MODELI RAČUNARSTVA - JEZIČNI PROCESORI 1

Siniša Srbljić, Sveučilište u Zagrebu

2. REGULARNI JEZICI

- 2.1. KONAČNI AUTOMATI
- 2.2. REGULARNI IZRAZI
- 2.3. SVOJSTVA REGULARNIH IZRAZA
- 2.4. GRAMATIKA

2. REGULARNI JEZICI

- REGULARNI JEZIK I KONAČNI AUTOMAT
  - jezik je regularan ako postoji konačni automat koji ga prihvaća
  - time je definirana istovjetnost konačnih automata i regularnih jezika
- REGULARNI JEZICI
  - jednostavni, prihvaćaju ih konačni automati

2.2. Regularni izrazi

- REGULARNI IZRAZI
  - koriste se za opis regularnih jezika
  - na osnovu definicije regularnih izraza
  - i na osnovu algoritma konstrukcije konačnog automata za jezik zadan regularnim izrazom
  - konstruiramo DKA generatorom automata

2.2.1. Definicija regularnih izraza

- REGULARNI JEZIK I REGULARNI IZRAZ
  - ako je jezik moguće opisati regularnim izrazima onda je regularan!
  - označava se L(r)
  - moguće je izgraditi DKA M tako da je L(M)=L(r)
  - npr. wcw^R, $w \in \{aa,ab,ba,bb\}$ **je** regularan
  - npr. wcw^R, w proizvoljan **nije** regularan
  - skup regularnih jezika je pravi podskup svih jezika
  - za neregularne jezike nije moguće izgraditi DKA

Definicija regularnih izraza

- ALGORITAM SINTEZE DKA
  - Regularni jezik K
  - Opis jezika K regularnim izrazima r: L(r)=K
  - Izgraditi ε-NKA M za koji vrijedi: L(M)=L(r)
  - Izgraditi NKA M’ za koji vrijedi: L(M’)=L(M)
  - Izgraditi DKA M’’ za koji vrijedi: L(M’’)=L(M’)
  - Izgraditi minimalni DKA M’’’ za koji vrijedi: L(M’’’)=L(M’’) =L(M’)=L(M)=L(r)=K

Definicija regularnih izraza

- REKURZIVNA PRAVILA ZA RI
  - $\emptyset$ jest RI i označava jezik $L(\emptyset)=\{\}$
  - ε jest RI i označava jezik $L(\varepsilon)= \{\varepsilon\}$
  - $\forall a \in \Sigma$, a jest RI i označava jezik $L(a)=\{a\}$
    (ista iznaka “a” znači znak, niz i RI)
  - ako su r i s RI koji označavaju L(r) i L(s) vrijedi:
    - $(r)\vee(s)$ jest RI i označava jezik $L((r)\vee(s))= L(r)\cup L(s)$
      (operator $\vee$ nekad se označava sa + ili |)
    - $(r)(s)$ jest RI i označava jezik $L((r)\,(s))= L(r)L(s)$
    - $(r)^*$ jest RI i označava jezik $L((r)^*)= (L(r))^*$

Definicija regularnih izraza

- Pravila asocijativnosti i prednosti
  - unarni operator * je lijevo asocijativan i najveće prednosti
  - binarni operator nadovezivanja je lijevo asocijativan i veće je prednosti od operatora $\vee$ (ili + ili |)
  - binarni operator $\vee$ je lijevo asocijativan i najniže je prednosti

Definicija regularnih izraza

- PRIMJER RI
  - RI 01 definira L(01) = {01}
  - RI $0\vee1$ definira $L(0\vee1)$ = {0,1}
  - RI $(0\vee1)(0\vee1)$ definira $L(0\vee1)(0\vee1)$ = {00,01,10,11}
  - RI 1* definira L(1*) = {ε,1,11,111,...,11111111,....}
  - RI $(0\vee1)^*$ definira $L((0\vee1)^*) = L(0\vee1)^*$ =
    = {ε,0,1,00,01,10,11,000,001,...,111,...,1110100111,....}
  - RI $(0\vee1)^*00(0\vee1)^*$ definira $L(0\vee1)^*\ 00(0\vee1)^*$ =
    = {00,000,001,100,0000,0100,1000,1100,0001,0010,...}
  - RI 0*1* definira L(0*1*) =
    = {ε,0,1,00,01,000,001,011,111,...,000000111,....}

Definicija regularnih izraza

- ISTOVJETNOST RI
  - RI r i s su istovjetni ako definiraju iste jezike:
    definira $L(r) = L(s)$
  - piše se $r = s$
  - npr. $(a)\vee((b)*(c) = a\vee b*c$;
  - $L(a\vee b*c) = \{a,c,bc,bbc,bbbc,...\}$

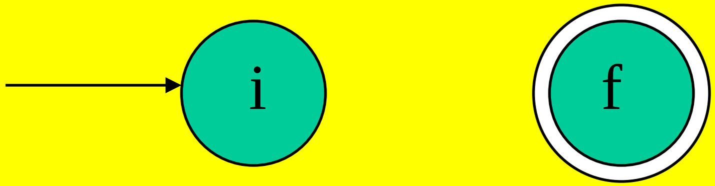
Definicija regularnih izraza

- SVOJSTVA RI
  - $r \vee s = s \vee r$ — $\vee$ jest komutativno
  - $r \vee (s \vee t) = (r \vee s) \vee t$ — $\vee$ jest asocijativno
  - $r(st) = (rs)t$ — nadovezivanje jest asocijativno
  - $r(s \vee t) = rs \vee rt$ — distributivnost nadovezivanja nad $\vee$
  - $\varepsilon r = r \varepsilon$ — ε je neutralni element nadovezivanja
  - $r^* = (r \vee \varepsilon)^*$ — relacija između * i $\vee$
  - $r^{**} = r^*$ — idempotentnost

2.2.2. Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI za “∅”
  - za bilo koji regularni izraz R moguće je izgraditi ε-NKA M tako da vrijedi L(M)=L(r)

  p1. za RI ∅ koji označava jezik L(∅)={} izgradimo:
  ε-NKA M=({i,f}, Σ, {}, i, {f})
  - ne prihvaća ni jedan niz

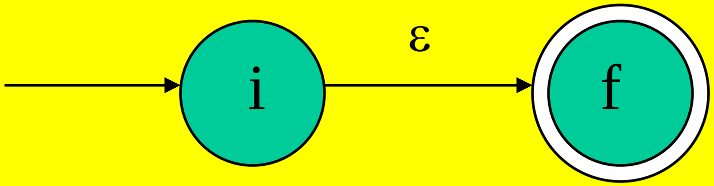


Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI za “ε”

  p2. za RI ε koji označava jezik $L(\varepsilon)=\{\varepsilon\}$ izgradimo:
  ε-NKA $M=(\{i,f\}, \Sigma, \{\delta(i,\varepsilon)=f\}, i, \{f\})$

  – prihvaća prazni niz ε

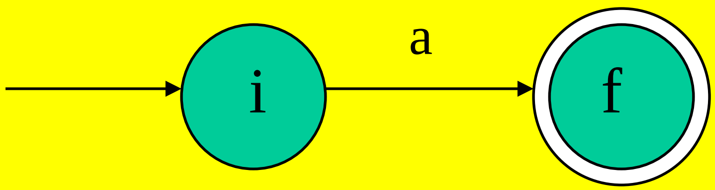


Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI za “a”

  p3. za RI a koji označava jezik $L(a)=\{a\}$ izgradimo: ε-NKA $M=(\{i,f\}, \Sigma, \{\delta(i,a)=f\}, i, \{f\})$

  – prihvaća niz a, ne prihvaća ε niti b od $b \in \Sigma$



Konstrukcija ε-NKA iz RI

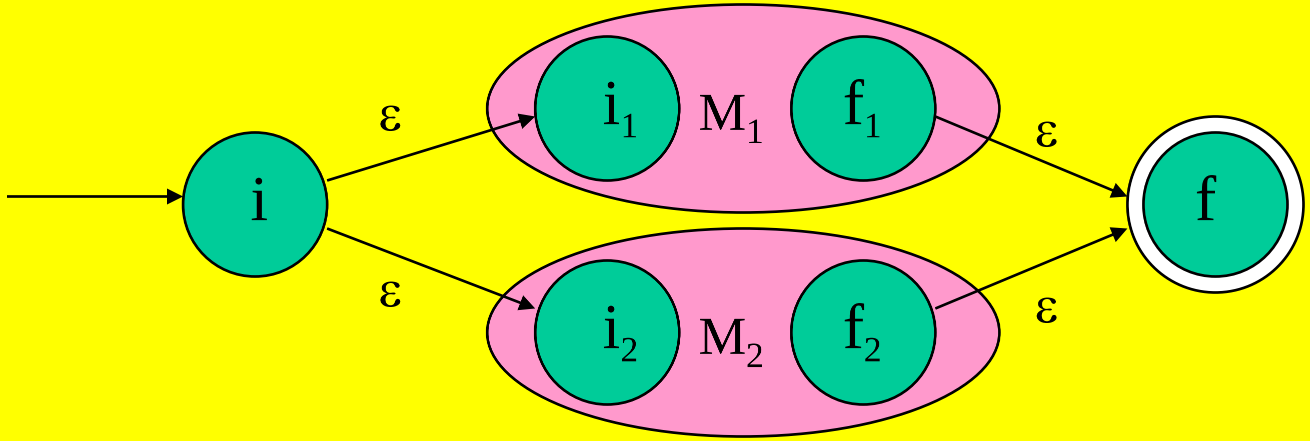
- KONSTRUKCIJA ε-NKA iz RI za $r_1 \vee r_2$

  p4. za RI $r_1 \vee r_2$ koji označava jezik $L(r_1 \vee r_2) = L(r_1) \cup L(r_2)$ izgradimo ε-NKA M postupkom:

  - izgradimo ε-NKA $M_1 = (Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ da je $L(M_1) = L(r_1)$
  - izgradimo ε-NKA $M_2 = (Q_2, \Sigma_2, \delta_2, i_2, \{f_2\})$ da je $L(M_2) = L(r_2)$
  - f_1 i f_2 nemaju prijelaza,
    tj. $\forall a \in (\Sigma_1 \cup \{\varepsilon\}) : \delta_1(f_1,a) = \emptyset$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\}) : \delta_2(f_2,b) = \emptyset$
  - stanja iz Q_1 i Q_2 imenujemo tako da je $Q_1 \cap Q_2 = \emptyset$
  - izgradimo ε-NKA $M = (Q_1 \cup Q_2 \cup \{i,f\}, \Sigma_1 \cup \Sigma_2, \delta, i, \{f\})$
  - f je novo prihvatljivo stanje, a f_1 i f_2 više nisu prihvatljiva

Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI za $r_1 \vee r_2$
  - konstruiramo δ:
    - $\delta(i, \varepsilon) = \{i_1, i_2\}$
    - $\delta(q, a) = \delta_1(q, a) \quad \forall q \in (Q_1 \setminus \{f_1\}) \quad i \quad \forall a \in (\Sigma_1 \cup \{\varepsilon\})$
    - $\delta(q, b) = \delta_2(q, b) \quad \forall q \in (Q_2 \setminus \{f_2\}) \quad i \quad \forall b \in (\Sigma_2 \cup \{\varepsilon\})$
    - $\delta(f_1, \varepsilon) = \delta(f_2, \varepsilon) = \{f\}$



Konstrukcija ε-NKA iz RI

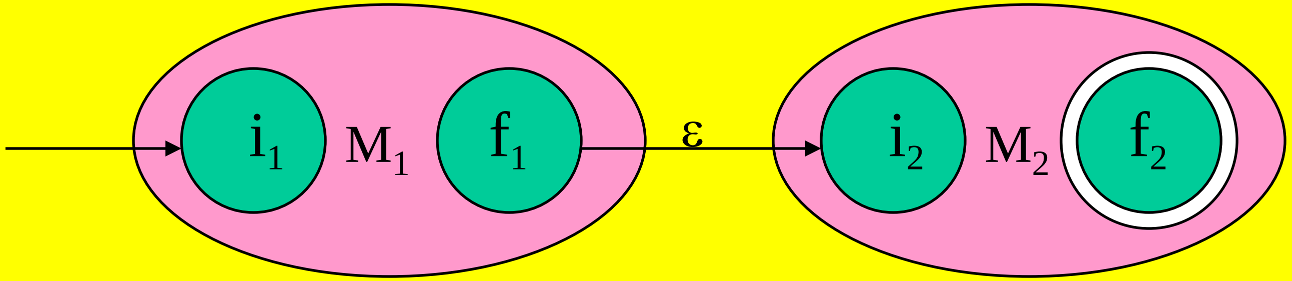
- KONSTRUKCIJA ε-NKA iz RI za r_1r_2

  p5. za RI r_1r_2 koji označava jezik $L(r_1r_2) = L(r_1)L(r_2)$ izgradimo ε-NKA M postupkom:

  - izgradimo ε-NKA $M_1 = (Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ da je $L(M_1) = L(r_1)$
  - izgradimo ε-NKA $M_2 = (Q_2, \Sigma_2, \delta_2, i_2, \{f_2\})$ da je $L(M_2) = L(r_2)$
  - f_1 i f_2 nemaju prijelaza,
    tj. $\forall a \in (\Sigma_1 \cup \{\varepsilon\}) : \delta_1(f_1,a) = \emptyset$ i $\forall b \in (\Sigma_2 \cup \{\varepsilon\}) : \delta_2(f_2,b) = \emptyset$
  - stanja iz Q_1 i Q_2 imenujemo tako da je $Q_1 \cap Q_2 = \emptyset$
  - izgradimo ε-NKA $M = (Q_1 \cup Q_2 \cup \{i,f\}, \Sigma_1 \cup \Sigma_2, \delta, i_1, \{f_2\})$
  - i_1 je novo početno stanje, a f_1 više nije prihvatljivo
  - i_2 više nije početno stanje, a f_2 je novo prihvatljivo stanje

Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI za r_1r_2
  - konstruiramo δ:
    - $\delta(q, a) = \delta_1(q, a) \quad \forall q \in (Q_1 \setminus \{f_1\}) \;$ i $\; \forall a \in (\Sigma_1 \cup \{\varepsilon\})$
    - $\delta(q, b) = \delta_2(q, b) \quad \forall q \in (Q_2 \setminus \{f_2\}) \;$ i $\; \forall b \in (\Sigma_2 \cup \{\varepsilon\})$
    - $\delta(f_1, \varepsilon) = \{i_2\}$



Konstrukcija ε-NKA iz RI

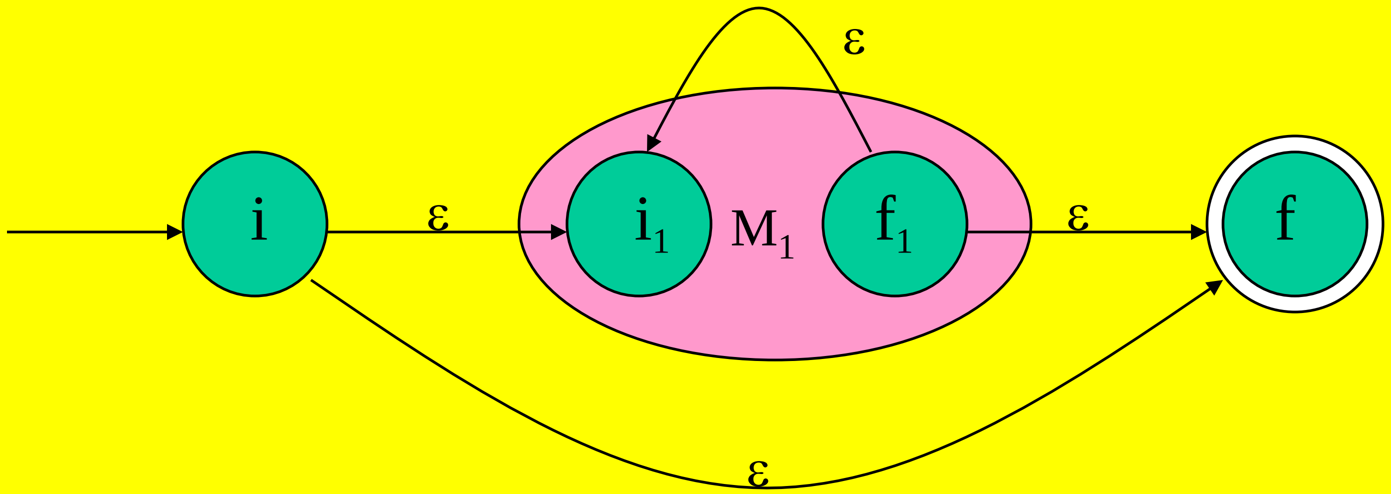
- KONSTRUKCIJA ε-NKA iz RI za r_1*

  p6. za RI r_1* koji označava jezik $L(r_1^*) = L(r_1)^*$
  izgradimo ε-NKA M postupkom:

  - izgradimo ε-NKA $M_1 = (Q_1, \Sigma_1, \delta_1, i_1, \{f_1\})$ da je $L(M_1) = L(r_1)$
  - f_1 nema prijelaza,
    tj. $\forall a \in \Sigma_1 : \delta_1(f_1, a) = \emptyset$
  - izgradimo ε-NKA $M = (Q_1 \cup \{i, f\}, \Sigma_1, \delta, i, \{f\})$
  - i_1 više nije početno stanje, a f_1 više nije prihvatljivo

Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI za r_1*
  - konstruiramo δ:
    - $\delta(i, \varepsilon) = \delta(f_1, \varepsilon) = \{i_1, f\}$
    - $\delta(q, a) = \delta_1(q, a) \quad \forall q \in (Q_1 \backslash \{f_1\}) \quad i \quad \forall a \in (\Sigma_1 \cup \varepsilon)$



Konstrukcija ε-NKA iz RI

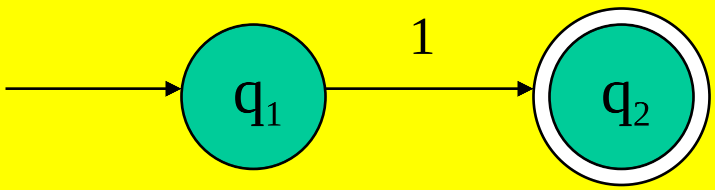
- KONSTRUKCIJA ε-NKA iz RI za r_1*

p7. za RI (r) koji označava jezik L((r)) = L(r)

- izgradimo ε-NKA M za RI (r)
- tako da uzmemo ε-NKA M_1 za RI r
- jer je $L(M_1) = L(M) = L(r) = L((r))$

