

# Az informatika számítástudományi alapjai

## 6. előadás

Vaszi György

[vaszil.gyorgy@inf.unideb.hu](mailto:vaszil.gyorgy@inf.unideb.hu)

I. emelet 110-es szoba

# A múlt órán

- Környezetfüggetlen grammatikák szabályainak egyszerűsítése, normálformák
  - törlő szabályok kiküszöbölése
  - láncszabályok kiküszöbölése
- Chomsky féle normálforma, Cocke-Younger-Kasami algoritmus
- Pumpálási lemma környezetfüggetlen nyelvekre

# Grammatik „egyszerűsítők“ normálformára

- Törlek nullát :  $A \rightarrow \lambda$

• Minden közműveltségű ember számára  
konstrukció  $G_1$  úgy, hogy  $L(G) = L(G_1) - \{\lambda\}$   
és  $G_1$  nem tartalmaz törlek nullát.

• Alapötlet:

$$\left. \begin{array}{l} A \rightarrow BCD \\ B \rightarrow \lambda \\ C \rightarrow \lambda \end{array} \right\}$$

$$\longleftrightarrow \left\{ \begin{array}{l} A \rightarrow BCD \\ A \rightarrow CD \\ A \rightarrow BD \\ A \rightarrow D \end{array} \right.$$

## Láncszabályok leírás

Láncszabály :  $A \rightarrow B$        $A, B \in N$

Minden  $G$  környezetfüggetlen grammatikához konstruálható  $G_1$  úgy, hogy  $L(G_1) = L(G)$  és  $G_1$  **nem tartalmaz láncszabályokat.**

Alapötlet :

$$\left. \begin{array}{l} A \rightarrow B \\ B \rightarrow C \\ C \rightarrow XY \end{array} \right\} \Leftrightarrow \left\{ A \rightarrow XY \right.$$

# A gyakorlaton részletesebben láttuk:

- Minden  $G$  környezetfüggetlen grammatika átalakítható  $G'$ -vé úgy, hogy  $L(G)=L(G')-\{\lambda\}$ , de  $G'$  szabályai között **nincsenek törlő szabályok**.
- Minden  $G$  környezetfüggetlen grammatika átalakítható  $G'$ -vé úgy, hogy  $L(G)=L(G')$ , de  $G'$  szabályai között **nincsenek láncszabályok**.

# A múlt órán

- Környezetfüggetlen grammatikák szabályainak egyszerűsítése, normálformák
  - törlő szabályok kiküszöbölése
  - láncszabályok kiküszöbölése
- Chomsky féle normálforma, Cocke-Younger-Kasami algoritmus
- Pumpálási lemma környezetfüggetlen nyelvekre

## Chomskyféle normál - forma

- Egy grammatika Chomskyféle normálformában van, ha ~~nincs~~ csak

$$A \rightarrow BC \text{ és } A \rightarrow a$$

$$a \in \Sigma \\ A, B, C \in N$$

alakú szabályokat tartalmaz.

- Minden  $G$ -re van olyan  $G_1$  Chomskyféle normálformaú, hogy  $L(G_1) = L(G) - \{\lambda\}$ .
- Alapötlet:  
$$A \rightarrow BCDE \} \iff \begin{cases} A \rightarrow BX \\ X \rightarrow CY \\ Y \rightarrow DE \end{cases}$$

# Mire jó a Chomsky normálforma:

- Cocke-Younger-Kasami algoritmus
  - adott egy  $G$  környezetfüggetlen grammatika Chomsky normálformában
  - adott egy  $w$  sztring

Az algoritmus eldönti, hogy generálható-e a  $w$  sztring a  $G$  nyelvtannal. (Előállítja a lehetséges levezetések is.)



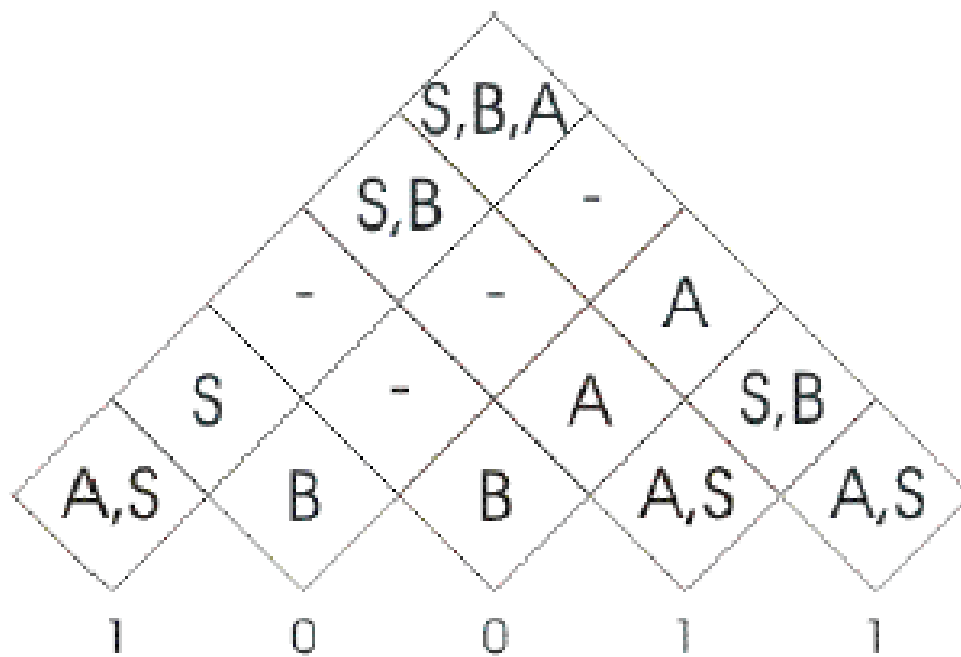
# Mire jó a Chomsky normálforma: Cocke-Younger-Kasami algoritmus

Tekintsük a következő grammatikát!

$G = (\{S, A, B\}, \{0, 1\}, S, H)$ , ahol  $H$  szabályai:

$\{S \rightarrow SA, S \rightarrow AB, A \rightarrow BS, B \rightarrow SA, A \rightarrow 1, S \rightarrow 1, B \rightarrow 0\}$

Bizonyítsuk be, hogy az 10011 szó benne van a grammatika által generált nyelvben,



*(miért kellett a Chomsky normálforma?)*

# A múlt órán

- Környezetfüggetlen grammatikák szabályainak egyszerűsítése, normálformák
  - törlő szabályok kiküszöbölése
  - láncszabályok kiküszöbölése
- Chomsky féle normálforma, Cocke-Younger-Kasami algoritmus
- Pumpálási lemma környezetfüggetlen nyelvekre

Pumpalemi lemma: Ha  $L$  környezetfüggetlen  
akkor létezik  $p$ , hogy ha  $s \in L$  és  $|s| > p$ ,  
akkor  $s$  felírható  $s = uvxyz$  alakba,  
ahol

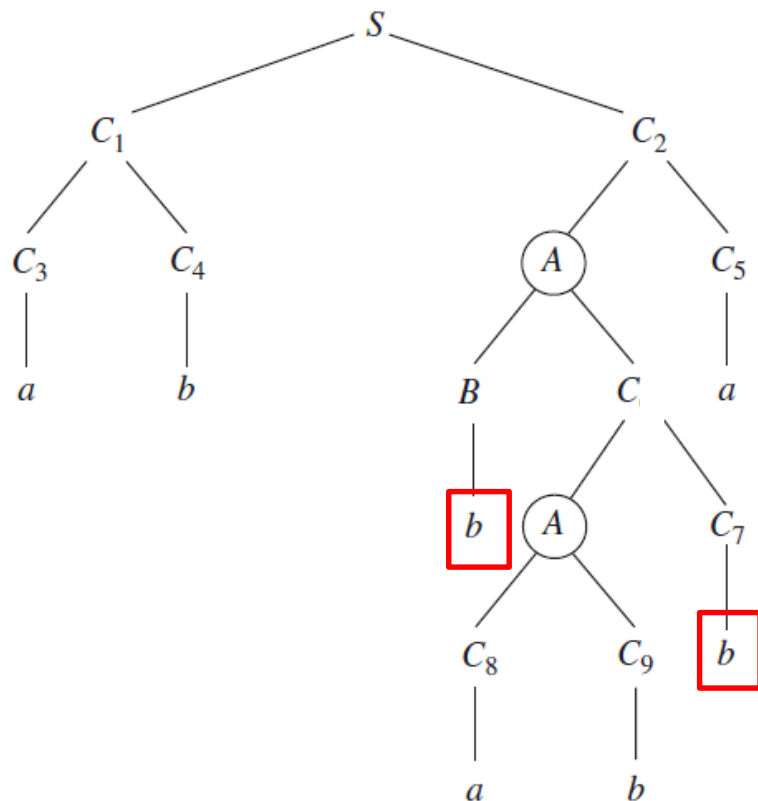
1.  $|vxy| \leq p$

2.  $|vy| > 0$

3.  $uv^i xy^i z \in L$  minden  $i \geq 0$ -ra

(környezetfüggetlen nyelv = környezetfüggetlen grammatikával  
generálható nyelv)

