Formalni jezici i jezični procesori I

REGULARNI IZRAZI

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

Uvod

- nastali 50-tih god. kao formalno sredstvo za opis sintakse programskih jezika
- namijenjeni su opisu traženih izraza ili uzoraka iz ulaznoga teksta
  - (0+1)1*
  - 0: {0}
  - 1: {1}
  - (0+1): ({0}+{1})
  - 1*: {1}*

Primjeri regularnih izraza

- broj: **[0-9]** 0, 5,...
  - []-potraži jednoznamenkasti broj iz liste 0-9
- cijeli broj: **[0-9]+** 5, 10, 35, 128,..
  - + potraži **jedno- ili više**znamenkasti broj
- predznak: **-?[0-9]+** -2, 51,...
  - ?- potraži **nijedan ili jedan** cjelobrojni višeznamenkasti
- decimalni: **[0-9]*\.[0-9]+** 0.0, 4.5, .31,...
  - *- potraži **nijednog ili više** izraza koji odgovaraju prethodnome

Primjeri regularnih izraza II

- slovo: **[A-Z]** A,B,C..,Z
  - potraži jedno **veliko tiskano** slovo iz liste A-Z
- riječ: **[Jj]adran** jadran, Jadran, jadranski,..
  - riječ koja sadrži **malo ili veliko početno** slovo
- niz riječi: **‘valovito i’**

  more valovito i umjereno valovito,...
  - potraži **točan niz riječi** u tekstu

Upotreba RI

- pretraživanje interneta
- u mnogim UNIX alatima: *grep*, *sed*, *awk*, *gawk*,...
- u programskim jezicima *Perl*, *Python*, *Java, JavaScript,*…
- programima za uređivanje teksta: *vi*, *find* izbornik u *MS Wordu*,...
- ...

Regularni izrazi

- regularni jezik L(r) opisujemo regularnim izrazima r
  - ako je jezik moguće opisati regularnim izrazom onda je jezik regularan
  - za bilo koji jezik L(r) definiran regularnim izrazom r moguće je izgraditi DKA M za koji vrijedi L(M)=L(r)
- regularni jezik je pravi podskup skupa svih jezika
- neregularne jezike nije moguće opisati regularnim izrazima i za njih ne možemo napraviti konačni automat
  - za neregularne jezike koriste se modeli potisnih automata i Turingovi strojevi (omogućavaju pamćenje)

Regularni jezici i DKA

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

Regularni jezici

$RJ \subseteq 2^{\Sigma^*}$

DKA

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

Konstrukcija minimalnog DKA za jezik K

REGULARNI JEZIK K

Opis jezika K regularnim izrazima r
L(r)=K

ε-NKA M za koji vrijedi
L(M)=L(r)

NKA M' za koji vrijedi
L(M')=L(M)

DKA M'' za koji vrijedi
L(M'')=L(M')

DKA M''' s minimalnim brojem stanja za koji vrijedi
L(M''')=L(M'')=L(M')=L(M)=L(r)=K

Pojednostavljivanje RI

- rekurzivna pravila
- precedenca operatora
- asocijativnost
- algebarski zakoni
  - komutativnost
  - asocijativnost
  - distributivnost
  - idempotentnost

Rekurzivna pravila

- $\emptyset$ je regularan izraz; $L(\emptyset)=\{\ \}$
- ε je regularan izraz; $L(\varepsilon)=\{\varepsilon\}$
- $\forall a\in\Sigma$, a je regularan izraz; $L(a)=\{a\}$
- ako su r i s regularni izrazi koji označavaju L(r) i L(s) onda:
  - (r)+(s) je regularan izraz [ili (r)|(s)]; $L((r)+(s))=L(r)\cup L(s)$ – unija jezika
  - (r)(s) je regularan izraz; $L((r)(s))=L(r)L(s)$ – nadovezivanje jezika, konkatenacija
  - (r)* je regularan izraz; $L((r)^*)=L(r)^*$ – Kleenov operator na jezikom L(r)

Precedenca operatora i asocijativnost

- unarni operator **a*:** lijevo asocijativan i najviše prednosti
- operator nadovezivanja **ab** : lijevo asocijativan i veće prednosti od +
- operator **a+b**: lijevo asocijativan i najmanje prednosti

- r i s su istovjetni r=s ukoliko označavaju iste jezike L(r)=L(s)
  - npr. (a)+((b)*(c)) je isto kao a+b*c
  - L(a+b*c)={ a, c, bc, bbc, bbbc, bbbbc, ..., bb..bbc, ...}

Algebarski zakoni

• $r+s=s+r$	+ - komutativan
• $r+(s+t)=(r+s)+t$	+ - asocijativan
• $(rs)t=r(st)$	nadovezivanje – asocijativno
• $r(s+t)=rs+rt$	distributivnost nadovezivanja
$(s+t)r=sr+tr$	nad +
• $\varepsilon r = r\varepsilon = r$	ε je neutralni el. nadovezivanja
• $r^*=(r+ \varepsilon)^*$	relacija operatora + i *
• $r^{**}=r^*$	* je idempotentan

Primjeri regularnih izraza

Binarna abeceda $\Sigma=\{0,1\}$

- regularni izraz 01: jezik $L(01)=\{01\}$
- regularni izraz 0+1: jezik $L(0+1)=\{0,1\}$
- regularni izraz (0+1)(0+1): jezik $L((0+1)(0+1))=\{00,01,10,11\}$
- regularni izraz 1*: jezik $L(1^*)=\{\varepsilon,1,11,111,...\}$
- regularni izraz (0+1)*: jezik $L((0+1)^*)=\{\varepsilon,0,1,00,01,10,11, 000, 001,1000, ...111101,...\}$
- regularni izraz (0+1)*00(0+1)* :barem 2 uzastopne nule jezik $L((0+1)^*00(0+1)^*)=\{00,000,100,001,1100,1000, 1001, ...110011001,...\}$
- regularni izraz 0*1* : proizvoljnom broju nula sljedi proizvoljan broj jedinica

  jezik $L(0^*1^*)=\{\varepsilon,0,1,00,01,011,000111,...00001111.... \}$

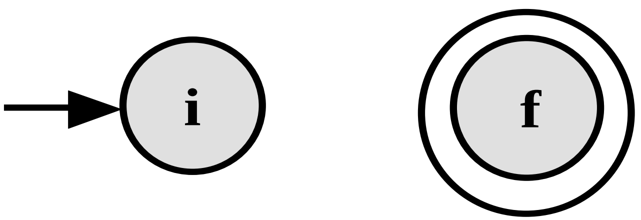
Primjeri regularnih izraza II

Binarna abeceda $\Sigma=\{0,1\}$

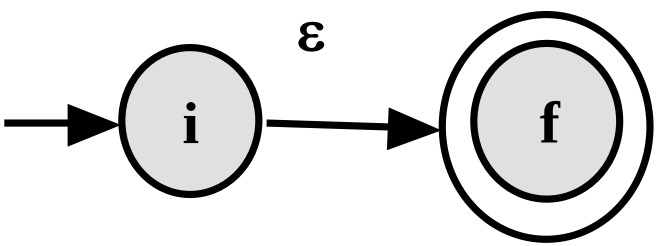
- 0^*10^* $L(0^*10^*)=\{w|w$ ima točno jednu jedinicu$\}$
- $\Sigma^*1\Sigma^*$ $L(\Sigma^*1\Sigma^*)=\{w|w$ barem jednu jedinicu$\}$
- $\Sigma^*001\Sigma^*$ $L(\Sigma^*001\Sigma^*)=\{w|w$ sadrži podniz 001$\}$
- $(\Sigma\Sigma)^*$ $L((\Sigma\Sigma)^*)=\{w|w$ je niz parne dužine$\}$
- $(\Sigma\Sigma\Sigma)^*$ $L((\Sigma\Sigma\Sigma)^*)=\{w|w$ je niz čija dužine je višekratnik od 3$\}$
- $(0+\varepsilon)1^*=01^*+1^*$ $L((0+\varepsilon)1^*)=\{w|w$ je niz jedinica koji može započeti s jednom 0$\}$

7 pravila konstrukcije ε-NKA iz regularnih izraza I

- **p1:** za regularan izraz ∅; L(∅)={ } konstruira se ε-NKA **M=({i,f}, Σ, {},i,{f})**

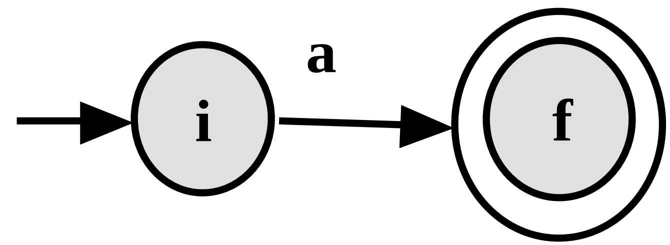


- **p2:** za regularan izraz ε; L(ε)={ε} konstruira se ε-NKA **M=({i,f}, Σ, {δ(i, ε)=f},i,{f})**
  – za bilo koji $b \in \Sigma$, $\delta(f, b)=\{\}$
  – M prihvaća isključivo prazni niz ε



7 pravila konstrukcije ε-NKA iz regularnih izraza II

- **p3:** za regularan izraz a; L(a)={a} konstruira se ε-NKA **M=({i,f}, Σ, {δ(i, a)=f},i,{f})**
  - za bilo koji $b \in (\Sigma \cup \{\varepsilon\})$ i $b \neq a$: $\delta(f, b)=\{\}$
  - M prihvaća isključivo niz a
  - M ne prihvaća niz ε