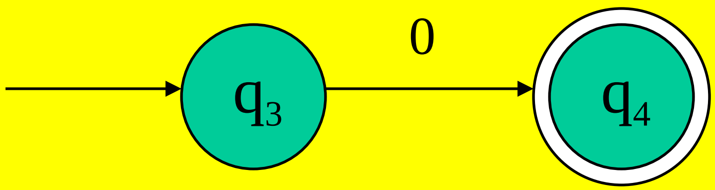
Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - PRIMJER
  - za RI $r = 01^* \vee 1$
    - RI r rastavimo na $r = r_1 \vee r_2$; $r_1 = 01^*$ i $r_2 = 1$
    - za $r_2 = 1$ izgradimo ε-NKA:



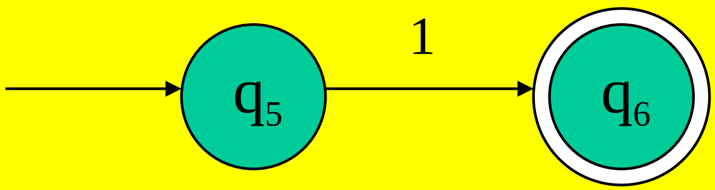
Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - PRIMJER
  - za RI $r = 01^* \vee 1$ nadalje
    - $r_1 = 01^*$ rastavimo na $r_1 = r_3 r_4$; $r_3 = 0$ i $r_4 = 1^*$
    - za $r_3 = 0$ izgradimo ε-NKA:



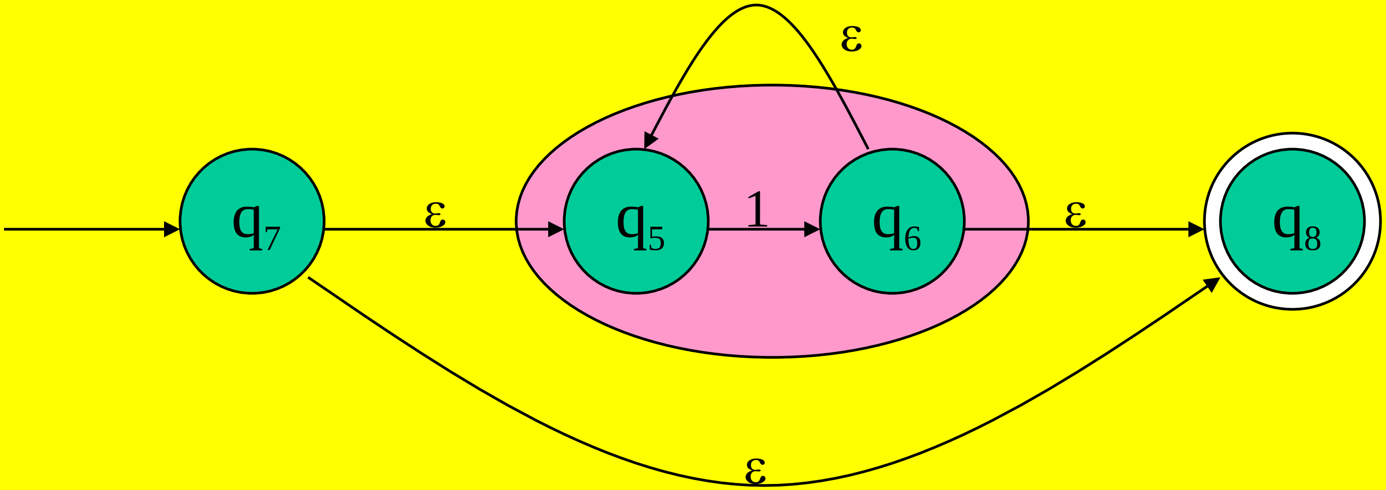
Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - PRIMJER
  - za RI $r = 01^* \vee 1$ nadalje
    - $r_4 = 1^*$ rastavimo na $r_4 = r_5^*$; $r_5 = 1$
    - za $r_5 = 1$ izgradimo ε-NKA:



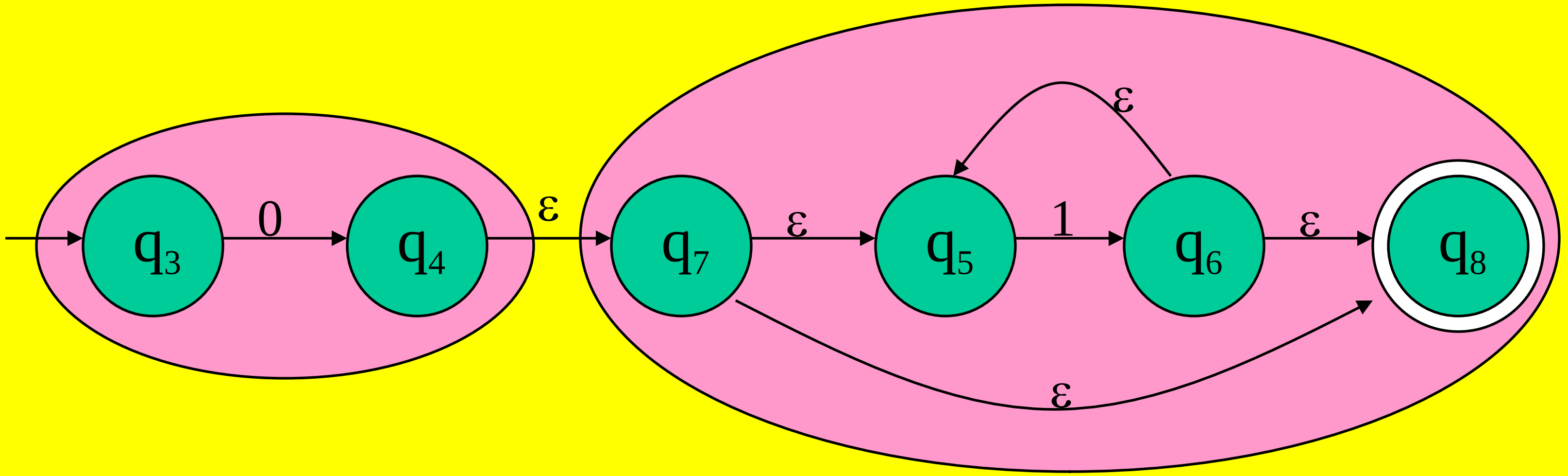
Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - PRIMJER
  - za RI $r = 01^* \vee 1$ nadalje
    - za $r_4 = 1^*$ izgradimo ε-NKA prema pravilu p6:



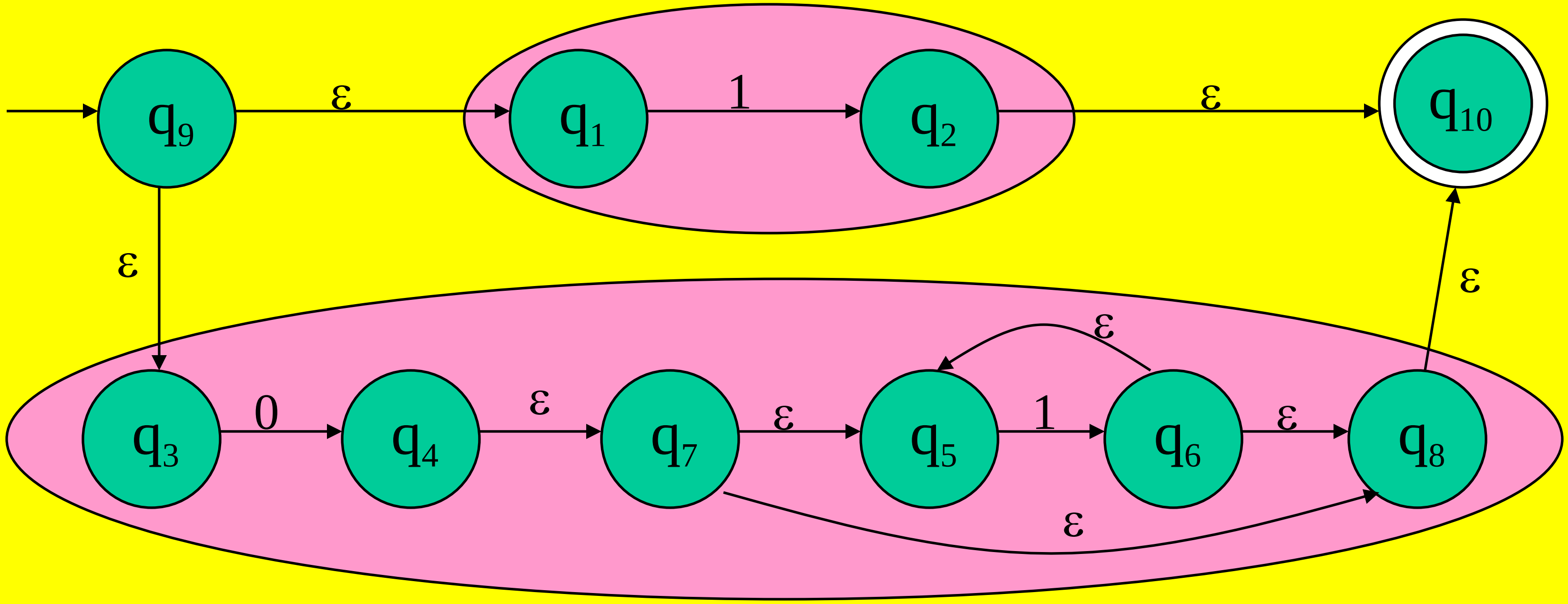
Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - PRIMJER
  - za RI $r = 01^* \vee 1$ nadalje
    - za $r_1 = r_3 r_4$ izgradimo ε-NKA prema pravilu p5:



Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - PRIMJER
  - za RI $r = 01^* \vee 1$ nadalje
    - za $r = r_1 \vee r_2$ izgradimo ε-NKA prema pravilu p4:



Konstrukcija ε-NKA iz RI

- KONSTRUKCIJA ε-NKA iz RI - SVOJSTVA
  - ε-NKA konstruiran iz RI ima svojstva
    - broj stanja ε-NKA nikad nije veći od $2|r|$ jer se u pojedinim koracima konstrukcije nikad ne stvara više od 2 nova stanja
    - ε-NKA $M = (Q, \Sigma, \delta, i, \{f\})$ ima samo jedno završno stanje f za koje vrijedi $\delta(f,a)=\emptyset\ \forall a \in (\Sigma \cup \{\varepsilon\})$,
odnosno nema prijelaza iz f
    - skup $\delta(q,a)$ sadrži najviše jedno stanje za ulazni znak a
skup $\delta(q, \varepsilon)$ sadrži najviše dva stanja za ulazni znak ε;
ima li čvor dvije grane obje su označene sa ε

2.2.3. Generator konačnog automata

- GENRATOR KA
  - KA gradi se za jezik zadan RI
  - generator ostvaruje dio ili cjelokupnu pretvorbu RI u DKA

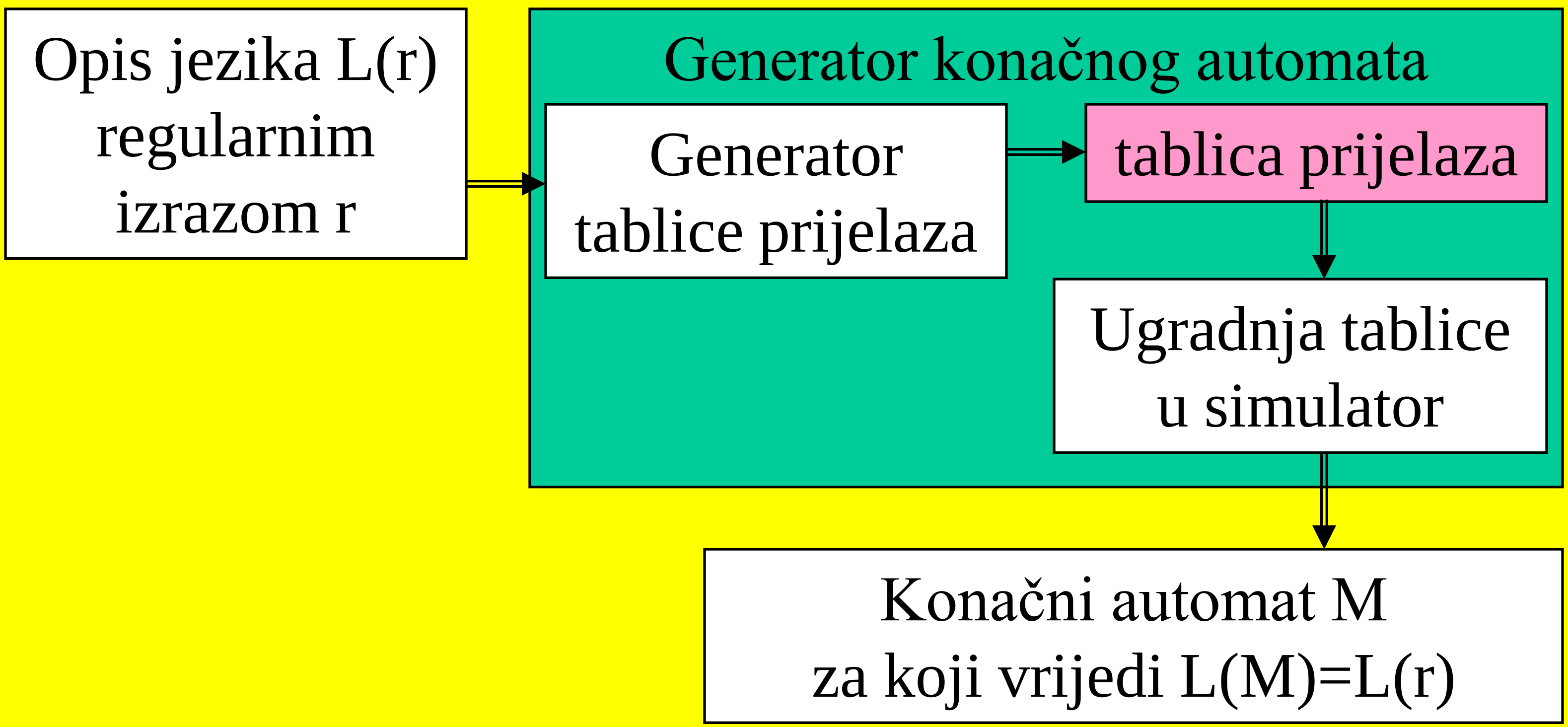
Opis jezika L(r) regularnim izrazom r	⇒	Generator konačnog automata	⇒	Konačni automat M za koji vrijedi L(M)=L(r)

Generator konačnog automata

- SIMULATOR KA
  - ovisno o željenom KA izgradi se program simulator
  - simulator radi na upisanoj tablici prijelaza
    (izravni način zapisa stanja, samo mijenjamo tablice)
  - simulator čita znakove ulaza iz ulaznog spremnika
    i računa prijelaz u stanje na osnovu tablice prijelaza

Generator konačnog automata

- GENRATOR KA
  - gradi tablicu prijelaza na temelju RI
  - tablica prijelaza ugradi se u program simulator



2.3. Svojstva regularnih jezika

- SKUP SVIH JEZIKA
  - jezici wcw^R i 0^{i^2}
    nisu regularni jer nije moguće izgraditi DKA
  - neka je 2^{Σ^*} označava skup svih jezika nad abecedom Σ, svaki jezik $L \subseteq \Sigma^*$ je član tog skupa $L \in 2^{\Sigma^*}$
  - regularni jezici su u skupu $RJ \subset 2^{\Sigma^*}$

Skup svih jezika 2^{Σ^*} nad abecedom Σ

Regularni jezici $RJ \subset 2^{\Sigma^*}$ $\Rightarrow$ DKA $(Q, \Sigma, \delta, q_0, F)$

2.3.1. Svojstva zatvorenosti regularnih jezika

- ZATVORENOST KLASE JEZIKA
  - definira se obzirom na operacije nad jezicima
  - klasa je zatvorena ako primjenom operacije dobijemo jezik u istoj klasi
  - regularni jezici zatvoreni su obzirom na više operacija
  - npr. za $L,N \in RJ$, $L \cup N \in RJ$ jer postoji M: $L(M) = L \cup N$
  - za opis svojstava RJ koristimo istovjetnost RJ, KA i RI

Svojstva zatvorenosti regularnih jezika

- UNIJA, NADOVEZIVANJE, KLEENE
  - RJ su zatvoreni obzirom na
    - **uniju,**
    - **nadovezivanje i**
    - **Kleeneov operator**
  - zatvorenost slijedi iz definicije regularnih izraza

Svojstva zatvorenosti regularnih jezika

- KOMPLEMENT
  - regularni jezici zatvoreni su obzirom na **komplement**
  - neka DKA $M = (Q, \Sigma, \delta, q_0, F)$ prihvaća $L(M) \in RJ$
  - za komplement jezika $L(M)^c$ izgradimo:
    DKA $M' = (Q, \Sigma, \delta, q_0, Q \backslash F)$
  - koji prihvaća jezik:
    $L(M') = \{w \mid \delta(q_0, w) \in Q \backslash F\} = \{w \mid \delta(q_0, w) \notin F\} =$
    $= \Sigma^* \backslash \{w \mid \delta(q_0, w) \in F\} = \Sigma^* \backslash L(M) = L(M)^c$
  - $L\ (M)^c \in RJ$

Svojstva zatvorenosti regularnih jezika

- PRESJEK
  - regularni jezici zatvoreni su obzirom na **presjek**
  - koristimo zatvorenost unije i komplementa, te DeMorganovo pravilo $L \cap N = ((L \cap N)^c)^c = (L^c \cup N^c)^c$
  - neka DKA $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ prihvaća $L(M_1) \in RJ$
  - neka DKA $M_2 = (Q_2, \Sigma_2, \delta_2, p_1, F_2)$ prihvaća $L(M_2) \in RJ$
  - tada DKA $M = (Q, \Sigma, \delta, q_0, F)$ za $L(M) = L(M_1) \cap L(M_2)$ gradimo po pravilima:
    - $Q = Q_1 \times Q_2$; $[q, p] \in Q$; $q \in Q_1$, $p \in Q_2$
    - $q_0 = [q_1, p_1]$
    - $F = F_1 \times F_2$; $[q, p] \in F$; $q \in F_1$, $p \in F_2$
    - $\delta([q, p], a) = [\delta_1(q,a), \delta_2(p,a)]$ $\forall q \in Q_1$, $\forall p \in Q_2$, $\forall a \in \Sigma$

