PROGRAMSKI PREVODIOCI

- Uvod -

O predmetu

- Nedeljni fond časova:
  2+2+1

- Nastavnik:
  Suzana Stojković, kancelarija 523,
  e-mail adresa: suzana.stojkovic@elfak.ni.ac.rs

- Asistenti:
  Ivica Marković, kancelarija 534,
  e-mail adresa: ivica.markovic@elfak.ni.ac.rs

  Nenad Petrović, kancelarija 331,
  e-mail adresa: nenad.petrovic@elfak.ni.ac.rs

  Teodora Đorđević, kancelarija 533
  e-mail adresa: teodora.djordjevic@elfak.ni.ac.rs

Sadržaj predmeta

- Predavanja
  - Uvod: vrste programskih pevodilaca, struktura kompilatora
  - Formalna jezici i formalne gramatike
  - Uredjaji za prepoznavanje jezika
  - **Komponenete kompilatora**

  - Vežbe:
    - Formalna definicija jezika i uredjaji za prepoznavanje jezika
    - Implementacija komponenti kompilatora
    - **Alati za automatkso generisanje leksičkih i sintaksnih analizatora**

Način ocenjivanja

- Laboratorijske vežbe: 20 poena
- I kolokvijium: 20 poena
- II kolokvijum: 20 poena

Pismeni deo ispita: 40 poena

- Aktivnost: 10 poena (On-line testovi u sklopu predavanja)
- Završni ispit: 30 poena

Literatura

- M. Stanković, S. Stojković, Ž. Tošić, *Programski prevodioci,* Univerzitet u Nišu, 2018.

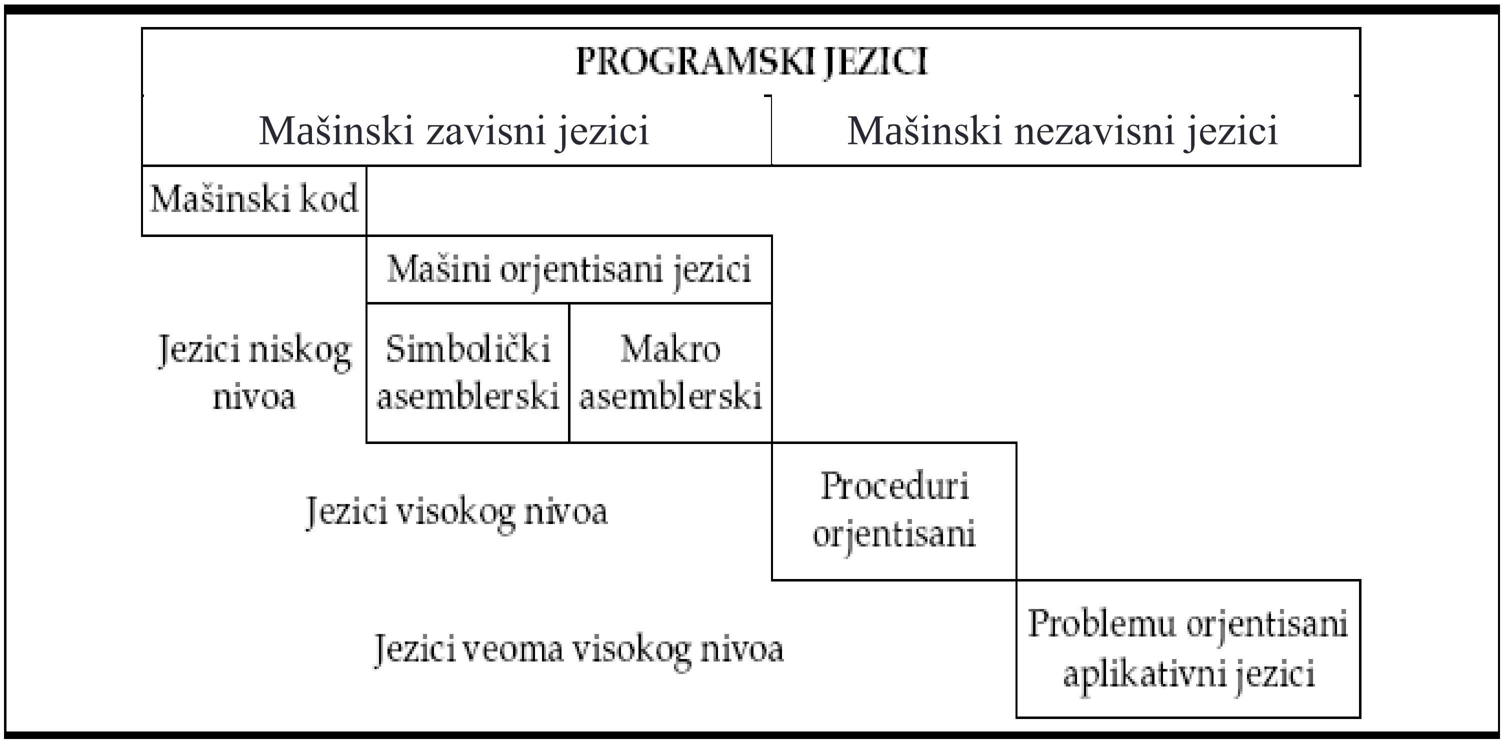
- A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman, *Compilers, Principles, Techniques, and Tool,* Second edition, Addison-Wesley, 2007.

Programski prevodioci

- Prevođenje – pretakanje teksta pisanog jednim jezikom u tekst pisan drugim jezikom.

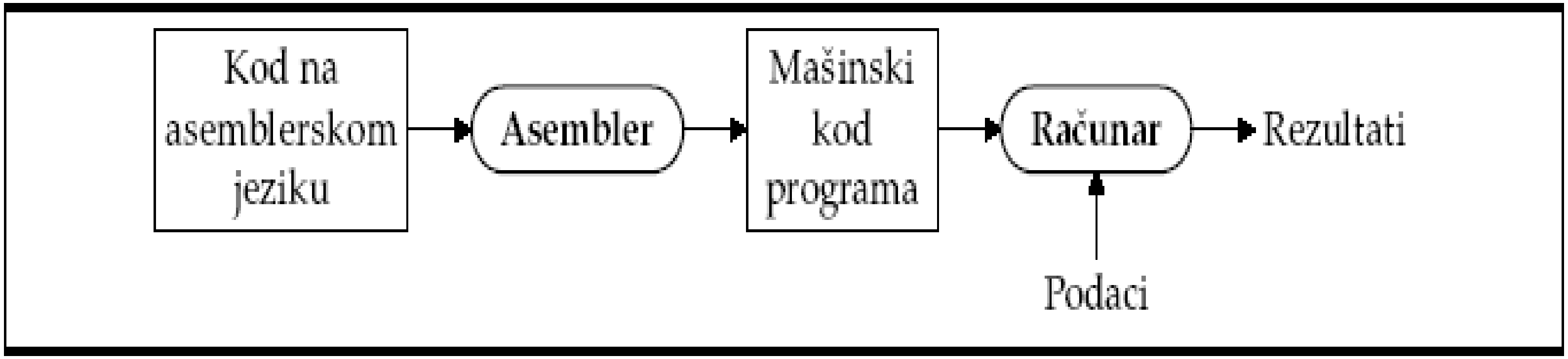
- Programski prevodilac – softver koji prevodi programe pisane jednim programskim jezikom u programe pisane drugim programskim jezikom
  - Uobičajeno: prevodi programe pisane jezikom višeg nivoa u programe pisane jezikom nižeg nivoa.