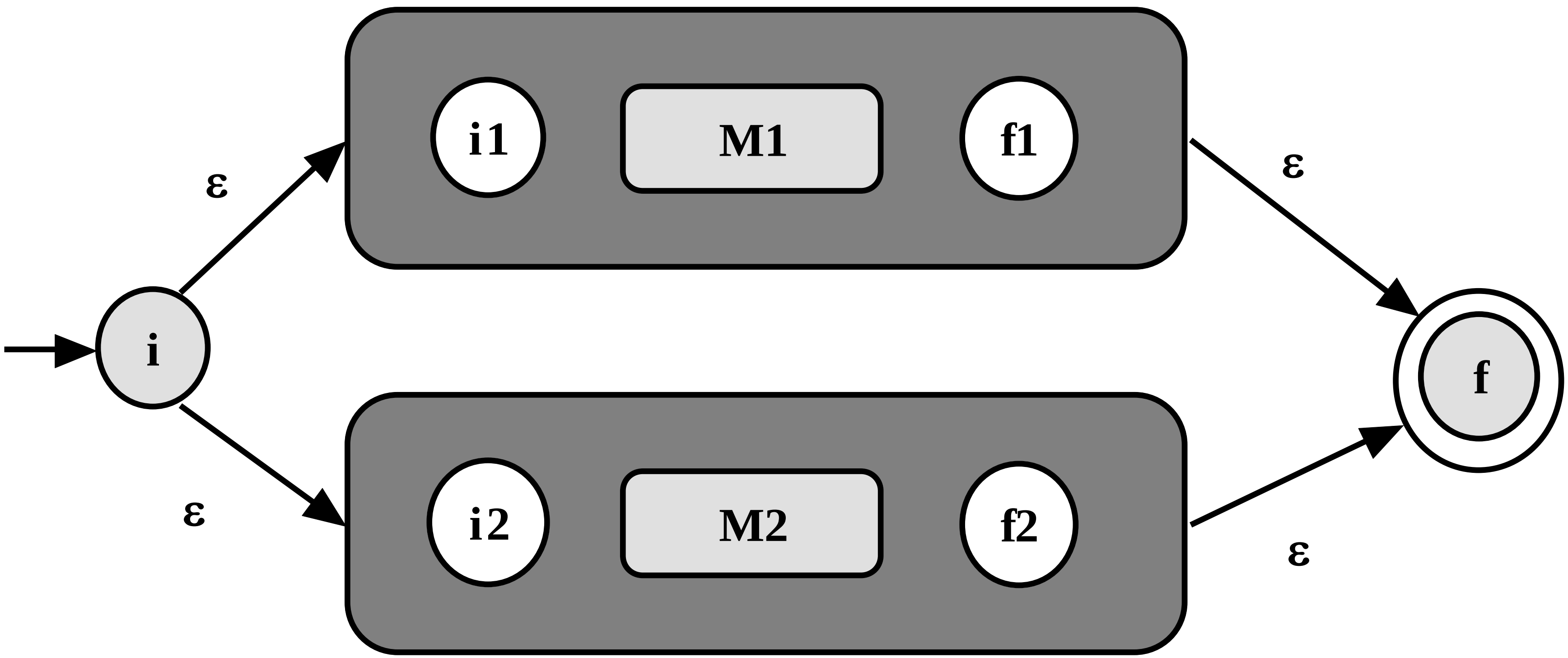


7 pravila konstrukcije ε-NKA iz regularnih izraza III

- **p4:** za regularan izraz r_1+r_2; $L(r_1+r_2)=L(r_1)\cup L(r_2)$ konstruira se ε-NKA **$M= (Q_1\cup Q_2\cup\{i,f\}, \Sigma_1 \cup \Sigma_2, \delta, i, \{f\})$**
  - ukoliko su prije izgrađeni ε-NKA $M_1=(Q_1, \Sigma_1, \delta_1, i_1,\{f_1\})$ i $M_2=(Q_2, \Sigma_2, \delta_2, i_2,\{f_2\})$ takvi da je $L(M_1)=L(r_1)$ i $L(M_2)=L(r_2)$
  - i nema prijelaza iz stanja f_1 i f_2 niti za jedan ulazni znak i $Q_1\cap Q_2 =\{\}$
- novo početno stanje je *i* a prihvatljivo stanje je *f*
  - stanja i_1 i_2 nisu više početna i stanja f_1 f_2 nisu više prihvatljiva
- i funkcija δ se određuje:
  - $\delta(i, \varepsilon)=\{i_1, i_2\}$
  - $\delta(f_1, \varepsilon)=\delta(f_2, \varepsilon)= \{f\}$
  - $\delta(q, a)= \delta_1(q, a)$: $\forall q\in(Q_1\setminus\{f_1\})$, $\forall a\in(\Sigma_1\cup \{\varepsilon\})$
  - $\delta(q, b)= \delta_2(q, b)$: $\forall q\in(Q_2\setminus\{f_2\})$, $\forall b\in(\Sigma_2\cup \{\varepsilon\})$

7 pravila konstrukcije ε-NKA iz regularnih izraza IV

- **p4 (II):**

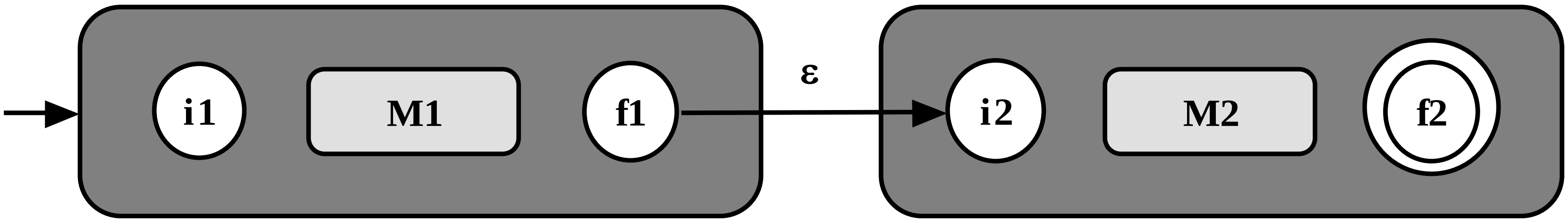


7 pravila konstrukcije ε-NKA iz regularnih izraza V

- **p5:** za regularan izraz r_1r_2; $L(r_1r_2)=L(r_1)L(r_2)$ konstruira se ε-NKA $\mathbf{M= (Q_1\cup Q_2 , \Sigma_1 \cup \Sigma_2, \delta, i_1, \{f_2\})}$
  - ukoliko su prije izgrađeni ε-NKA $M_1=(Q_1, \Sigma_1, \delta_1, i_1,\{f_1\})$ i $M_2=(Q_2, \Sigma_2, \delta_2, i_2,\{f_2\})$ takvi da je $L(M_1)=L(r_1)$ i $L(M_2)=L(r_2)$
  - i nema prijelaza iz stanja f_1 i f_2 niti za jedan ulazni znak i $Q_1\cap Q_2 =\{\}$
- novo početno stanje je i_1 a prihvatljivo stanje je f_2
  - stanje i_2 nije više početno i stanje f_1 nije više prihvatljivo
- i funkcija δ se određuje:
  - $\delta(f_1, \varepsilon)= \{i_2 \}$
  - $\delta(q, a)= \delta_1(q, a)$: $\forall q\in(Q_1\setminus\{f_1\})$, $\forall a\in(\Sigma_1\cup \{\varepsilon\})$
  - $\delta(q, b)= \delta_2(q, b)$: $\forall q\in(Q_2\setminus\{f_2\})$, $\forall b\in(\Sigma_2\cup \{\varepsilon\})$

7 pravila konstrukcije ε-NKA iz regularnih izraza VI

- **p5 (II):**

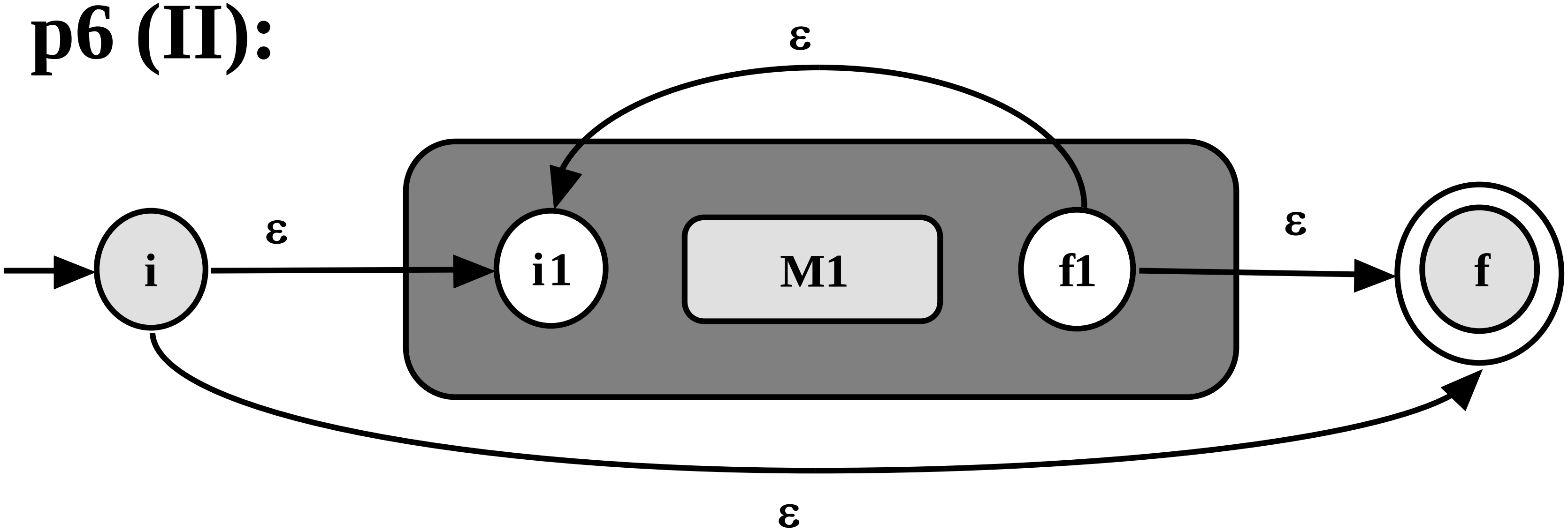


7 pravila konstrukcije ε-NKA iz regularnih izraza VII

- **p6:** za regularan izraz r_1*; $L(r_1^*)=L(r_1)^*$ konstruira se ε-NKA **$M= (Q_1 \cup \{i,f\}, \Sigma_1, \delta, i, \{f\})$**
  - ukoliko je prije izgrađeni ε-NKA $M_1=(Q_1, \Sigma_1, \delta_1, i_1,\{f_1\})$ takav da je $L(M_1)=L(r_1)$ i nema prijelaza iz stanja f_1 niti za jedan ulazni znak
  - novo početno stanje je *i* a prihvatljivo stanje je *f* (stanje i_2 nije više početno i stanje f_1 nije više prihvatljivo)
- i funkcija δ se određuje:
  - $\delta(i, \varepsilon)= \delta(f_1, \varepsilon)= \{i_1,f\}$
  - $\delta(q, a)= \delta_1(q, a)$: $\forall q\in(Q_1\backslash\{f_1\})$, $\forall a\in(\Sigma_1\cup \{\varepsilon\})$