Svojstva zatvorenosti regularnih jezika

- PRESJEK - PRIMJER

	a	b	c	⊥	
q1	q1	q1	q2	1	
q2	q2	q2	q2	0	prihvaća a i b

	a	b	c	⊥	
p1	p1	p2	p1	1	
p2	p2	p2	p2	0	prihvaća a i c

	a	b	c	⊥
[q1, p1]	[q1, p1]	[q1, p2]	[q2, p1]	1
[q1, p2]	[q1, p2]	[q1, p2]	[q2, p2]	0
[q2, p1]	[q2, p1]	[q2, p2]	[q2, p1]	0
[q2, p2]	[q2, p2]	[q2, p2]	[q2, p2]	0

prihvaća nizove a

Svojstva zatvorenosti regularnih jezika

- SUPSTITUCIJA
  - regularni jezici zatvoreni su obzirom na **supstituciju**
  - neka je $R \subseteq \Sigma^*$; $R \in RJ$
  - pridružimo znaku $a \in \Sigma$ RJ $R_a \subseteq \Delta^*$
    tako da niz $a_1 a_2 ... a_n$ zamijenimo nizom $w_1 w_2 ... w_n$
  - dobiveni jezik f(R) je regularan
  - dovoljno je R i R_a opisati regularnim izrazima
  - svojstvo supstitucije (zamjene) omogućava pojednostavljeno zapisivanje **regularnih definicija**

Svojstva zatvorenosti regularnih jezika

- SUPSTITUCIJA - PRIMJER
  - neka je $R \in RJ$ zadan s $r = 0^*(0 \vee 1)1^*$
  - pridružimo znaku 0 jezik $f(0)=a$, a znaku 1 $f(1)=b^*$
  - $f(R)$ je zadan regularnim izrazom:

    $$f(R) = f(0^*(0 \vee 1)1^*) = (f(0))^*((f(0)) \vee (f(1)))(f(1))^*) =$$
    $$= a^*(a \vee b^*)(b^*)^* = a^*(a \vee b^*)b^* = (a^*a \vee a^*b^*)b^* =$$
    $$= a^*ab^* \vee a^*b^*b^* = a^*ab^* \vee a^*b^* = a^+b^* \vee a^*b^* =$$
    $$= (a^+ \vee a^*)b^* = a^*b^*$$
  - $f(R) \in RJ$

2.3.2. Regularne definicije

- REGULARNE DEFINICIJE
  - imenima dodjeljujemo RI, odnosno zamjenjujemo s RI
  - oblik regularnih definicija je:

    $d_1 \to r_1$

    $d_2 \to r_2$

    ....

    $d_n \to r_n$
  - d_i su znakovi imena, a r_i su RI nad $\Sigma \cup \{d_i\}$
  - r_i je moguće napisati nad Σ

    ako sve znakove d_i zamijene s RI

Regularne definicije

- REGULARNE DEFINICIJE - PRIMJER
  - programske varijable definiramo nad $\Sigma=\{a,b,\ldots z,A,B,\ldots Z,0,1,\ldots 9\}$
  - korištenjem definicija pišemo:

    (i) $\underline{\text{slovo}} \rightarrow A \vee B \vee \ldots \vee Z \vee a \vee b \vee \ldots \vee z$

    (ii) $\underline{\text{brojka}} \rightarrow 0 \vee 1 \vee \ldots \vee 9$

    (iii) $\underline{\text{varijabla}} \rightarrow \underline{\text{slovo}}\ (\underline{\text{slovo}} \vee \underline{\text{brojka}})^*$
  - RI $\underline{\text{varijabla}}$ zadan je nad $\Sigma \cup \{\underline{\text{slovo}}, \underline{\text{brojka}}\}$
  - RI $\underline{\text{varijabla}}$ moguće je napisati korištenjem isključivo Σ ako se umjesto $\underline{\text{slovo}}$, $\underline{\text{brojka}}$ uvrste izrazi (i) i (ii)

    $\underline{\text{varijabla}} \rightarrow (A \vee B \vee \ldots \vee z)\ ((A \vee B \vee \ldots \vee z) \vee (0 \vee 1 \vee \ldots \vee 9))^*$

Regularne definicije

- REGULARNE DEFINICIJE - PRIMJER
  - konstante bez predznaka definiramo nad $\Sigma=\{0,1,2,3,4,5,6,7,8,9,E,.,+,-\}$
  - korištenjem definicija pišemo:

    $\underline{\text{broj}} \rightarrow 0\vee 1\vee 2\vee 3\vee 4\vee 5\vee 6\vee 7 \vee 8\vee 9$

    $\underline{\text{brojke}} \rightarrow \underline{\text{broj}}\ \underline{\text{broj}}^*$

    $\underline{\text{decimale}} \rightarrow .\ \underline{\text{brojke}} \vee \varepsilon$

    $\underline{\text{eksponent}} \rightarrow (E(+ \vee - \vee \varepsilon)\ \underline{\text{brojke}}) \vee \varepsilon$

    $\underline{\text{konstanta}} \rightarrow \underline{\text{brojke}}\ \underline{\text{decimale}}\ \underline{\text{eksponent}}$

2.3.3. Svojstvo napuhavanja

- SVOJSTVO NAPUHAVANJA (Pumping lemma)
  - koristi se za
    - dokazivanje neregularnosti nekih jezika
    - dokazivanje algoritama nepraznosti RJ
    - beskonačnost RJ itd.
  - za RJ postoji DKA M= $M = (Q, \Sigma, \delta, q_0, F)$
  - neka M ima **n** stanja
  - promatrajmo $a_1a_2...a_m$ niz duljine m>n
  - za i=1,2,...,m neka vrijedi $\delta(q_0, a_1a_2...a_i) = q_i$
  - kako je n<m, nije moguće da sva stanja u nizu $q_0,q_2,q_3,...,q_n$ budu različita, barem jedno se **ponavlja**

Svojstvo napuhavanja

- PONAVLJANJE STANJA
  - za neka dva indeksa **j** i **k** stanja u nizu vrijedi $0 \leq j < k \leq n$ i $q_i = q_k$
  - za niz $z = a_1 a_2 \ldots a_m$ vrijedi $1 \leq | a_{j+1} a_{j+2} \ldots a_k | \leq n$
  - pišemo: z = uvw, gdje je
    - podniz $u = a_1 a_2 \ldots a_j$
    - podniz $v = a_{j+1} a_{j+2} \ldots a_k$
    - podniz $u = a_{k+1}$ak+2$_2 \ldots a_m$

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots q_{j-1} \xrightarrow{a_j} q_j = q_k \xrightarrow{a_{k+1}} q_{k+1} \xrightarrow{a_{k+2}} \cdots q_{m-1} \xrightarrow{a_m} q_m$$

$$q_{j+1} \xrightarrow[a_{j+2}]{} q_{j+2} \xrightarrow[a_{j+3}]{} q_{j+3} \xrightarrow[a_{j+4}]{} \cdots q_{k-2} \xrightarrow[a_{k-1}]{} q_{k-1}$$

Svojstvo napuhavanja

- PONAVLJANJE STANJA
  - ako je $q_m \in F$, **uvw** se prihvaća, kao i **uw**
  - kako je $q_i = q_k$ postoji slijed prijelaza iz q_0 u q_m
    $\delta(q_0, uw) = \delta(\delta(q_0, u), w) = \delta(q_j, w) = \delta(q_k, w) = q_m$
  - prihvaća se niz uvvw
    $\delta(q_0, uvvw) = \delta(\delta(q_0, u), vvw) = \delta(q_j, vvw) =$
    $= \delta(\delta(q_j, v), vw) = \delta(q_k, vw) = \delta(q_j, vw) =$
    $= \delta(\delta(q_j, v), w) = \delta(q_k, w) = q_m$
  - prihvaćaju se svi nizovi uv^iw, $i \geq 0$

Svojstvo napuhavanja

- NAPUHAVANJE
  - bilo koji dovoljno dugački niz $z \in L(M)$ može se rastaviti na podnizove: $z=uvw$
  - podniz **v** moguće je proizvoljan broj puta ponoviti (napuhati), jer je $uv^iw \in L(M)$, a M odgovarajući DKA
  - ako RJ sadrži dovoljno dugačak niz $z=uvw$, onda taj jezik sadrži beskonačni skup nizova uv^iw
  - broj ponavljanja i ne mora biti velik

Svojstvo napuhavanja

- DOKAZIVANJE NEREGULARNOSTI

  (i) ako je L regularan, postoji **n** takav da je moguće

  - bilo koji niz $z \in L$ gdje je $|z|>n$
    rastaviti na podnizove $z=uvw$ tako da je:

    $$|uv| \leq n \quad \text{i} \quad 1 \leq |v|$$

  - za bilo koji $i \geq 0$ niz $uv^iw \in L$
  - pokazuje se da n nije veći od broja stanja minimalnog DKA koji prihvaća jezik L

Svojstvo napuhavanja

- DOKAZIVANJE NEREGULARNOSTI PRIMJER
  - jezik $K = \left\{ 0^{\ell^2} \middle| \ell \in N; \ell \geq 1 \right\}$ nije regularan
    - pretpostavi se da je L regularan jezik
    - neka **n** odgovara (i) i neka je $z = 0^{n^2}$ niz jezika L: $|z|=n^2$, $|z|>n$
    - prema (i) niz **z** rastavlja se na podnizove uvw; $1\leq |v|\leq |uv|\leq n$
    - treba utvrditi da li je niz uv^iw element jezika L za bilo koji **i**
    - ako je **i**=2 onda i $|v|\leq |uv|\leq n$ onda je
      $|uvw| = |z| = n^2<|uv^2w| = (n^2+|v|) \leq (n^2+n)$
    - budući da je $(n^2+n) < (n+1)^2$ vrijedi:
      $n^2 < |uv^2w| < (n+1)^2$ tj. $|uv^2w|$ nije kvadrat cijelog broja
    - bez obzira na n i na podjelu uvw, uv^2w nije član jezika
      posljedično L je neregularan

