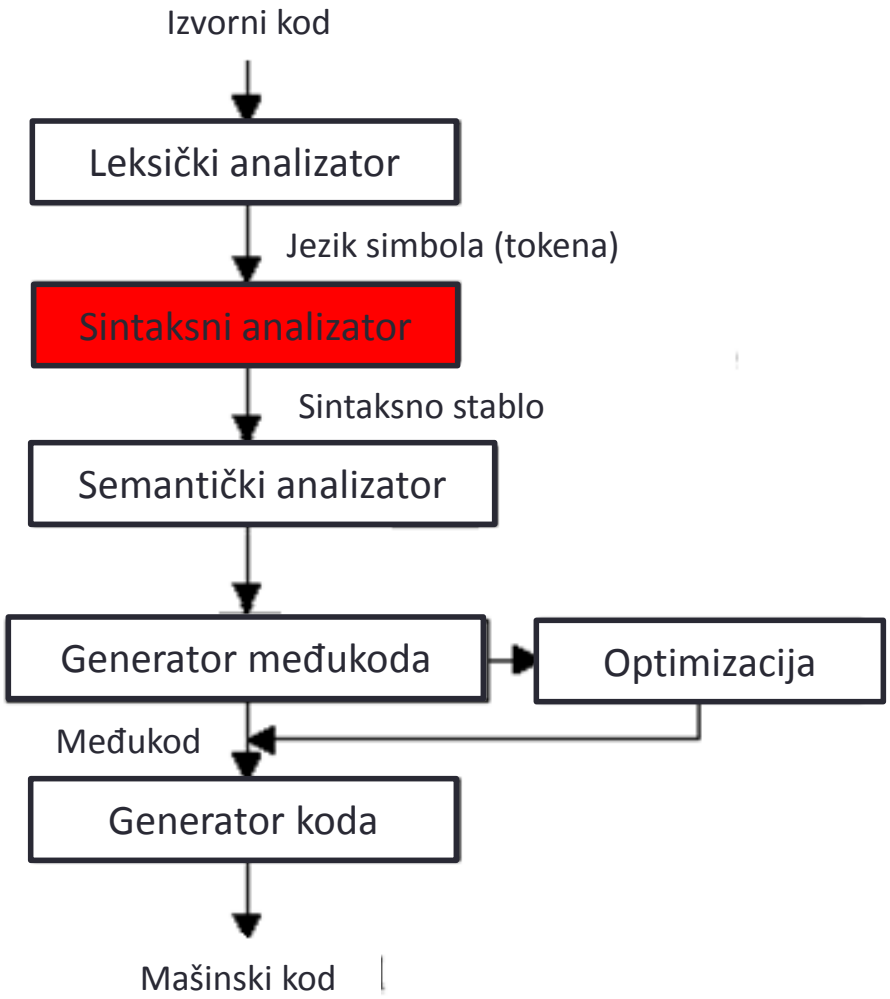
PROGRAMSI PREVODIOCI
- Sintaksna analiza -

Struktura kompilatora



Sintaksni analizator

- Proverava da li je program napisan u skladu sa formalnim opisom jezika.
- Ulaz: niz tokena koje dobija od leksičkog analizatora.
- Izlaz: sintaksno stablo koje pokazuje kako je program izgrađen.

Metode za generisanje sintaksnog stabla

- Top-down
  - S leva na desno (krajnje levo izvođenje)
  - S desna na levo (krajnje desno izvođenje)
- Bottom-up

Generisanje sintaksnog stabla – primer 1

Neka je data gramatika G kojom je definisan zapis identifikatora u jednom programskom jeziku:

$$G(\{<id>, <slovo>, <cifra>\}, \{a,b,c,\ldots,z,0,1,\ldots 9\}, <id>, P)$$

P: $<id> \rightarrow <slovo> | <id><slovo> | <id><cifra>$

$<slovo> \rightarrow a | b | c | \ldots | z$

$<cifra> \rightarrow 0 | 1 | 2 | \ldots | 9$

Kreirati sintaksno stablo identifikatora: b11

Top-down algoritam sleva na desno

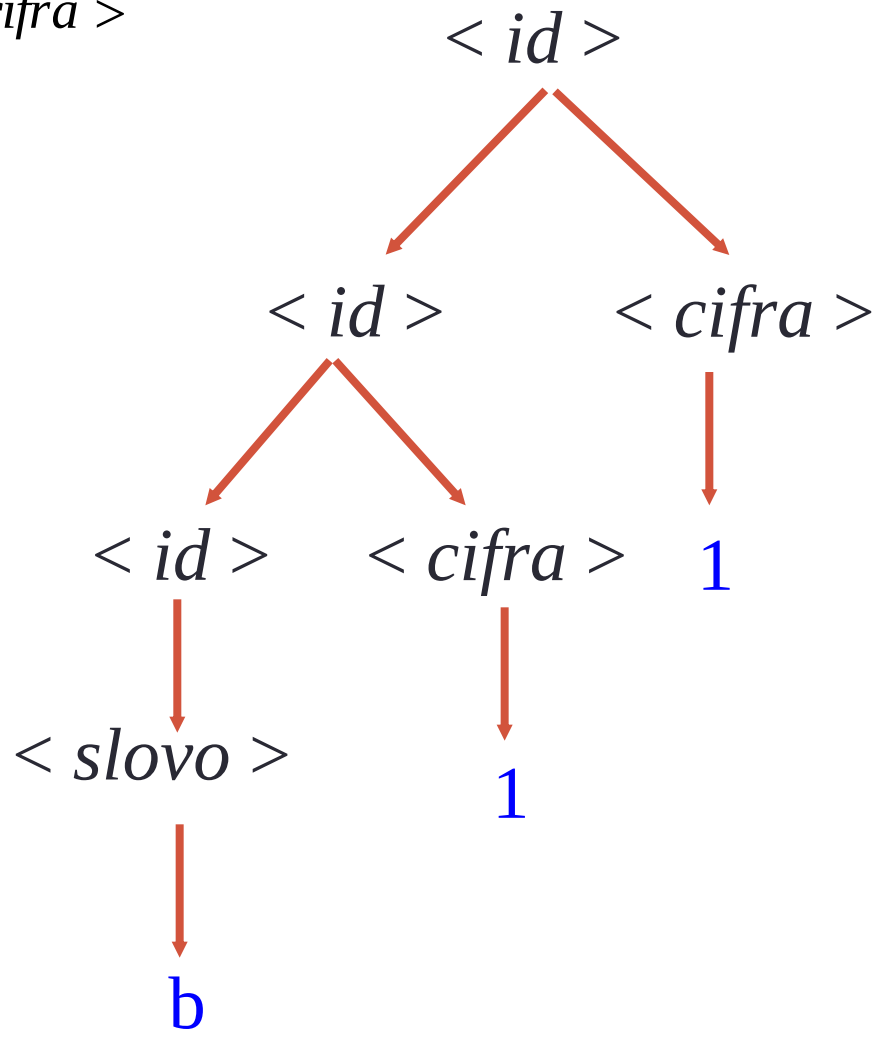
$$< id > \rightarrow < slovo > | < id >< slovo > | < id >< cifra >$$