7 pravila konstrukcije ε-NKA iz regularnih izraza VIII

- **p6 (II):**



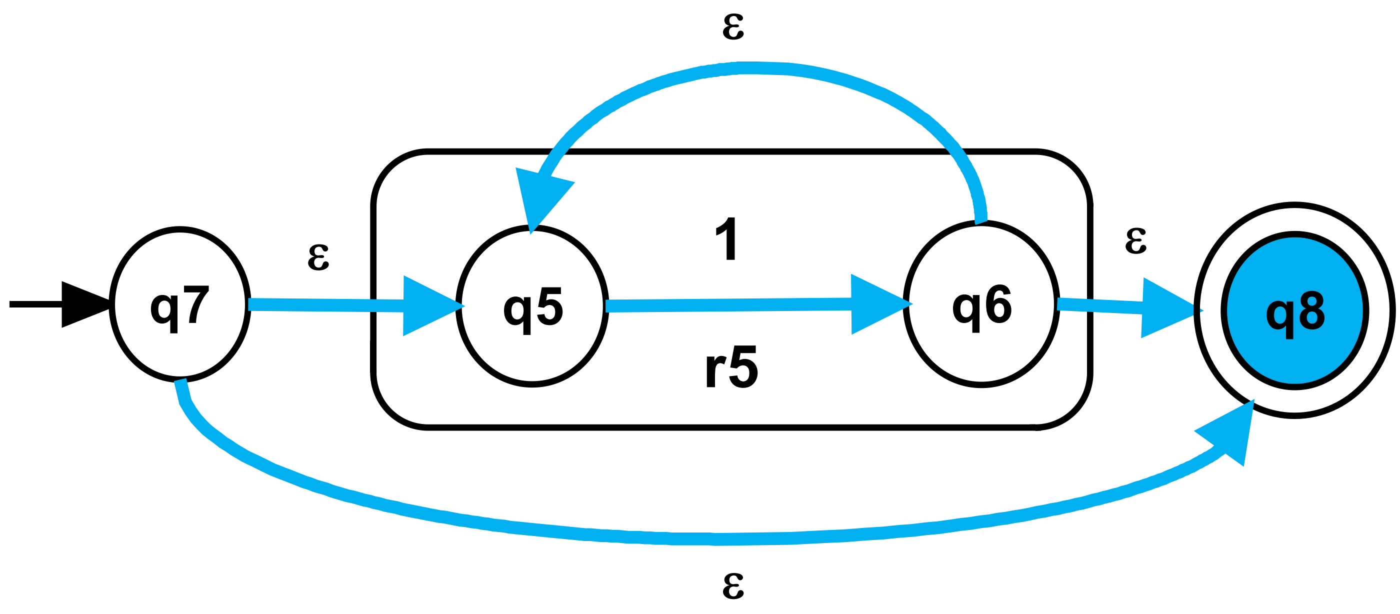
- **p7:** budući da je L((r))=L(r)za ε-NKA M regularnog izraza r uzima se ε-NKA M_1 regularnog izraza r jer je $L(M_1)$ =L(r)=L((r))

Primjer

- pokazuje vezu regularnih izraza i konačnih automata (ε-NKA)
- za regularni izraz **r=01*+1** konstuira se ε-NKA
- postoji 7 pravila konstrukcije

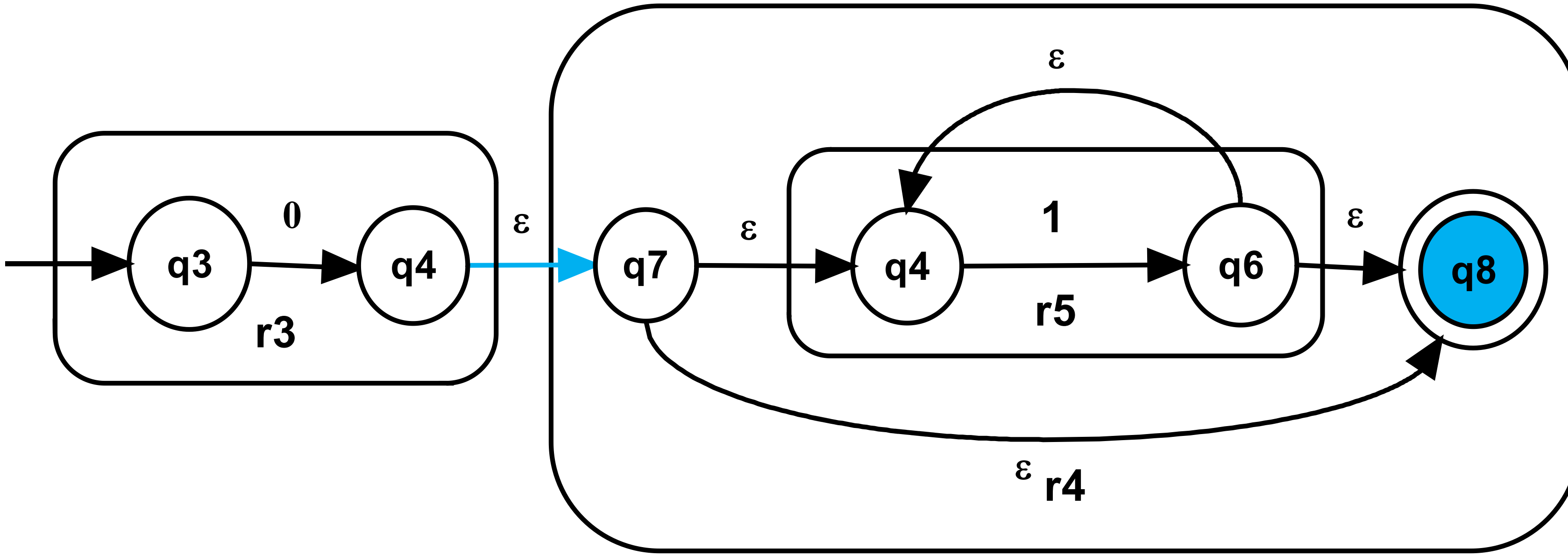
Primjer:
za r=01*+1 konstuiraj ε-NKA II

- r4=1*=(r5)*



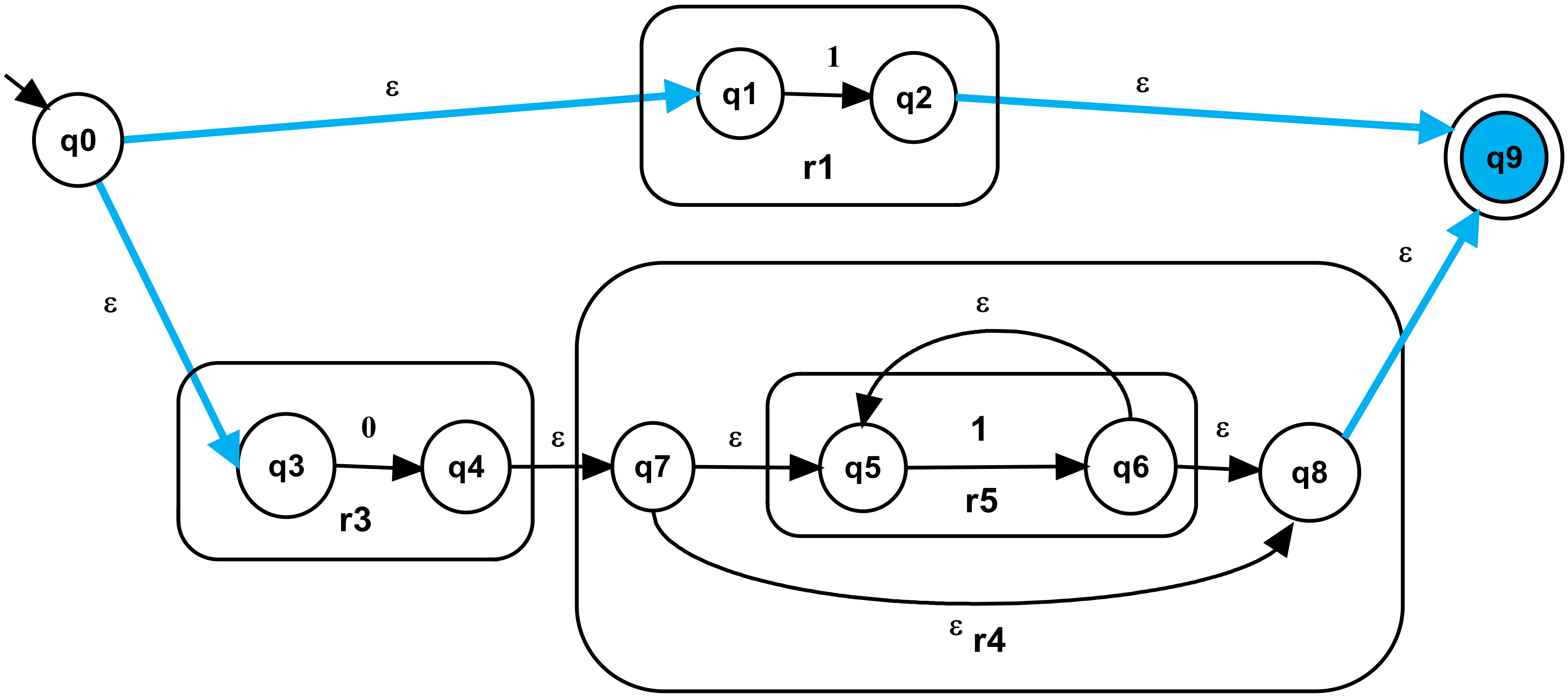
Primjer:
za r=01*+1 konstuiraj ε-NKA III