# Levezetési fák közelebbről

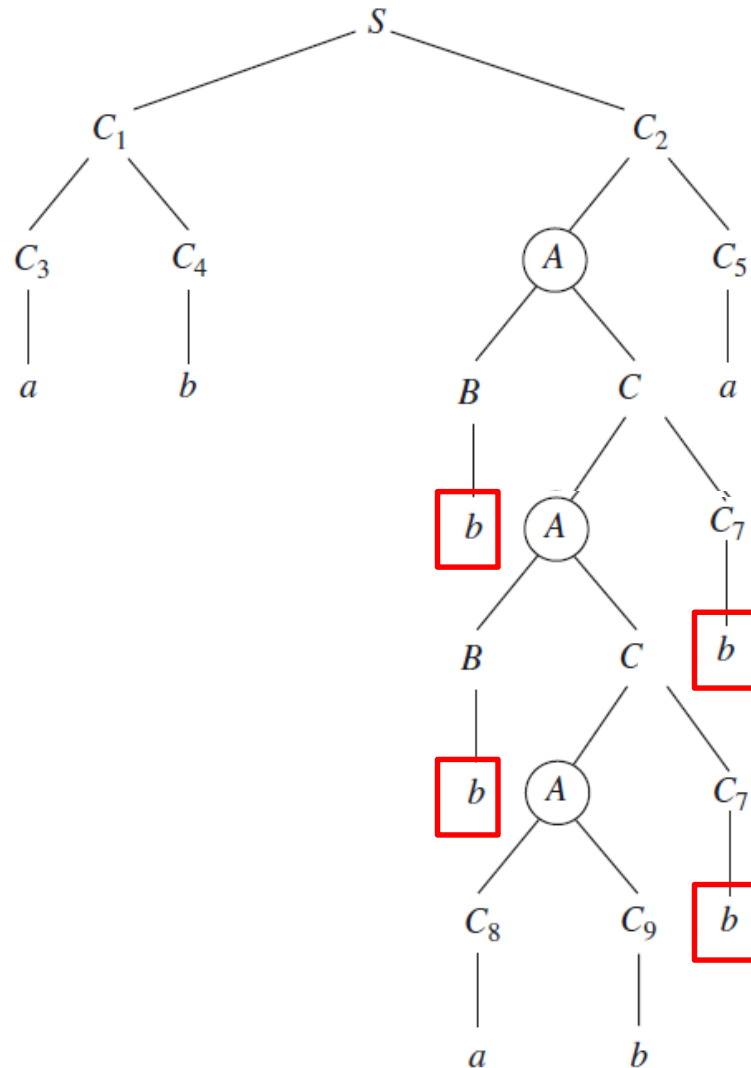


$u = (ab)(b)(ab)(b)(a)$

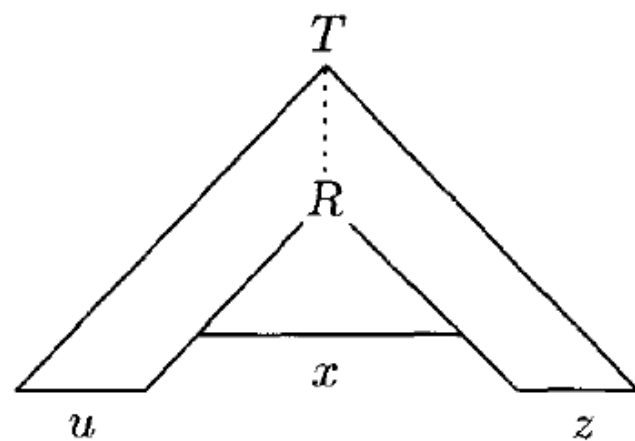
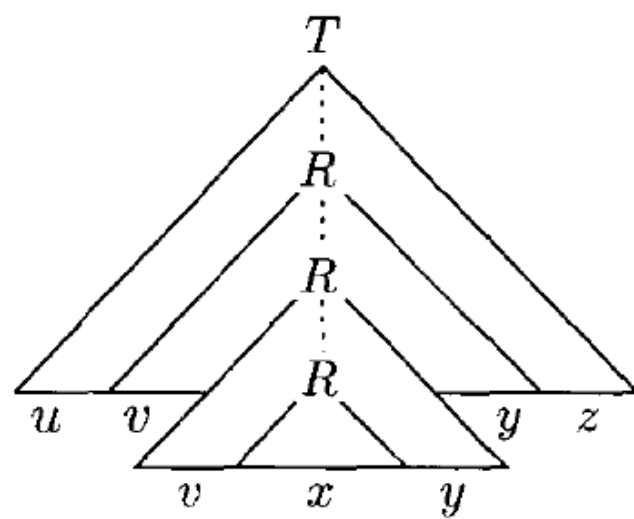
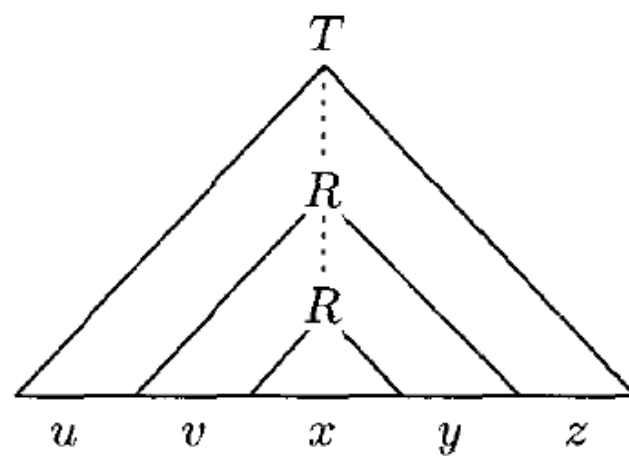
van  $A \rightarrow BC$ ,  $C \rightarrow AC_7$  és

$A \rightarrow C_8C_9$  szabály

(meg egy csomó más szabály)



$u = (ab)(b)(b)(ab)(b)(b)(a)$



## Reidmund

$L = \{a^n b^n c^n \mid n \geq 0\}$  wenn Vöngset-  
fingsetten. Wissen:

Ha  $L$  Vöngsetfingsetten uoh  $a$ ,  $b$ ,  $c$ ,  $p$ ,  
uoh under  $s \in L$ ,  $|s| > p$  u ~~te~~ pumpil-  
hete uoh. Veggjör  $s = a^p b^p c^p - t$ .

Nu leht pumpil-, leht  $L$  uoh  
leht Vöngsetfingsetten.

# A múlt órán

- Környezetfüggetlen grammatikák szabályainak egyszerűsítése, normálformák
  - törlő szabályok kiküszöbölése
  - láncszabályok kiküszöbölése
- Chomsky féle normálforma, Cocke-Younger-Kasami algoritmus
- Pumpálási lemma környezetfüggetlen nyelvekre

# MA

- Veremautomaták, környezetfüggetlen nyelvek elfogadása veremautomatával
- Determinisztikus veremautomaták, determinisztikus környezetfüggetlen nyelvek
- Szintaktikai elemzés/parsing veremautomatával
- Felülről lefelé/top-down elemzés, LL(k) grammatikák



Leggen G een cingrethjingssetten ugeloten, sa-  
derhai :  $S \rightarrow a S a \mid b S b \mid c$

geven

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

Undersøik vi, om  $L(G)$  er regulær.

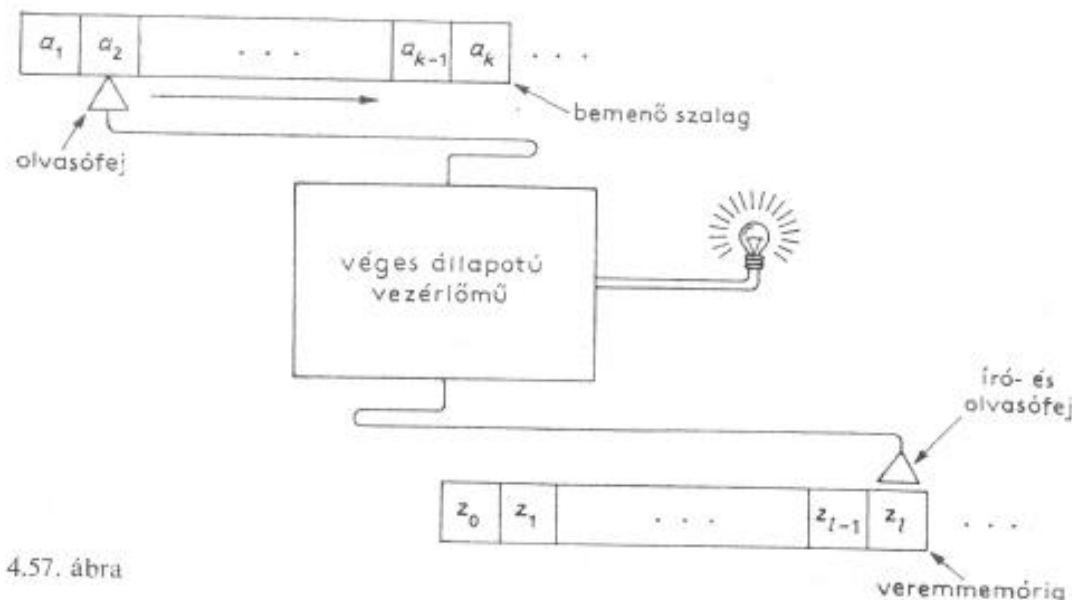


$L(G)$  er ikke et reg  
automatisk elfgaddi.

