

# Programski prevodioci (vežbe)

---

Formalne gramatike  
Konačni automati

# Formalne gramatike

Formalna gramatika je uređena četvorka:

$$G=(V_N, V_T, S, P)$$

gde je:

$V_N$  - skup neterminalnih simbola,

$V_T$  - skup terminalnih simbola

$S$  - startni simbol gramatike

$P$  - skup smena.

$V = V_N \cup V_T$  - azbuka jezika,

$V_N \cap V_T = \emptyset$ .

$P: X \rightarrow Y, X \in V^* V_N V^*, Y \in V^*$

---

# Pravila za definisanje formalnih gramatika

- Gramatika mora da bude redukovana.
    - Redukovana gramatika je gramatika koja ne sadrži beskorisne smene.
    - Beskorisne smene su one koje sadrže simbole koji ne mogu da budu izvedeni iz startnog simbola gramatike, ili smene koje sadrže “beskonačne” simbole.
  - Gramatika ne sme da bude nejednoznačna.
    - Kod nejednoznačnih gramatika za jedan ulazni niz je moguće kreirati veci broj sintaksnih stabala.
-

# Primer neredukovane gramatike

$G = (\{W, X, Y, Z\}, \{a\}, W, P)$

P:  $W \rightarrow aW$  (1)

$W \rightarrow Z$  (2)

Neupotrebljiva jer je  $Z$   
beskonačan simbol.

$W \rightarrow X$  (3)

$Z \rightarrow aZ$  (4)

Neupotrebljiva iz istog razloga.

$X \rightarrow a$  (5)

$Y \rightarrow aa$  (6)

Neupotrebljiva jer ne postoji  
izvođenje  $W \Rightarrow \alpha Y \beta$ .

# Primer nejednoznačne gramatike

$G = (\{lraz\}, \{\mathbf{const}, +, *\}, lraz, P)$

P:  $lraz \rightarrow lraz + lraz$

$lraz \rightarrow lraz * lraz$

$lraz \rightarrow \mathbf{const}$

Kreirati moguća sintaksna stabla izraza

**$const * const + const$**

---

# Ekvivalentna jednoznačna gramatika

$G = (\{Izraz, Proizvod, Faktor\}, \{\mathbf{const}, +, *\}, Izraz, P)$

P:

- $Izraz \rightarrow Izraz + Proizvod$
- $Izraz \rightarrow Proizvod$
- $Proizvod \rightarrow Proizvod * Faktor$
- $Proizvod \rightarrow Faktor$
- $Faktor \rightarrow \mathbf{const}$

Kreirati sintaksno stablo izraza koji je naveden u prethodnom primeru.

---

# Tipovi gramatika

- Gramatike tipa 0
  - Bez ograničenja
- Gramatike tipa 1
  - Dužina reči koja se nalazi na levoj strani smene  $\leq$  od dužine reči koja se nalazi na desnoj strani smene
- Gramatike tipa 2
  - Na levoj strani smene nalazi se samo jedan neterminalni simbol (beskonteksne gramatike)
- Gramatike tipa 3
  - Smene su oblika:  $A \rightarrow a$ ,  $A \rightarrow aB$  ili  $A \rightarrow \varepsilon$  (gde  $A \in V_N$ ,  $a \in V_T$ , a  $\varepsilon$  je prazan simbol)

# Zadatak

- Kom tipu pripadaju i koji jezik definišu gramatike zadate sledećim skupom smena:

G1:      (1)       $I \rightarrow a C a$                       (4)       $a B \rightarrow B a$   
             (2)       $C \rightarrow a C B a$                       (5)       $b B \rightarrow b b$   
             (3)       $C \rightarrow b$

G2:      (1)       $S \rightarrow S a$                       (3)       $A \rightarrow a A b$   
             (2)       $S \rightarrow a A$                       (4)       $A \rightarrow a b$

G3:      (1)       $S \rightarrow a S$                       (4)       $C \rightarrow a C$   
             (2)       $S \rightarrow a B$                       (5)       $C \rightarrow a$   
             (3)       $B \rightarrow b C$