Klasifikacija programskih jezika



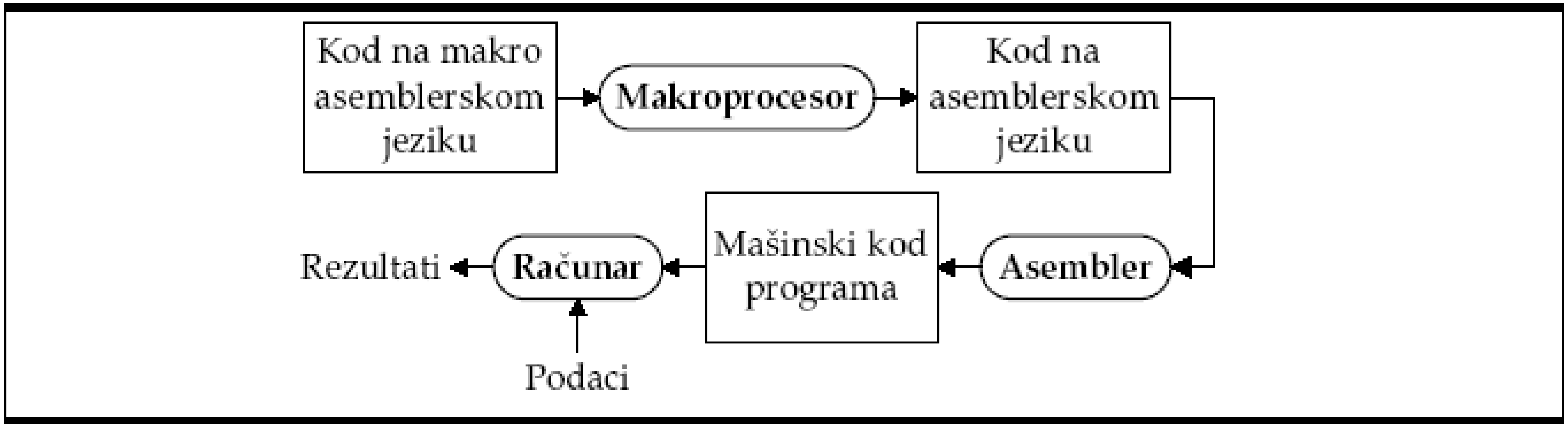
Slika 1.1 Podela programskih jezika

Asembleri



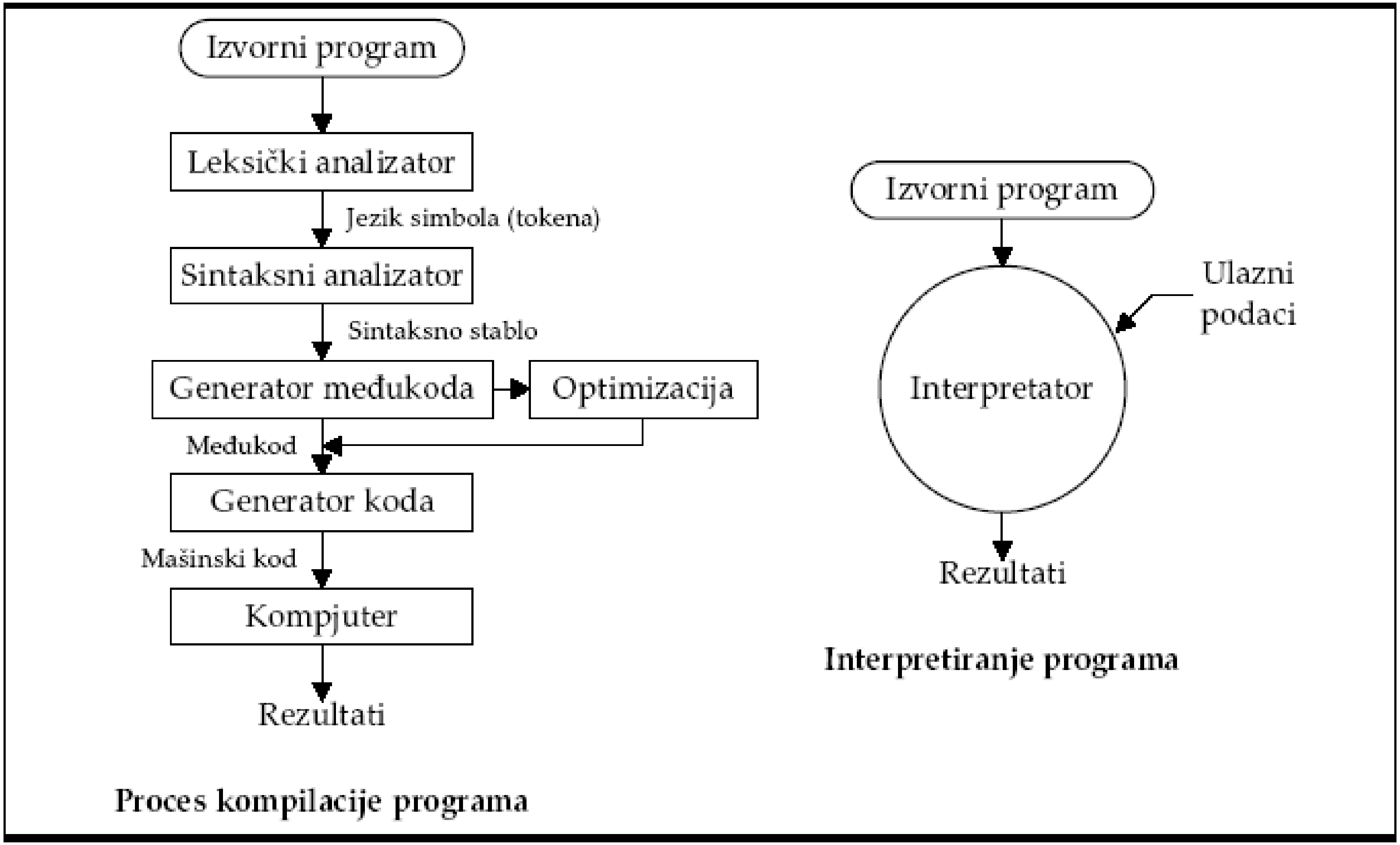
Slika 1.2 Asembleri

Makroprocesori



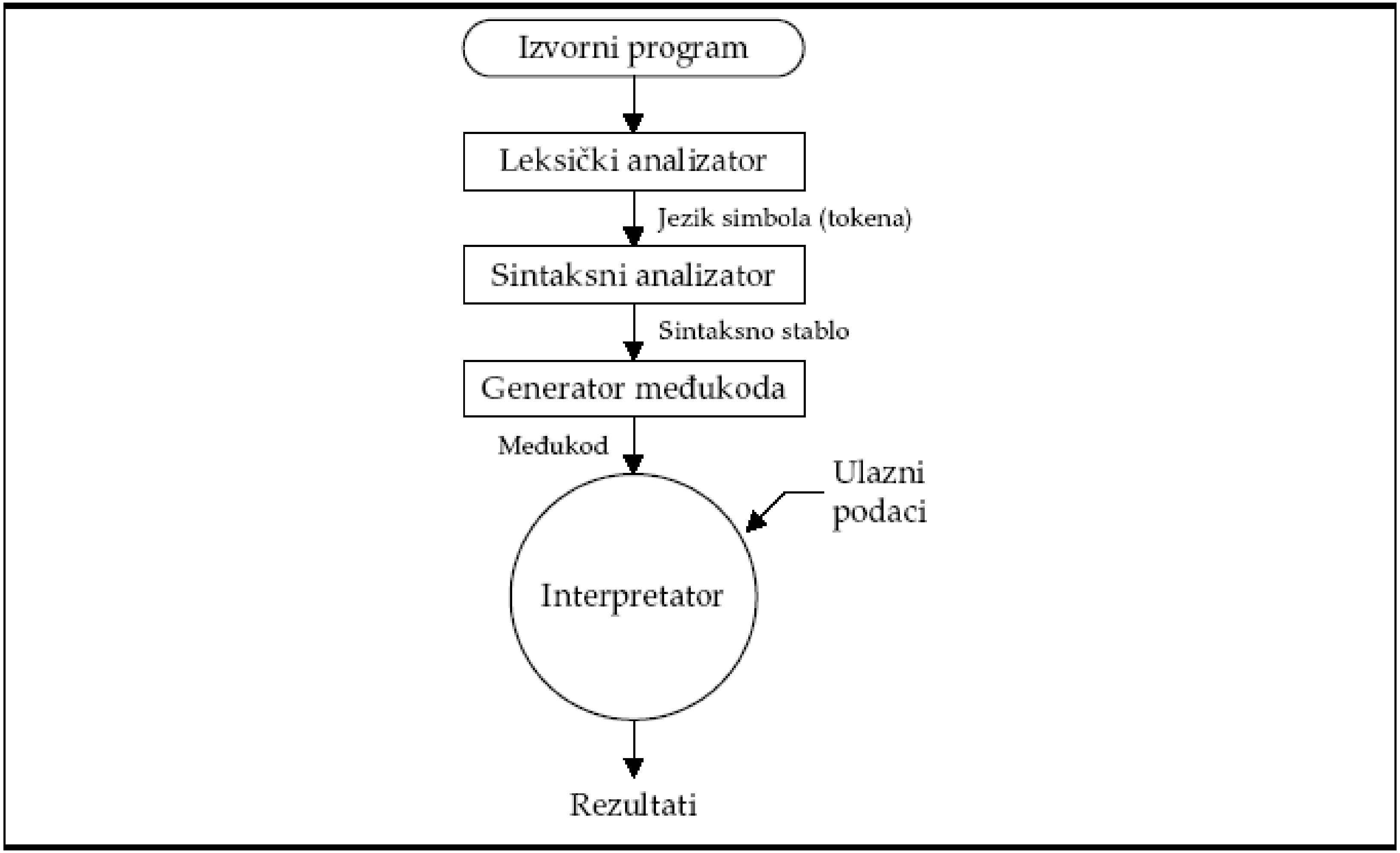
Slika 1.3 Makroprocesor

Prevodioci jezika višeg nivoa



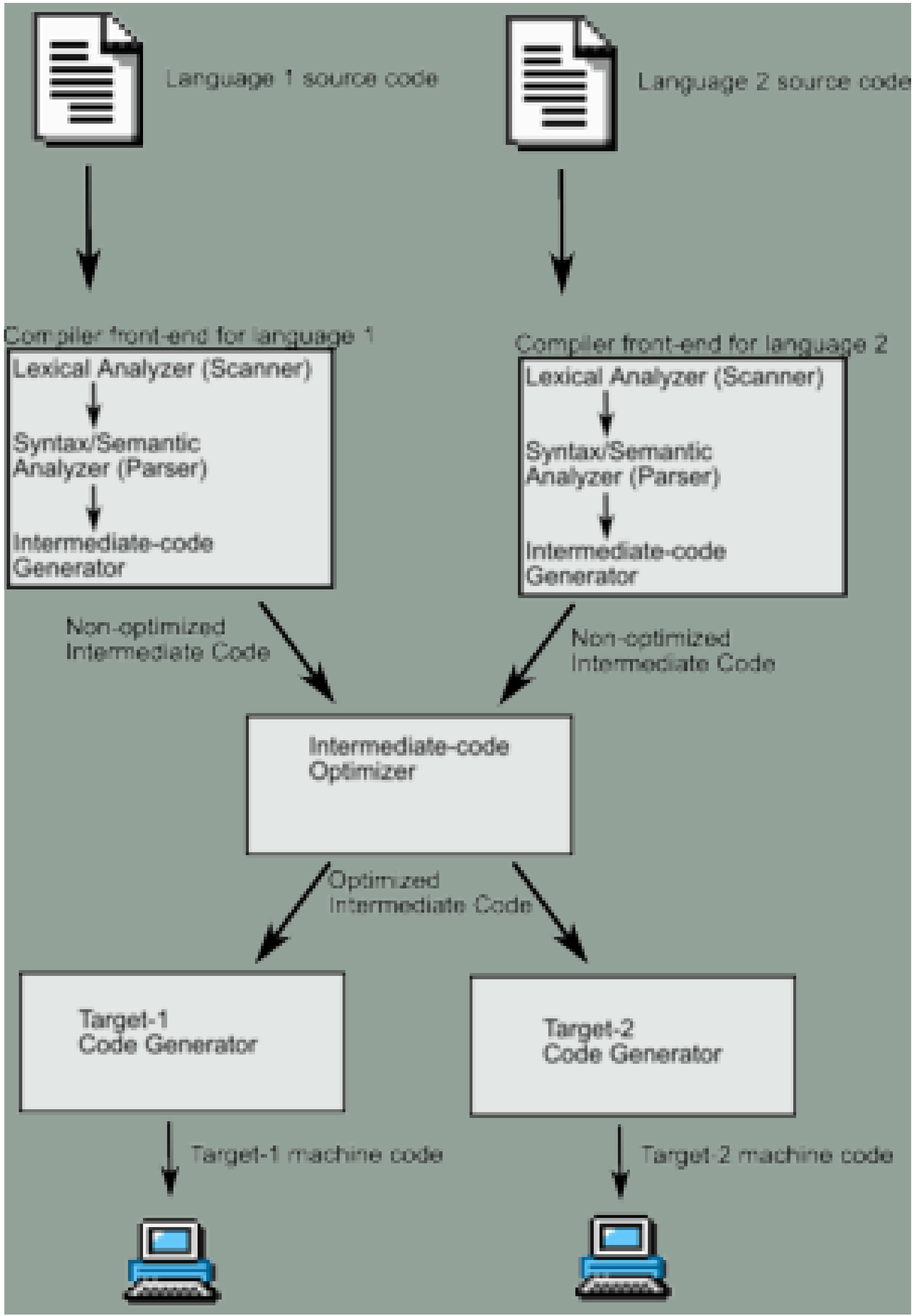
Slika 1.4 Kompilator i interpretator

Hibridni prevodioci



Slika 1.5 *Hibridni sistemi*

Multijezički prevodioci



Leksička analiza

- Leksički analizator izdvaja reči iz ulaznog koda i određuje njihovo značenje:

- Primer:

Pos = init + r * 60

Leksički analizator

id = id + id * const

1.	Pos	…
2.	init	…
3.	r	…

Sintaksna analiza

- Proverava da li je ulazni kod napisan u skladu sa pravilima jezika.

Pravila

- Primer pravila:
  1. Svaki identifikator je izraz
  2. Svaka konstanta je izraz
  3. Ako su *izraz1* i *izraz2* izrazi tada je su izrazi i:

     izraz1+ izraz2

     izraz1* izraz2

     (izraz1)

  4. Ako je *id1* identifikator i *izraz1* izraz, tada je

     id1 := izraz1

     naredba.
  1. Ako je *izraz1* izraz i *naredba1* naredba tada su naredbe i:

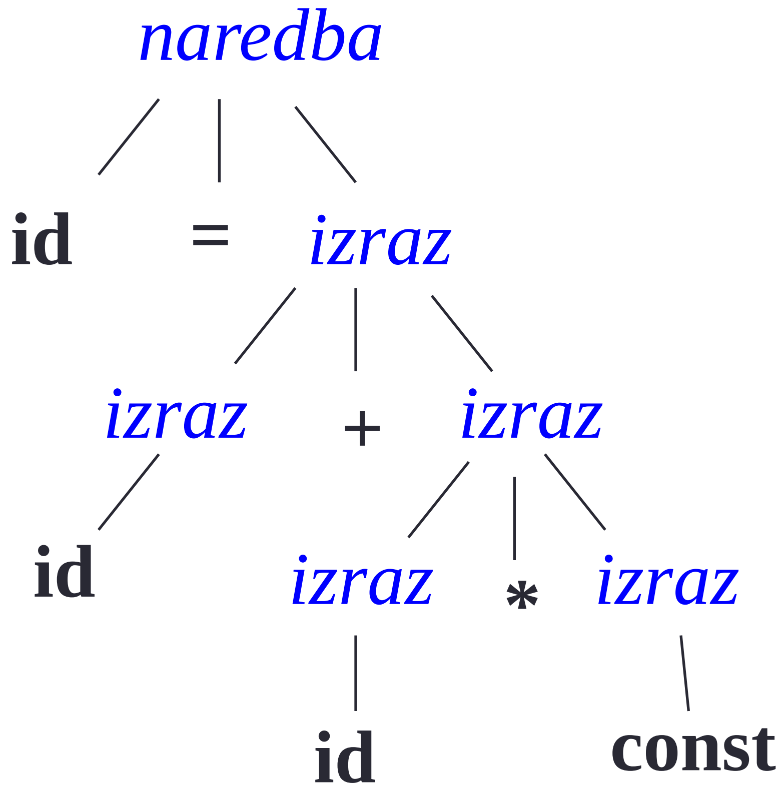
     while (*izraz1*) *naredba1*

     if (*izraz1*) *naredba1*

Sintakna analiza

id = id + id * const

Sintaksni analizator



Semantička analiza