- R1=01*=r3r4



Primjer:
za r=01*+1 konstuiraj ε-NKA IV

r=01*+1=r1+r2



Regularni izrazi

- http://www.regexbuddy.com/
- http://regexpal.com/
- http://www.nregex.com/nregex/default.aspx
- http://osteele.com/tools/reanimator/

Primjena RI

- **grep**
  - Unix alat za pretraživanje uzoraka u datotekama ili na standardnom ulazu
- **Perl**
  - je skriptni (interpreter) programski jezik
    - prevodioc može prevesti Perl u C i napraviti izvršni kod
  - slobodno dostupan
  - izvorno namijenjen za obradu teksta, razvio se u cjeloviti programski alat: sistemsko programiranje, internetno programiranje,...

grep

- *Globally Search for Regular Expressions and Print*
- *General Regular Expression Printer*
- *Global Regular Expression Parser*

- osnovna namjena je prikaz svih redaka datoteke koji odgovaraju navedenom regularnom izrazu:

```
$ grep regularni_izraz datoteka.txt
```

Primjeri RI - grep

- primjeri izraza za pretraživanje unutar tekstualne datoteke primjer.txt:
  - $ grep Jadran primjer.txt

    traži niz znakova koji sadrži riječ Jadran: Jadran, Jadranka,..
  - $ grep jugo primjer.txt

    traži niz znakova koji sadrži niz jugo: jugo, jugozapadni, jugoistočnjak, ..
  - $ grep “jugo “ primjer.txt

    traži niz znakova koji sadrži samo riječ jugo_
  - $ grep ^[Nn]a primjer.txt

    traži nizove “Na” i “na” na početku retka

Perl

- *Practical Extraction and Report Language*
- Larry Wall, autor
  - *"There's more than one way to do it"*
  - *"Easy things should be easy and hard things should be possible"*.
  - 1987: početak razvoja
- prva linija Perl programa
  - put do perl interpretera
  - `#!/usr/bin/perl` ili `#!/usr/local/bin/perl`

Regularni izrazi u PERL-u

- sintaksa regularnih izraza zasniva se na Unix alatu **sed**
- koriste se za:
  - **pronalaženje izraza u ulaznome nizu / tekstu**

    `$red =~ /izraz/;`

    - operator `=~` testira da li je niz `/izraz/` pronađen
  - **zamjenu pronađenog izraza novim**

    `$red =~ s/izraz/zamjena/;`

    - potraži prvi izraz `/izraz/` i zamijeni ga izrazom `/zamjena/`

Primjeri Perl RI I

- pronađi riječi Ana ili Anika

```
$line =~ /An(a|ika)/;
```

- zamijeni velika slova malima (tr-translate)

```
$line =~ tr/A-Z/a-z/;
```

- zamijeni niz ab*c s ABC globalno

```
$line =~ s/ab*c/ABC/g;
```

  – /g u cijeloj datoteci, a ne samo prvog koji odgovara

- zamijeni sve konsonante znakom &

```
$line =~ tr/AEIOUaeiou/\&/cs;
```

  – c- complement, s- komprimira u jedan & znak

Primjeri Perl RI II

- ispiši cijenu u kunama i lipama

```
while($line =<>){
  while($line =~ m/\$ *([0-9]+)\.?([0-9])*/g {
     print “Kune:”, $1, “Lipe:”, $2 \n;
  }}
```

  - m/ - pregledava više linija na ulazu (ignorira znak za novi redak)
  - \n – newline -znak za novi redak
  - $1 i $2 su interne varijable kojima je privremeno pridružena vrijednost regularnog izraza u ( )
  - =<> - određuje standardni ulaz (vrijednost retka na ulazu je pridružena varijabli $line)

Primjeri Perl programa

```
#!/usr/bin/perl
print ("Molim postavi pitanje: \n");
  $upit = <STDIN>;
if ($upit =~ /[Mm]olim/) {
  print ("Hvala sto si pristojan!\n");
 } else {
  print ("Ovo nije bilo pristojno!\n"); }
```

Rad:

```
$ ./pristojnost.pl                 #poziv Perl prg.
Molim postavi pitanje:
Mogu li pitati molim?
Hvala sto si pristojan!
```

Kako provjeriti e-adresu?

- najjednostavnije
  - ima li znak @ (called: the **at sign**, **amphora**, **asperand**, or **at symbol**)
  - **r1+@r2+**
  - **^\S+@\S+$**
    - \S-svaki znak osim praznine NONBLANK
    - + ponovi se 1 ili više puta
    - ^ i $ početak i kraj retka
- ograničeni znakovi
  - **^[A-Z0-9+_.-]+@ [A-Z0-9.-]+$**
    - [A-Z0-9+_.-] sva slova brojke i znakovi _ . -

Kako provjeriti e-adresu? II

- svi znakovi
  - **^[\w!#$$&’*+/=?`{|}~^.-]+@ [A-Z0-9.-]+$**
  - svi dozvoljeni znakovi e-adrese prema RFC 2822
  - posebno paziti * | ` imaju dodatno značenje za SQL
  - najprije adresu pretvorili low

- bez početnih, završnih ili uzastopnih točaka
  - **^[\w!#$&’*+/=?`{|}~^]+(?:\. [!#$&’*+/=?`{|}~^-]+)*@ [A-Z0-9-]+(?:\.[A-Z0-9-] +)*$**
    - **?:\. točno jedna točka**

Kako provjeriti e-adresu? III

- glavna domena 2-6 znakova
  - **^[\w!#$&'*+/=?`{|}~^]+(?:\. [!#$&'*+/=?`{|}~^-]+)*@ (?:[A-Z0-9-]+\.)[A-Z] {2,6}$**

- **i zatim sve to CASE SENSITIVE**
  - ....

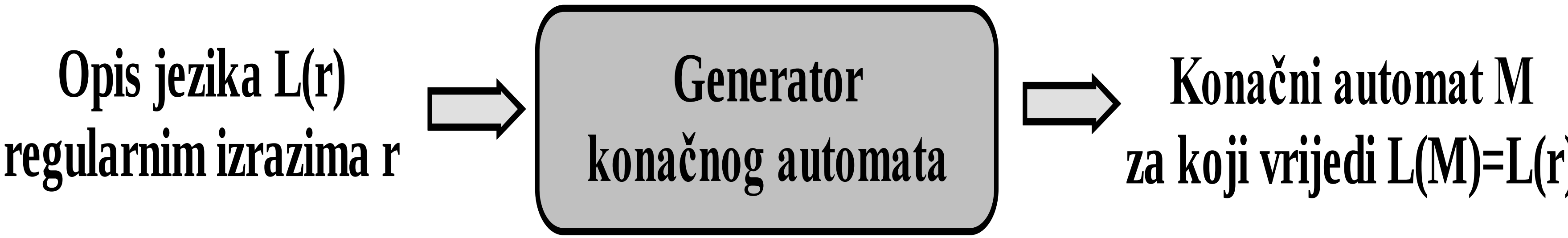
RI u praksi

- provjeravanje formata datuma
- provjeravanje formata IP adrese
- provjeravanje unosa “govoreće” šifre
- provjera formata računa ili katrice
- .....

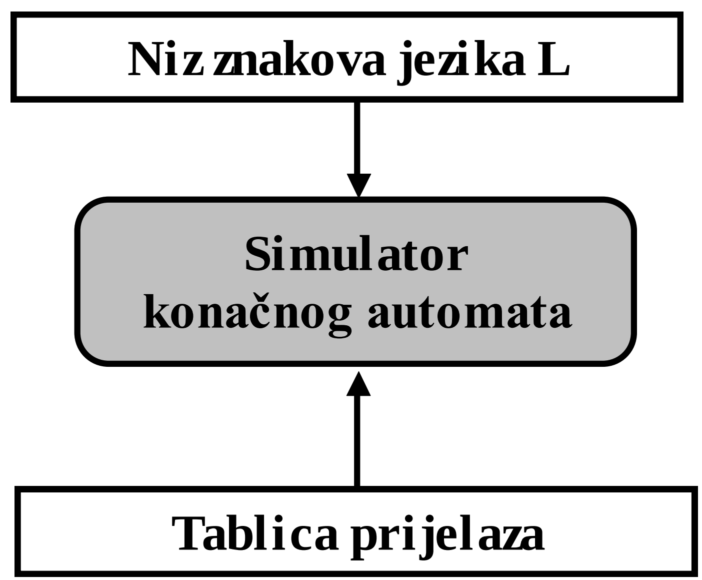
- praktična primjena Perl Formalni II

Generator konačnog automata

- KA se gradi za jezik zadan regularnim izrazima



- za automat se izgradi odgovarajući simulator
  - izravni način zapisa stanja
  - posredni način zapisa stanja