(Hvis det er ?)

# Veremautomata

A véges automataát egészíteni ki egy  
verem-memóriával  $\rightarrow$  veremautomata



4.57. ábra

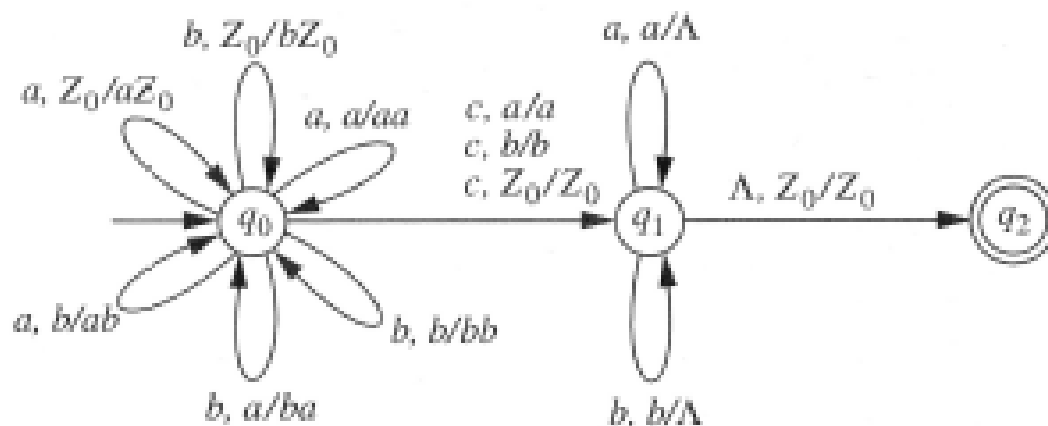
A' állapot átmenet:

(bemeneti  $a_i$  szimbólum,  $i$ -edik állapot, a verem teteje)  $\rightarrow$  (új állapot, a verem tetején lévő szimbólum cseréje)

Rélda, veremautomata

**Table 7.1** | Transition table for Example 7.1

Move number	State	Input	Stack symbol	Move(s)
1	$q_0$	$a$	$Z_0$	$(q_0, aZ_0)$
2	$q_0$	$b$	$Z_0$	$(q_0, bZ_0)$
3	$q_0$	$a$	$a$	$(q_0, aa)$
4	$q_0$	$b$	$a$	$(q_0, ba)$
5	$q_0$	$a$	$b$	$(q_0, ab)$
6	$q_0$	$b$	$b$	$(q_0, bb)$
7	$q_0$	$c$	$Z_0$	$(q_1, Z_0)$
8	$q_0$	$c$	$a$	$(q_1, a)$
9	$q_0$	$c$	$b$	$(q_1, b)$
10	$q_1$	$a$	$a$	$(q_1, \Lambda)$
11	$q_1$	$b$	$b$	$(q_1, \Lambda)$
12	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
(all other combinations)				none



Kezdő állapot:  $q_0$

Elfogadó állapot:  $q_2$

Kezdetben a  
verem alján  
lévő betű:

$Z_0$

Előrejel: a  $abcba$ ,  $ab$ ,  $acaa$  szavakra?

# Verevanszometer, definíció /1

$$M = (Q, T, \Gamma, q_0, z_0, \delta, F)$$

ahol:

$Q$  - állapothalmaz

$T$  - bemeneti ábécé

$\Gamma$  - kimeneti ábécé

$q_0 \in Q$  kezdő-állapot

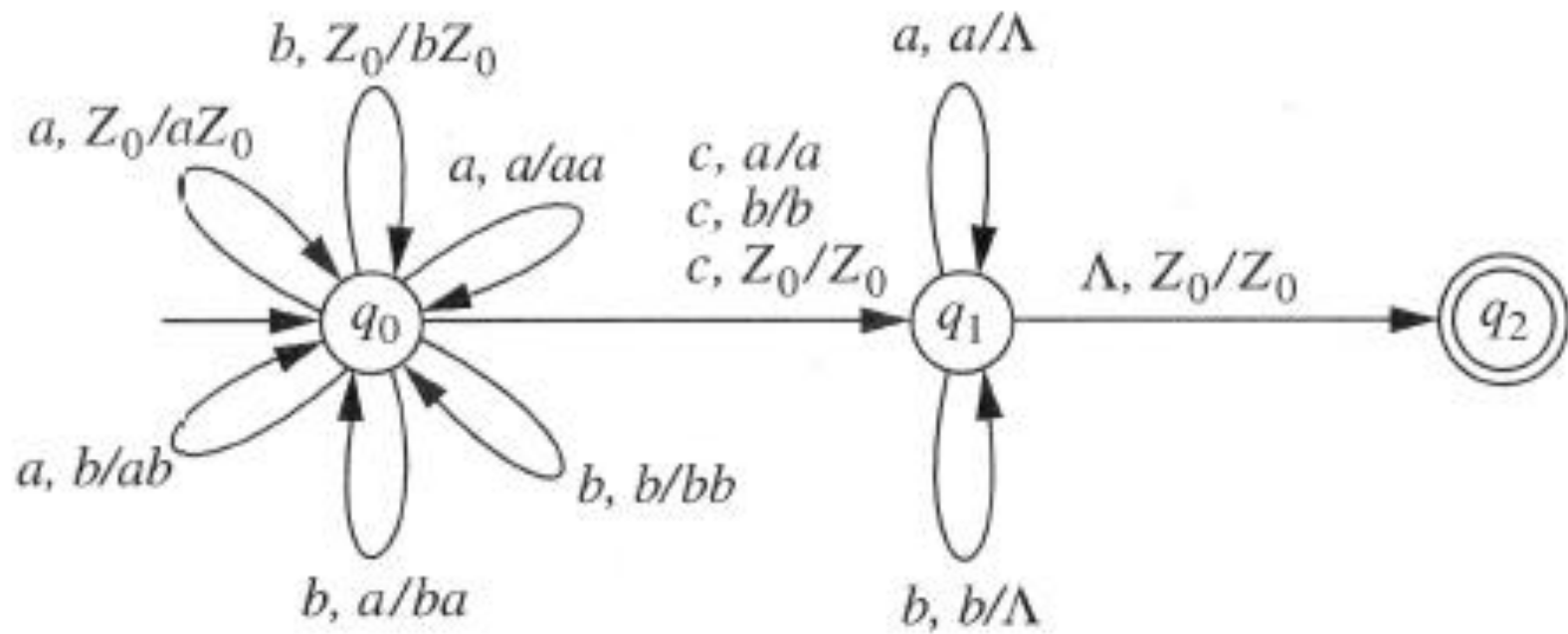
$z_0 \in \Gamma$  kezdeti verevanszót

$\delta$  - állapot átmenet reláció

$F \subseteq Q$  végállapotok halmaza



# Re'ldai'nd



$$\delta(q_0, b, a) = (q_0, ba)$$

a uerend  
leni b-t

$$\delta(q_1, b, b) = (q_1, \Lambda)$$

a uerend  
ei ueni b-t

# Konfiguration, Konfigurationen

Konfiguration :  $(p, x, \alpha)$   
                   $\uparrow \quad \uparrow \quad \nwarrow$   
                   $p \in Q \quad x \in T^* \quad \alpha \in \Gamma^*$

Konfiguration  
äquivalent :  $(p, x, \alpha) \rightarrow (q, y, \beta)$

- wa
- $x = a x', a \in T \cup \{\lambda\}$
  - $\alpha = \gamma \alpha', \gamma \in \Gamma, \alpha' \in \Gamma^*$
  - $\beta = \delta \alpha', \delta \in \Gamma^*$       $\gamma'$
  - $(q, \delta) \in \delta(p, a, x)$
-

Elfogadott szavak, né

kezdő

$$M = (Q, T, P, q_0, z_0, \delta, F)$$

Az  $M$ -alakkal elfogadott szavak:

$$L(M) = \{ w \in T^* \mid (q_0, w, z_0) \rightarrow \dots \rightarrow (q_f, \lambda, \alpha) \}$$

$\uparrow$   
 $q_f \in F$

(**Létezik** olyan konfiguráció átmenet sorozat, ami a kezdő konfigurációból a szó elolvasása közben elfogadó állapotot tartalmazó konfigurációba visz)



A veremautomaták által megadható nyelvek azok,  
amelyek környezetfüggetlen grammatikával  
generálhatóak.