$$< slovo > \rightarrow a \,|\, b \,|\, c \,|\, \ldots \,|\, z$$

$$< cifra > \rightarrow 0 \,|\, 1 \,|\, 2 \,|\, \ldots \,|\, 9$$

b 1 1

$$< id > \rightarrow < id >< cifra > \rightarrow$$
$$< id >< cifra >< cifra > \rightarrow$$
$$< slovo >< cifra >< cifra > \rightarrow$$
$$b < cifra >< cifra > \rightarrow$$
$$b\,1 < cifra > \rightarrow b\,11$$



Top-down algoritam sdesna na levo

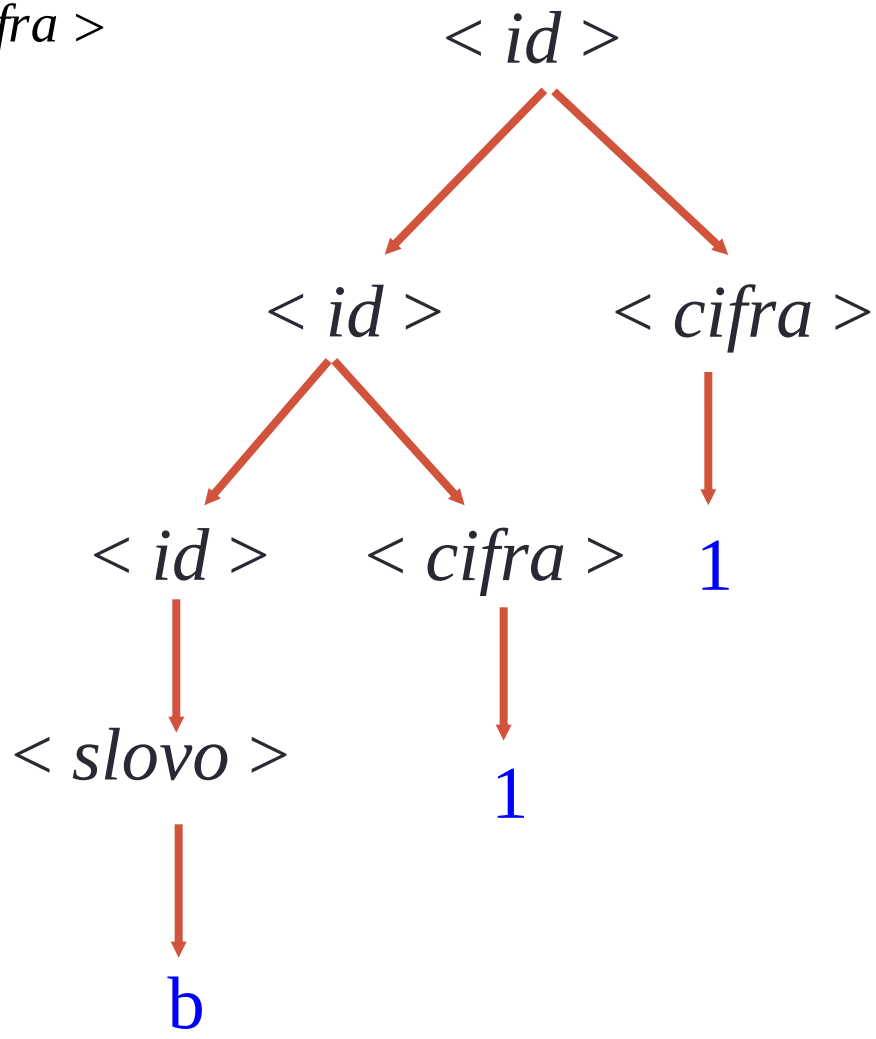
$$< id > \longrightarrow < slovo > | < id >< slovo > | < id >< cifra >$$

$$< slovo > \longrightarrow a \,|\, b \,|\, c \,|\, \ldots \,|\, z$$

$$< cifra > \longrightarrow 0 \,|\, 1 \,|\, 2 \,|\, \ldots \,|\, 9$$

b 1 1

$$< id > \longrightarrow < id >< cifra > \longrightarrow$$
$$< id > 1 \longrightarrow$$
$$< id >< cifra > 1 \longrightarrow$$
$$< id > 11 \longrightarrow$$
$$< slovo > 11 \longrightarrow b\,11$$



Generisanje sintaksnog stabla – primer 2

Sledećom skupom smena je data gramatika kojom je definisan izraz u jednom programskom jeziku:

$$<izraz> \rightarrow <id> | <izraz> + <izraz> | <izraz> * <izraz>$$

Kreirati sintaksno stablo izraza:

$$<id> * <id> + <id>$$

Top-down algoritam sleva na desno

$$< izraz > \rightarrow < id > | < izraz > + < izraz > | < izraz > * < izraz >$$

$$< id > * < id > + < id >$$

$< izraz > \rightarrow$

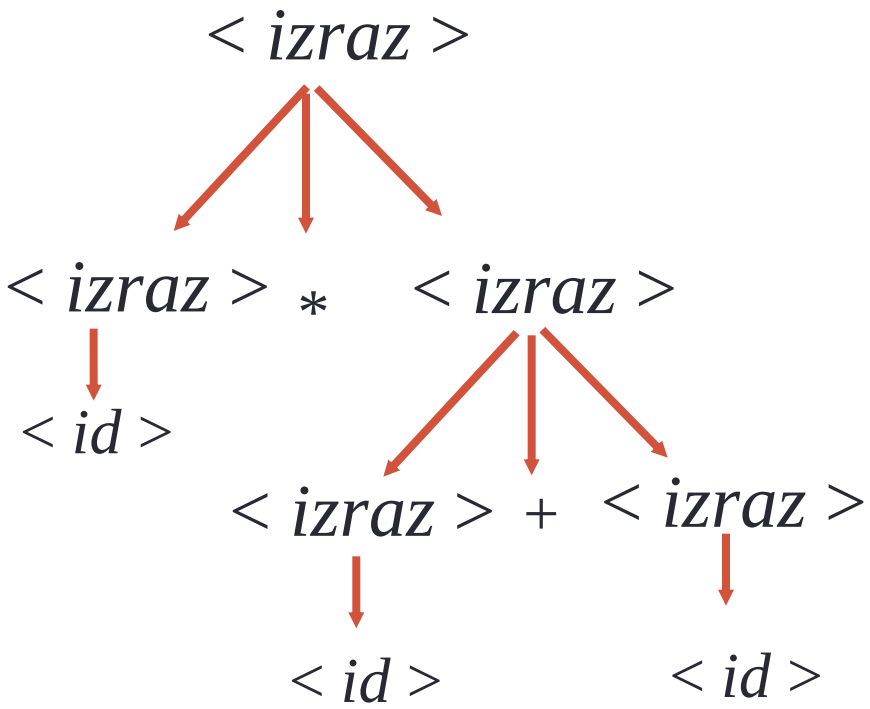
$< izraz > * < izraz > \rightarrow$

$< id > * < izraz > \rightarrow$

$< id > * < izraz > + < izraz > \rightarrow$

$< id > * < id > + < izraz > \rightarrow$

$< id > * < id > + < id >$



Top-down algoritam sdesna na levo

$$< izraz > \rightarrow < id > | < izraz > + < izraz > | < izraz > * < izraz >$$

$$< id > * < id > + < id >$$

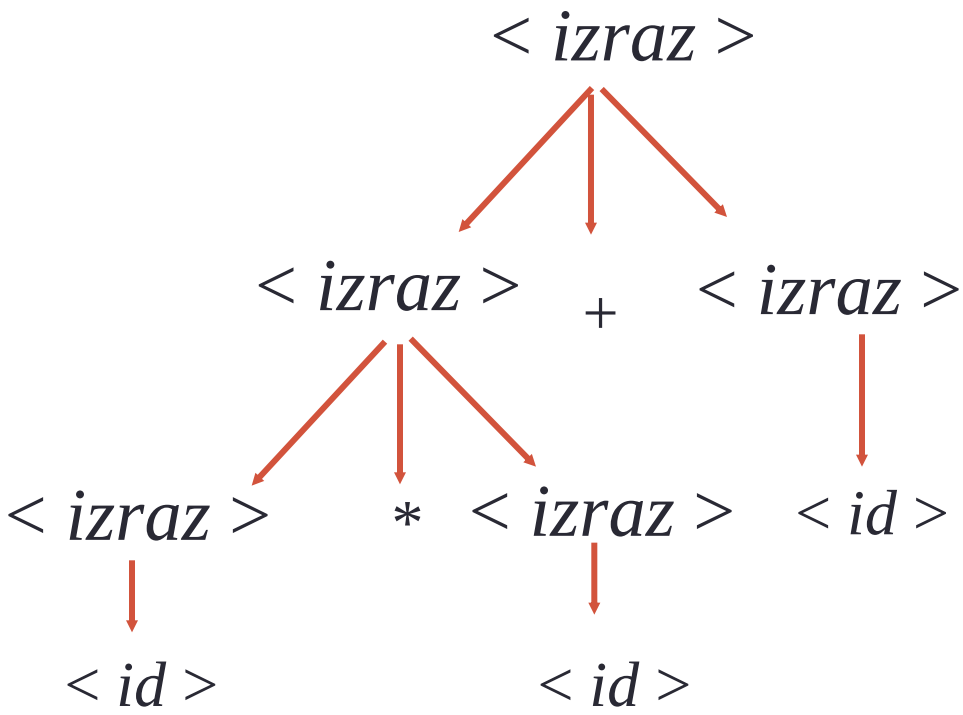
$< izraz > \rightarrow < izraz > + < izraz > \rightarrow$

$< izraz > + < id > \rightarrow$

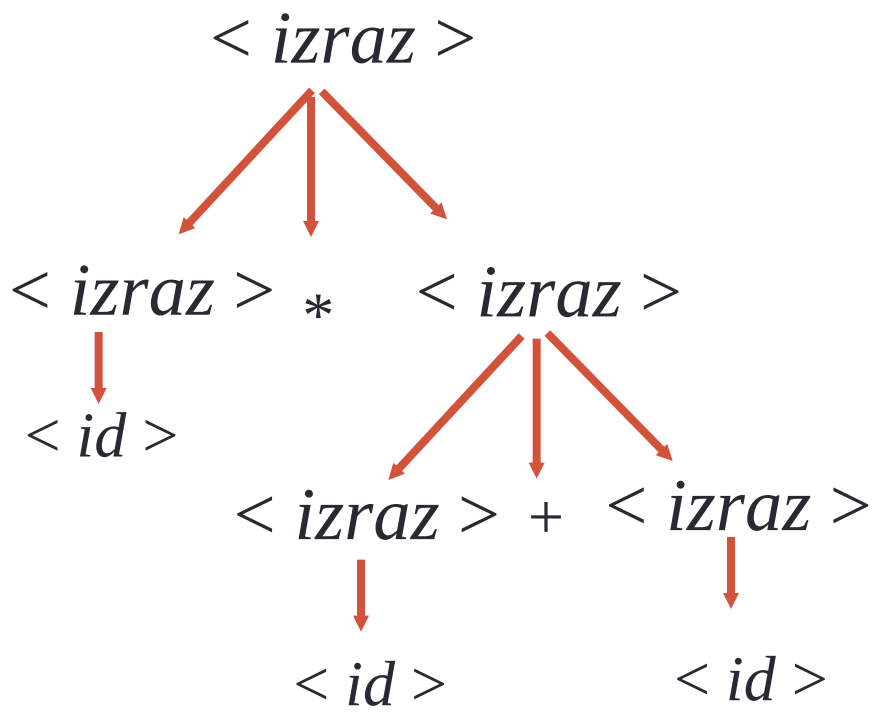
$< izraz > * < izraz > + < id > \rightarrow$