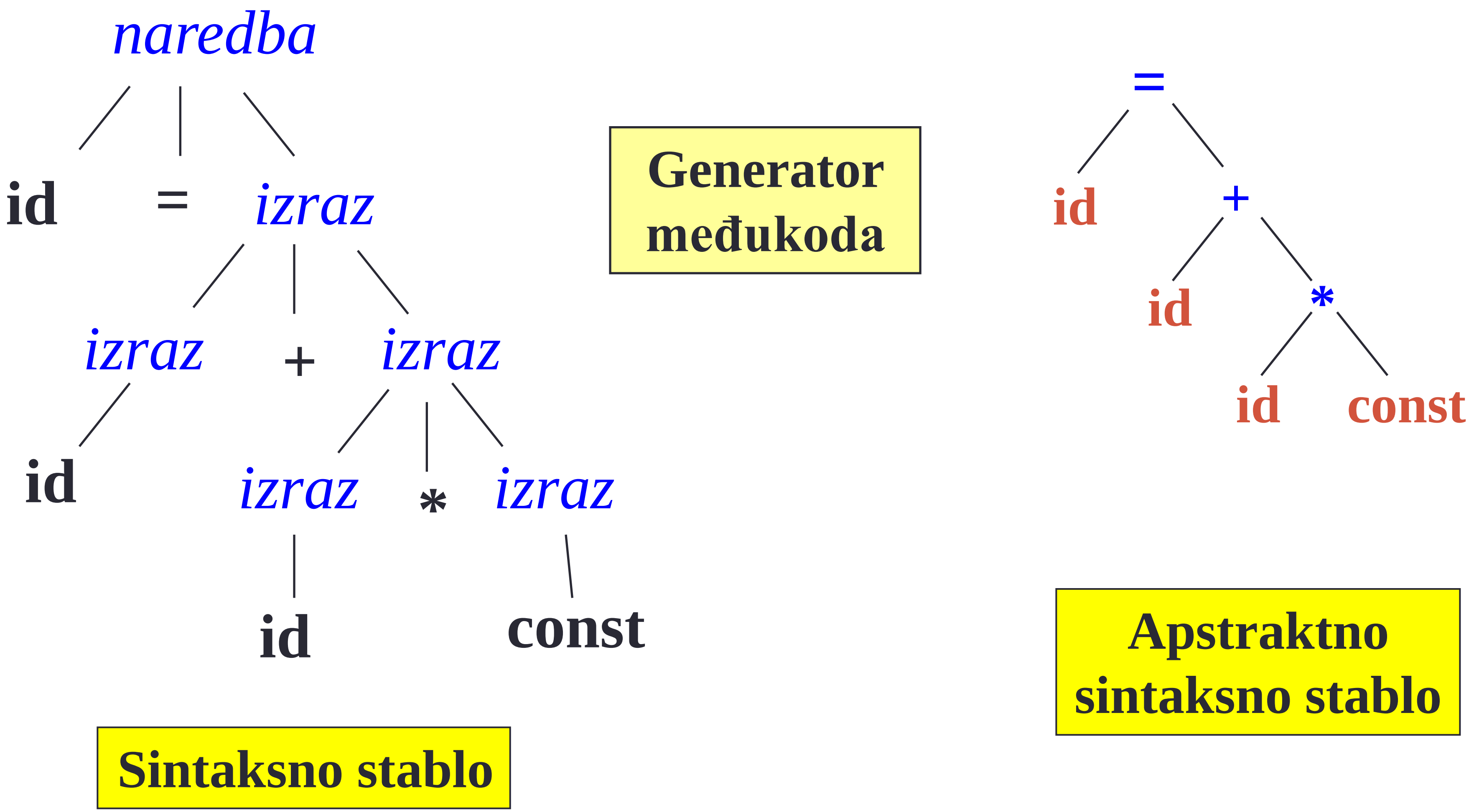
- Proverava da li su prepoznate sintaksne celine medjusobno usaglašene.

Primer:

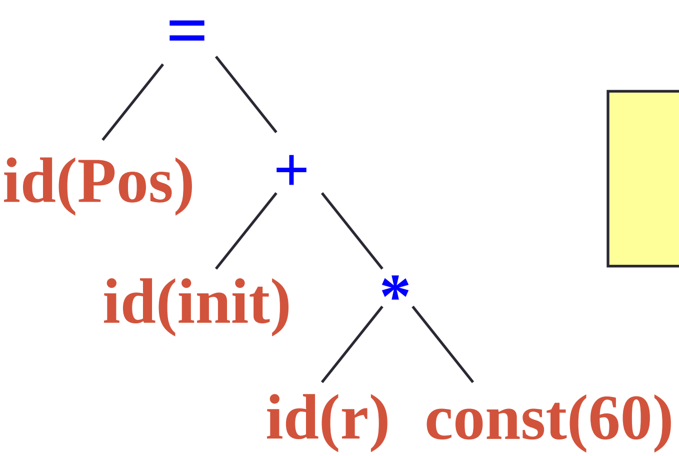
```
1. int init, Pos;
2. bool r=true;
3. Pos = init + r * 60
```

	Name	Type	Def.
1.	Pos	int	3
2.	init	int	-
3.	r	bool	2

Generisanje međukoda visokog nivoa



Generisanje međukoda niskog nivoa



Generator međukoda

temp1 := r * 60

temp2 := init + temp1

Pos := temp2

Optimizacija međukoda

temp1 := r * 60

temp2 := init + temp1

Pos := temp2

Optimizacija međukoda

temp1 := r * 60

Pos := init + temp1

Generisanje koda

temp1 := r * 60

Pos := init + temp1

Generator koda

```
MOV  r, R1
MUL  60, R1
MOV  R1, temp1
MOV  init, R1
ADD  temp1, R1
MOV  R1, Pos
```

Optimizacija izlaznog koda

```
MOV   r, R1
MUL   60, R1
MOV   R1, temp1
MOV   init, R1
ADD   temp1, R1
MOV   R1, Pos
```

Optimizator koda

```
MOV   r, R1
MUL   60, R1
ADD   init, R1
MOV   R1, Pos
```

PROGRAMSKI PREVODIOCI

- Formalni jezici i formalne gramatike -

Azbuka

- Simbol (znak ili slovo) je osnovni (nedeljivi) element jezika.
- Apstraktna (formalna) azbuka, ili samo azbuka, je svaki konačan neprazan skup elemenata V.
- Npr. Azbuku V čine sledeći simboli:
  - Mala i velika slova abecede A,a,B,b,C,c ...
  - Specijalni znaci +, -,*, :=, ...
  - Reči kao što su **if**, **while, class**, ...