---

# Mašine za prepoznavanje jezika

- Tjuringova mašina
    - Prepoznaje reči jezika definisanih gramatikama tipa 0
  - Linearno-ograničeni automati
    - Prepoznaju reči jezika definisanih gramatikama tipa 1
  - Magacinski automati
    - Prepoznaju reči jezika definisanih gramatikama tipa 2
  - Konačni automati
    - Prepoznaju reči jezika definisanih gramatikama tipa 3
-

# Načini definisanja konačnih automata

- Formalni opis uređenom petorkom:

$$M=(V,Q,q_0,F,g)$$

gde je:

$V$  – ulazna azbuka,

$Q$  – skup stanja,

$q_0$  – početno stanje,

$F$  – skup završnih stanja,

$g$  – skup preslikavanja ( $g: Q \times V \rightarrow Q$ )

# Primer konačnog automata

## ■ Primer1:

$M = (\{0, 1\}, \{A, B, C, D, F, G\}, A, \{C, F, G\}, g)$

Gde je g:

$g(A, 0) = B$

$g(A, 1) = D$

$g(B, 0) = C$

$g(B, 1) = F$

$g(C, 0) = C$

$g(C, 1) = F$

$g(D, 0) = F$

$g(D, 1) = E$

$g(E, 0) = F$

$g(E, 1) = E$

$g(F, 0) = G$

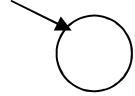

$g(F, 1) = G$

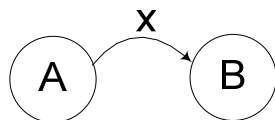
$g(G, 0) = G$

$g(G, 1) = G$

# Načini definisanja konačnih automata

## ■ Grafom prelaza:

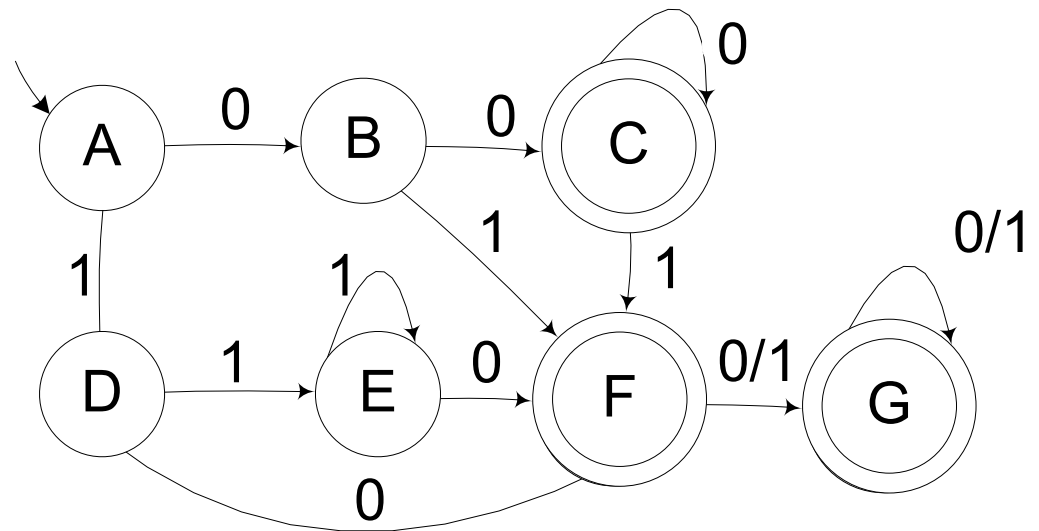
- ❑ Ulazna azbuka  $V$  - skup simbola kojima su označeni potezi u grafu.
- ❑ Skup stanja  $Q$  – čvorovi u grafu.
- ❑ Početno stanje – ulazni čvor (obeležen ulaznom granom )
- ❑ Skup završnih stanja – završni čvorovi obeleženi sa 
- ❑ Skup preslikavanja konačnog automata definiše se potezima u grafu. Preslikavanju  $g(A,x)=B$  odgovara potez



# Konačni automat iz primera 1 predstavljen grafom prelaza

$M = (\{0, 1\}, \{A, B, C, D, E, F, G\}, A, \{C, F, G\}, g)$

$g(A, 0) = B$	$g(E, 0) = F$
$g(A, 1) = D$	$g(E, 1) = E$
$g(B, 0) = C$	$g(F, 0) = G$
$g(B, 1) = F$	$g(F, 1) = G$
$g(C, 0) = C$	$g(G, 0) = G$
$g(C, 1) = F$	$g(G, 1) = G$
$g(D, 0) = F$	
$g(D, 1) = E$	