$< izraz > * < id > + < id > \rightarrow$

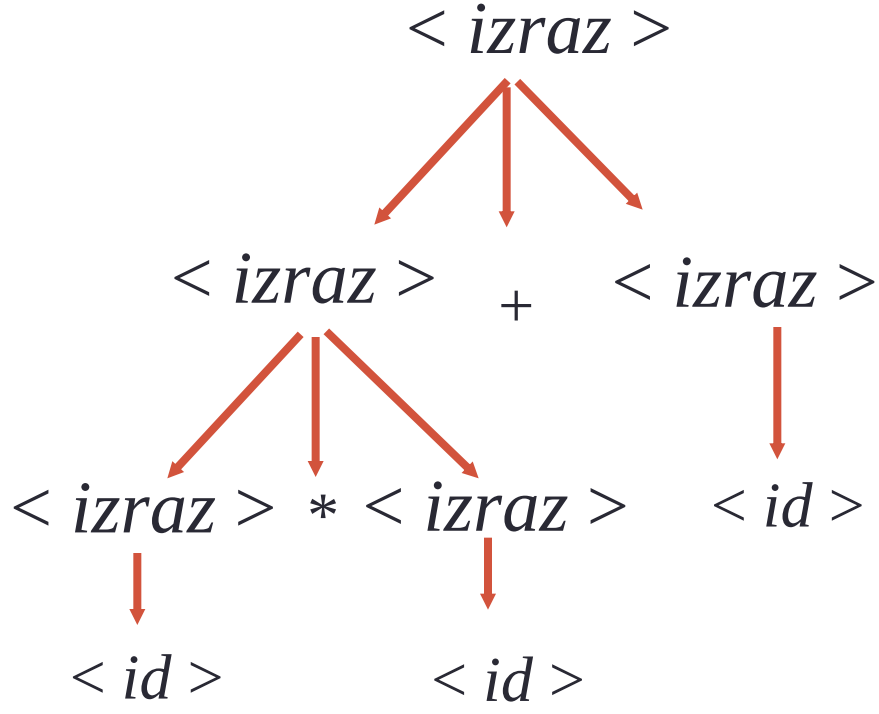
$< id > * < id > + < id >$



Generisana sintaksna stabla



Sleva na desno



Sdesna na levo

ZAKLJUČAK: NEJEDNOZNAČNA GRAMATIKA!

Ekvivalentna jednoznačna gramatika

$$< izraz > \longrightarrow < term > | < izraz > + < term >$$

$$< term > \longrightarrow < factor > | < term > * < factor >$$

$$< factor > \longrightarrow < id >$$

Top-down algoritam sleva na desno

$$< izraz > \longrightarrow < term > | < izraz > + < term >$$

$$< term > \longrightarrow < factor > | < term > * < factor >$$

$$< factor > \longrightarrow < id >$$

$$< id > * < id > + < id >$$

$< izraz > \longrightarrow < izraz > + < term > \longrightarrow$

$< term > * < factor > + < term > \longrightarrow$

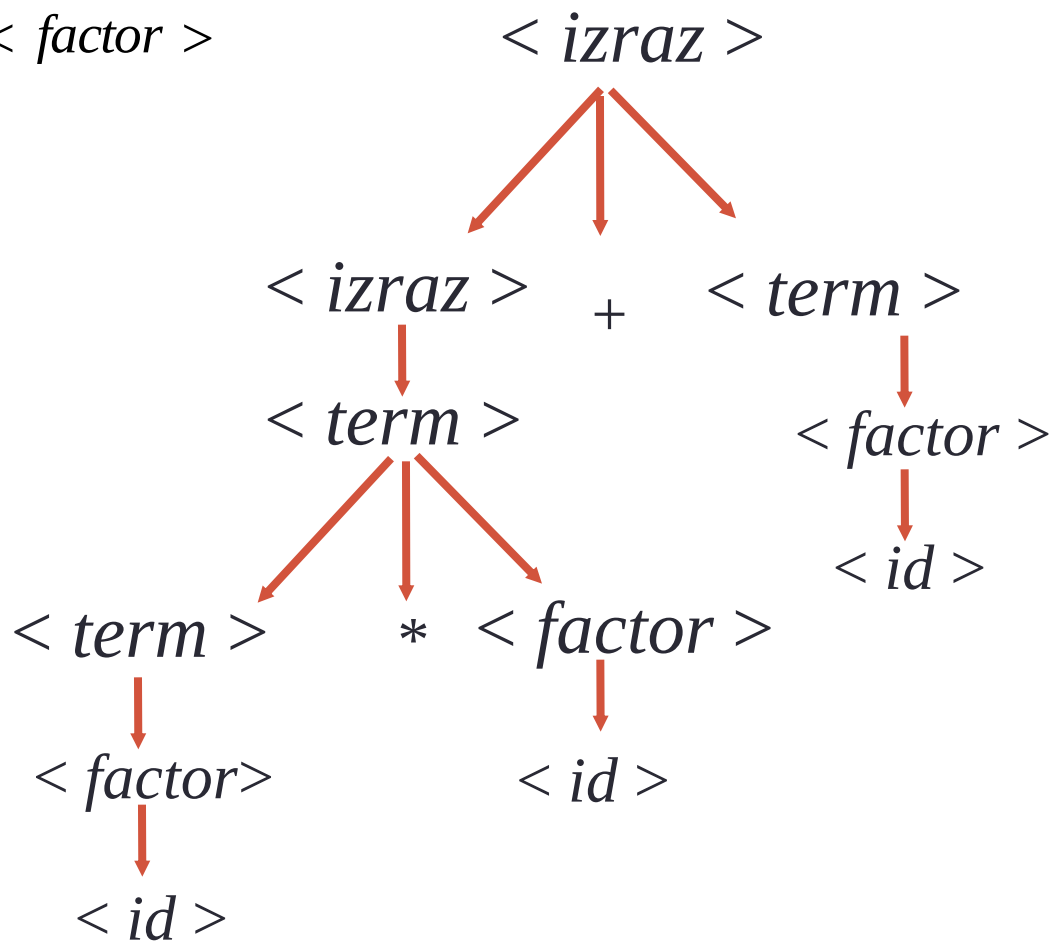
$< factor > * < factor > + < term > \longrightarrow$