Reči

- Niz (niska ili reč) – konačan broj redom napisanih simbola azbuke V.
- Niz koji ne sadrži nijedan simbol naziva se prazna reč i označava sa ε

- Primer: V= {a,b,c}
- Reči: ε, a, b, c, aa, bb, cc, ab, ac, abc, aabc,

- Reči su uređeni nizovi tako da je ab $\neq$ ba

Formalna definicija reči

1. ε reč nad azbukom V
2. Ako je x reč azbuke V i ako je a element azbuke V tada je i xa reč azbuke V.
3. y je reč nad azbukom V ako i samo ako je dobijen pomoću pravila 1. i 2.

Za označavanje reči koriste se obično završna mala slova abecede: *u, v, w, x, y, z*

Dužina reči

- Dužina reči – broj simbola u nizu
- Oznaka: $|x|$ je dužina reči x.
- Za $x = abc$ $\quad |x| = 3$.
- $|\varepsilon| = 0$

Operacije nad rečima: Spajanje (konkatenacija) proizvod reči

- Ako su x i y dve reči azbuke V, proizvod ili spajanje reči je operacija kojom se stvara nova reč tako što se na jednu reč nadovezuje druga reč.

  $x = aA$, $y = ab$

  $z = xy = aAab$

- ε je neutralni element za operaciju množenja (nadovezivanja) reči.

  $\varepsilon x = x\varepsilon = x$

Operacije nad rečima: Eksponent

$$\underbrace{xxx\ldots x}_{\text{n puta}} = x^n$$

$$x^i = x^{i-1}x$$

$$x^0 = \varepsilon$$

$$x^1 = x$$

$$x^2 = xx$$

$$x^3 = xxx$$

Delovi reči

prefiks reči x	Niz koji se dobija kada se izbaci nula ili više simbola na kraju reči *x*, pr. ban je prefiks reči banana.
sufiks reči x	Niz koji se dobija izbacivanjem nula ili više početnih simbola reči *x*. nana je sufiks reči banana
podniz reči x	reč koja se dobija kada se izbaci neki prefiks i/ili neki sufiks reči *x*. ana je podniz reči banana Svaki prefiks i svaki sufiks reči *x* su podnizovi reči *x*, dok svaki podniz reči *x* ne mora da bude ni sufiks ni prefiks reči *x*. Za svaku reč *x* i *x* i ε su prefiksi, sufiksi i podnizovi reči *x*

Delovi reči

Pravi prefiks, pravi sufiks i pravi podniz reči *x*	Svaki neprazan niz *y* koji je prefiks, sufiks ili podniz reči *x*, takav da je *x* različito od *y*.
Podsekvenca reči *x*	Svaki niz koji se dobija izbacivanjem nula ili više sukcesivnih simbola iz reči *x*. baa je podsekvenca niza banana

Formalni jezik

- Formalnim jezikom L nad azbukom V naziva se bilo koji skup reči nad tom azbukom.

- Prema ovoj definiciji formalni jezik je i prazan skup reči kao i skup { ε } koji sadrži samo reč ε.

Primeri jezika

1. Neka azbuku V čine sva slova naše azbuke:

   V={а,б,в,г,д,ђ,ж,з,и,...,ш}

   Sve reči srpskog jezika predstavljaju jedan formalni jezik definisan nad ovom azbukom.

Operacije nad jezicima

OPERACIJA	DEFINICIJA OPERACIJE
Unija jezika L i M L U M	$L \bigcup M = \{x \mid x \in L \vee x \in M\}$
Nadovezivanje konkatenacija L i M LM	$LM = \{xy \mid x \in L \wedge y \in M\}$
Potpuno zatvaranje L*	$L^* = \bigcup_{i=0}^{\infty} L^i$
Pozitivno zatvaranje L^+	$L^+ = \bigcup_{i=1}^{\infty} L^i$

V*

V = {a,b,c}

V* = {ε,a,b,c,aa,bb,cc,ab,ac,bc,ca,cb,abc,…}

$V^+ = V^* \setminus \varepsilon$

Formalni jezik nad azbukom V je bilo koji podskup skupa V*.

$$L \subseteq V^*$$

L^T

- Transponovana reč reči x u oznaci x^T definiše se na sledeći način:
  1. $\varepsilon^T = \varepsilon$
  2. $(xa)^T = ax^T$

Pr. Ako je $x = abc$ tada je $x^T = cba$

Transponovani jezik jezika L je skup svih transponovanih reči jezika L.

Primer

Neka je L={A, B, C,…,Z, a, b, c, …, z} i
D= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.
Kako se slova azbuke mogu posmatrati kao reči dužine 1, onda je svaki od skupova L i D i formalni jezik.

$L \cup D$	skup slova i cifara
LD	skup svih reči koje se sastoje od slova iza kog stoji cifra
L^4	skup svih četvoroslovnih reči
L^*	skup svih nizova slova uključujući i ε
$L(L \cup D)^*$	skup nizova slova i cifara koji počinju slovom.
D^+	skup svih nizova od jedne ili više cifara

Opis i prepoznavanje jezika

- Formalna gramatika je sredstvo za opis jezika na konačan način.
- Gramatika jezika opisuje kako se generišu reči koje pripadaju određenom jeziku.

- Prepoznavanje jezika je problem utvrđivanja da li određena reč pripada jeziku opisanom zadatom gramatikom.
- Ovaj problem se rešava pomoću uređaja za prepoznavanje jezika ili automata

Opis jezika (primer sa proslog casa)

1. Svaki identifikator je izraz
2. Svaka konstanta je izraz
3. Ako su *izraz1* i *izraz2* izrazi tada je su izrazi i:

   izraz1+ izraz2

   izraz1* izraz2

   (izraz1)

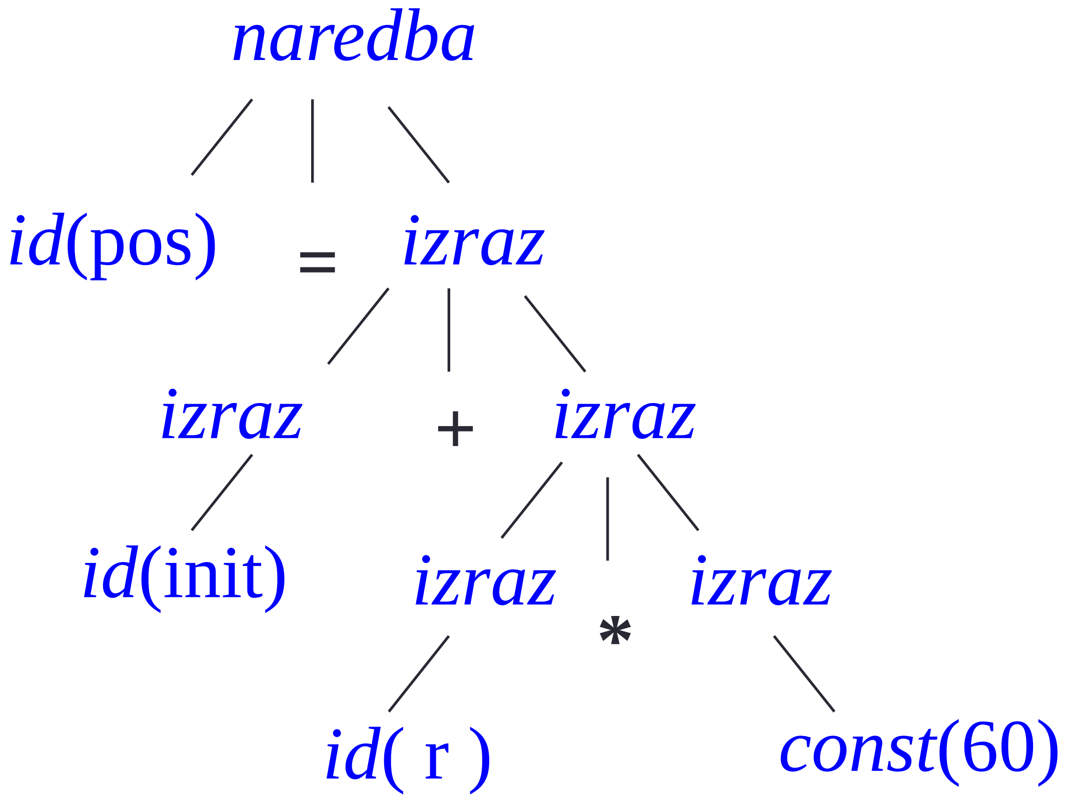
Opis jezika (primer sa proslog casa)