# Načini definisanja konačnih automata


## ■ Tablicom prelaza:

- Ulazna azbuka  $V$  – oznake kolona u tabeli prelaza.
- Skup stanja  $Q$  – oznake vrsta u tabeli prelaza.
- Početno stanje – oznaka vrste koja je obeležena strelicom ( $\rightarrow$ ) ili oznaka prve vrste ukoliko vrsta koja odgovara startnom simbolu nije posebno obeležena.
- Skup završnih stanja – U tabeli prelaza se dodaje posebna kolona u koju se upisuje identifikator prihvatanja-odbijanja. U toj koloni upisane su jedinice u vrstama koje odgovaraju završnim stanjima, a nule u vrstama koje odgovaraju stanjima koja nisu završna.
- Elementi tabele definišu preslikavanja automata.

# Konačni automat iz primera 1 predstavljen tablicom prelaza

$M = (\{0, 1\}, \{A, B, C, D, E, F, G\}, A, \{C, F, G\}, g)$

$g(A, 0) = B$        $g(E, 0) = F$   
 $g(A, 1) = D$        $g(E, 1) = E$   
 $g(B, 0) = C$        $g(F, 0) = G$   
 $g(B, 1) = F$        $g(F, 1) = G$   
 $g(C, 0) = C$        $g(G, 0) = G$   
 $g(C, 1) = F$        $g(G, 1) = G$   
 $g(D, 0) = F$   
 $g(D, 1) = E$



	0	1	prih/odb
A	B	D	0
B	C	F	0
C	C	F	1
D	F	E	0
E	F	E	0
F	G	G	1
G	G	G	1

---

# Deterministički i nedeterministički konačni automati

- Prethodna definicija konačnog automata je, u stvari definicija determinističkog konačnog automata.
- Nedeterministički konačni automati definisani su istom uređenom petorkom kao i deterministički konačni automati, jedina razlika je u tome što su preslikavanja koja vrše nedeterministički konačni automati oblika:

$$g: Q \times V \rightarrow S, \text{ gde je } S \subset Q.$$

Dakle, nedeterministički konačni automat pod dejstvom istog ulaznog simbola može da pređe u više različitih stanja.

---



---

# Algoritam za prepoznavanje reči korišćenjem deterministiških automata

```
DKA(  $t_0 t_1 \dots t_{n-1}$  )
{
    tekuce_stanje =  $q_0$ ;    //  $q_0$  - pocetno stanje
    tekuci_ulaz =  $t_0$ ;
    while ( !(kraj_ulazne_reci) )
    {
        tekuce_stanje = g(tekuce_stanje, tekuci_ulaz );
        tekuci_ulaz = sledeci_znak_ulazne_sekvence;
    }
    if ( tekuce_stanje  $\in F$  )    //  $F$  - skup završnih stanja
        ulazna_rec_je_prepoznata;
    else
        ulazna_rec_nije_prepoznata;
}
```

---

# Algoritam za prepoznavanje reči korišćenjem nedeterminističkih automata

```
NDKA(  $t_0 t_1 \dots t_{n-1}$  )
{
    tekuci_skup_stanja =  $\{q_0\}$ ;
    tekuci_ulaz =  $t_0$ ;
    while ( !(kraj_ulazne_reci) )
    {
        novi_skup_stanja =  $\emptyset$ ;
        foreach(  $q$  in tekuci_skup_stanja )
            novi_skup_stanja = novi_skup_stanja  $\cup$   $g(q, tekuci\_ulaz)$ ;
        tekuci_skup_stanja = novi_skup_stanja;
        tekuci_ulaz = sledeci_znak_ulazne_sekvence;
    }
    if ( tekuci_skup_stanje  $\cap F \neq \emptyset$  )
        ulazna_rec_je_prepoznata;
    else
        ulazna_rec_nije_prepoznata;
}
```

