$
  - $T3= T1 \cup T2$
  - u skup produkcija $P3=P1 \cup P2$ se dodaje **$S3 \rightarrow S1 | S2$**

Zatvorenost s obzirom na nadovezivanje

- nadovezivanje KNJ-a je KNJ
- gramatika G1=(V1, T1, P1, S1) generira jezik L(G1) i G2=(V2, T2, P2, S2) generira jezik L(G2)
- gramatika G4=(V4, T4, P4, S4) koja generira jezik L(G4)=L(G1)L(G2) se konstruira:
  - $V4=V1\cup V2\cup\{S4\}$, $V1\cap V2=\emptyset$ i $S4\notin V1$ i $S4\notin V2$
  - $T4= T1\cup T2$
  - u skup produkcija $P4=P1\cup P2$ se dodaje **$S4\rightarrow S1S2$**

Zatvorenost s obzirom na Kleenov operator *

- KNJ su zatvoreni s obzirom na Kleenov operator
- gramatika G1=(V1, T1, P1, S1) generira jezik L(G1)
- gramatika G5=(V5, T5, P5, S5) koja generira jezik L(G5)=L(G1)* se konstruira:
  - V5=V1$\cup${S5}, S5$\notin$V1
  - T5= T1
  - u skup produkcija P5=P1 se dodaje **S5$\rightarrow$S1S5|ε**

Zatvorenost s obzirom na supstituciju I

- KNJ su zatvoreni s obzirom na substituciju
- gramatika G=(V, T, P, S) generira KNJ L(G)
- **supstitucija:**
  - neka se svi završni znakovi $\mathbf{a_i}$ gramatike G zamijene nizovima KNJ L(**Gi**) gdje je 1≤i≤k, gdje je k kardinalni broj skupa završnih znakova T
  - gramatika Gi=(Vi, Ti, Pi, Si) generira jezik L(Gi)
- supstitucijaom dobivamo L' koji je KNJ i za njega je moguće konstruirati gramatiku G'=(V', T', P', S')

Zatvorenost s obzirom na supstituciju II

- konstrukcija gramatike G'=(V', T', P', S'):
  - u skup završnih znakova V' se stave svi nezavršni znakovi gramatike G i svi nezavršni znakovi gramatike Gi
    - $V' = V \cup V1 \cup V2 \cup .. \cup Vk$, $V \cap Vi = \emptyset$, $Vi \cap Vj = \emptyset$ $1 \leq i \leq k$, $i \neq j$
  - u skupu završnih znakova su isključivo znakovi gramatika Gi
    - $T' = T1 \cup T2 \cup .. \cup Tk$
  - početni nezavršni znak S'=S početni nazavršni znak G
  - u skup produkcija $P' = P1 \cup P2 \cup .. \cup Pk$ dodaju se produkcije iz G u obliku:
    - u gramatici G bilo koji završni ai zamijeni se početnim Si gramatike Gi

Primjer: zatvorenost KNJ na supstituciju I

- zadan je L: nizovi s jednakim brojem znakova a i b
  – jezik L generira gramatika G=({S}, {a,b}, P, S), $S \to aSbS|bSaS|\varepsilon$
- znak a zamijeni se nizovima jezika $L1=\{0^n1^n|n\geq1\}$
  – jezik L1 generira gramatika $G1=(\{S_1\}, \{0,1\}, P, S_1)$
  $S_1 \to 0S_11|01$
- znak b zamijeni se nizovima jezika $L2=\{ww^R|w$ niz jezika $(0+2)^*\}$
  – jezik L2 generira gramatika $G2=(\{S_2\}, \{0,2\}, P, S_2)$
  $S_2 \to 0S_20|2S_22|\varepsilon$

Primjer: zatvorenost KNJ na supstituciju II

- supstitucijom znakova a i b nizovima jezika L1 i L2 nastaje L' kojeg generira gramatika G':