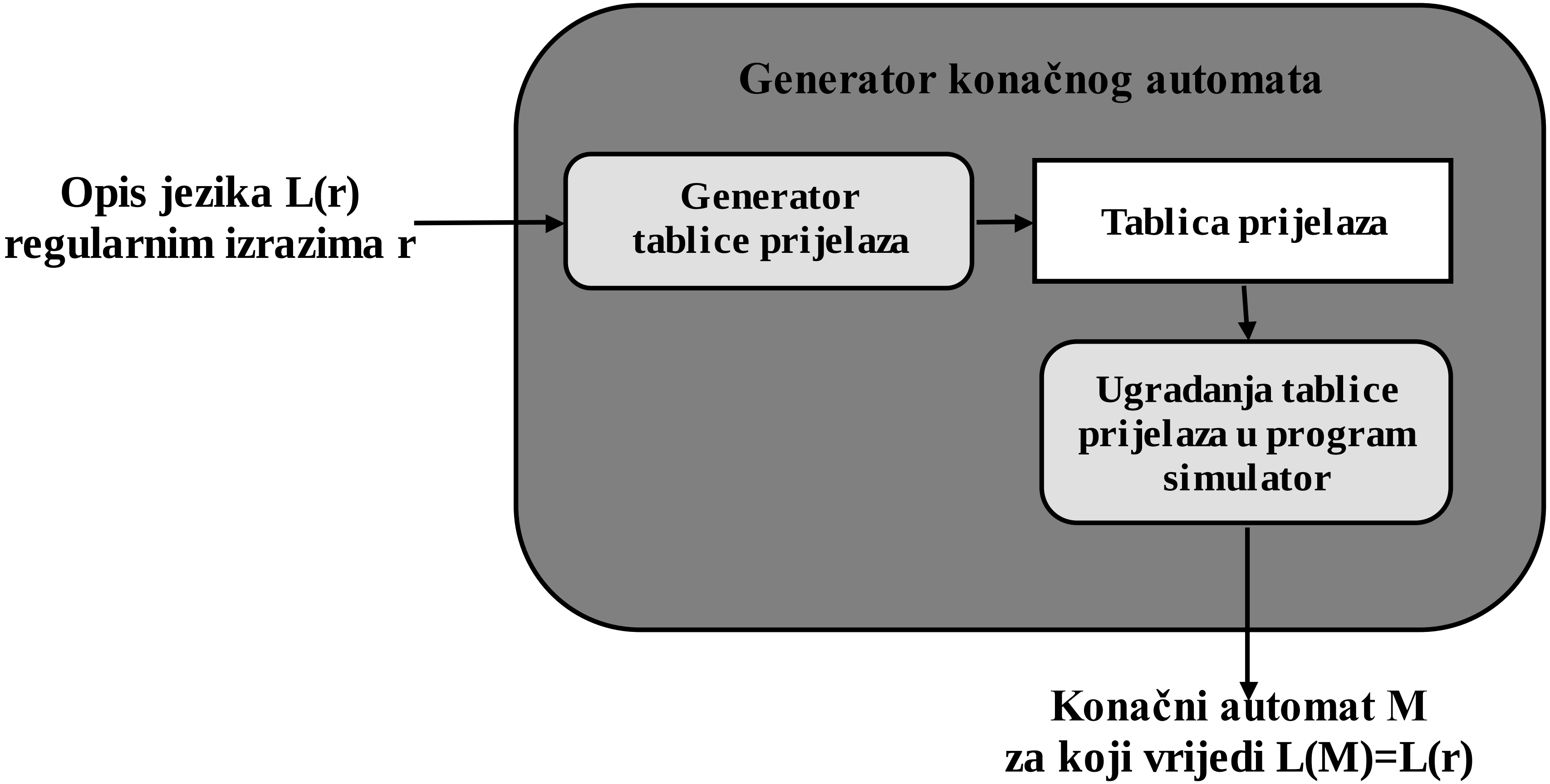
Izravni način zapisa stanja KA

```
Tablica [PP, 0]=NP;
Tablica [PP, 1]=PN;
Tablica [PP, ⊥]=1;
Tablica [NP, 0]=PP;
Tablica [NP, 1]=NN;
Tablica [NP, ⊥]=0;
Tablica [PN, 0]=NN;
Tablica [PN, 1]=PP;
Tablica [PN, ⊥]=0;
Tablica [NN, 0]=PN;
Tablica [NN, 1]=NP;
Tablica [NN, ⊥]=1;
Stanje =PP;
Pročitaj(Znak);
```

Neizravni način -tablica prijelaza

```
Dok(Znak != ⊥)
{
  Stanje = Tablica[Stanje,
  Znak];
  Pročitaj(Znak);
}
Ispiši(Tablica[Stanje,⊥],
  Stanje);
```

Struktura generatora konačnog automata



Svojstva regularnih jezika

Konačni automati (RI) : ograničenja

- jezik 0^n1^n $L=\{0^n1^n \mid n\geq 0\}$ nije regularan jezik automat koji ga prihvaća mora **pamtiti** koliko 0 smo pročitali da bi pročitali isti broj 1
- jezik $C=\{w \mid w$ ima jednak broj 0 i 1$\}$ **NIJE REGULARAN**
- jezik $D=\{w \mid w$ ima jednak broj podnizova 01 i 10$\}$

REGULARAN

- **Kako odrediti regularnost?**
  - Svojstva regularnih jezika
  - Dokazivanje regularnosti jezika

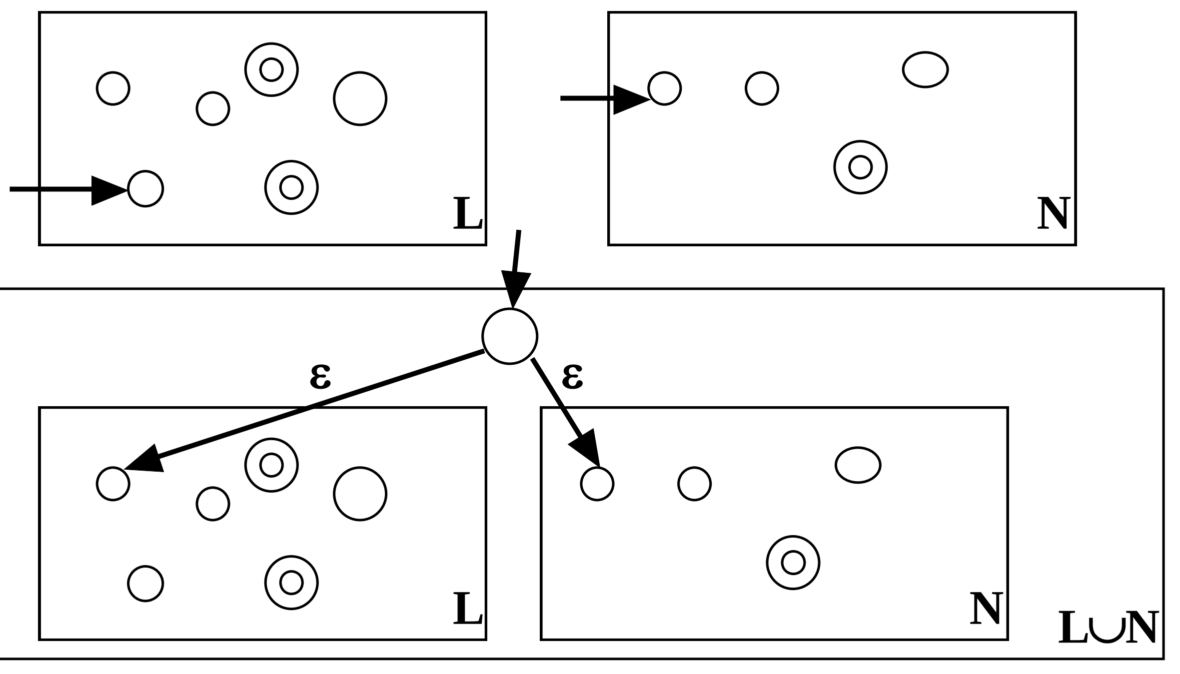
Svojstva zatvorenosti regularnih jezika

- zatvorenost klase jezika definira se s obzirom na pojedine operacije nad jezicima:
  - unije, nadovezivanja, Kleenovog operatora – zatvorenost slijedi iz neposredno iz definicije regularnih izraza
    - unija regularnih jezika L i N je regularni jezik za koji je moguće izgraditi DKA M takav da vrijedi $L(M) = L \cup N$
  - komplement, presjek, supstitucija
- *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