---

# Konačni automati i regularni izrazi

## ■ Zadatak:

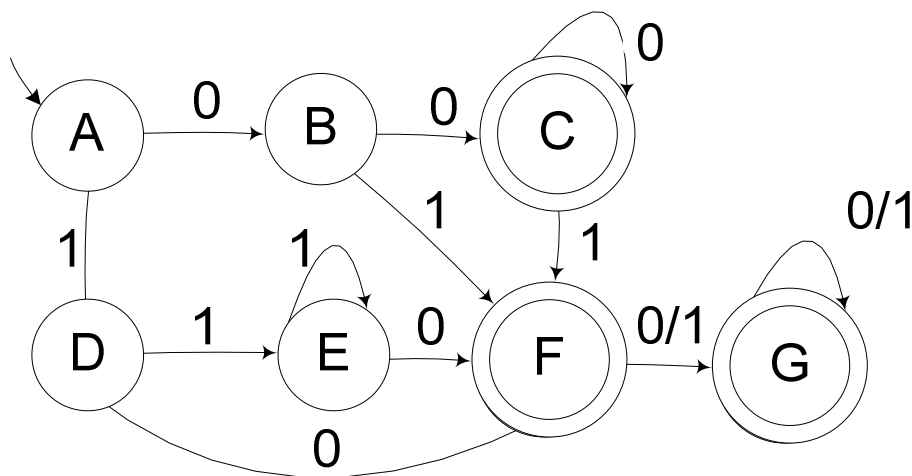
Odrediti regularni izraz jezika koji se prepoznaje konačnim automatom iz primera 1.

### Korak1: Definisanje jednačina stanja

Za svako stanje u konačnom automatu piše se jednačina stanja koja definiše načine na koje se može doći do posmatranog stanja.

---

# Jednačine stanja konačnog automata iz primera 1.



1)  $A = A\lambda$

2)  $B = A0$

3)  $C = B0 + C0$

4)  $D = A1$

5)  $E = D1 + E1$

6)  $F = B1 + C1 + D0 + E0$

7)  $G = F0 + F1 + G0 + G1$

# Izvođenje regularnog izraza jezika na osnovu jednačina stanja

## Korak2: Izvođenje regularnog izraza

Regularni izraz jezika definiše opšti oblik skupa reči koje konačni automat prevode iz početnog u neko od završnih stanja. Definirati regularni izraz znači odrediti funkcionalnu zavisnost izlaznih stanja i ulazne grane u automat. Tj. izraziti svako završno stanje  $q_j$  u obliku  $q_j = f(q_0 \lambda)$ .

Pravila za izvođenje regularnih izraza:

- Jednačina oblika  $q_i = q_j a + q_j b$  se transformiše u  $q_i = q_j (a + b)$
- Jednačina oblika  $q_i = q_i a + q_j b$  se transformiše u  $q_i = q_j b a^*$

## Izvođenje regularnog jezika koji se prepoznaje konačnim automatom iz primera 1.

- 1)  $A = A\lambda$
- 2)  $B = A0$
- 3)  $C = B0 + C0$
- 4)  $D = A1$
- 5)  $E = D1 + E1$
- 6)  $F = B1 + C1 + D0 + E0$
- 7)  $G = F0 + F1 + G0 + G1$

Iz 1) i 4) dobijamo:

$$10) D = A\lambda 1$$

Iz 5) i 10):

$$E = A\lambda 11 + E1$$

tj.

$$11) E = A\lambda 111^*$$

---

Iz jednačina 1) i 2) sledi:

$$8) B = A\lambda 0$$

Iz 3) i 8):

$$C = A\lambda 00 + C0$$

tj.

$$9) C = A\lambda 000^*$$

Zamenom 8), 9), 10) i 11) u 6) dobijamo:

$$F = A\lambda 01 + A\lambda 000^*1 + A\lambda 10 + A\lambda 111^*0$$

tj.

$$12) F = A\lambda (01 + 000^*1 + 10 + 111^*0)$$

---

---

Izvođenje regularnog jezika koji se prepoznaje konačnim automatom iz primera 1.