$< id > * < factor > + < term > \longrightarrow$

$< id > * < id > + < term > \longrightarrow$

$< id > * < id > + < factor > \longrightarrow$

$< id > * < id > + < id >$



Top-down algoritam sdesna na levo

$< izraz > \rightarrow < term > | < izraz > + < term >$

$< term > \rightarrow < factor > | < term > * < factor >$

$< factor > \rightarrow < id >$

$< id > * < id > + < id >$

$< izraz > \rightarrow < izraz > + < term > \rightarrow$

$< term > * < factor > + < term > \rightarrow$

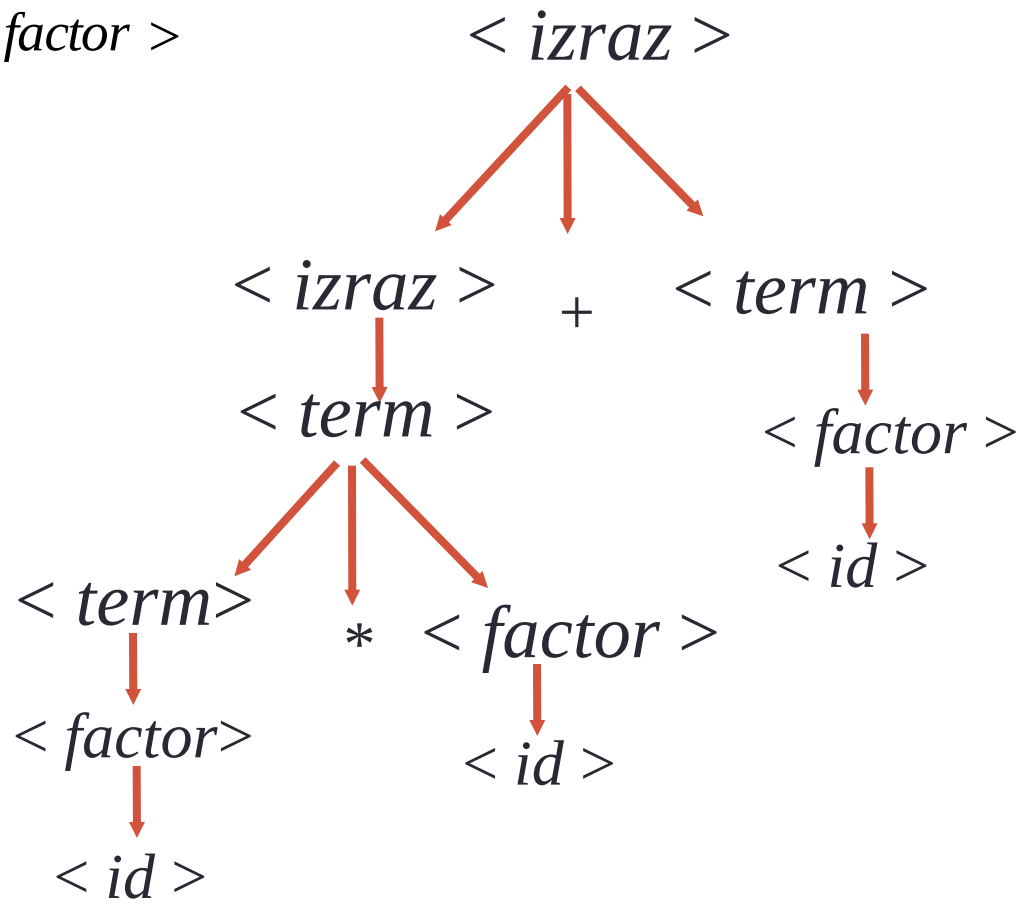
$< factor > * < factor > + < term > \rightarrow$

$< id > * < factor > + < term > \rightarrow$

$< id > * < id > + < term > \rightarrow$

$< id > * < id > + < factor > \rightarrow$

$< id > * < id > + < id >$



Osnovni top-down algoritam

1. U izvedenu sekvencu upisati startni simbol gramatike, proglasiti ga za tekuci u izvedenoj sekvenci i pročitati prvi simbol iz ulaznog koda.
2. Ukoliko je tekuci simbol u izvedenoj sekvenci neterminalni simbol, zameniti ga desnom stranom prve smene na čijoj je levoj strani taj neterminalni simbol.
3. Ukoliko je tekuci simbol u izvedenoj sekvenci terminalni simbol jednak tekućem ulaznom simbolu, prihvati ga (preći na analizu sledećeg simbola).
4. Ukoliko je tekuci simbol u izvedenoj sekvenci terminalni simbol različit od tekućeg ulaznog simbola, poništiti dejstvo poslednje primenjene smene. Ukoliko postoji još koja smena za preslikavanje istog neterminalnog simbola, pokušati sa primenom sledeće smene, u suprotnom vratiti se još jedan korak nazad.
5. Ukoliko se vraćanjem došlo do startnog simbola i ne postoji više smena za njegovo preslikavanje, ulazni kod sadrži sintaksnu grešku.
6. Ukoliko nakon prihvatanja poslednjeg ulaznog simbola, ni u izvedenoj sekvenci nema neobradjenih simbola, kod je sintaksno korektan.

Top-down analiza – primer 3

- Neka je data gramatika G:

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

$$P:$$

$$S \rightarrow aAd \mid aB$$

$$A \rightarrow b \mid c$$

$$B \rightarrow ccd \mid ddc$$

- Proveriti da li reč $accd$ pripada jeziku definisanom gramatikom G.

Top-down analiza – primer 3

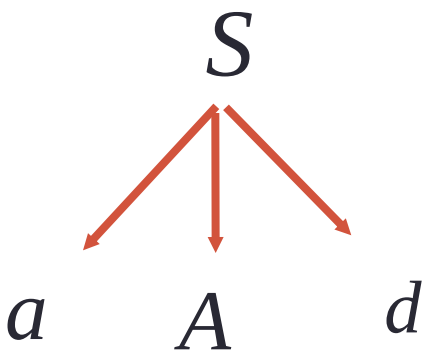
$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$

P:

$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



$\underline{a}\ c\ c\ d$

$\underline{S} \xrightarrow{S_1} \underline{a}Ad$

Top-down analiza – primer 3

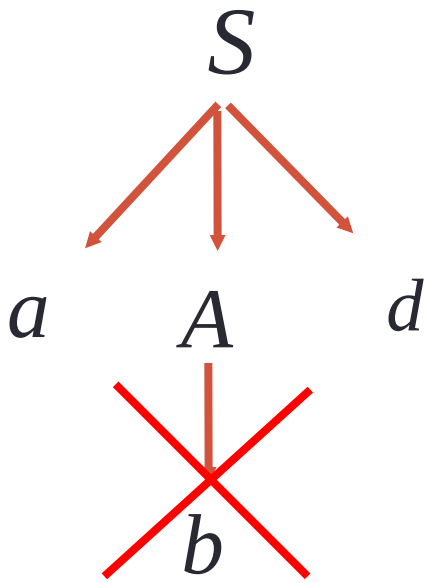
$$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$$