1. V'={S}$\cup${S1}$\cup${S2} ={S, S1, S2}
2. T'= {0,1}$\cup${0,2}={0, 1, 2}
3. S'=S
4. u skup produkcija P'=$\{S_1 \rightarrow 0S_1 1|01\} \cup \{S_2 \rightarrow 0S_2 0|2S_2 2|\varepsilon\}$ dodaju se produkcije gramatike u kojima se **a** zamijeni nezavršnim S1 a **b** nezavršnim S2: $S \rightarrow S_1 S S_2 S | S_2 S S_1 S | \varepsilon$

- gramatika G'($\{S, S_1, S_2\}$, {0, 1, 2},P, S) ima produkcije

$S \rightarrow S_1 S S_2 S | S_2 S S_1 S | \varepsilon$

$S_1 \rightarrow 0S_1 1|01$

$S_2 \rightarrow 0S_2 0|2S_2 2|\varepsilon$

Primjer presjek

- **jezik $L=\{ a^i b^i c^i \mid i \geq 1 \}$ nije KN** (dokaz svojstvom napuhavanja)
- L je presjek jezika L1 i L2 $L= L1 \cap L2$
- L1 i L2 su KNJ gdje je $L1=\{ a^i b^i c^j \mid i \geq 1, j \geq 1 \}$ i

  $L2=\{ a^j b^i c^i \mid i \geq 1, j \geq 1 \}$
- Gramatika G1=({S,A,B}, {a,b,c}, P, S), generira L1 i ima

  $P: S \to AB \quad A \to aAb|ab \quad B \to cB|c$
- Gramatika G2=({S,C,D}, {a,b,c}, P, S), generira L2 i ima

  $P: S \to CD \quad C \to aC|a \quad D \to bDc|bc$
- dokazano s **kontradikcijom**: L1 i L2 su KNJ a

  $L= L1 \cap L2$ nije KNJ

Komplement

- ako je iz prošlog primjera $L = L1 \cap L2$ onda po DeMorganovom pravilu

$$L^c = (L1 \cap L2)^c = (L1^c \cup L2^c)^c$$

- a upravo smo pokazali da

$$L1 \cap L2 \text{ nije KNJ} \Rightarrow (L1 \cap L2)^c \text{ nije KNJ}$$

dokaz kontradikcijom

- **KNJ nisu zatvoreni na presjek i komplement**

Presjek kontekstno neovisnog i regularnog jezika je KNJ I

- kontekstno neovisni jezik (KNJ) L1 prihvaća PA

$$M1= (Q1, \Sigma, \Gamma, \delta 1, q0, Z1,F1)$$

- regularni jezik (RJ) L2 prihvaća DKA

$$M2=(Q2, \Sigma, \delta 2, p0, F2)$$

- moguće je izgraditi PA M'= (Q', Σ, Γ,δ',q'0, Z',F')

koji prihvaća prihvatljivim stanjem jezik $L' = L1 \cap L2$

Presjek kontekstno neovisnog i regularnog jezika je KNJ II

- Q’=Q2xQ1
- q’0=[p0,q0]
- F’=F2xF1
- skup δ’([p,q],a,X) sadrži ([p’,q’],γ) ako i samo ako
  - za funkciju prijelaza DKA M2 δ2(p, a)=p’ i
  - za funkciju prijelaza PA M1 (q’,γ) ∈ δ1(q, a, X)
  - ako je a = ε onda je p’=p

uvažava se obje funkcija prijelaza a na stog ionako piše samo PA !

Primjer: presjek KNJ i RJ I

- PA M1= ({q1,q2},{0,1},{N,K}, δ1, q1, K,{q2})

1) δ1(q1,0,K)={(q1,NK)} 2) δ1(q1,0,N)={(q1,NN)}

3) δ1(q1,1,N)={(q2,ε)} 4) δ1(q2,1,N)={(q2,ε)}

  – prihvaća jezik L(M1)={0^n1^m |n≥1, m≥1, n>m } prihvatljivim stanjem q2

- DKA M2=({p1,p2,p3},{0,1}, δ2, p1, {q3})