Hogyan lehet felírni  
környezetfüggetlen  
nyelvi veremautomatát adni?

Például:  $S \rightarrow [S] / SS / \lambda$

Move Number	State	Input	Stack Symbol	Move
1	$q_0$	$\Lambda$	$Z_0$	$(q_1, SZ_0)$
2	$q_1$	$\Lambda$	$S$	$(q_1, [S]), (q_1, SS), (q_1, \Lambda)$
3	$q_1$	$[$	$[$	$(q_1, \Lambda)$
4	$q_1$	$]$	$]$	$(q_1, \Lambda)$
5	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
	(all other combinations)			none

Kezdőállapot:  $q_0$

Elfogadó állapot:  $q_2$

Kezdeti veremtartalom:  $Z_0$

(vegyünk egy jó és egy rossz példát)

Move Number	State	Input	Stack Symbol	Move
1	$q_0$	$\Lambda$	$Z_0$	$(q_1, SZ_0)$
2	$q_1$	$\Lambda$	$S$	$(q_1, [S]), (q_1, SS), (q_1, \Lambda)$
3	$q_1$	$[$	$[$	$(q_1, \Lambda)$
4	$q_1$	$]$	$]$	$(q_1, \Lambda)$
5	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
(all other combinations)				none

$(q_0, [ [] [] ], Z_0)$

$\vdash (q_1, [ [] [] ], SZ_0)$	$S$	
$\vdash (q_1, [ [] [] ], [S] Z_0)$	$\Rightarrow$	$[S]$
$\vdash (q_1, [ [] [] ], S] Z_0)$		
$\vdash (q_1, [ [] [] ], SS] Z_0)$	$\Rightarrow$	$[SS]$
$\vdash (q_1, [ [] [] ], [S] S] Z_0)$	$\Rightarrow$	$[ [S] S]$
$\vdash (q_1, [ [] [] ], S] S] Z_0)$		
$\vdash (q_1, [ [] [] ], ] S] Z_0)$	$\Rightarrow$	$[ [] S]$
$\vdash (q_1, [ [] ], S] Z_0)$		
$\vdash (q_1, [ [] ], [S] ] Z_0)$	$\Rightarrow$	$[ [] [S] ]$
$\vdash (q_1, [ [] ], S] ] Z_0)$		
$\vdash (q_1, [ [] ], ] ] Z_0)$	$\Rightarrow$	$[ [] [] ]$
$\vdash (q_1, [ ], ] Z_0)$		
$\vdash (q_1, \Lambda, Z_0)$	$\Rightarrow$	$[ [] ] []$
$\vdash (q_2, \Lambda, Z_0)$		

## A 'statische'

Gegeben  $G = (N, T, S, P)$ , konstruieren  $M$

$$M = (\{q_0, q_1, q_2\}, T, N \cup T \cup \{z_0\}, q_0, \delta, \{q_2\})$$

mit  $\delta$ :

- $\delta(q_0, \lambda, z_0) = \{(q_1, Sz_0)\}$
- $\delta(q_1, \lambda, A) = \{(q_1, \alpha) \mid A \rightarrow \alpha \in P\}$  wobei  $A \in N - z$
- $\delta(q_1, a, a) = \{(q_1, \lambda)\}$  wobei  $a \in T - z$
- $\delta(q_1, \lambda, z_0) = \{(q_2, z_0)\}$

$$\text{Geben } L(G) = L(M).$$

Azar:

te'el:

$L$  Einyeraltfiggellen  $\Rightarrow$  Van alga  $M$  ueremmar-  
fanta, dunnise

$$L = L(M)$$

Prüfungsausschuss:

haid an el'le' Gusskuss.

Azar:

Lékel:

$L$  Vöngretfíggeten  $\Rightarrow$  Van alga  $M$  ueremar-  
fanta, durne

$$L = L(M)$$

Prigiti öttel:

haid an elöle'ö' Gushkünd.

Iga-e en "vissafelli"i? Azor iga-e, uen  
här uel ueremautomata elfgadet gelsen  
Vöngret fíggeten? Iga, de set uen  
Prigiti öttel.

Lemma 1/2

$L$  környezetfüggetlen  $\Leftrightarrow$  Van olyan végesautomata  $M$ , hogy  
 $L = L(M)$

— \* —

Vagyis a környezetfüggetlen nyelvvel a végesautomata hasonló, szereket jellemez "mint a reguláris nyelvvel a véges automata."

Van-e lehetséges analógia  $\rightarrow$  pl. determinizmus

# MA

- Veremautomaták, környezetfüggetlen nyelvek elfogadása veremautomatával
- Determinisztikus veremautomaták, determinisztikus környezetfüggetlen nyelvek
- Szintaktikai elemzés/parsing veremautomatával
- Felülről lefelé/top-down elemzés, LL(k) grammatikák