P:

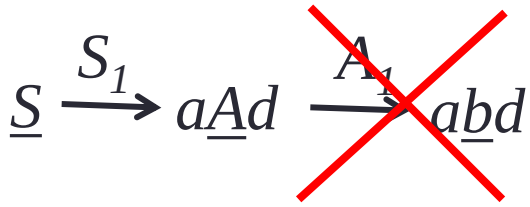
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



a c c d



Top-down analiza – primer 3

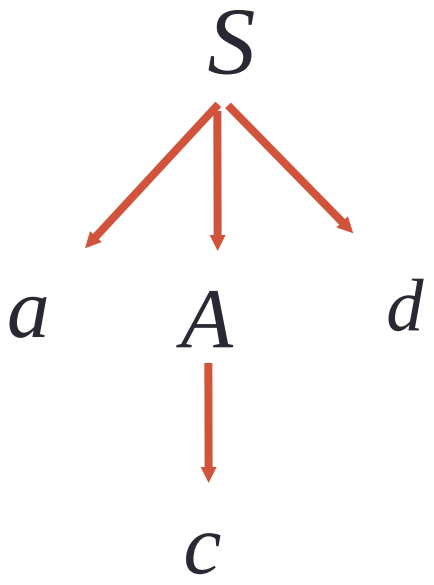
$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$

P:

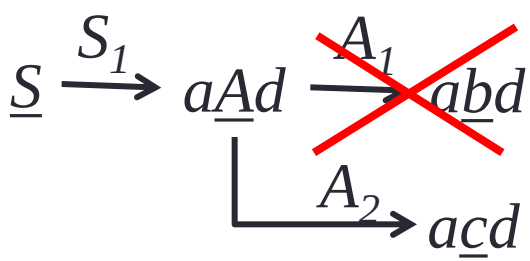
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



a c c d



Top-down analiza – primer 3

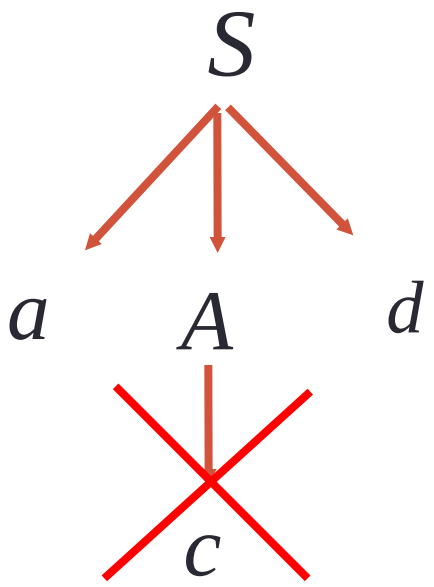
$$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$$

P:

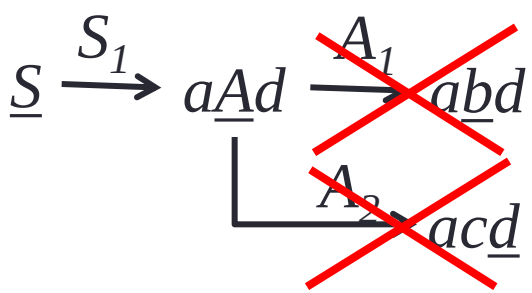
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



$a\ c\ \underline{c}\ d$



Top-down analiza – primer 3

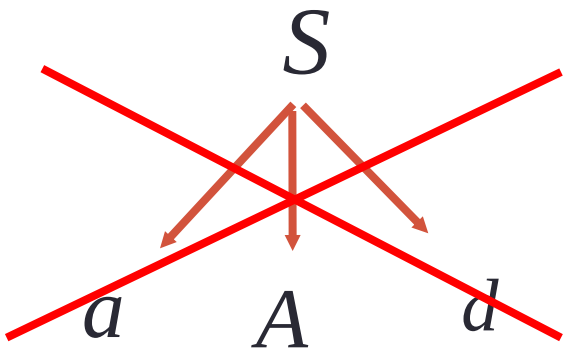
$$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$$

P:

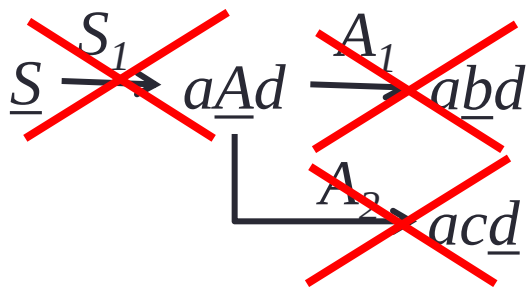
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



a c c d



Top-down analiza – primer 3

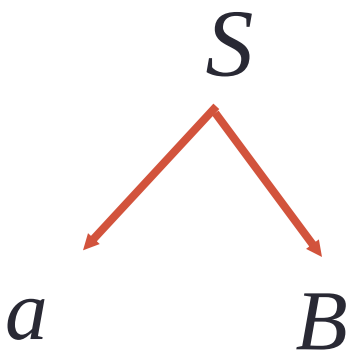
$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$

P:

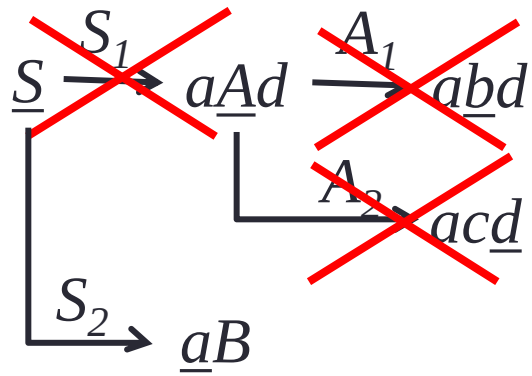
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