Zatvorenost s obzirom na uniju

- regularni jezici L i N
- L∪N je regularan ako je moguće izgraditi

DKA M takav da mu je jezik L(M)= L∪N



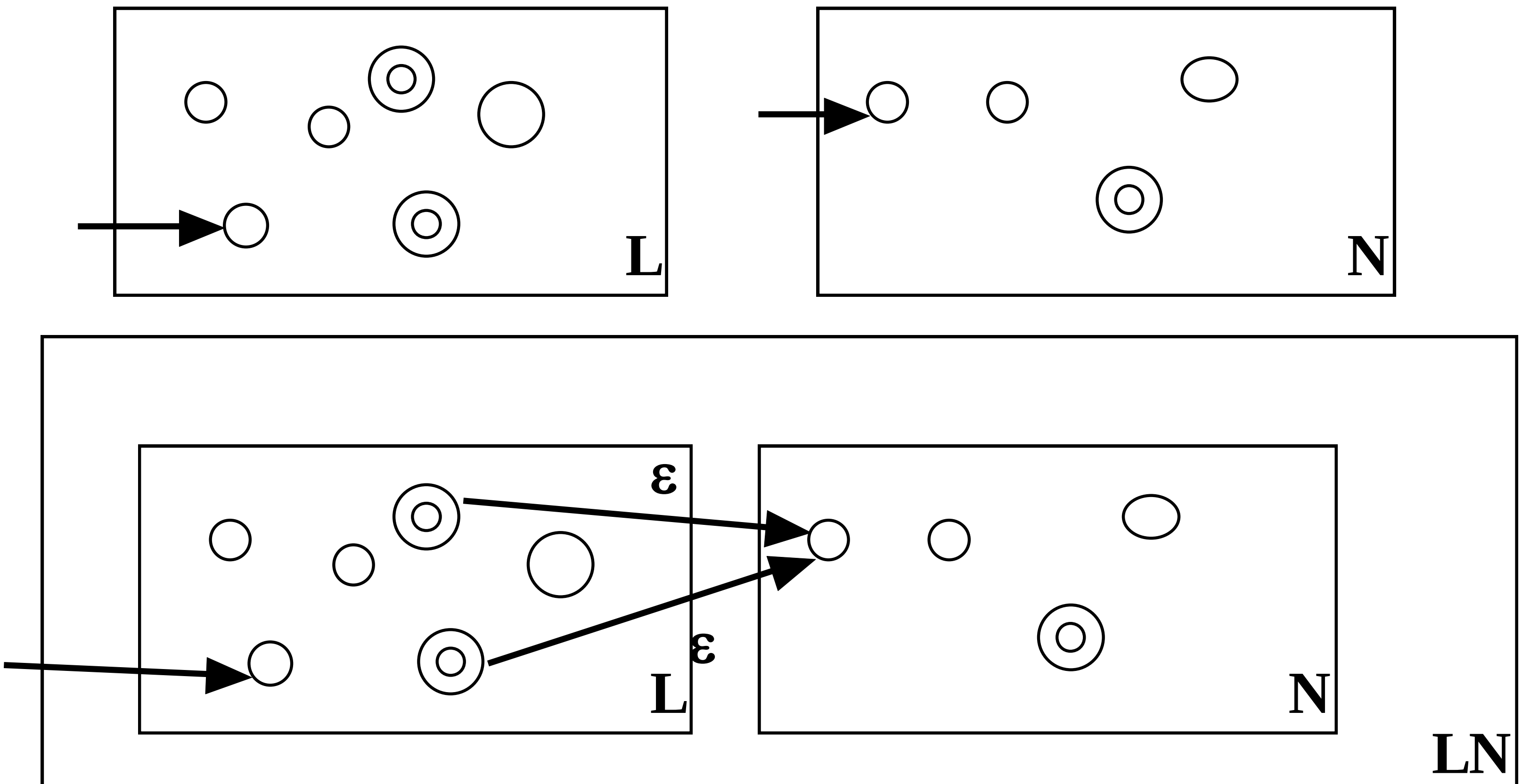
Zatvorenost s obzirom na uniju I

- $L=(Q_1, \Sigma, \delta_1, q_1, F_1)$ i $N=(Q_2, \Sigma, \delta_2, q_2, F_2)$
- za $L \cup N$ je $M=(Q, \Sigma, \delta, q_0, F)$
  - $\Sigma = \Sigma$
  - $Q = \{q_0\} \cup Q_1 \cup Q_2$
  - q_0 – novo početno stanje
  - $F = F_1 \cup F_2$

$$\delta(q,a) = \begin{cases} \delta_1(q,a), q \in Q_1 \\ \delta_2(q,a), q \in Q_2 \\ \{q_1, q_2\}, q = q_0 \;\text{ i }\; a = \varepsilon \\ \varnothing, \qquad q = q_0 \;\text{ i }\; a \neq \varepsilon \end{cases}$$

Zatvorenost s obzirom na nadovezivanje

- regularni jezici L i N: LN je regularan ako je moguće izgraditi DKA M takav da mu je jezik L(M)= LN



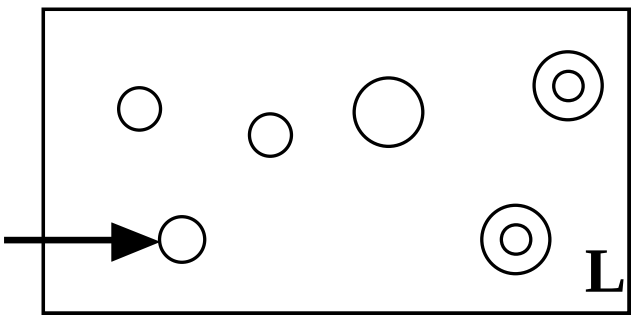
Zatvorenost s obzirom na nadovezivanje I

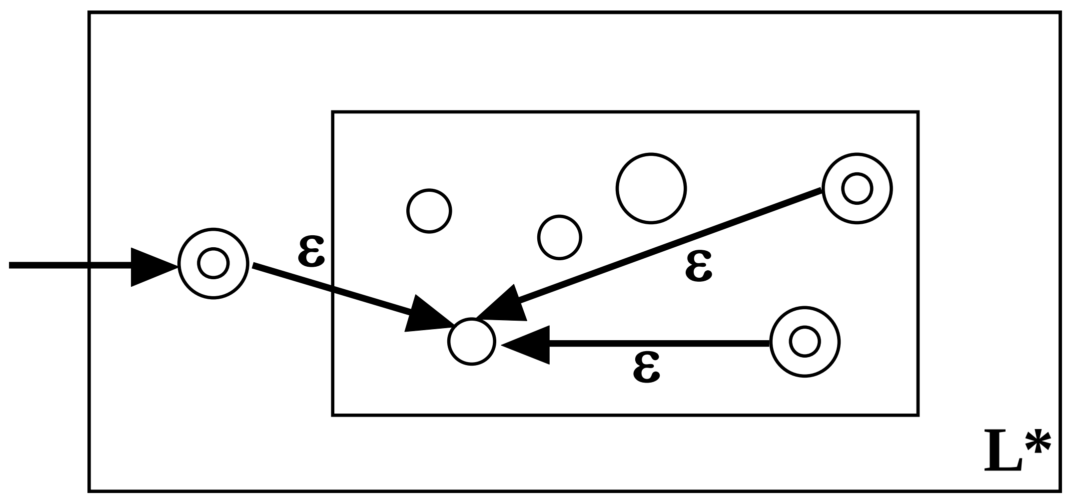
- $L=(Q_1, \Sigma, \delta_1, q_1, F_1)$ i $N=(Q_2, \Sigma, \delta_2, q_2, F_2)$
- za LN je $M=(Q, \Sigma, \delta, q_0, F)$
  - $\Sigma = \Sigma$
  - $Q = Q_1 \cup Q_2$
  - $q_0 = q_1$ početno stanje L
  - $F = F_2$

$$\delta(q,a) = \begin{cases} \delta_1(q,a), & q \in Q_1 \text{ i } q \notin F_1 \\ \delta_1(q,a), & q \in F_1 \text{ i } a \neq \varepsilon \\ \delta_1(q,a) \bigcup \{q_2\}, & q \in F_1 \text{ i } a = \varepsilon \\ \delta_2(q,a), & q \in Q_2 \end{cases}$$

Zatvorenost s obzirom na *

- regularni jezik L L* je regularan ako je moguće izgraditi DKA M takav da mu je jezik L(M)= L*
  - iz svih prihvatljivih stanja ε-prijelaz u početno stanje





Zatvorenost s obzirom na * II

- L1=(Q_1, Σ, δ_1, q_1, F_1) prihvaća jezik A onda L=(Q, Σ, δ, q_0, F) prihvaća jezik A*
- za LN je M=(Q, Σ, δ, q_0, F)
  - Σ= Σ
  - Q= $Q_1 \cup \{q0\}$
  - q_0 = početno stanje L
  - F= $F_1 \cup \{q0\}$

$$\delta(q,a)=\begin{cases}\delta_1(q,a), & q\in Q_1 \text{ i } q\notin F_1\\ \delta_1(q,a), & q\in F_1 \text{ i } a\neq\varepsilon\\ \delta_1(q,a)\bigcup\{q_1\}, & q\in F_1 \text{ i } a=\varepsilon\\ \{q_1\}, & q=q_0 \text{ i } a=\varepsilon\\ \varnothing, & q=q_0 \text{ i } a\neq\varepsilon\end{cases}$$

Zatvorenost s obzirom na komplement

- DKA $M=(Q, \Sigma, \delta, q_0, F)$; $L(M)$
- $L(M)^c$ – komplement jezika $L(M)$
- moguće je izgraditi DKA $M'=(Q, \Sigma, \delta, q_0, Q \setminus F)$ koji prihvaća $L(M')$
- $L(M') =\{w| \delta(q_0,w)\in(Q\setminus F)\}=\{w|\delta(q_0,w)\notin F\}=$

$= \Sigma^*\setminus\{w| \delta(q_0,w)\in F\}=\Sigma^*\setminus L(M)=L(M)^c$

- regularni jezici su zatvoreni s obzirom na operaciju komplementa

Zatvorenost s obzirom na presjek

- Dokaz (DeMorganovo pravilo):

$$L \cap N = ((L \cap N)^c)^c = (L^c \cup N^c)^c$$

- DKA $M_1=(Q_1, \Sigma_1, \delta_1, q_1, F_1)$ i DKA $M_2=(Q_2, \Sigma_2, \delta_2, p_1, F_2)$ onda je moguće izgraditi DKA $M=(Q, \Sigma, \delta, q_0, F)$ koji prihvaća $L(M)=L(M_1) \cap L(M_2)$:
  - $Q = Q_1 \times Q_2$, stanje $[q,p] \in Q$, $q \in Q_1$, $p \in Q_2$
  - $q_0 = [q_1, p_1]$
  - $F = F_1 \times F_2$, stanje $[q,p] \in F$, $q \in F_1$, $p \in F_2$
  - $\delta([q, p], a)=[\delta_1(q, a), \delta_2(p, a)]$, $\forall\, q \in Q_1$, $\forall\, p \in Q_2$ i $\forall\, a \in \Sigma$

Primjer: presjek

DKA M_1: $L(M_1)$={w|w je niz znakova a i b}

	a	b	c	
→q1	q1	q1	q2	1
q2	q2	q2	q2	0

DKA M_2:$L(M_2)$={w|w je niz znakova a i c}

	a	b	c	
→p1	p1	p2	p1	1
p2	p2	p2	p2	0

$L(M)=L(M_1) \cap L(M_2)$: L(M)={w|w je niz znakova a}

	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

Zatvorenost s obzirom na supstituciju

- $R \subseteq \Sigma^*$ je regularni jezik
- znaku $a \in \Sigma$ pridružimo $R_a \subseteq \Delta^*$, Δ - je abeceda
- ako niz $a_1, a_2, .. a_n$, $a_i \in \Sigma$ regularnog jezika R zamijenimo nizom $w_1, w_2, .. w_n$, $w_i \in R_{a1}$ gdje je w_i proizvoljni niz jezika R_{ai} onda je jezik f(R) regularan
- supstitucija omogućuje jednostavno zapisivanje regularnih izraza regularnim definicijama