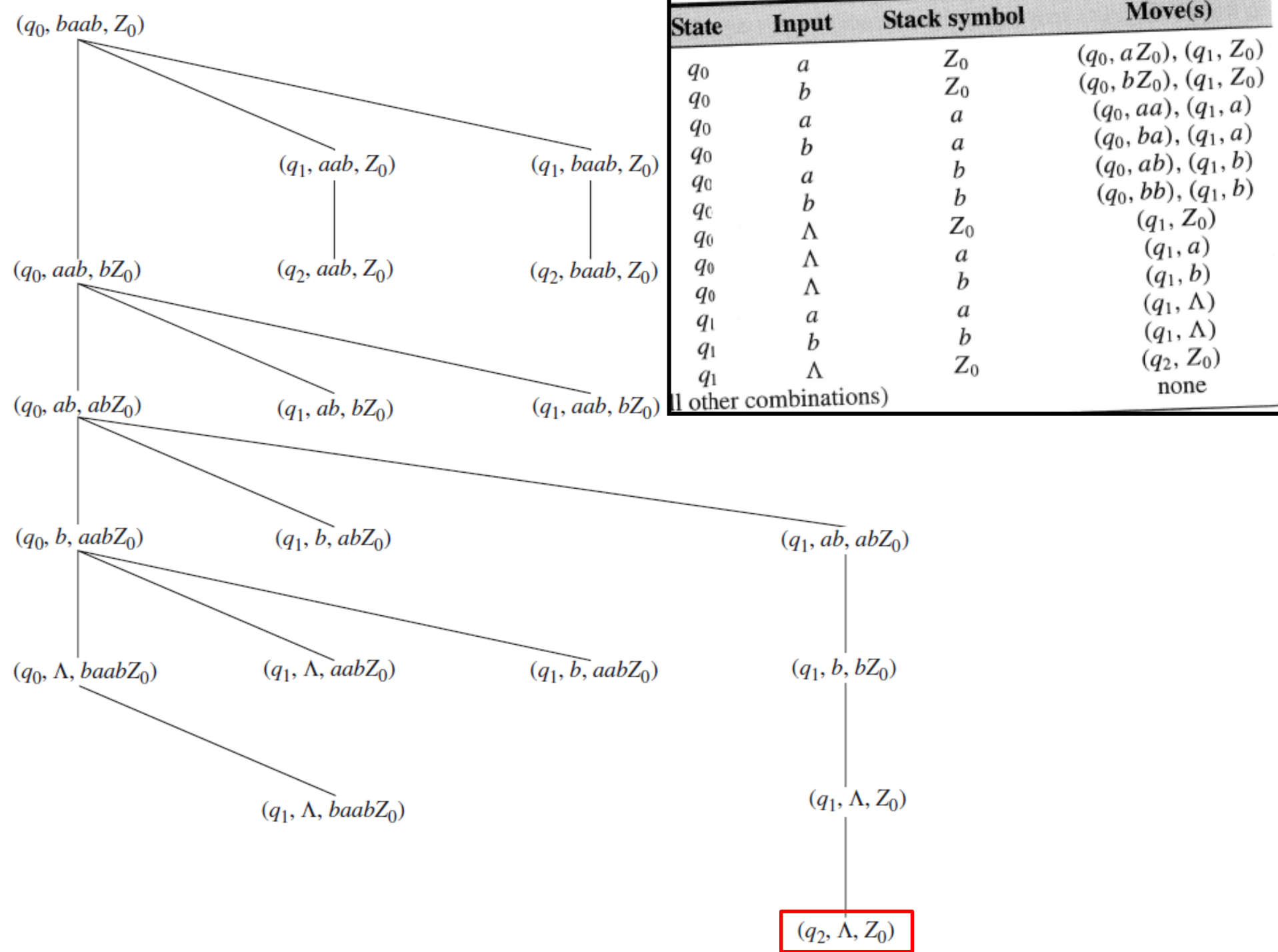
Re'la , palindro'na'z

$$M = (\{q_0, q_1, q_2\}, \{a, b\}, \{a, b, z_0\}, q_0, z_0, \delta, \{q_2\})$$

**Table 7.21** Transition table for  $M$

Move number	State	Input	Stack symbol	Move(s)
1	$q_0$	$a$	$Z_0$	$(q_0, aZ_0), (q_1, Z_0)$
2	$q_0$	$b$	$Z_0$	$(q_0, bZ_0), (q_1, Z_0)$
3	$q_0$	$a$	$a$	$(q_0, aa), (q_1, a)$
4	$q_0$	$b$	$a$	$(q_0, ba), (q_1, a)$
5	$q_0$	$a$	$b$	$(q_0, ab), (q_1, b)$
6	$q_0$	$b$	$b$	$(q_0, bb), (q_1, b)$
7	$q_0$	$\Lambda$	$Z_0$	$(q_1, Z_0)$
8	$q_0$	$\Lambda$	$a$	$(q_1, a)$
9	$q_0$	$\Lambda$	$b$	$(q_1, b)$
10	$q_1$	$a$	$a$	$(q_1, \Lambda)$
11	$q_1$	$b$	$b$	$(q_1, \Lambda)$
12	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
(all other combinations)				none

Lege a leenest baab. (irgiz kl a lehenest)  
 congruaitat!



## Deterministic veremautomata

- 1.) Bármely  $q \in Q$ ,  $a \in T \cup \{\lambda\}$ ,  $X \in \Gamma$  hívmossa,  
 $\delta(q, a, X)$  legfeljebb egy elem.
- 2.) Ha  $\delta(q, \lambda, X) \neq \emptyset$ , akkor  $\delta(q, a, X) = \emptyset$   
↑  
mivel  $a \in T$ -ra!  
( $a \neq \lambda$ )

Ha egy állapot-veremszimbólum  
párhoz van „lambda” (input olvasás nélküli)  
átmenet, akkor csak „lambda” átmenet van.

# Egy nemdeterminisztikus veremautomata

$$G = (\Sigma, S, P)$$

$$S \rightarrow aSa$$

$$S \rightarrow bSb$$

$$S \rightarrow c$$

$\longleftrightarrow$

$$M = (Q, \Sigma, \Gamma, q_0, \delta, F, q_2)$$

$$\delta(q_0, \lambda, z_0) = \{(q_1, Sz_0)\}$$

$$\delta(q_1, \lambda, S) = \{(q_1, aSa), (q_1, bSb)\}$$

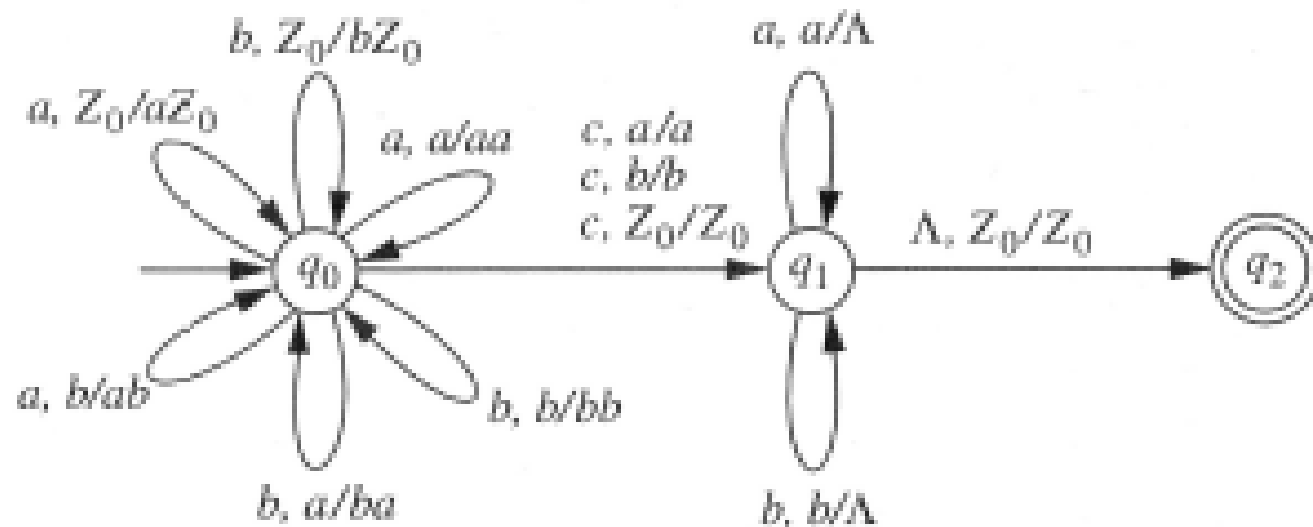
$$\delta(q_1, a, a) = \{(q_1, \lambda)\}$$

$$\delta(q_1, b, b) = \{(q_1, \lambda)\}$$

$$\delta(q_1, \lambda, z_0) = \{(q_2, z_0)\}$$

determinisztikus változat

Move number	State	Input	Stack symbol	Move(s)
1	$q_0$	$a$	$Z_0$	$(q_0, aZ_0)$
2	$q_0$	$b$	$Z_0$	$(q_0, bZ_0)$
3	$q_0$	$a$	$a$	$(q_0, aa)$
4	$q_0$	$b$	$a$	$(q_0, ba)$
5	$q_0$	$a$	$b$	$(q_0, ab)$
6	$q_0$	$b$	$b$	$(q_0, bb)$
7	$q_0$	$c$	$Z_0$	$(q_1, Z_0)$
8	$q_0$	$c$	$a$	$(q_1, a)$
9	$q_0$	$c$	$b$	$(q_1, b)$
10	$q_1$	$a$	$a$	$(q_1, \Lambda)$
11	$q_1$	$b$	$b$	$(q_1, \Lambda)$
12	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
(all other combinations)				none



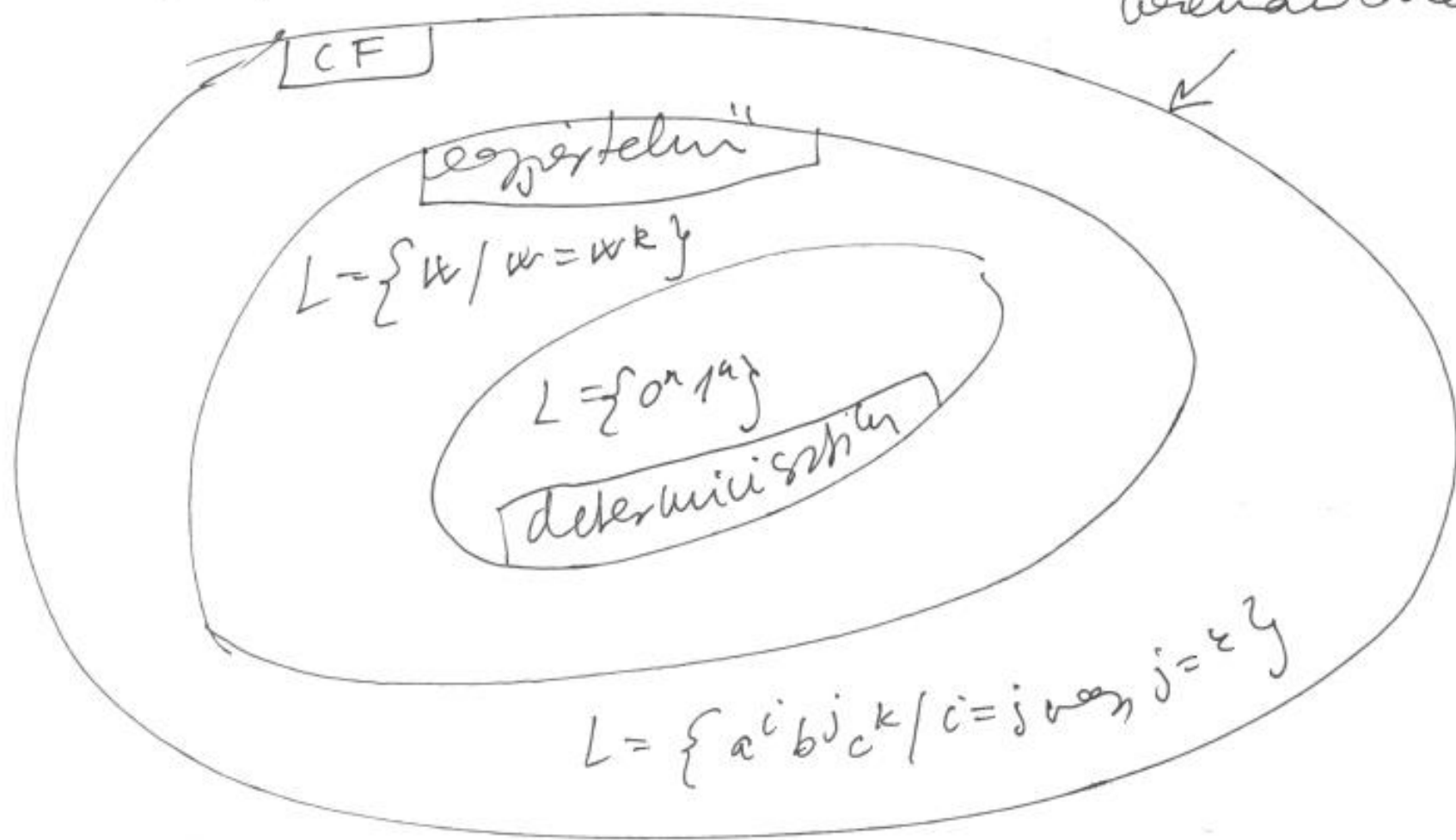
hírdőniz a  $abcba$ ,  $ab$ ,  $acaa$  kénen telen?