1) δ2(p1,0)=p1 2) δ2(p2,0)=p2 3) δ2(p3,0)=p3

4) δ2(p1,1)=p2 5) δ2(p2,1)=p3 6) δ2(p3,1)=p3

  – u jeziku L(M2) su barem dve 1

Primjer: presjek KNJ i RJ II

- **Presjek**: $L3=L(M1)\cap L(M2) =\{0^n1^m \mid n\geq 2, m\geq 2, n>m \}$ i prihvaća ga PA $M' = (Q', \Sigma, \Gamma, \delta', q'0, Z', F')$
- $Q'=\{[p1,q1], [p1,q2], [p2,q1], [p2,q2], [p3,q1], [p3,q2]\}$
- $q'0=[p1,q1]$
- $F'=\{[p3,q2]\}$ i
- δ'

1) $\delta'([p1,q1],0,K)=\{([p1,q1],NK)\}$
2) $\delta'([p1,q1],0,N)=\{([p1,q1],NN)\}]$
3) $\delta'([p1,q1],1,N)=\{([p2,q2],\varepsilon)\}$
4) $\delta'([p2,q2],1,N)=\{([p3,q2],\varepsilon)\}$
5) $\delta'([p3,q2],1,N)=\{([p3,q2], \varepsilon)\}$

Primjer: presjek KNJ i RJ III

- prihvaćanje niza 00011 iz L3
  - PA M1: $(q1,00011,K) \succ (q1,0011,NK) \succ (q1,011,NNK) \succ (q1,11,NNNK) \succ (q2,1,NNK) \succ (q2,\varepsilon,NK)$ i $q2 \in F1$
  - DKA M2: $\delta(p1,00011) = \delta(p1,0011) = \delta(p1,011) = \delta(p1,11) = \delta(p2,1) = p3$ i $p3 \in F2$
  - PA M': $([p1,q1],00011,K) \succ ([p1,q1],0011,NK) \succ ([p1,q1],011,NNK) \succ ([p1,q1],11,NNNK) \succ ([p2,q2],1,NNK) \succ ([p3,q2],\varepsilon,NK)$ i $[p3,q2] \in F'$

Presjek KNJ i RJ je KNJ: $L3=L(M1) \cap L(M2)$ je KNJ

Svojstvo napuhavanja KNJ I

- slično svojstvu napuhavanja RJ
- dokazuje se kontekstna neovisnost jezika
- zasniva se na broju čvorova generativnog stabla i broju nezavršnih znakova gramatike
- *Def.:* za dovoljno dugački niz znakova broj unutrašnjih čvorova generativnog stabla veći je od kardinalnog broja skupa nezavršnih znakova gramatike
  - značí da je više čvorova označeno istim nezavršnim znakom gramatike

Svojstvo napuhavanja KNJ II

- kaže da uvijek postoje dva kratka podniza koja je oba moguće ponavljati neograničen ali jednak broj puta