Iz 7) sledi:

$$13) \quad G = F(0+1) + G(0+1) = F(0+1)(0+1)^*$$

Zamenom 12) u 13) dobijamo:

$$14) \quad G = A\lambda(01 + 000^*1 + 10 + 111^*0)(0+1)(0+1)^*$$

Konačno, jezik koji je definisan gramatikom je unija jezika koje prepoznaju stanja C, F i G:

$$L = 000^* + (01 + 000^*1 + 10 + 111^*0) + (01 + 000^*1 + 10 + 111^*0)(0+1)(0+1)^*$$

---

# Veza konačnih automata i gramatika tipa 3

- ❑ Ulazna azbuka odgovara skupu terminalnih simbola gramatike.
- ❑ Skup stanja odgovara skupu neterminalnih simbola gramatike.
- ❑ Početno stanje je startni simbol gramatike.
- ❑ Preslikavanje oblika  $g(q_i, a) = q_j$  odgovara smeni gramatike  $q_i \rightarrow aq_j$ .
- ❑ Završna stanja su oni neterminalni simboli za koje postoji smena oblika  $q_i \rightarrow \varepsilon$ .



---

Definisanje konačnog automata za prepoznavanje reči jezika definisanog gramatikom tipa 3.

■ Zadatak:

Definisati konačni automat za prepoznavanje označenih celih brojeva. Označeni ceo broj definisan je sledećom gramatikom:

$\langle \text{OznaceniCeoBroj} \rangle ::= + \langle \text{NeoznaceniCeoBroj} \rangle$

$\langle \text{OznaceniCeoBroj} \rangle ::= - \langle \text{NeoznaceniCeoBroj} \rangle$

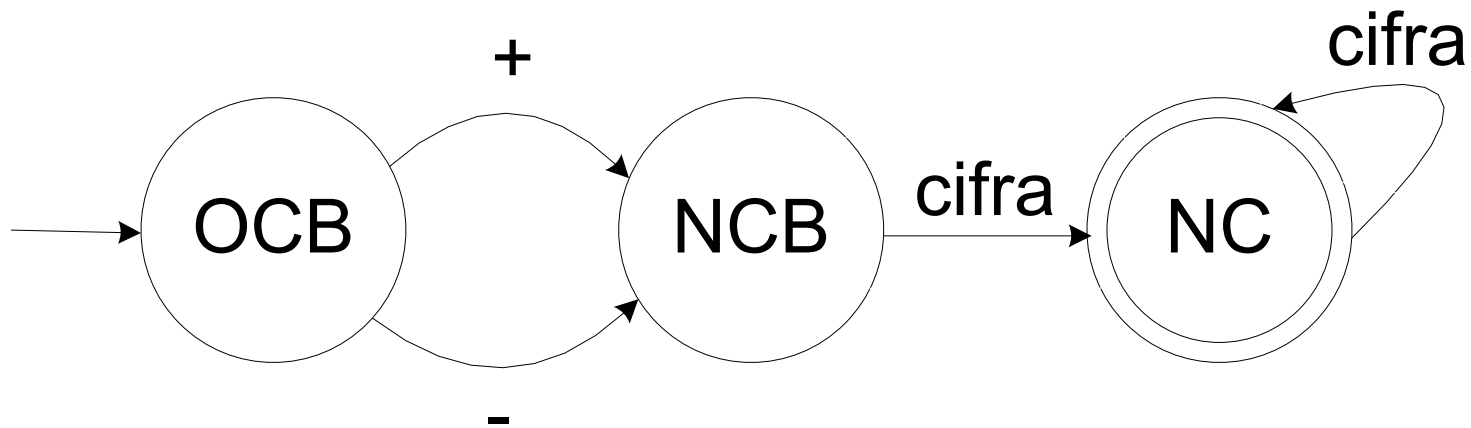
$\langle \text{NeoznaceniCeoBroj} \rangle ::= \mathbf{cifra} \langle \text{NizCifara} \rangle$

$\langle \text{NizCifara} \rangle ::= \mathbf{cifra} \langle \text{NizCifara} \rangle$

$\langle \text{NizCifara} \rangle ::= \varepsilon$

---

Definisanje konačnog automata za prepoznavanje reči jezika definisanog gramatikom tipa 3.



# Definisanje konačnog automata za prepoznavanje reči jezika definisanog regularnim izrazom

## ■ Zadatak:

Definisati konačni automat za prepoznavanje označenih realnih brojeva u eksponencijalnom zapisu. Eksponencijalni zapis realnog broja definisan je sledećim regularnim izrazom:

$(+ \mid -) \text{cifra} \text{cifra}^* \cdot \text{cifra}^* (E \mid e) (+ \mid -) \text{cifra} \text{cifra}^*$