## Übungen:

A 'ltala' kor nen aga, kor under özyet -  
független nyelvhez lehet dekor minősíteni  
heremant ontat csikálni.

Például  $L = \{w \in \{a, b\}^* \mid w = w^R\}$  palindromák,  
a "körvonal" is megjelölve.

Wangreduktionssette  $\equiv$  van de deterministische  
vermenigvuldiging



## Aufgaben

- $L = \{0^n 1^n \mid n \geq 1\}$  = deterministisch (richtig?)
- $L = \{w \in \{a,b\}^* \mid w = w^R\}$  = nicht deterministisch, da  
"palindrom":  $S \rightarrow aSa \mid bSb$   
 $S \rightarrow a \mid b$
- $L = \{a^i b^j c^k \mid i=j \text{ oder } j=k\}$  = nicht  
palindrom



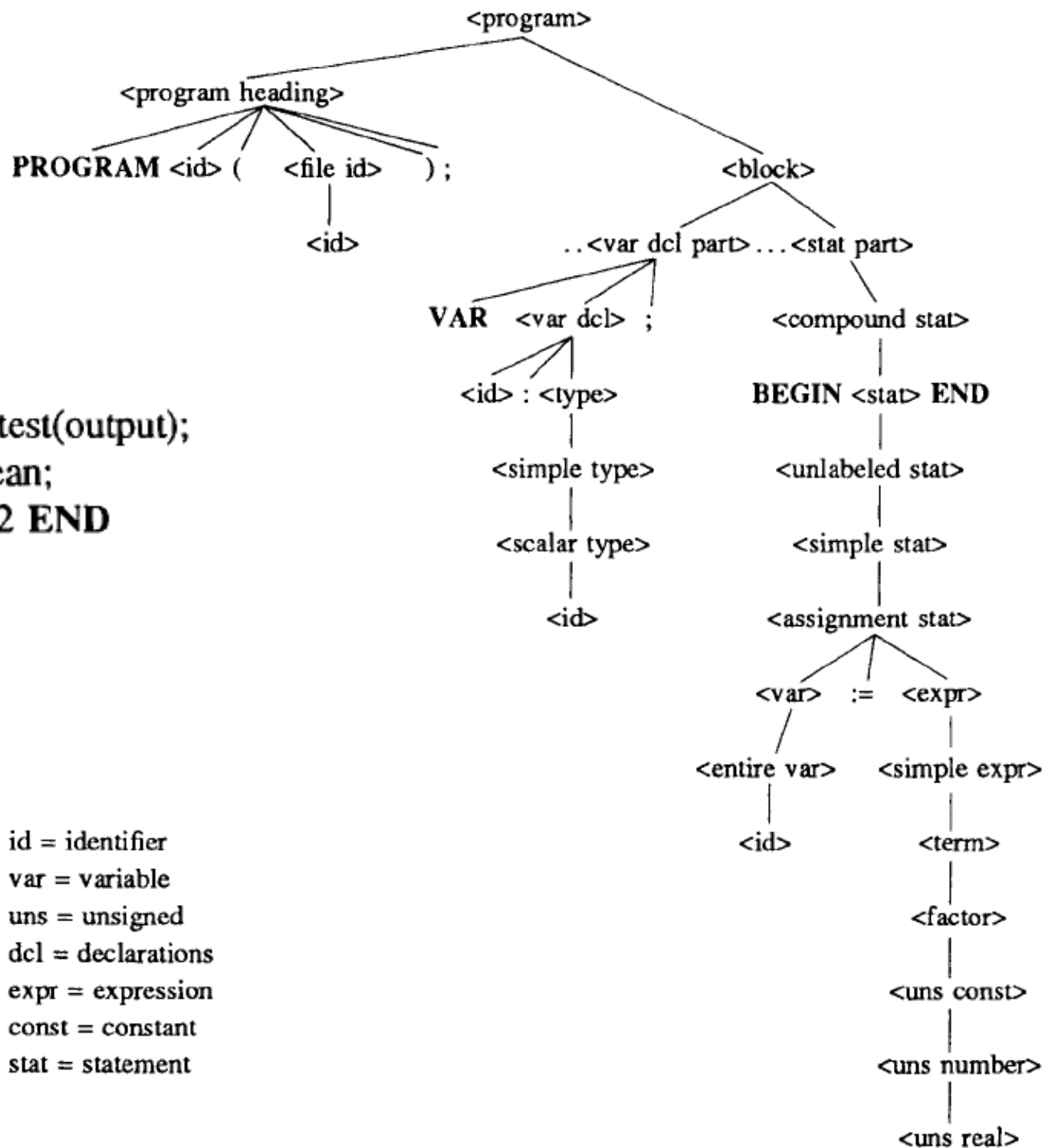


## En av egna sidorna bildas av:

- A programeri ydhet niutaxisa kishatd  
Gingret figgetla grammatikaral
- A compilereruet ellenoniri kull farditakar, es,  
kann a program niutaxi emileg kelly-e

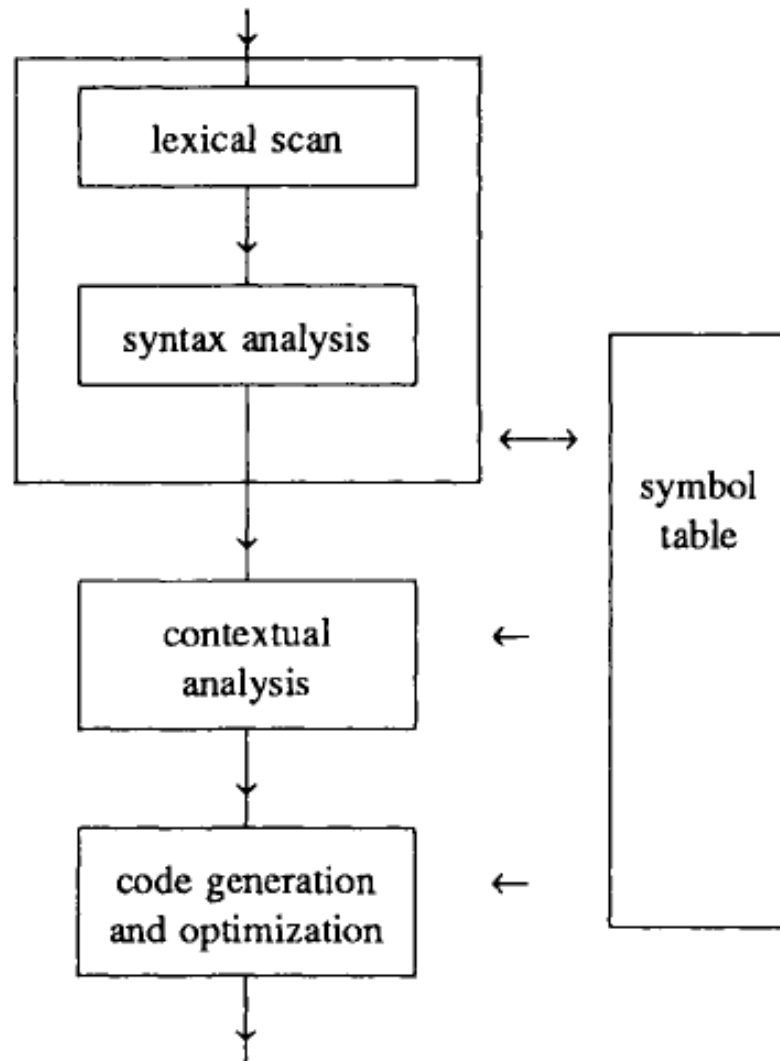
# MA

- Veremautomaták, környezetfüggetlen nyelvek elfogadása veremautomatával
- Determinisztikus veremautomaták, determinisztikus környezetfüggetlen nyelvek
- Szintaktikai elemzés/parsing veremautomatával
- Felülről lefelé/top-down elemzés, LL(k) grammatikák