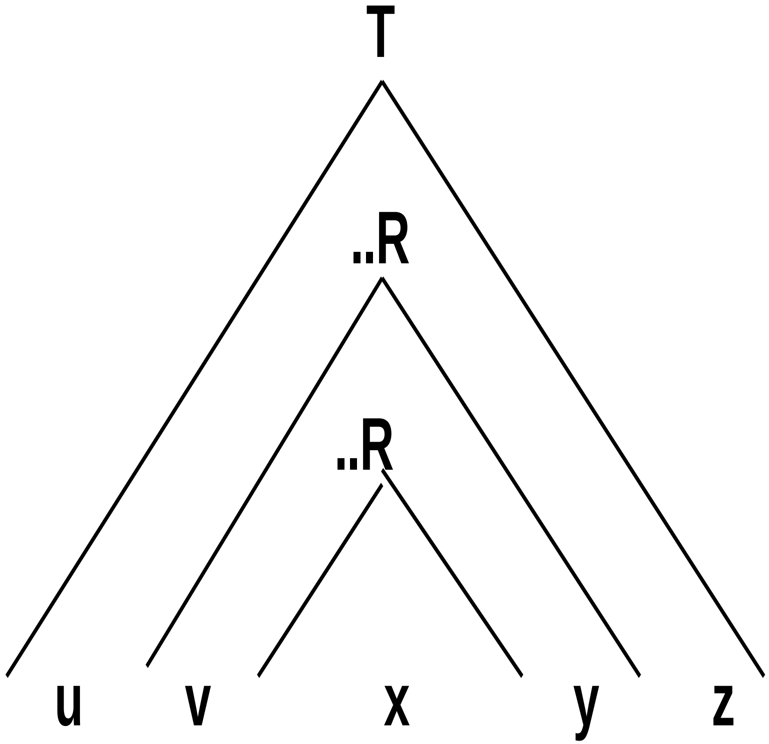
Svojstvo napuhavanja KNJ III

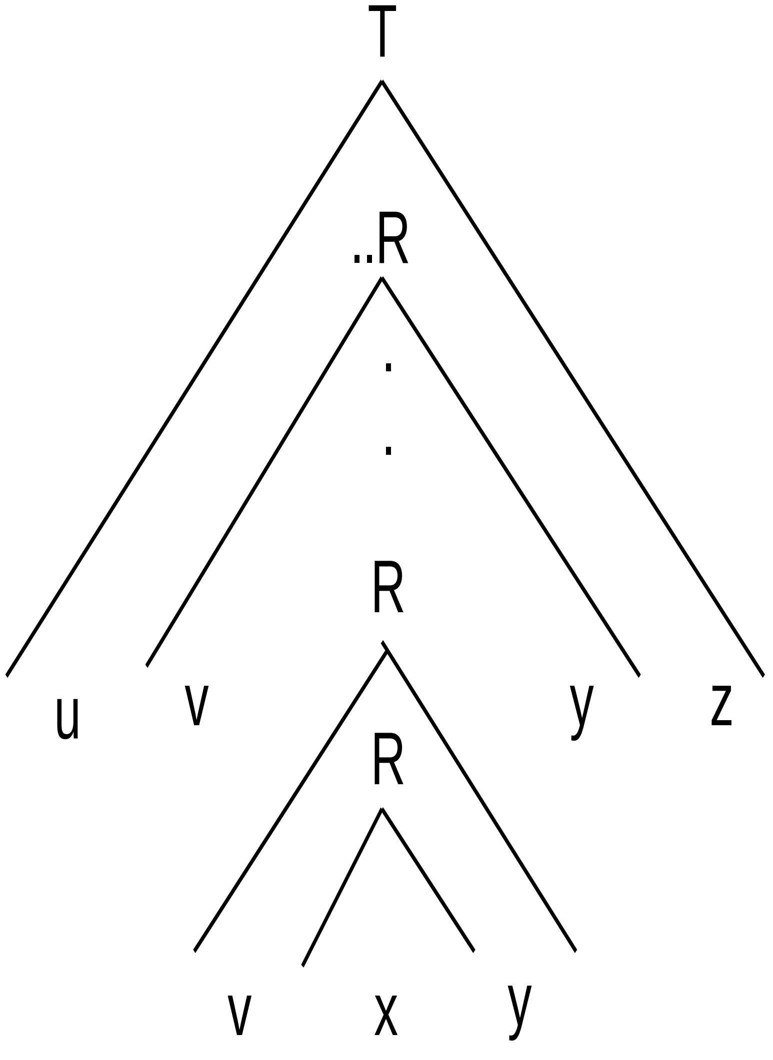
- neka je L KNJ, postoji konstanta n koja ovisi isključivo o jeziku L takva da ako je niz z iz jezika L i $|z| \geq n$, onda je niz z moguće napisati kao niz uvwxy za koji vrijedi
  - a) $|vx| \geq 1$
  - b) $|vwx| \leq n$
  - c) za bilo koji $i \geq 1$vrijedi da je niz uv^iwx^iy iz jezika L

Dokaz napuhavanja za KNJ I

- gramatika G generira KNJ A
- moramo pokazati da svaki niz s dovoljne duljine možemo napuhati i da je još uvijek iz jezika A
- s je jako dugi niz iz A i za s je moguće izgraditi generativno drvo
  - budući da je s jako dug => drvo je jako visoko => postoji dug put od korijena do listova
  - na tom putu postoji simbol R
  - R se ponavlja zbog pigeonhole principa
  - niz s možemo podijeliti na uvxyz i stablo parsiranja je onda:

Dokaz napuhavanja za KNJ II





Dokaz da L nije KNJ I

- koristi se svojstvo napuhavanja
- $L=\{ a^i b^i c^i \mid i \geq 1 \}$pretpostavimo da je KN i izaberemo konstantu n: $z=a^n b^n c^n$ i $|z| \geq n$
- ako z zapišemo z=uvwxy koji zadovoljava svojstvo napuhavanja
- moramo odrediti neprazne znakove v i x u nizu $a^n b^n c^n$ tako da ih je moguće ponoviti i da je zadovoljeno $|vwx| \leq n$ (b)

Dokaz da L nije KNJ II

- 1. slučaj: x i v sadrže samo iste simbole znači x sadrži samo a ili samo b ili samo c isto vrijedi i za v
  - onda uv^2wx^2y ne može sadržavati jednak broj a-jeva, b-jeva i c-jeva=> znači nije niz iz jezika iz L
- 2. slučaj: x i v sadrže različite simbole (a b i c)
  - onda broj a-jeva, b-jeva i c-jeva može bit jednak ali nisu poredani po redu

- kontradikcijom dokazano da L nije KNJ

Dokaz da D nije KNJ I

- D={ ww | w $\in$ {0,1}* }
- pretpostavimo da je KN i kontradikcijom sa svojstvom napuhavanja pokažemo da nije KNJ
- uzmemo niz 0^p10^p1 iz D s duljinom većom od p
  - niz možemo napumpati 00000 0 1 0 000001

    u v x y z
  - niz se može napumpati pa nije dobar kandidat
- uzmemo niz $s=0^p1^p0^p1^p$ iz D s duljinom većom od p
  - svojstvo napuhavanja kaže: da niz možemo napumpati s $|vxy| \le p$
  - napuhani dio mora biti točno u sredini

Dokaz da D nije KNJ II

- uzmemo niz $s=0^p1^p0^p1^p$ iz D s duljina većom od p
  - svojstvo napuhavanja kaže: da niz možemo napumpati s $|vxy| \leq p$
  - napuhani dio mora biti točno u sredini
    - ako je u 1. dijelu uv^2xy^2z prebaci 1 u prvu polovicu 2. dijela niza => nije oblika ww => nije u D
    - ako je u 2. dijelu uv^2xy^2z prebaci 0 na kraj 1. dijela => nije oblika ww => nije u D
    - ako je točno u sredini $\mathbf{0^p1^i0^j1^p}$ barem jedan od **i** i **j** je različit od p => nije oblika ww => nije u D
  - D nije KNJ

Primjer

- zadan je jezik L u kojem su nizovi koji imaju jednak broj a i b znakova
  - gramatika G generira jezik L: $S \to aSbS \mid bSaS \mid \varepsilon$
- supstitucija
  - znak **a** zamijeni se nizovima jezika $L1=\{0^n 1^n \mid n \geq 1\}$
    - jezik L1 generira gramatika G1: $S_1 \to 0S_11 \mid 01$
  - znak b zamijeni se nizovima jezika $L2=\{ww^R \mid$ w je niz $(0+2)^*$ a w^R niz napisan u obrnutom redoslijedu$\}$
    - jezik L2 generira gramatika G2: $S_2 \to 0S_20 \mid 2S_22 \mid \varepsilon$

Primjer II

- zamjenom a i b nizovima jezika L1 i L2 dobivamo jezik L' kojeg generira gramatika G'

1. $V' = \{S\} \cup \{S_1\} \cup \{S_2\} = \{S, S_1, S_2\}$
2. $T' = T1 \cup T2 \cup .. \cup Tk = \{0,1\} \cup \{0,2\} = \{0,1,2\}$
3. $S' = S$
4. skupu produkcija $P' = P1 \cup P2 \cup .. \cup Pk$

Formalni jezici i jezični procesori I

KONTEKSTNO NEOVISNI JEZICI

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