$\underline{a}\ c\ c\ d$



Top-down analiza – primer 3

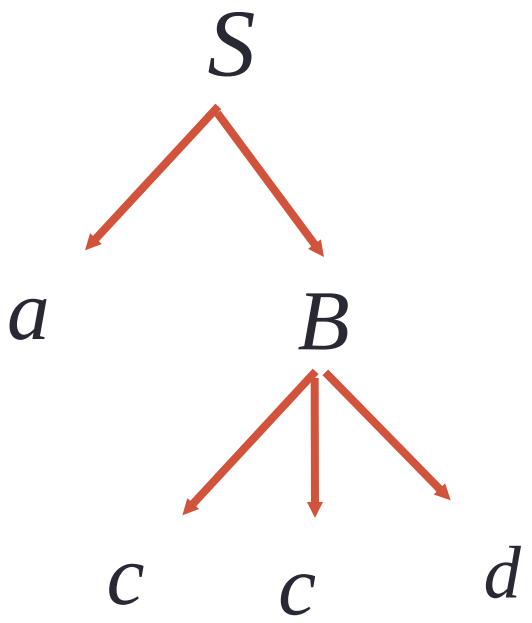
$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$

P:

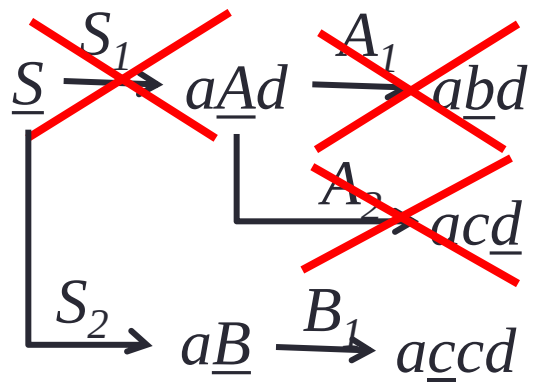
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



a c c d



Top-down analiza – primer 3

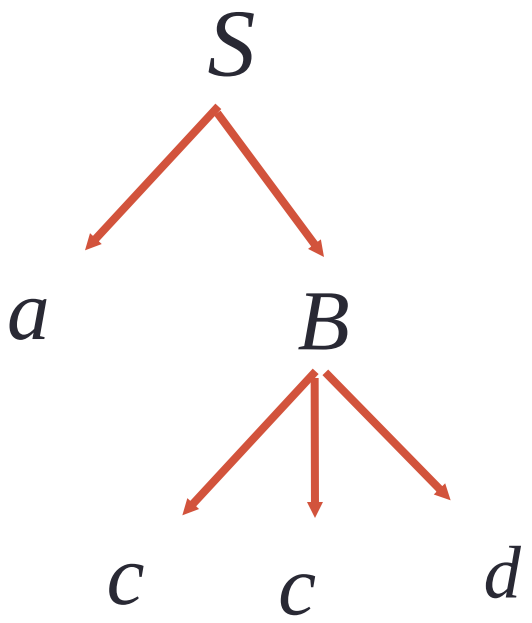
$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$

P:

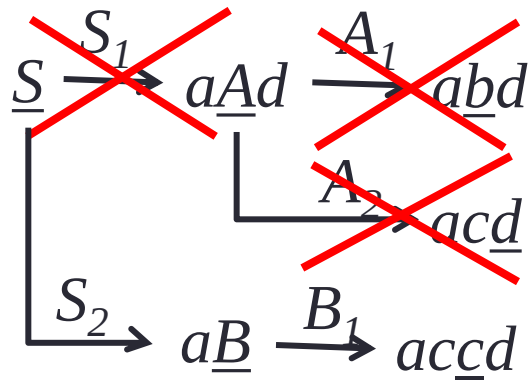
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



a c c d



Top-down analiza – primer 3

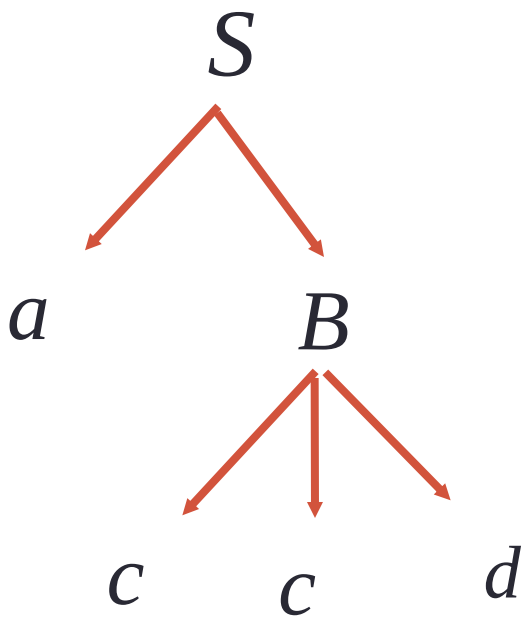
$G = (\{S, A, B\}, \{a, b, c, d\}, S, P)$

P:

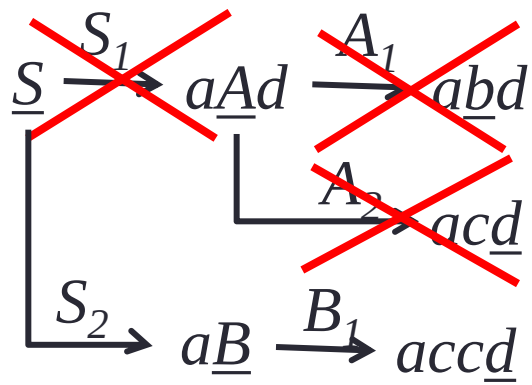
$S \rightarrow aAd \mid aB$

$A \rightarrow b \mid c$

$B \rightarrow ccd \mid ddc$



a c c d



Levo-rekurzivne smene

- Smene u kojima se simbol sa leve strane pojavljuje i na desnoj strani smene se nazivaju rekurzivnim smenama.

- Levo-rekurzivne smene su smene kod kojih je prvi simbol na desnoj strani jednak simbolu sa level strane smene

- Primeri:

$$\langle izraz \rangle \rightarrow \langle izraz \rangle + \langle term \rangle$$

$$\langle niz\_parametara \rangle \rightarrow \langle niz\_parametara \rangle \, , \langle parametar \rangle$$