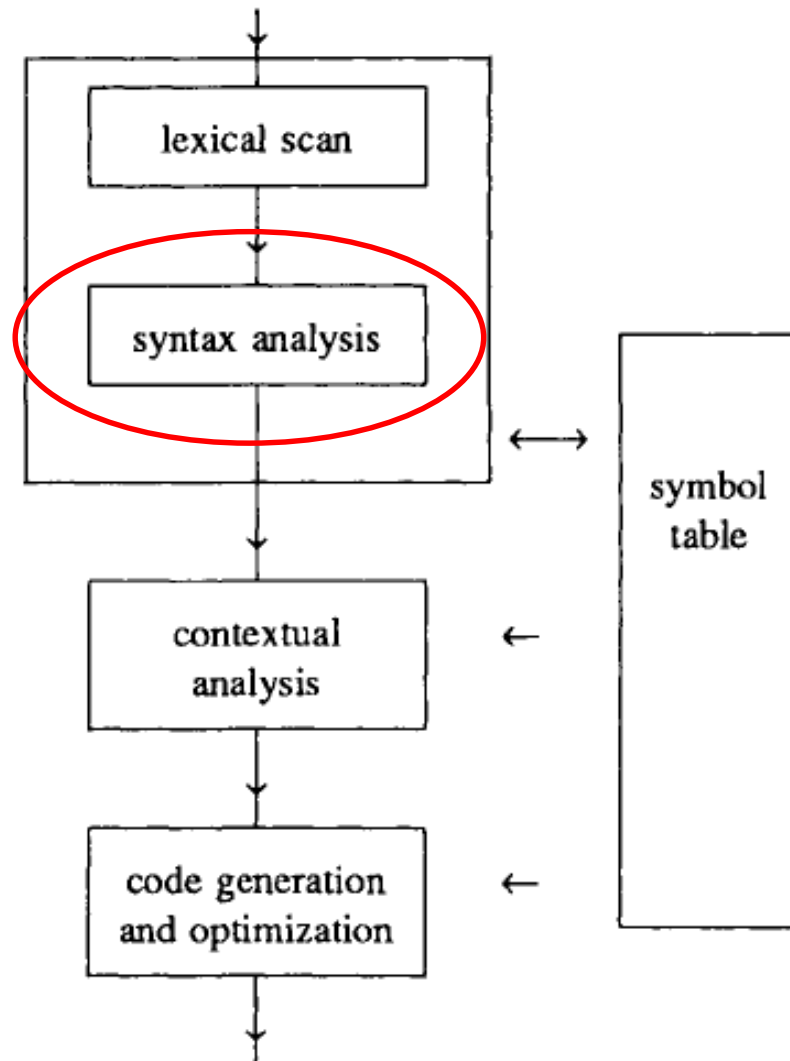


**PROGRAM** test(output);  
**VAR** b: boolean;  
**BEGIN** b := 2 **END**

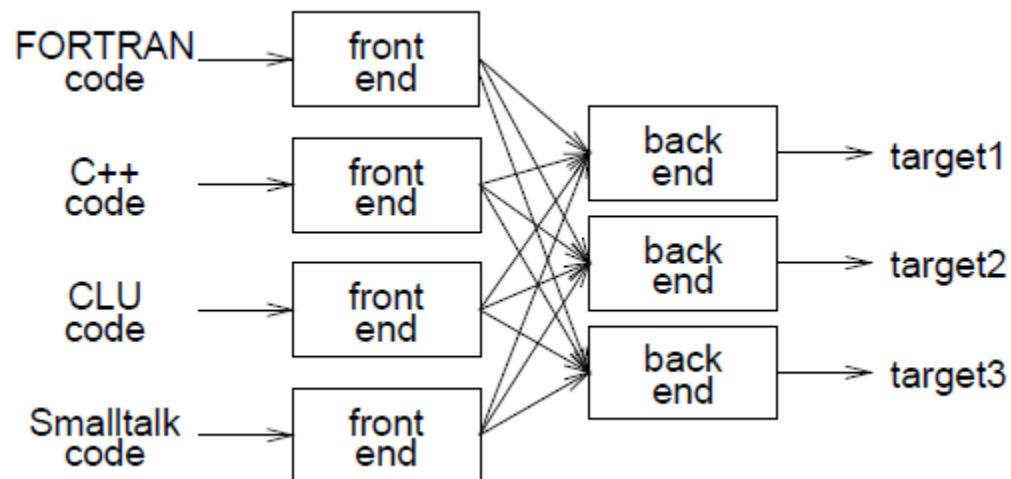
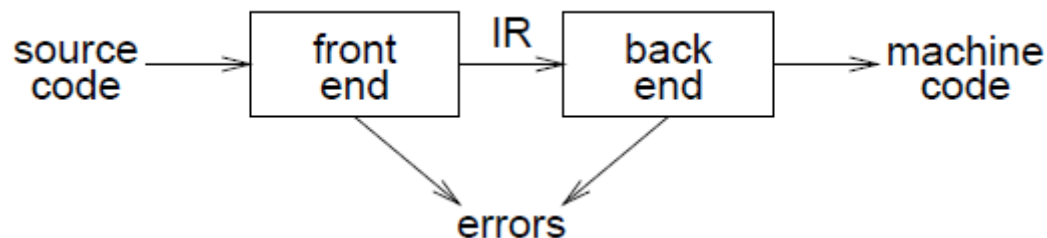
# Egy compiler/fordítóprogram moduljai



# Egy compiler/fordítóprogram moduljai

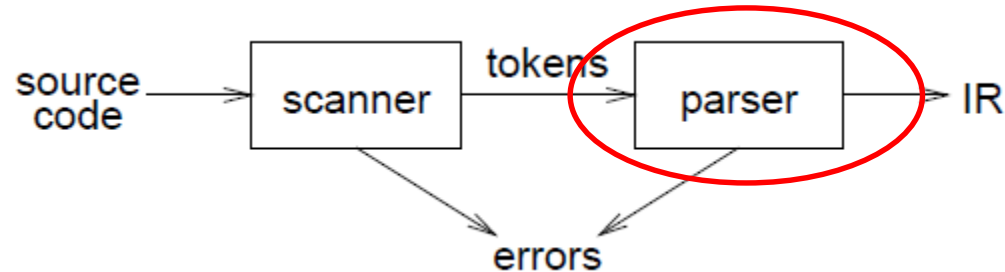


# Másképpen ábrázolva...



**IR:** Intermediate Representation, azaz “köztes reprezentáció”

# Minket az első fázis („*front end*”) érdekel:

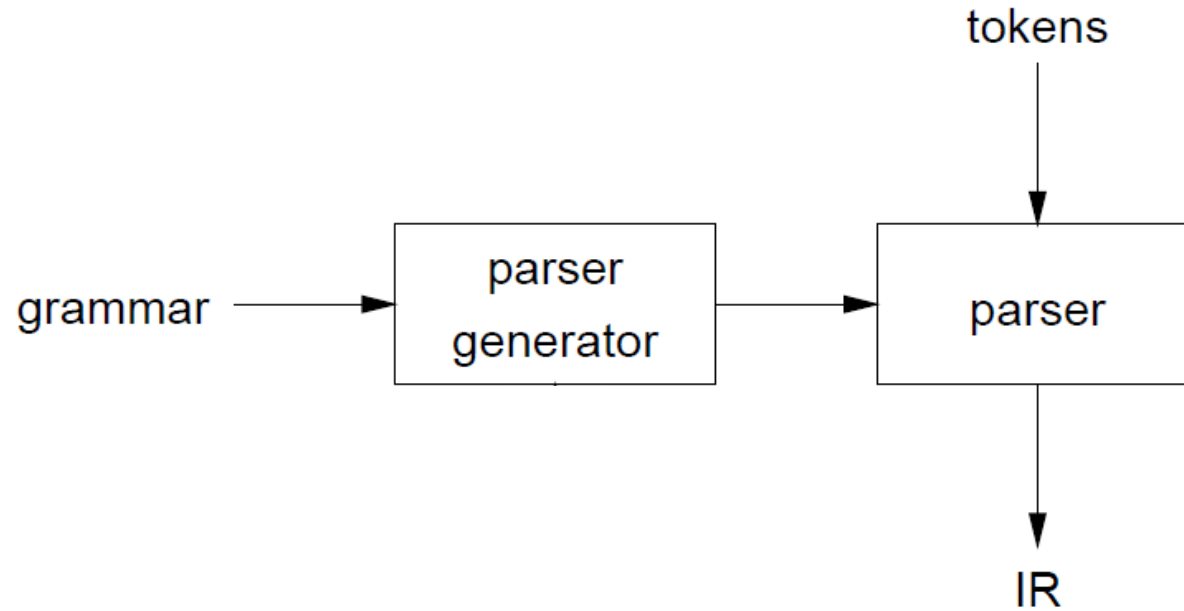


scanner:

- lexikális analízis (a program szövegét alkotó karakter sorozatban azonosítja a kulcsszavakat, változókat, műveleti jeleket, stb. – tokenek)
- (reguláris grammatikák, véges automaták)

parser:

- a token sorozat alapján a szöveg szerkezetét reprezentáló fát épít
- (környezetfüggetlen grammatikák, veremautomaták)
- egyebek...