Svojstvo napuhavanja

- ALGORITMI ODLUČIVANJA
  - **nepraznost** regularnog jezika
    - L(M) je neprazan ako DKA M prihvaća niz duljine |z|<n; n je broj stanja DKA M
    - ako je u skupu dohvatljivih stanja jedno prihvatljivo, L(M) je neprazan
  - **beskonačnost** regularnog jezika
    - L(M) je beskonačan ako DKA M prihvaća niz duljine n≤|z|<2n; n je broj stanja DKA M
    - promatra se graf DKA M i dobije se DKA M’ tako da se izuzmu neprihvatljiva stanja za koje ne postoji staza u prihvatljivo
    - L(M) je beskonačan ako graf DKA M’ ima barem jednu zatvorenu petlju

2.4. Gramatika

- REGULARNA GRAMATIKA
  - regularna gramatika generira regularne jezike
  - gramatiku formalno definiramo na svojstvima kontekstno neovisnih gramatika
  - definiraju se svojstva regularne gramatike
  - oblik produkcija ograničen je tako da se jamči generiranje regularnih jezika

2.4.1. Formalna gramatika

- FORMALNA GRAMATIKA
  - koristi se u analizi i generiranju nizova znakova formalnog jezika
  - npr. promatrajmo gramatiku koja generira 16 rečenica
  - s nezavršnim elementima označenim <>:
    - < Rečenica >,<Subjektni skup>,<Predikat>, <Objektni skup>,<Subjekt>,<Objekt>,<Atribut>
  - i sa završnim elementima - šest riječi:
    - djevojke, mačke, gledaju, zbunjene, uplašene, . (točka)

Formalna gramatika

- FORMALNA GRAMATIKA
  - gramatika gradi rečenice primjenom pravila - produkcija

    (1) <Rečenica>→<Subjektni skup><Predikat><Objektni skup>.

    (2) <Subjektni skup>→<Atribut><Subjekt>

    (3) <Objektni skup>→<Atribut><Objekt>

    (4) <Predikat>→gledaju

    (5) <Subjekt>→djevojke

    (6) <Subjekt>→mačke

    (7) <Atribut>→zbunjene

    (8) <Atribut>→uplašene

    (9) <Objekt>→djevojke

    (10) <Objekt>→mačke

Formalna gramatika

- PRIMJENA PRODUKCIJA
  - pravila koristimo kod zamjene nezavršnih dok ne dobijemo samo završne znakove:
    - <Rečenica>
    - <Subjektni skup><Predikat><Objektni skup>.
    - <Atribut><Subjekt><Predikat><Objektni skup>.
    - <Atribut><Subjekt><Predikat><Atribut><Objekt>.
    - <Atribut><Subjekt>**gledaju**<Atribut><Objekt>.
    - <Atribut>**djevojke gledaju**<Atribut> **mačke**.
    - **zbunjene djevojke gledaju uplašene mačke**.

Formalna gramatika

- PRIMJENA PRODUKCIJA
  - relacijom ⇒ označimo primjenu jednog pravila:

<Rečenica>

⇒<Subjektni skup><Predikat><Objektni skup>.

⇒<Atribut><Subjekt><Predikat><Objektni skup>.

⇒<Atribut><Subjekt><Predikat><Atribut><Objekt>.

⇒<Atribut><Subjekt>**gledaju**<Atribut><Objekt>.

⇒<Atribut>**djevojke gledaju**<Atribut><Objekt>.

⇒<Atribut>**djevojke gledaju**<Atribut> **mačke**.

⇒**zbunjene djevojke gledaju** <Atribut> **mačke**.

⇒**zbunjene djevojke gledaju uplašene mačke**.

Formalna gramatika

- PRIMJENA PRODUKCIJA
  - relacijom ⇒* ili $\overset{*}{\Rightarrow}$ označimo primjenu više pravila:

    <Rečenica>

    ⇒*<Atribut><Subjekt><Predikat><Atribut><Objekt>.

    ⇒***zbunjene djevojke gledaju uplašene mačke**.
  - ili:

    <Rečenica>

    ⇒* **uplašene djevojke gledaju zbunjene mačke**.

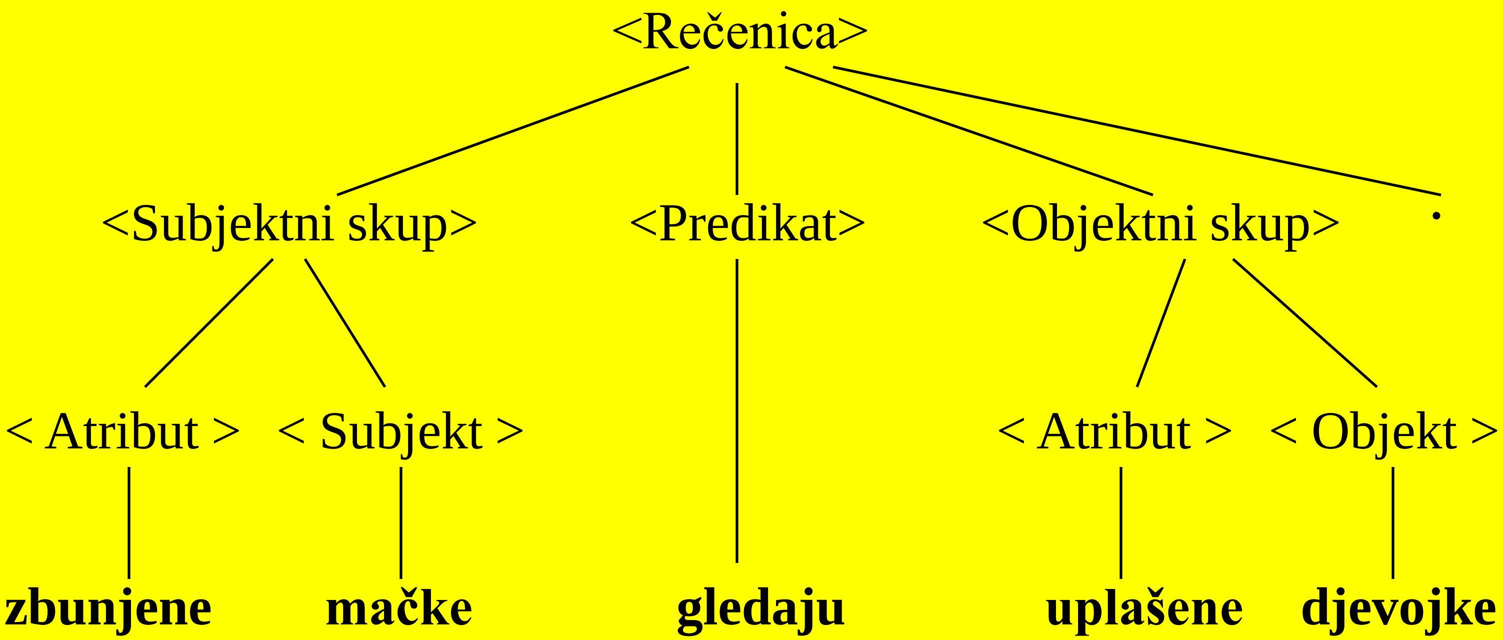
    <Rečenica>

    ⇒* **uplašene mačke gledaju zbunjene djevojke** .

    itd.

Formalna gramatika

- PRIMJENA PRODUKCIJA - STABLO
  - primjena produkcija prikazuje se stablom:



Formalna gramatika

- KONTEKSTNO NEOVISNA GRAMATIKA
  - je uređena četvorka:

    $$G = (V, T, P, S)$$

    V - konačni skup nezavršnih znakova

    T - konačni skup završnih znakova $V \cap T = \emptyset$

    P - konačni skup produkcija oblika $A \rightarrow \alpha$;

    - $A \in V$
    - α je niz: $\alpha \in (V \cup T)^*$, uključuje prazni niz ε

    S - početni nezavršni znak
  - koristi se u definiranju sintakse programskih jezika
  - produkcije su najčešće u BNF (Backus-Naur Form)

Formalna gramatika

- OZNAKE U FORMALNOJ GRAMATICI

  1) A,B,C...S su nezavršni znakovi gramatike

  2) a,b,c...0,1,2,..., **mačke** su završni znakovi gramatike

  3) X,Y,Z su završni ili nezavršni znakovi

  4) u, v, w, x, y i z označavaju nizove završnih znakova

  5) α, β, γ označavaju nizove završnih i nezavršnih znakova

  – ima li više produkcija za isti nezavršni znak koristimo |, npr. $A \rightarrow a$ i $A \rightarrow b$ piše se $A \rightarrow a|b$

Formalna gramatika

- FORMALNA GRAMATIKA PRIMJER

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

  – generira nizove završnih znakova nizovima produkcija

    1) $E \Rightarrow a$

    2) $E \Rightarrow E+E \Rightarrow^* a+a$

    3) $E \Rightarrow E*E \Rightarrow^* a*a$

    4) $E \Rightarrow E*E \Rightarrow (E)*E \Rightarrow (E+E)*E \Rightarrow^* (a+a)*a$