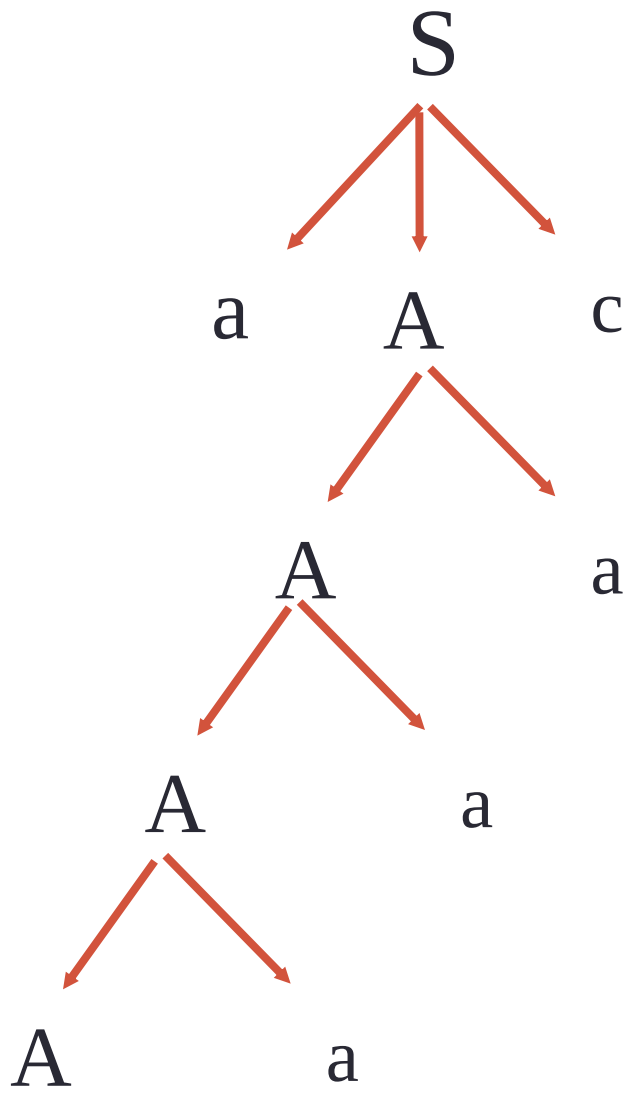
Problem levo-rekurzivnih smena

$$S \rightarrow aAc$$

$$A \rightarrow Aa$$

$$A \rightarrow \varepsilon$$

Rešenje: Elimimisati levo-rekurzivne smene!



Eliminacija leve rekurzije

- Deo gramatike sa rekurzivnim smenama:

$$A \rightarrow A\alpha \mid \beta$$

- Fraze izvedene iz neterminalnog simbola:

$$A \rightarrow A\alpha \rightarrow A\alpha\alpha \rightarrow \ldots \rightarrow S\alpha^{n} \rightarrow \beta\alpha^{n}$$

- Ekvivalentni skupovi smena:

$$A \rightarrow \beta A' \qquad A \rightarrow \beta \mid \beta A'$$

$$A' \rightarrow \alpha A' \mid \varepsilon \qquad A' \rightarrow \alpha \mid \alpha A'$$

Eliminicija leve rekurzije (opšti slučaj)

- Polazni skup smena:

$$A \longrightarrow A\alpha_1 \mid A\alpha_2 \mid \ldots \mid A\alpha_n \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_m$$

- Transformisani skup smena:

$$A \longrightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_m \mid \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_m A'$$

$$A' \longrightarrow \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_m \mid \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_m A'$$

ili:

$$A \longrightarrow \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_m A'$$

$$A' \longrightarrow \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_m A' \mid \varepsilon$$

Eliminicija leve rekurzije (primer)

- Polazni skup smena:

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid a$$

- Transformisani skup smena:

$$E \rightarrow T \mid TE'$$

$$E \rightarrow +T \mid +TE'$$

$$T \rightarrow F \mid FT'$$

$$T' \rightarrow *F \mid *FT'$$

$$F \rightarrow (E) \mid a$$

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid a$$

Osnovni bottom-up algoritam

1. Pročitati prvi simbol iz ulaznog koda i upisati ga u radni magacin.
2. Ponavljati sledeće korake dok se ne dođe do kraja ulaznog koda:
   a) Ukoliko se na vrhu radnog magacina nalazi desna strana neke smene, redukovati frazu sa vrha radnog magacina (tj. frazu sa vrha radnog magacina zameniti simbolom sa leve strane odgovarajuće smene);
   b) u suprotnom pročitati novi simbol iz ulaznog koda i smestiti ga u radni magacin.
3. Kada se dođe do kraja ulaznog koda, ukoliko je u radnom magacinu samo startni simbol gramatike, analiza je uspešno završena (kod je prihvaćen).
4. Kada se dođe do kraja ulaznog koda, ukoliko kod nije redukovan na startni simbol gramatike, a usput su izvršene neke redukcije, vratiti se na korak kada je izvršena poslednja redukcija, poništiti njeno dejstvo i preći na korak 2.
5. Ako se došlo do kraja ulaznog koda i pri tom nije izvršena ni jedna redukcija, analiza je završena neuspešno, ulazni kod je sintaksno neispravan.

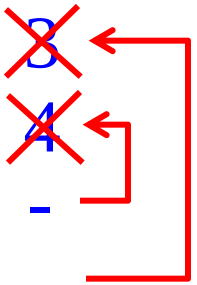
Analiza odozdo prema gore (Bottom Up)

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

P:

1. $S \rightarrow bA$
2. $S \rightarrow bB$
3. $B \rightarrow ab$
4. $B \rightarrow aaB$
5. $A \rightarrow aab$
6. $A \rightarrow aaA$

Ulazni niz	Magacin	Smene
$\underline{b}\ a\ a\ a\ a\ b\ \#$	-	-
$b\ \underline{a}\ a\ a\ a\ b\ \#$	b	-
$b\ a\ \underline{a}\ a\ a\ b\ \#$	ba	-
$b\ a\ a\ \underline{a}\ a\ b\ \#$	baa	-
$b\ a\ a\ a\ \underline{a}\ b\ \#$	$baaa$	-
$b\ a\ a\ a\ a\ \underline{b}\ \#$	$baaaa$	-
$b\ a\ a\ a\ a\ b\ \underline{\#}$	$baaaab$	~~3~~
$b\ a\ a\ a\ a\ b\ \underline{\#}$	$baaaB$	~~4~~
$b\ a\ a\ a\ a\ b\ \underline{\#}$	baB	-
$b\ a\ a\ a\ a\ b\ \underline{\#}$	$baaaB$	-
$b\ a\ a\ a\ a\ b\ \underline{\#}$	$baaaab$	5
$b\ a\ a\ a\ a\ b\ \underline{\#}$	$baaA$	6
$b\ a\ a\ a\ a\ b\ \underline{\#}$	bA	1
$b\ a\ a\ a\ a\ b\ \underline{\#}$	S	



Sintaksno stablo