1. Ako je *id1* identifikator i ako je *izraz1* izraz tada je *id1 := izraz1* *naredba.*
2. Ako je *izraz1* izraz i *naredba1* naredba tada su naredbe i:

while (*izraz1*) do *naredba1*

if (*izraz1*) then *naredba1*

Sintaksno stablo



Elementi gramatike

- Terminalni simboli – Osnovni simboli od kojih se satoje reči jezika. Npr. ključne reči nekog programskog jezika **if**, **then**, **else** su terminalni simboli.
- Neterminalni simbol – Pomoćni (sintaksni) simboli kojima se označavaju skupovi reči. Neterminali se uvode da bi se lakše definisao jezik koji se generiše gramatikom, kao i da bi se lakše definisala hierahijska struktura jezika.
- Startni simbol – Neterminalni simbol iz kojeg se izvode sve reči jezika koji se definiše.
- Produkciono pravilo (smena) – definiše način na koji se stvaraju nizovi koji mogu da se sastoje od neterminala i terminala. U opštem slučaju pravila su oblika:

  x ::= y ili x -> y

Notacija

- Terminalni simboli:
  - Slova abecede a,b,c,...
  - Simboli operatora *, +, -, ..
  - Specijalni znaci: (,), <, >, …
  - Cifre 0,1,2, ...9.
  - reči napisane boldiranim fontom kao što su **id** ili **if.**
- Neterminalni simboli:
  - Velika slova A, B, C, ...
  - Reči napisane malim slovima italik: *expr*, *stmt* ili između zagrada: <expr>, <stmt>
- Produkciona pravila
  - x ::= y ili x -> y

Direktno izvođenje i direktna redukcija

$$z = z_1 x z_2, \quad x \rightarrow y, \quad z' = z_1 y z_2$$

Niz z' je direktno izveden iz niza z. Ovaj postupak se naziva direktno izvođenje i označava se sa:

$$z_1 x z_2 \Rightarrow z_1 y z_2$$

Niz z' se redukuje na niz z.

Važi i: $x \Rightarrow y, \qquad x \Rightarrow x$

Izvođenje

$$x \Rightarrow u_1, \quad u_1 \Rightarrow u_2, \ldots, u_n \Rightarrow y$$

Kažemo da se reč y izvodi iz reči x, i da se y redukuje na x. Izvođenje je višestruko primenjeno direktno izvođenje i označava se sa: $x \xrightarrow{*} y$

Formalne gramatike
Noam Chomsky

G = (V_n, V_t, S, P)

Važi: $$V = V_n \cup V_t \quad i \quad V_n \cap V_t = \phi$$

P je skup smena oblika:

$$x \to y, \quad gde\ je \quad x \in V^* V_n V^* \wedge y \in V^*$$

Jezik: $$L(G) = \{ w \mid S \xrightarrow{\ *\ } w,\ w \in V_t^* \}$$

Gramatika – primer 1

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

$P:$

1. $A \to CD$
2. $C \to aCa$
3. $C \to bCB$
4. $BD \to bD$
5. $Ba \to aB$
6. $Bb \to bB$
7. $C \to \varepsilon$
8. $D \to \varepsilon$

$$L(G) = \{ww \mid w \in \{a, b\}^*\}$$

Jezik L(G) sadrži samo reči parne dužine, pri čemu je prva polovina reči jednaka drugoj. Primer izvođenja:

$$A \xrightarrow{1} CD \xrightarrow{2} aCaD \xrightarrow{3} abCBaD \xrightarrow{7}$$

$$abBaD \xrightarrow{5} abaBD \xrightarrow{4} ababD \xrightarrow{8} abab$$

Gramatika – primer 2

1. Svaki identifikator je izraz
2. Svaka konstanta je izraz
3. Ako su *izraz1* i *izraz2* izrazi tada je su izrazi i:

   izraz1+ izraz2

   izraz1* izraz2

   (izraz1)

Gramatika – primer 2

1. Svaki identifikator je izraz
2. Svaka konstanta je izraz
3. Ako su *izraz1* i *izraz2* izrazi tada je su izrazi i:

   izraz1+ izraz2

   izraz1* izraz2

   (izraz1)

$G = (V_n, V_t, S, \mathrm{P}), V_t = \{\mathbf{id}, \mathbf{const}, +, *, (,)\}, V_n = \{izraz\}, S = izraz$

P: $izraz \rightarrow \mathbf{id}$

$izraz \rightarrow \mathbf{const}$

$izraz \rightarrow izraz + izraz$

$izraz \rightarrow izraz * izraz$

$izraz \rightarrow (izraz)$

Tipovi gramatika
Gramatike tipa 0

G = (V$_n$, V$_t$, S, P) u kojoj su sve smene iz skupa P oblika:

$$x \rightarrow y, \quad gde\ je \quad x \in V^* V_n V^* \wedge y \in V^*$$

Primer:

$$V_n = \{S\} \quad i \quad V_t = \{0,1\}$$

$$P = \{S \rightarrow 0S1, S \rightarrow 01\}$$

$$L(G) = \{0^n 1^n \mid n \geq 1\}.$$

Primer izvođenja:

$$S \rightarrow 0S1 \rightarrow 00S11 \rightarrow 000S111 \rightarrow \ldots \rightarrow 0^n 1^n$$

Gramatike tipa 1.
Konteksna gramatika

$$Za \quad x \rightarrow y \quad vazi \quad |y| \geq |x|$$

Kako je |x| >= 1 sledi da je i |y| >= 1, što znači da na desnoj strani pravila ne može da bude prazan niz ε.

Gramatike tipa 2.
Beskonteksne gramatike

Ako u gramatici G svaka smena ima oblik:

$$A \rightarrow y, \quad A \in V_n, \quad y \in V^*$$

Za ove gramatike se koristi i naziv *Bekusova normalna forma* BNF i najčešće se koriste za opis sintakse programskih jezika.

Primer: Sledeća gramatika definiše proste aritmetičke izraze:

$$G = (\{E, A\}, \{(,), +, -, *, /, \mathbf{id}\}, E, P)$$

$$P:$$

$$E \rightarrow EAE \mid (E) \mid -E \mid \mathbf{id}$$

$$A \rightarrow + \mid - \mid * \mid /$$

Gramatike tipa 3.
Regularne gramatike

Gramatika G je gramatika tipa 3 ako je svaka njena smena oblika:

$$A \to aB \vee A \to a, \quad A, B \in V_n \wedge a \in V_t \bigcup \{\varepsilon\}$$

Za ove gramatike se koriste još i nazivi *Regularne gramatike*, *Gramatike sa konačnim brojem stanja* i *Automatne gramatike*.

Služe za opis leksičkih elemenata jezika.

Relacije između jezika

$$L(3) \subseteq L(2) \subseteq L(1) \subseteq L(0)$$

Gramatike tipa 0 su najopštije i praktično su sinonim za algoritam. Sva algoritamska preslikavanja se mogu opisati gramatikama tipa 0.

Gramatike tipa 3 pokrivaju najuži skup jezika ali je za ove jezike najjednostavnije rešiti problem prepoznavanja. Prepoznaju se pomoću konačnih automata.

Rečenične forme

- Svi nizovi koji nastaju u postupku generisanja jezika su rečenične forme tog jezika.
- Sve rečenične forme jezika se redukuju na startni simbol.

$$G = (\{E, A\}, \{(,),+,-,*,/,\mathbf{id}\}, E, P)$$

$$P: \quad E \to EAE \mid (E) \mid -E \mid \mathbf{id}; \quad A \to + \mid - \mid * \mid /$$

$$E \to \boxed{EAE} \to \boxed{E + E} \to \boxed{id + E}$$

Normalne forme gramatika

- Dve gramatike su ekvivalentne ako generišu isti jezik.

- Pod normalnim formama podrazumevamo standardni način zadavanja skupa ekvivalentnih gramatika.

NF za konteksne gramatike

Ako je G konteksna gramatika onda postoji njoj ekvivalentna konteksna gramatika G1 u kojoj svaka smena ima oblik:

$$xAy \rightarrow xry \quad A \in V_n, \quad x, y \in V^*, \quad r \in V^+$$

Na osnovu ovog svojstva izveden je i naziv konteksne gramatike. Vidi se da se neterminal *A* zamenjuje nizom *r* samo ako se nađe u kontekstu reči *x* i *y*.

NF za beskonteksne gramatike (1)

Svaka beskonteksna gramatika G ima ekvivalentnu gramatiku G1 u kojoj svaka smena ima jedan od sledećih oblika:

$$S \to \varepsilon, \quad A \to BC, \quad A \to a$$

$$A, B, C \in V_n, \quad a \in V_t$$

Ova normalna forma se često naziva Normalna forma Čomskog.

NF za beskonteksne gramatike (2)

Svaka beskonteksna gramatika G ima ekvivalentnu gramatiku G1 u kojoj svaka smena ima jedan od sledećih oblika:

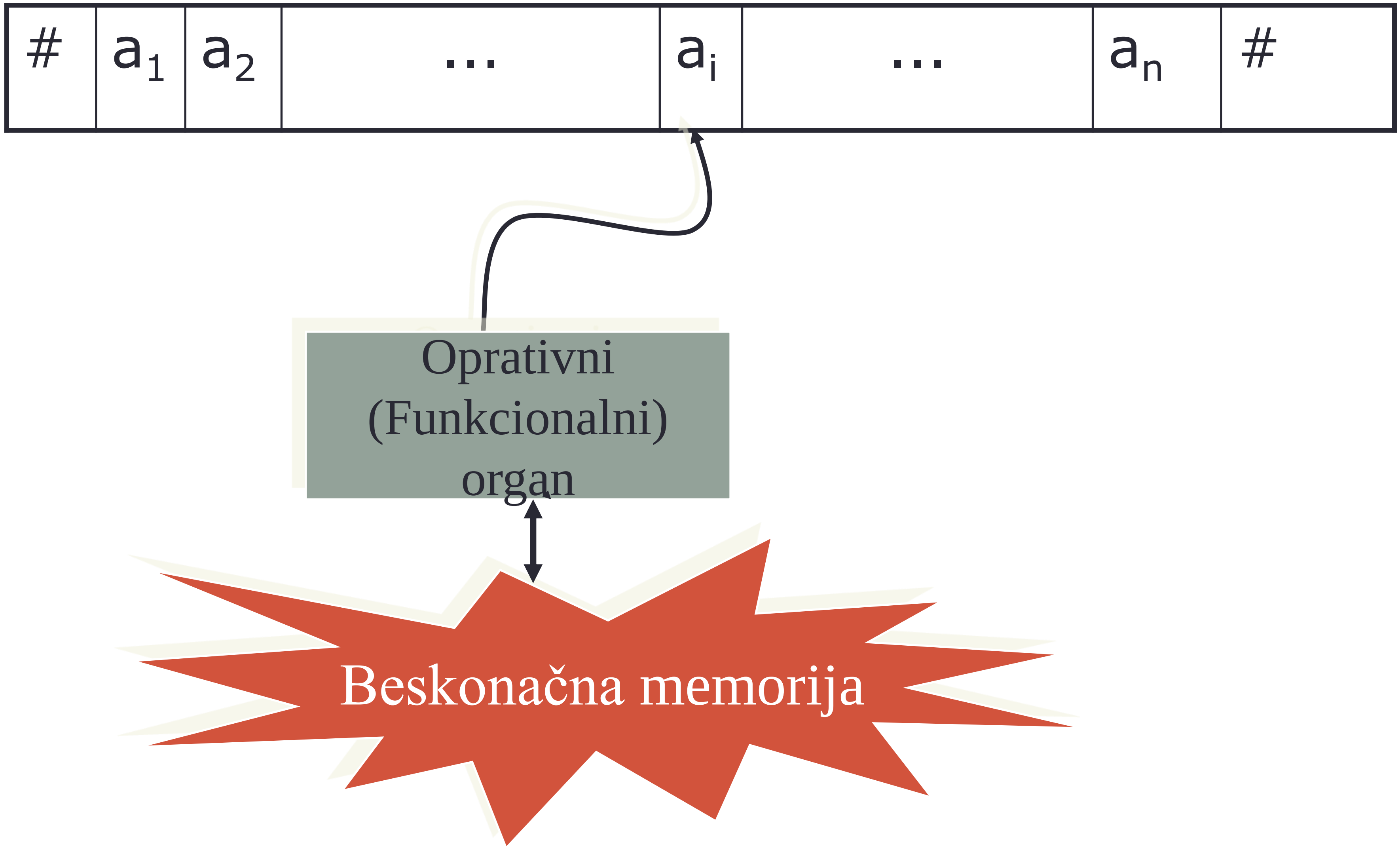
$$S \to \varepsilon, \quad A \to a, \quad A \to aB, \quad A \to aBC$$

$$A, B, C \in V_n, \quad a \in V_t$$

PROGRAMSKI PREVODIOCI

- Uređaji za prepoznavanje jezika -

Automati



#
a1
a2
...
ai
...
an
#
Oprativni (Funkcionalni) organ
Beskonačna memorija

Podela automata

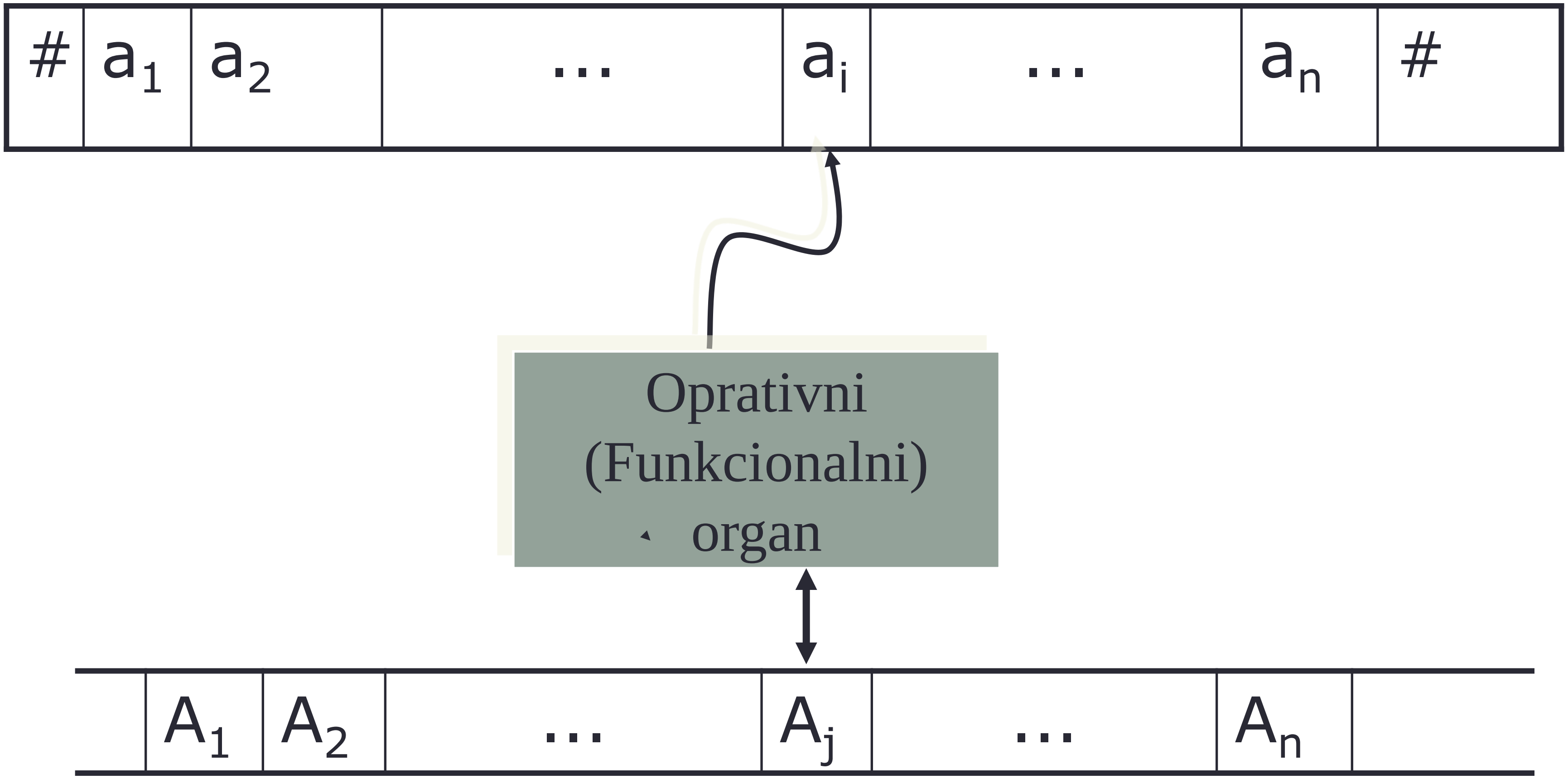
- Prema mogućim smerovima kretanja ulazne glave
  - Jednosmerne
  - Dvosmerne
- Prema broju stanja u koje može da pređe iz tekućeg stanja pod dejstvom istog ulaznog simbola:
  - Determinističke
  - Nedeterminičke

- 2N Dvosmerni nedeterministički
- 1N Jednosmerni nedeterministički
- 2D Dvosmerni deterministički
- 1D Jednosmerni deterministički

Tipovi automata

- Prema organizaciji radne memorije:
  - Tjuringova mašina – radna memorija je beskonačna traka
  - Linearno ograničeni automati – radna memorija je konačna traka
  - Magacinski automati – radna memorija je magacin
  - Konačni automati – ne postoji radna memorija

Tjuringova mašina



2N Tjuringova mašina

- Operativni organ menja stanje.
- Ulazna glava se pomera.
- Na radnu traku se upisuje novi simbol.
- Pomera se radna glava.

2N Tjuringova mašina

$TM=(Q, V, S, g, q_0, b, F)$

Q – Skup stanja operativnog organa

V – Skup simbola na ulaznoj traci

S – Skup simbola na radnoj traci

F – Skup završnih, krajnjih stanja (Podskup skupa Q)

q_0 – Početno stanje operativnog organa

b – Oznaka za prazno slovo

g – Preslikavanje

$$Q \times (V \cup \#) \times S \rightarrow \{Q \times (S \setminus \{b\}) \times \{-1,0,+1\} \times \{-1,0,+1\}\}$$

2N Tjuringova mašina

$$Ako\,(P,Y,d_1,d_2)\in g(q,a,X)$$

Ako se mašina nalazi u stanju *q*, pri čemu ulazna glava čita simbol *a,* a radna glava čita simbol *X*, mašina prelazi u novo stanje *P*, upisuje na radnu traku slovo *Y* i pomera ulaznu glavu u smeru d_1, a radnu glavu u smeru d_2.

+1	Pomeranje za jedno mesto ulevo
0	Nema pomeranja
-1	Pomeranje za jedno mesto udesno

$$d_1, d_2 \in \{-1, 0, +1\}$$

Tjuringova mašina

- Mašina je uspešno prepoznala ulazni niz ako se u trenutku kada se ulazna glava pozicionira na granični simbol # TM nađe u nekom od završnih stanja (elemenata skupa F).

Tjuringova mašina

- TM je deterministička ako se svaki element (q, a, X) preslikava u najviše jedan element (P, Y, d_1, d_2).

- Četri klase TM 2N, 1N, 2D, 1D međusobno su ekvivalentni i svaka od njih raspoznaje jezike tipa nula.

- TM je pojam ekvivalentan pojmu algoritma.

Linearno ograničeni automat

Linearno ograničen automat je 2N Tjuringova mašina za koju može da se odredi konstanta *k* takva da se prilikom prepoznavanja reči:

$$w \in V^*, \quad |w| = n$$

na radnoj traci upisuje niz *r* pri čemu je:

$$|r| \leq kn$$

Za prepoznavanje svih reči jezka dovoljna je radna traka *k* puta duža od najduže reči jezika.