  – ako a označava konstantu ili varijablu, G generira aritmetičke izraze programskog jezika

Formalna gramatika

- FORMALNA GRAMATIKA OZNAKE
  - za $G = (V, T, P, S)$ definira se relacija $\Rightarrow_G$ nad nizovima iz skupa $(V \cup T)^*$
  - ako je $A \rightarrow \beta$ iz P, ako su α, i γ iz $(V \cup T)^*$, tada vrijedi:
    $$\alpha A \gamma \Rightarrow_G \alpha\beta\gamma$$
  - definiramo relaciju $\Rightarrow_G^*$ tako da vrijedi:
    $\alpha_1 \Rightarrow_G \alpha_2 \Rightarrow_G \alpha_3 \ldots\ldots \alpha_{m-1} \Rightarrow_G \alpha_m = \alpha_1 \Rightarrow_G^* \alpha_m$
  - $\Rightarrow_G^*$ je refleksivno i tranizitivno okruženje $\Rightarrow_G$
  - znade li se G, skraćeno se piše $\Rightarrow$ i $\Rightarrow^*$
  - za i produkcija piše se $\alpha \Rightarrow^i \beta$

Formalna gramatika

- FORMALNA GRAMATIKA I JEZICI
  - $G = (V, T, P, S)$ generira jezik
    $$L(G) = \{w | w \in T^*; S \Rightarrow_G^* w\}$$
  - niz je u jeziku $L(G)$ ako vrijedi
    - u nizu su isključivo završni znakovi $w \in T^*$;
    - niz je moguće generirati iz početno nezavršnog znaka S
  - G_1 i G_2 su istovjetne ako vrijedi $L(G_1) = L(G_2)$

Formalna gramatika

- KONTEKSTNO NEOVISNI JEZICI
  - kontekstno neovisna gramatika generira kontekstno neovisne jezike
  - klasa kontekstno neovisnih jezika sadrži jezike koji nisu regularni, npr. $N=\{wcw^R\}$je neregularan
  - gramatika: $G = (\{S\}, \{a, b, c\}, \{S \rightarrow aSa|bSb|c\}, S)$
  - generira jezik

    $S \Rightarrow aSa \Rightarrow aaSaa \Rightarrow aabSbaa \Rightarrow aabbSbbaa \Rightarrow \ldots$

    ili $S \Rightarrow^* wSw^R$

    konačno: $S \Rightarrow^* wSw^R \Rightarrow wcw^R$; $L(G)=\{wcw^R\}=N$

Formalna gramatika

- KONTEKSTNO NEOVISNI JEZICI
  - može se pokazati da za regularni jezik postoji kontekstno neovisna gramatika
  - regularni jezik je dio skupa kontekstno neovisnih jezika
  - klasa kontekstno neovisnih jezika je skup jezika širi od skupa regularnih jezika jer je $N=\{wcw^R\}$ neregularan

Skup kontekstno neovisnih jezika KNJ

Regularni jezici $RJ \subset KNJ$

Formalna gramatika

- GENERATIVNO STABLO
  - za G = (V, T, P, S) stablo je generativno ako

    1) čvorove označimo znakovima $V \cup T \cup \{\varepsilon\}$
    2) korijen stabla označen je početnim nezavršnim znakom S
    3) unutrašnji čvorovi označeni su nezavršnim $A \in V$
    4) za čvor A i djecu $X_1, X_2, \ldots X_n$ vrijedi produkcija iz P

    $$A \to X_1 X_2 \ldots X_n$$

    5) znakom ε označava se isključivo list stabla;

    taj list je jedino dijete svog roditelja

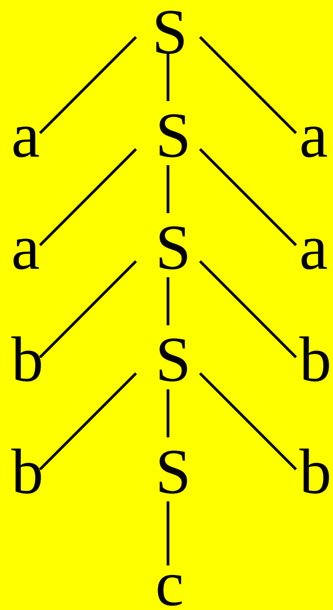
    6) listovi stabla označeni su znakovima skupa $T \cup \{\varepsilon\}$

    čitani slijeva na desno čine generirani niz jezika L(G)

  - $S \Rightarrow_G^* w$ vrijedi samo ako postoji generativno stablo

Formalna gramatika

- GENERATIVNO STABLO PRIMJER



2.4.2. Regularna gramatika

- REGULARNA GRAMATIKA
  - ujedno je i kontekstno neovisna gramatika
  - konstruirajmo gramatiku za regularni jezik zadan DKA
  - time dokazujemo da je gramatika regularna

Regularna gramatika

- KONSTRUKCIJA GRAMATIKE PREMA DKA
  - za regularni jezik zadan s DKA $M = (Q, \Sigma, \delta, q_0, F)$ gradi se kontekstno neovisna gramatika $G = (V, T, P, S)$ tako da je $L(M) = L(G)$
  - primjenjujemo pravila:
    - $T = \Sigma$; završni znakovi gramatike su ulazni znakovi automata
    - $V = Q$; nezavršni znakovi su stanja automata
    - $S = q_0$; početno stanje je početni nezavršni znak
    - na temelju prijelaza DKA $\delta(A, a) = B$ gradimo produkciju

      $$A \to aB$$
    - za prihvatljiva stanja $A \in F$ gradimo produkcije $A \to \varepsilon$

Regularna gramatika

- KONSTRUKCIJA G iz DKA PRIMJER
  - za DKA

    $M = (\{S,A,B\}, \{a,b\}, \delta, S, \{S,B\})$

    gradi se kontekstno neovisna gramatika

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

    tako da je L(M) = L(G), pri čemu su δ i P:

	a	b	⊥			
S	A	B	1	$S \to aA$	$S \to bB$	$S \to \varepsilon$
A	B	A	0	$A \to aB$	$A \to bA$	
B	S	A	1	$B \to aS$	$B \to bA$	$B \to \varepsilon$

Regularna gramatika

- KONSTRUKCIJA G iz DKA PRIMJER
  - za niz aba postoji slijedeći niz prijelaza:

$$S \xrightarrow{a} A \xrightarrow{b} A \xrightarrow{a} B \;; \qquad B \in F$$

  - gramatika generira niz aba:

$$S \Rightarrow aA \Rightarrow abA \Rightarrow abaB \Rightarrow aba$$

Regularna gramatika

- KONSTRUKCIJA G iz DKA PRIMJER
  - drugi primjeri nizova su:

$\varepsilon:$	$S \xrightarrow{\varepsilon} S;$	$B \in F$	$S \Rightarrow \varepsilon$
$a:$	$S \xrightarrow{a} A;$	$A \notin F$	$S \Rightarrow aA \Rightarrow$
$aa:$	$S \xrightarrow{a} A \xrightarrow{a} B;$	$B \in F$	$S \Rightarrow aA \Rightarrow aaB \Rightarrow aa$
$bbb:$	$S \xrightarrow{b} B \xrightarrow{b} A \xrightarrow{b} A;$	$A \notin F$	$S \Rightarrow bB \Rightarrow bbA \Rightarrow bbbA \Rightarrow$
$bbba:$	$S \xrightarrow{b} B \xrightarrow{b} A \xrightarrow{b} A \xrightarrow{a} B;$	$B \in F$	$S \Rightarrow bB \Rightarrow bbA \Rightarrow bbbA \Rightarrow bbbaB \Rightarrow bbba$

Regularna gramatika

- ISTOVJETNOST G i DKA
  - prihvaća li DKA isti jezik koji generira G, DKA i G su istovjetni: L(DKA) = L(G)
  - na temelju pravila vrijedi:
    (i) $A \Rightarrow^* wB$ ako je $\delta(A, w) = B$
  - neka je $C = \delta(q_0, v)$ prihvatljivo stanje, prihvaća se niz v
  - na temelju (i) vrijedi: $q_0 \Rightarrow^* vC$
  - ako je C prihvatljivo postoji produkcija $C \rightarrow \varepsilon$
  - vrijedi: $q_0 \Rightarrow^* vC \Rightarrow v$
    ako DKA prihvaća v, tada ga G generira!

Regularna gramatika

- ISTOVJETNOST G i DKA
  - neka G generira v, tada je $q_0 \Rightarrow^* vC \Rightarrow v$
  - pošto je na temelju (i) $\delta(q_0, v) = C$ i postoji $C \rightarrow \varepsilon$
    tada je C prihvatljivo
  - ako G generira v, tada ga DKA prihvaća!