Primjer

- R regularni jezika zadan izrazom: $r=0^*(0+1)1^*$
- supstitucija: $0 \leftrightarrow f(0)=a$ i $1 \leftrightarrow f(1)=b^*$ onda je
- $f(R) = f(0^*(0+1)1^*)=f(0)^*(f(0)+f(1))f(1)^*=$

$$=a^*(a+b^*)(b^*)^*= a^*(a+b^*)b^*=$$
$$=(a^*a+a^*b^*)b^*$$

- $b^*b^*=b^*,\ a^*a=a^+,\ a^++a^*=a^*$

$$(a^*a+a^*b^*)b^*=a^*ab^*+a^*b^*b^*=a^+b^*+a^*b^*=$$
$$= (a^++a^*)b^*=a^*b^*$$

Svojstvo napuhavanja

- engl. pumping lema
- DKA $M=(Q, \Sigma, \delta, q_0, F)$ ima n stanja
- ulazni niz $a_1,a_2,..a_m$; $m > n$;
- $\forall$ i= 1..m : neka je $\delta(q_0, a_1,a_2,..a_i)= q_i$;
- budući da DKA ima samo **n** različitih stanja, nije moguće da je **n+1** stanja u nizu stanja $q_0,q_1,..q_n$ različito
  - 0 ... n .. i .. m
  - pigeonhole principle
- s obzirom da je m > n **neka od stanja** u nizu $q_0,q_1,..q_n$ **se moraju ponoviti** (barem jedno stanje) odnosno

Pigeonhole principle

10 golubova

9 rupa

$\Rightarrow$

u jednoj rupi 2 goluba

The first statement of the principle is believed to have been made by Dirichlet in 1834 under the name *Schubfachprinzip* **("drawer principle" or "shelf principle")**. http://en.wikipedia.org/

Svojstvo napuhavanja I

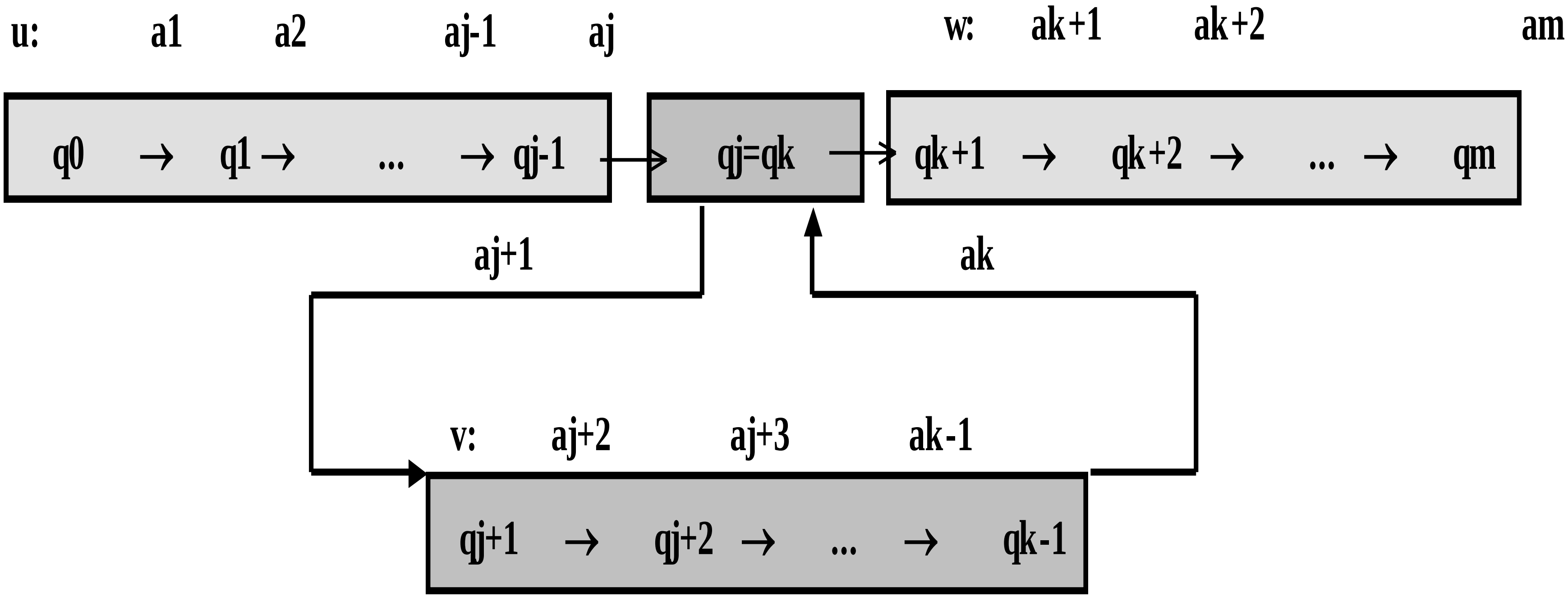
- za svaki ulazni niz $a_1,a_2,..a_m$ može se odrediti
- dva indeksa j i k: $0 \leq j<k \leq n$ i $q_j=q_k$:
  - $j<k$ i $k \leq n$ za duljinu niza $a_{j+1},a_{j+2},..a_k$ vrijedi $1 \leq | a_{j+1},a_{j+2},..a_k| \leq n$
- Uvodi se oznaka z
  - ako $a_1,a_2,..a_m = z = uvw$ (u,v,w, podnizovi)

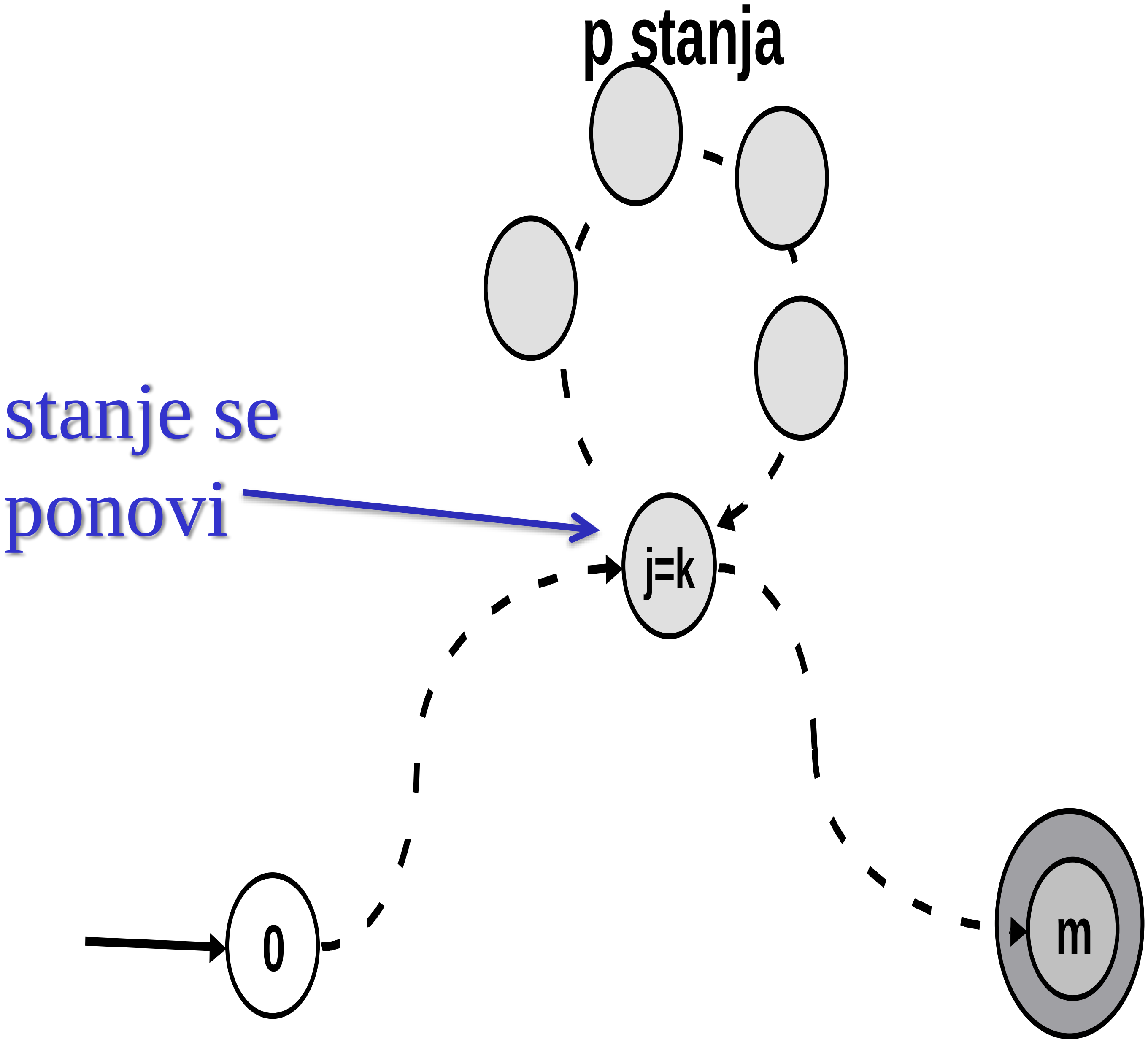
    $u= a_1,a_2,..a_j;$

    $v= a_{j+1},a_{j+2},..a_k;$

    $w= a_{k+1},a_{k+2},..a_m;$

Svojstvo napuhavanja II




p stanja
stanje se ponovi
j=k
0
m

Svojstvo napuhavanja III

- Dokaz prihvaćanja niza uvw
  - $q_{k=} q_{j;}$ $\delta(q_0, uw) = \delta(\delta(q_0, u), w) = \delta(q_j, w) = \delta(q_k, w) = q_m$
  - $q_{k=} q_{j;}$ $\delta(q_0, uvvw) = \delta(\delta(q_0, u), vvw) = \delta(q_j, vvw) = \delta(\delta(q_j, v), vw) = \delta(q_k, vw) = \delta(\delta(q_j, v), w) = \delta(q_k, w) = q_m$
- prihvaćaju se svi nizovi uv^iw; $i \geq 0$
- svaki dugački niz $z \in L(M)$ moguće je rastaviti na podnizove $z=uvw$
  - podniz v je moguće proizvoljan broj puta napuhati (ponoviti) te je $uv^iw \in L(M)$

Primjena svojstva napuhavanja

- dokazivanje neregularnosti jezika
  - ako jeziku ne možemo dokazati svojstvo napuhavanja onda je jezik neregularan
  - svojstvo napuhavanja kaže da se svi nizovi jezika mogu napuhati ako prelaze određenu duljinu napuhavanja
    - to znači da svaki niz sadrži podniz koji se može ponavljati bezbroj puta, a da rezultirajući niz bude u jeziku = regularan
  - dokazivanje ispravnosti algoritama kojima se utvrđuje nepraznost regularnog jezika, beskonačnost regularnog jezika, itd.