Linearno ograničeni automat

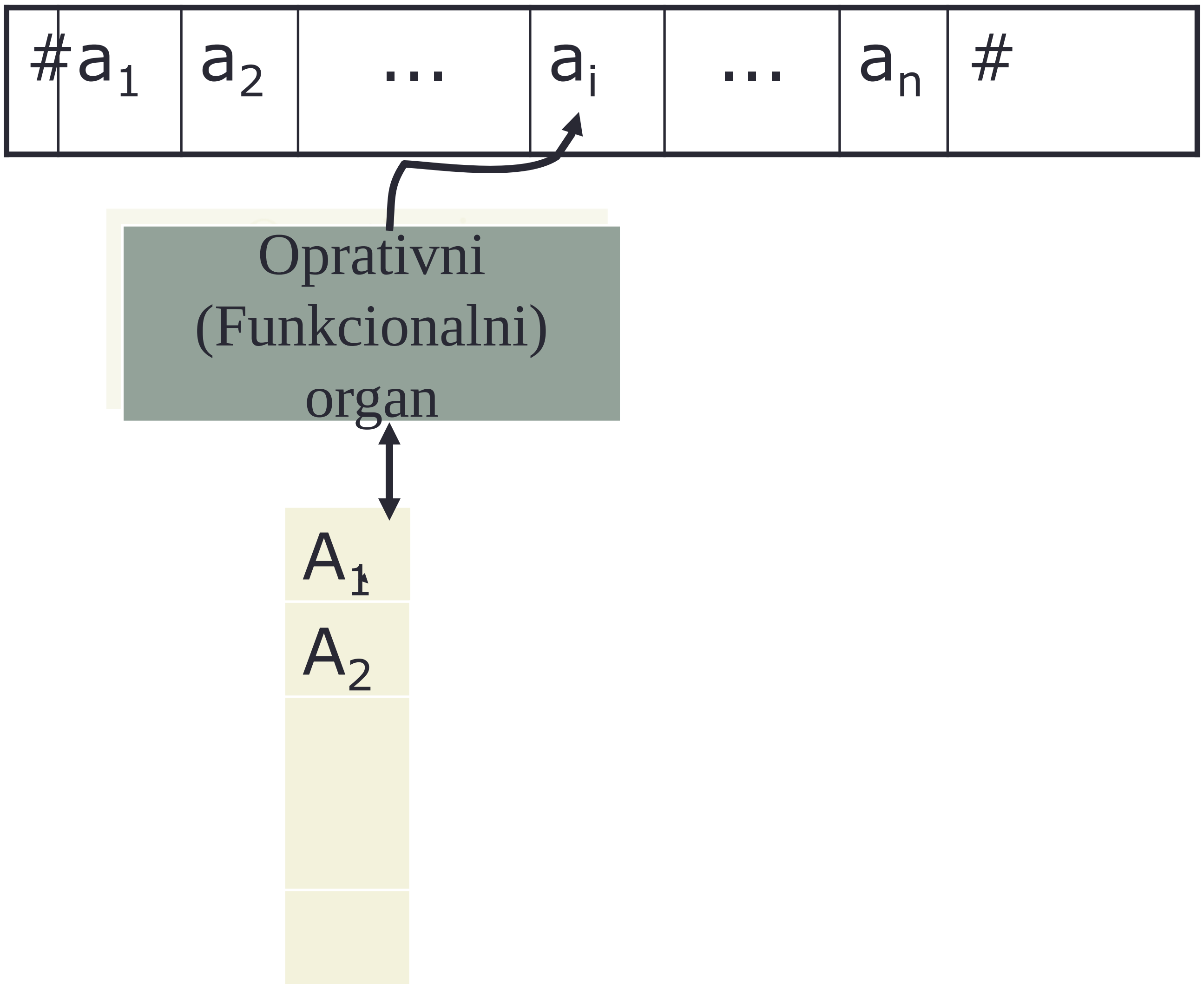
- Klase 2N i 1N LOA međusobno su ekvivalentne i svaka od njih prepoznaje jezike tipa 1. (Za svaki jezik tipa 1. može se definisati nedeterministički LOA koji ga prepoznaje)

- Klase 2D i 1D LOA međusobno su ekvivalentne.

- Ekvivalentnost 2N i 2D LOA do sada nije dokazana.

- Svaka klasa LOA ne prepoznaje sve jezike tipa 1.

Magacinski automat



2N Magacinski automat

$MA=(Q, V, S, g, q_0, Z_0, F)$

Q – Skup stanja operativnog organa

V – Skup simbola na ulaznoj traci

S – Skup magacinskih simbola

F – Skup završnih, krajnjih stanja (Podskup skupa Q)

q_0 – Početno stanje operativnog organa

Z_0 – Početni simbol na vrhu magacina

g – Preslikavanje

$$Q \times (V \cup \#) \times S \to \{Q \times S^* \times \{-1, 0, +1\}\}$$

Magacinski automat

Konfiguracija magacinskog automata:

$$(q, \# a_1 a_2 \cdots \overline{a}_i \cdots a_n \#, X_1 X_2 \cdots X_m)$$
$$a_i \in V, i = 1,2,\cdots,n$$
$$X_m - Vrh\ magacina$$

Ako je $(p, r, d) \in g(q, a_i, X_m)$

$$p \in Q, \quad r \in S^*, \quad d \in \{-1, 0, +1\},$$

$$(q, \# a_1 a_2 \cdots \overline{a}_i \cdots a_n \#, X_1 X_2 \cdots X_m) \mapsto$$
$$(p, \# a_1 a_2 \cdots \overline{a}_{i+d} \cdots a_n \#, X_1 X_2 \cdots X_{m-1} r)$$

U magacin može da se upiše reč, da se ništa ne upiše i da se izbaci jedno slovo kada je $r = \varepsilon$.

Primer

$$S \to 0S1$$

$$S \to a$$

- Jezik koji je definisan ovom gramatikom:

$$S \to 0S1 \to 00S11 \to \ldots \to 0^n a 1^n, \text{ gde } n \geq 0$$

Primer

$$S \rightarrow 0S1$$

$$S \rightarrow a$$

$$Q = \{q_0, q_1\}$$

$$X = \{0, 1, a\}$$

$$Y = \{A, B\}$$

$$Z_0 = A$$

$$q_0 - \text{pocetno stanje}$$

$$g:$$

$$\{(q_0, \#, A) \rightarrow (q_0, A),$$

$$(q_0, 0, A) \rightarrow (q_0, BA),$$

$$(q_0, a, A) \rightarrow (q_1, \varepsilon),$$

$$(q_1, 1, B) \rightarrow (q_1, \varepsilon),$$

$$(q_1, \#, \varepsilon)\ kraj\ preslikavanja$$

Na kraju prepoznavanja magacin je ispražnjen.

Primeri prepoznavanja

$(q_0, \overline{\#}000a111\#, A) \mapsto$

$(q_0, \#\overline{0}00a111\#, A) \mapsto$

$(q_0, \#, 0\overline{0}0a111\#, BA) \mapsto$

$(q_0, \#, 00\overline{0}a111\#, BBA) \mapsto$

$(q_0, \#, 000\overline{a}111\#, BBBA) \mapsto$

$(q_1, \#, 000a\overline{1}11\#, BBB) \to$

$(q_1, \#, 000a1\overline{1}1\#, BB) \mapsto$

$(q_1, \#, 000a11\overline{1}\#, B) \mapsto$

$(q_1, \#, 000a111\overline{\#}, \varepsilon)$

$(q_0, \overline{\#}a\#, A) \mapsto$

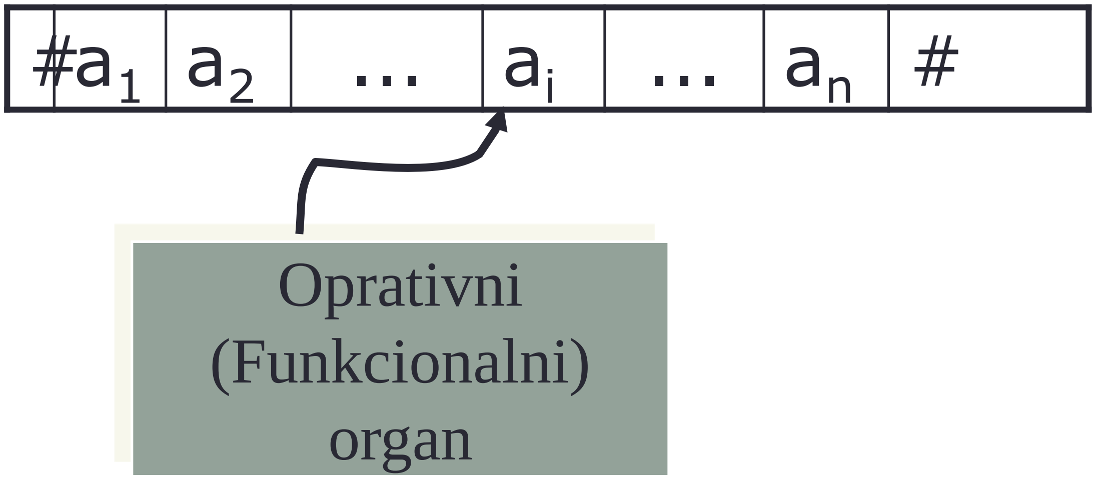
$(q_0, \#\overline{a}\#, A) \mapsto$

$(q_1, \#a\overline{\#}, \varepsilon)$

Magacinski automat

- 1N magacinski automat prepoznaje jezik L ako i samo ako je L beskonteksni jezik tipa 2.
- 2N i 2D magacinskim automatima mogu se prepoznavati svi beskontesni i neki, ali ne svi, konteksni jezici.
- Postoje beskonteksni jezici koji se ne mogu prepoznati 1D magacinskim automatima.
- Magacinski automati se koriste u sintaksnoj analizi programskih jezika.

Konačni automati



Na osnovu stanja operastivnog organa i simbola koji čita ulazna glava, menja se stanje automata i ulazna glava pomera za jedno mesto ulevo ili udesno ili ostaje na svom mestu.