Regularna gramatika

- KONSTRUKCIJA NKA ZA JEDNOSTAVNI G
  - koristimo gramatiku s produkcijama oblika

    $A \rightarrow aB$ i $C \rightarrow \varepsilon$
  - to su upravo oblici koji nastaju konstrukcijom gramatike na osnovu DKA; pravila su:
    - $\Sigma = T$; završni znakovi gramatike su ulazni znakovi automata
    - $Q = V$; nezavršni znakovi su stanja automata
    - $q_0 = S$; početno stanje je početni nezavršni znak
    - na temelju produkcije $A \rightarrow aB$ gradi se prijelaz DKA

      $$\delta(A, a) = \delta(A, a) \cup B$$

      jer su moguće višestruke produkcije iz A,a
    - ako postoji produkcija $A \rightarrow \varepsilon$, stanje A je prihvatljivo; $A \in F$

Regularna gramatika

- KONSTRUKCIJA NKA PRIMJER
  - jezik G generira programske varijable

    $G = (\{V,B\}, \{s,b\}, \{V \rightarrow sB,\ B \rightarrow bB|sB|\varepsilon\}, V)$
  - nizovi započinju slovom i nastavljaju se proizvoljnom kombinacijom brojki i slova
  - gradimo NKA $M = (\{V,B\}, \{s,b\}, \delta, V, \{B\})$;

    $\delta = \{\}$ $A \rightarrow aB$ i $C \rightarrow \varepsilon$

	s	b	⊥
V	B	∅	0
B	B	B	1

Regularna gramatika

- DESNO LINEARNA GRAMATIKA
  - **desno linearna** gramatika:
    - ima najviše jedan nezavršni znak na desnoj strani
    - $A \rightarrow wB$ ili $A \rightarrow w$
    - $A, B \in V$; $w \in T^*$
- LIJEVO LINEARNA GRAMATIKA
  - **lijevo linearna** gramatika:
    - ima najviše jedan nezavršni znak na lijevoj strani
    - $A \rightarrow Bw$ ili $A \rightarrow w$
  - lijevo i desno linearne gramatike su **regularne**!

Regularna gramatika

- KONSTRUKCIJA NKA IZ LINEARNIH G
  - L je regularan ako postoji desno linearna algebra G_D

    $L = L(G_D)$
  - L je regularan ako postoji lijevo linearna algebra G_L

    $L = L(G_L)$
  - za bilo koju G_D ili G_L moguće je izgraditi DKA M

    $L(M) = L(G)$
  - složene produkcije moguće je preurediti na jednostavne
  - primijene se pravila za konstrukciju NKA

Regularna gramatika

- KONSTRUKCIJA NKA IZ DLG
  - Složenu produkciju

    $A \rightarrow abbS$
  - rastavimo na jednostavne:

    $A \rightarrow a[bbS]$

    $[bbS] \rightarrow b[bS]$

    $[bS] \rightarrow bS$
  - generirali smo nove nezavršne znakove

    $[bbS]$ i $[bS]$
  - nove tri produkcije generiraju isti međuniz

Regularna gramatika

- KONSTRUKCIJA NKA IZ DESNO LIN. G
  - Složenu produkciju

    $$S \rightarrow bc$$

  - rastavimo na jednostavne:

    $$S \rightarrow bc[\varepsilon]$$
    $$[\varepsilon] \rightarrow \varepsilon$$
    $$S \rightarrow b[c\varepsilon]$$
    $$[c\varepsilon] \rightarrow c[\varepsilon]$$

  - generirali smo nove nezavršne znakove

    $[\varepsilon]$ i $[c\varepsilon]$

  - nove produkcije generiraju isti međuniz

Regularna gramatika

- KONSTRUKCIJA NKA IZ DESNO LIN. G
  - Složenu produkciju

    $S \rightarrow A$
  - rastavimo na jednostavne:

    $S \rightarrow$ desne strane svih produkcija od A
  - ako postoje produkcije $A \rightarrow cA$ i $A \rightarrow \varepsilon$ bit će

    $S \rightarrow cA$

    $S \rightarrow \varepsilon$
  - jer S preko A generira sve međunizove od A

Regularna gramatika

- KONSTRUKCIJA NKA IZ DESNO LIN. G
  - Za primjer gramatike

$$
\begin{array}{ll}
S \to aA & A \to abbS \\
S \to bc & A \to cA \\
S \to A & A \to \varepsilon
\end{array}
$$

  - dobije se:

$$
\begin{array}{lll}
S \to aA & S \to a[bbS] & [bbS] \to b[bS] \\
S \to b[c\varepsilon] & S \to cA & [bS] \to bS \\
[c\varepsilon] \to c[\varepsilon] & S \to \varepsilon & A \to cA \\
[\varepsilon] \to \varepsilon & A \to a[bbS] & A \to \varepsilon
\end{array}
$$

Regularna gramatika

- KONSTRUKCIJA NKA IZ DESNO LIN. G
  - dobijemo automat:

	a	b	c	⊥
S	A, [bbS]	[cε]	A	1
[cε]	∅	∅	[ε]	0
[ε]	∅	∅	∅	1
A	[bbS]	∅	A	1
[bbS]	∅	[bS]	∅	0
[bS]	∅	S	∅	0

Regularna gramatika

- KONSTRUKCIJA ε-NKA IZ LIJEVO LIN. G
  - neka je $G = (V, T, P, S)$ lijevo linearna gramatika
  - ε-NKA M': L(M')=L(G) konstruira se

    1) izgradi se desno linearna gramatika $G' = (V, T, P', S)$

    $$P' = \{A \to \alpha^R \mid A \to \alpha \in P\}$$

    G' generira obrnute nizove $L(G')=L(G)^R$

    2) na temelju DLG G' konstruira se NKA M:

    $$L(M) = L(G')=L(G)^R$$

    3) na temelju NKA M izgradi se ε-NKA M':

    $$L(M') = L(M)^R = L(G')^R = L(G)^{RR} = L(G)$$

Regularna gramatika

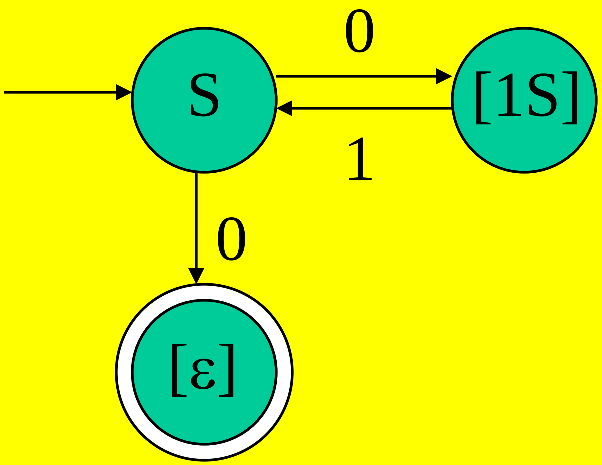
- KONSTRUKCIJA ε-NKA IZ LIJEVO LIN. G
  - ε-NKA M’ izgradi se postupkom:
    - preuredi se NKA M da ima samo jedno prihvatljivo stanje
    - ako ih ima više, generira se novo, stara više nisu prihvatljiva, a stara se s novim povežu ε-prijelazima
    - početno stanje ε-NKA M’ je prihvatljivo stanje NKA M
    - prihvatljivo stanje ε-NKA M’ je početno stanje NKA M
    - funkcija prijelaza ε-NKA M’ gradi se zamjenom smjera usmjerenih grana u dijagramu stanja NKA M
  - izgrađeni ε-NKA M’ prihvaća jezik L(M’)
  - u L(M’) su nizovi obrnutog redoslijeda od L(M)

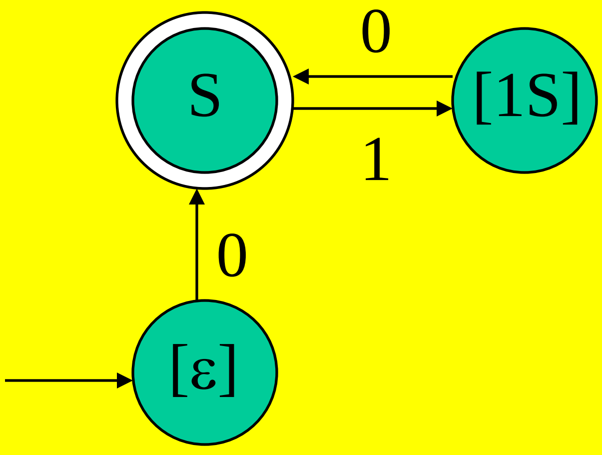
Regularna gramatika

- PRIMJER KONSTRUKCIJE ε-NKA IZ LLG
  - zadana je LLG G = ({S}, {0,1}, P, S)
    $P = \{S \rightarrow S10 \mid 0\}$
  - G generira jezik 0(10)* s desna na lijevo
  - izgradimo DLG G': $P' = \{S \rightarrow 01S \mid 0\}$
  - G' generira jezik (01)*0 s lijeva na desno
  - za DLG G' konstruiramo NKA M koji prihvaća (01)*0
  - iz NKA M konstruiramo ε-NKA M'
    - zamjenom prihvatljivog i početnog stanja
    - okretanjem smjera prijelaza

Regularna gramatika

- PRIMJER KONSTRUKCIJE ε-NKA IZ LLG
  - dobiveni M i M’ su:





Regularna gramatika

- KONSTRUKCIJE LLG IZ ε-NKA
  - LLG gramatiku G konstruiramo postupkom:
    - zada se jezik L
    - konstruira se ε-NKA M koji prihvaća $L(M) = L^R$
    - na temelju ε-NKA M konstruira se DLG G koja generira $L(G) = L(M) = L^R$
    - desne strane produkcija napišu se obrnutim redoslijedom
  - izgrađena G’ je LLG: $L(G') = L(G)^R = L(M)^R = L$