Neregularnost jezika

- nema odgovarajućeg **konačnog** automata
- ako je L – regularan jezik onda postoji cjelobrojna konstanta n: da je moguće bilo koji niz z iz L: $|z|>n$ rastaviti na podnizove $z=uvw$: $1 \le |v|$ i $|uv| \le n$ te za bilo koji $i \ge 0$ niz uv^iw je također element L

- pokazuje se da n nije veći od broja stanja minimalnog DKA koji prihvaća jezik L

Primjer 0^n1^n: neregularnost jezika

- za jezik $L=\{0^n1^n \mid n\geq 0\}$ pomoću napuhavanja pokažimo da nije regularan
- na početku pretpostavimo: da je regularan i iskoristimo svojstvo napuhavanja
  - duljina napuhvanja p: niz $s = 0^p1^p$, niz s ima duljinu veću od p, i možemo zapisati s=xyz i za $i\geq 0$ niz xy^iz je u jeziku L
  - ako se y sastoji samo od 0: niz xyyz ima više 0 od 1 i nije iz L: **kontradikcija**
  - ako se y sastoji samo od 1: niz xyyz ima više 1 od 0 i nije iz L: **kontradikcija**
  - ako se y sastoji samo od 0 i 1: niz xyyz može imati isti broj 0 i 1 ali opet nije iz L, jer je loš redosljed može se pojaviti 1 prije 0: **kontradikcija** **NIJE REGULARAN**

Primjer 0^{l^2} :neregularnog jezika

- Dokaz regularnosti jezika $K=\{0^{l^2} \mid l \text{ je cijeli broj i } l\geq 1\}$
  - nizovi 0 čija duljina je kvadrat cijelog broja, unarni jezik
- pretpostavka K je regularan
  - n cjelobrojna konstanta i z= 0^{n^2} niz jezika za koji vrijedi
    - $|z|=n^2$ i $|z|>n$.
  - niz z rastavimo $z=uvw$ $1\leq|v| \leq|uv| \leq n$; uv^iw je element jezika K za bilo koji cijeli broj i
  - ako se uzme i=2 i $|v| \leq|uv| \leq n$ onda je $|uvw|=|z|=n^2 \leq |uv^2w|=(n^2+|v|) \leq (n^2 +n)$
  - budući da je $(n^2+n) \leq (n +1)^2$ onda vrijedi
    - $n^2 \leq |uv^2w| \leq (n +1)^2$, odnosno uv^2w nije kvadrat niti jednog cijelog broja
  - bez obzira na veličinu n i duljinu niza z i bez obzira na podjele na podnizove, niz uv^2w nije element jezika K: **NIJE REGU**.

Primjer neregularnog jezika 0^{l^2} - 2. način

- ako je p duljina napuhavanja onda je niz s= 0^{p^2}
- s=xyz razbijemo u 3 podniza po svojstvu napuhavanja
  - za svaki i≥1 s=xy^iz niz S je u K
    - za p= 0,1,2,3,4,… dobivamo nizove duljine 0, 1, 4, 9, 16, 25, 36, 49, ... veliki prostori između kvadrata brojeva
- ako imamo dva niza xy^iz i $xy^{i+1}z$ koji se razlikuju samo za ponavljanje jednog y , i njihove duljine se razlikuju upravo za duljinu y
  - ako je **i** jako veliki broj onda y^i i y^{i+1} ne mogu biti kvadrati brojeva jer su si preblizu, znači nisu u K, kontradikcija i K nije regularan

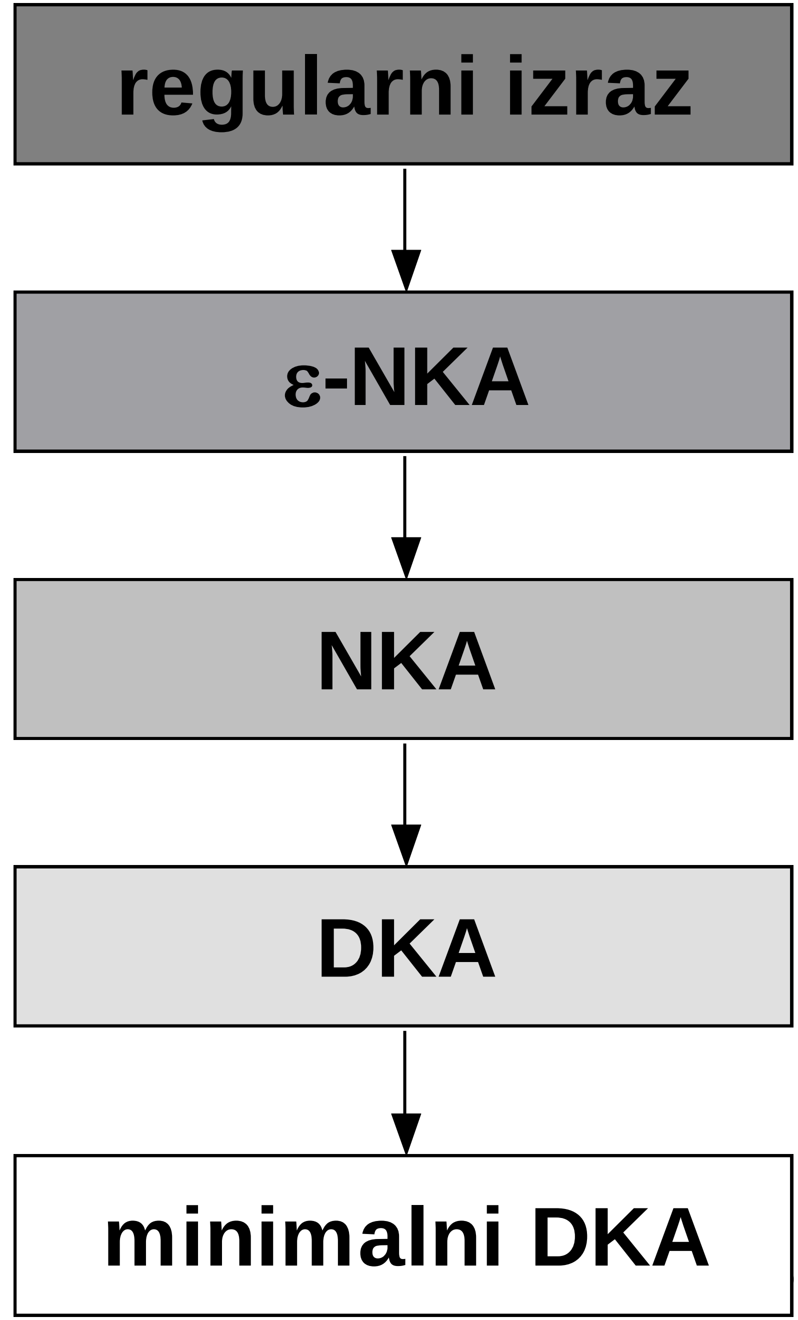
Primjer neregularnog jezika III

- to možemo pokazati kontradikcijom
- ako je $\mathbf{m = n^2}$ (savršen kvadrat broja) ondaako računamo razliku dvaju susjednih $(n+1)^2 - n^2$
  - $(n+1)^2-n^2=n^2+2n+1-n^2=2n+1=2\sqrt{m}+1$
- ako koristimo svojstvo napuhavanja $\mathbf{m = |xy^iz|}$
  - $|y| < 2\sqrt{|xy^iz|}+1$
  - onda izračunamo i:
    - $|y| \leq |s| = p^2$; ako je $i = p^4$ onda
    - $|y| \leq p^2 = \sqrt{p^4} < 2\sqrt{p^4}+1 \leq 2\sqrt{|xy^iz|}+1$

Nepraznost i beskonačnost regularnog jezika

- **Nepraznost:**
  - Regularni jezik L(M) je neprazan ako i samo ako DKA M s n stanja prihvaća niz **z** duljine manje od **n** ($|z|<n$)
  - ako je u skupu dosegljivih stanja M barem jedno prihvatljivo stanje L(M) je neprazan
- **Beskonačnost**:
  - Regularni jezik L(M) je beskonačan ako i samo ako M prihvaća niz duljine l, gdje je $n \leq l < 2n$
  - ako je u dijagramu stanja M (bez neprihvatljivih stanja) barem jedna zatvorena petlja L(M) je beskonačan

Postupak



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.
- J. Goyvaerts, S. Levithan, Regular Expressions Cookbook, O'Riley, 2009.

Regularne definicije

- r_i su regularni izrazi nad abecedom $\Sigma \cup \{d_1, d_2, .. d_{i-1}\}$, a d_i su znakovi
- regularne definicije su oblika:
  - $d_1 \rightarrow r_{1,}\ d_2 \rightarrow r_2\ ,..\ d_n \rightarrow r_n$
- abecedu regularnog izraza čine znakovi skupa Σ i znakovi $d_1, d_2, .. d_{i-1}$ koji su prethodno definirani regularnim izrazima
- regularni izraz rj se definira nad abecedom Σ tako da se svi znakovi $d_1, d_2, .. d_{j-1}$ zamijene regularnim izrazima

Primjer I

Programske varijable definiraju se regularnim izrazima nad abecedom Σ={A,B,..,Z,a.b,..,z,0,1,..9}

- slovo$\rightarrow$A+B+C+...+Z+a+b+c+..+z
- brojka $\rightarrow$0+1+2+..+9
- varijabla $\rightarrow$slovo(slovo+brojka)*

- uvrštavanjem

varijabla $\rightarrow$ (A+B+C+...+Z+a+b+c+..+z)((A+B+C+...+Z+a+b+c+..+z)+(0+1+2+..+9))*

Primjer II

- Konstante bez predznaka zadane su regularnim definicijama nad abecedom Σ={0,1,..9, E, .,+,-} (ili |=+)
  - broj → 0|1|2|3|4|5|6|7|8|9
  - brojke → broj broj*
  - decimale → .brojke|ε
  - eksponent → (E(+|-|ε)brojke)|ε
  - konstanta → brojke decimale eksponent