Konačni automat

MA=(Q, V, g, q_0,F)

Q – Skup stanja operativnog organa

V – Skup simbola na ulaznoj traci

F – Skup završnih, krajnjih stanja (Podskup skupa Q)

q_0 – Početno stanje operativnog organa

g – Preslikavanje

$$Q \times \{V \cup \#\} \to Q$$

Za neki ulazni niz $w \in V^*$ kazemo da je prepoznat automatom ako je $g(q_0, w) \in F$.

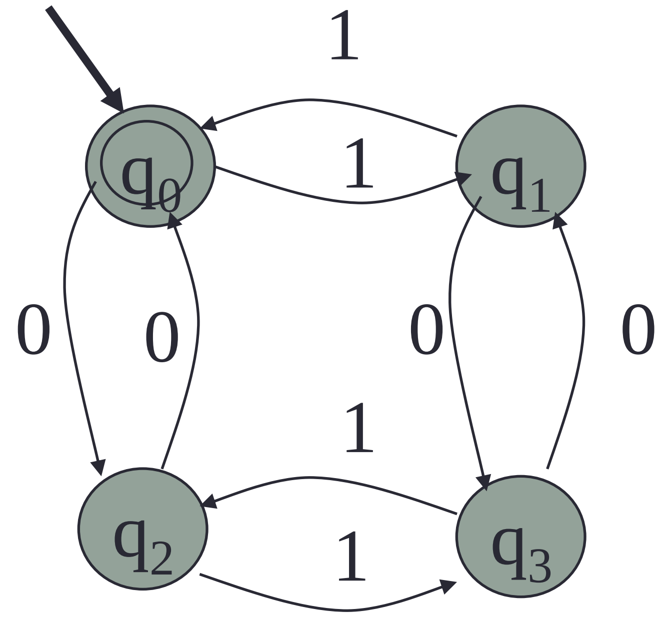
Primer

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

$$Q = \{q_0, q_1, q_2, q_3\},$$

$$V = \{0,1\},$$

$$F = \{q_0\}$$



$g:$

$g(q_0,0) = q_2,$ $\qquad g(q_0,1) = q_1,$

$g(q_1,0) = q_3,$ $\qquad g(q_1,1) = q_0,$

$g(q_2,0) = q_0,$ $\qquad g(q_2,1) = q_3,$

$g(q_3,0) = q_1,$ $\qquad g(q_3,1) = q_2.$

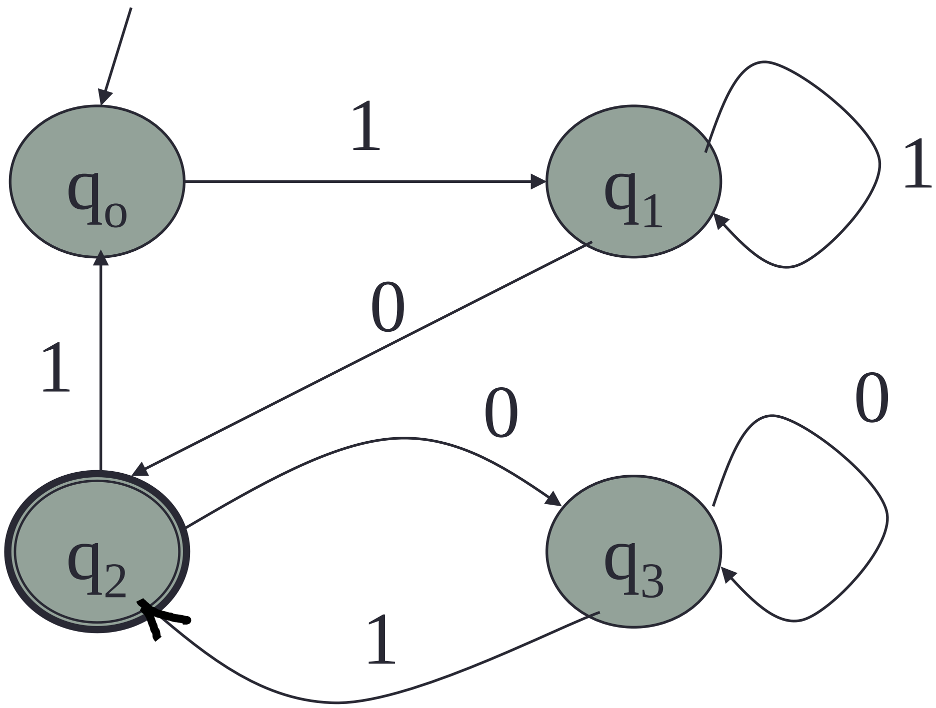
Može se dokazati da ovaj KA prepoznaje reči koje se sastoje od parnog broja jedinica i parnog broja nula.

Regularni izrazi

Skup svih reči koje prepoznaje KA nazivamo regularnim i označavaćemo ih sa $T(k)$:

$$T(k) = \{x \mid g(q_0, x) \in F\}$$

Primer



Jezik koji prepoznaje KA određujemo analitički kao događaj krajnjeg stanja u funkciji početnog stanja.

$$q_{kraj} = f(q_{poc}) \qquad q_2 = f(q_0)$$

Jednačine stanja

1. $q_0 = q_0\varepsilon \cup q_2 1$
2. $q_1 = q_1 1 \cup q_0 1$
3. $q_2 = q_1 0 \cup q_3 1$
4. $q_3 = q_2 0 \cup q_3 0$

Transformacije koje se koriste u izvođenju regularnih izraza:

$$q_i = q_i a \cup q_j b \Rightarrow q_j b a^*$$

$$q_i = q_j a \cup q_j b \Rightarrow q_j (a \cup b)$$

Rešenje

$4'. \qquad q_3 = q_2 00^*$

$4' \to 3 \qquad q_2 = q_1 0 \cup q_2 00^* 1 \qquad (3')$

$2'. \qquad q_1 = q_0 11^*$

$2' \to 3' \qquad q_2 = q_0 11^* 0 \cup q_2 00^* 1 \qquad (3'')$

$1 \to 3'' \qquad q_2 = q_0 \varepsilon 11^* 0 \cup q_2 111^* 0 \cup q_2 00^* 1$

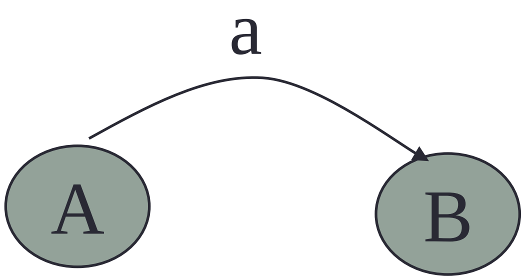
$q_2 = q_0 \varepsilon 11^* 0 (111^* 0 \cup 00^* 1)^*$

$R(G) = 11^* 0 (111^* 0 \cup 00^* 1)^*$

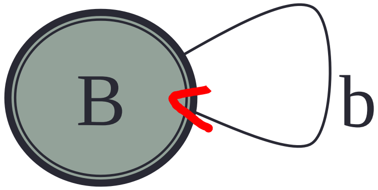
Gramatike tipa 3 i KA

Postoji direktna veza između gramatika tipa 3. i KA:

- Q - V_n
- V - V_t
- g - Skup smena
- q_0 - Startni simbol S.



$$A \rightarrow aB$$

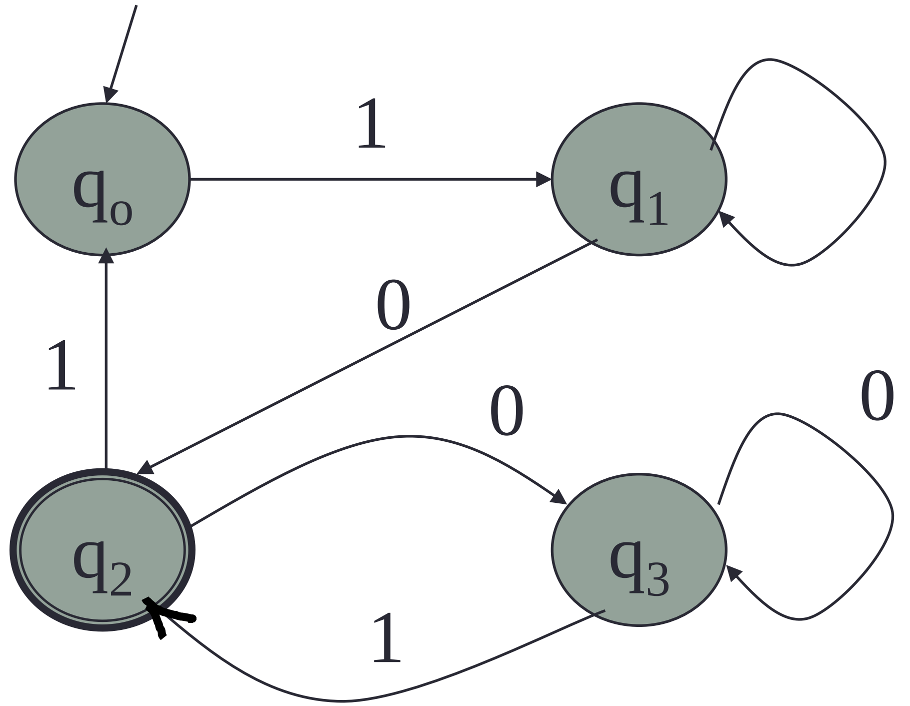


$$B \rightarrow bB \qquad B \rightarrow bB$$

$$B \rightarrow \varepsilon \qquad B \rightarrow b$$

Gramatika jezika

1. $Q_0 \to 1Q_1,$
2. $Q_1 \to 1Q_1,$
3. $Q_1 \to 0Q_2,$
4. $Q_2 \to 1Q_0,$
5. $Q_2 \to 0Q_3,$
6. $Q_2 \to \varepsilon\ ,$
7. $Q_3 \to 0Q_3,$
8. $Q_3 \to 1Q_2.$



$$V_n = \{Q_0, Q_1, Q_2, Q_3\}$$

$$V_t = \{0,1\}$$

$$S = Q_0$$