A parser konstrukcióját szeretnénk automatizálni.



# MA

- Veremautomaták, környezetfüggetlen nyelvek elfogadása veremautomatával
- Determinisztikus veremautomaták, determinisztikus környezetfüggetlen nyelvek
- Szintaktikai elemzés/parsing veremautomatával
- Felülről lefelé/top-down elemzés, LL(k) grammatikák

# Simularkai elemzés: (Parsing)

## Levegő felől / top-down

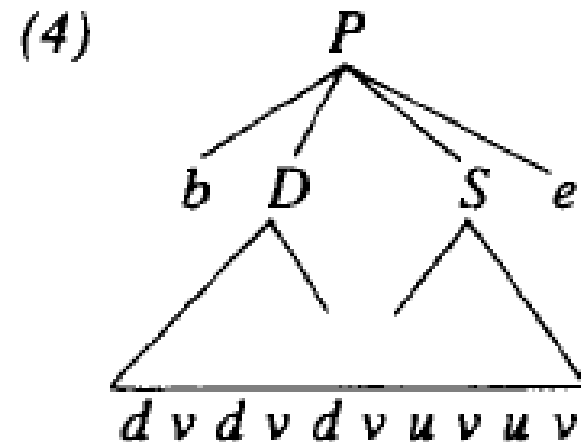
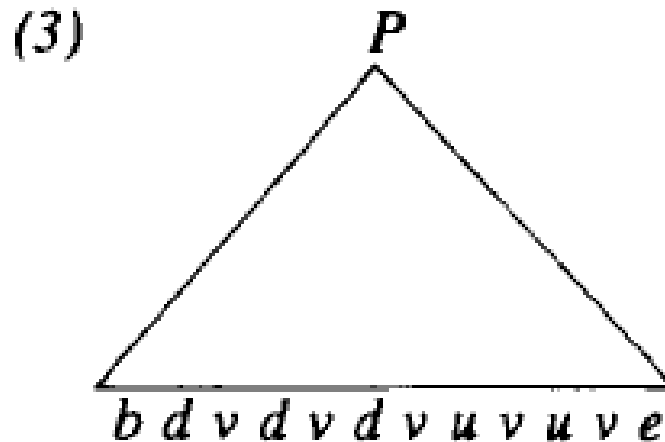
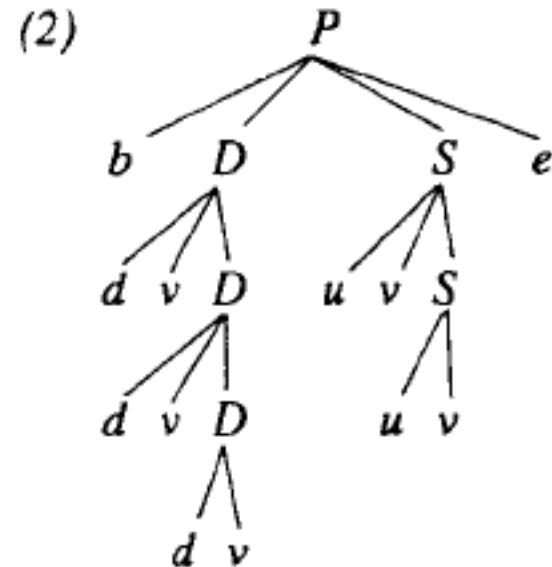
(1)  $P \rightarrow bDSe$

$D \rightarrow dvD$

$D \rightarrow dv$

$S \rightarrow uvS$

$S \rightarrow uv$



Például:  $S \rightarrow [S] / SS / \lambda$

Move Number	State	Input	Stack Symbol	Move
1	$q_0$	$\Lambda$	$Z_0$	$(q_1, SZ_0)$
2	$q_1$	$\Lambda$	$S$	$(q_1, [S]), (q_1, SS), (q_1, \Lambda)$
3	$q_1$	$[$	$[$	$(q_1, \Lambda)$
4	$q_1$	$]$	$]$	$(q_1, \Lambda)$
5	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
	(all other combinations)			none

Kezdőállapot:  $q_0$

Elfogadó állapot:  $q_2$

Kezdeti veremtartalom:  $Z_0$

(vegyünk egy jó és egy rossz példát)

Move Number	State	Input	Stack Symbol	Move
1	$q_0$	$\Lambda$	$Z_0$	$(q_1, SZ_0)$
2	$q_1$	$\Lambda$	$S$	$(q_1, [S]), (q_1, SS), (q_1, \Lambda)$
3	$q_1$	$[$	$[$	$(q_1, \Lambda)$
4	$q_1$	$]$	$]$	$(q_1, \Lambda)$
5	$q_1$	$\Lambda$	$Z_0$	$(q_2, Z_0)$
(all other combinations)				none

$(q_0, [ [] [] ], Z_0)$

$\vdash (q_1, [ [] [] ], SZ_0)$	$S$	
$\vdash (q_1, [ [] [] ], [S] Z_0)$	$\Rightarrow$	$[S]$
$\vdash (q_1, [ [] [] ], S] Z_0)$		
$\vdash (q_1, [ [] [] ], SS] Z_0)$	$\Rightarrow$	$[SS]$
$\vdash (q_1, [ [] [] ], [S] S] Z_0)$	$\Rightarrow$	$[ [S] S]$
$\vdash (q_1, [ [] [] ], S] S] Z_0)$		
$\vdash (q_1, [ [] [] ], ] S] Z_0)$	$\Rightarrow$	$[ [] S]$
$\vdash (q_1, [ [] ], S] Z_0)$		
$\vdash (q_1, [ [] ], [S] ] Z_0)$	$\Rightarrow$	$[ [] [S] ]$
$\vdash (q_1, [ [] ], S] ] Z_0)$		
$\vdash (q_1, [ [] ], ] ] Z_0)$	$\Rightarrow$	$[ [] [] ]$
$\vdash (q_1, [ ], ] Z_0)$		
$\vdash (q_1, \Lambda, Z_0)$	$\Rightarrow$	$[ [] ] []$
$\vdash (q_2, \Lambda, Z_0)$		

Hogyan lehetne ezt determinisztikus-  
cusan csinálni?

Példának:

Grammatika:

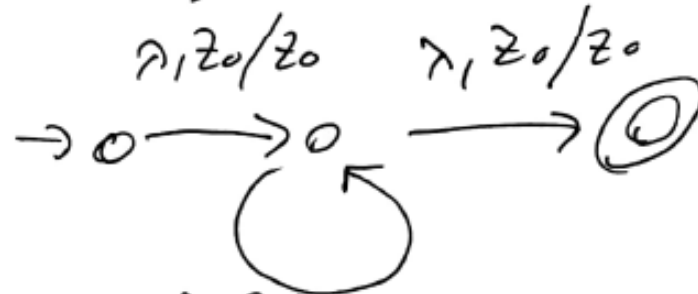
$S \rightarrow aAc$

$S \rightarrow b$

$A \rightarrow aSc$

$A \rightarrow b$

A szintaktikai elemő:



$\lambda, S/c aAc$

$\lambda, S/b$

$\lambda, A/aSc$

$\lambda, A/b$

$a, a/\lambda$

$b, b/\lambda$

$c, c/\lambda$

(előrejelzés)

LL(1) grammar, LL(1) elemző-  
felhő

$S \rightarrow aAc \mid b$

$A \rightarrow aSc \mid b$

	$a$	$b$	$c$
$S$	$S \rightarrow aAc$	$S \rightarrow b$	-
$A$	$A \rightarrow aSc$	$A \rightarrow b$	-

Hogyan lehet elhárítani a leveleket,  
leveleséit is?

	$a$	$b$	$c$
$S$	$S \rightarrow aAc$	$S \rightarrow b$	-
$A$	$A \rightarrow aSc$	$A \rightarrow b$	-

aabcc  
↑  
aabcc  
↑  
aabcc  
↑  
aabcc  
↑  
aabcc  
↑  
~~a~~~~a~~bcc  
↑  
aa**b**cc  
↑  
aabcc  
↑

20

520

$$\underline{aAc \in L} \Rightarrow S \rightarrow aAc$$

Ac 20

$$\boxed{\alpha \text{ SCC } z_0} \rightarrow A \rightarrow a \text{ Sc}$$

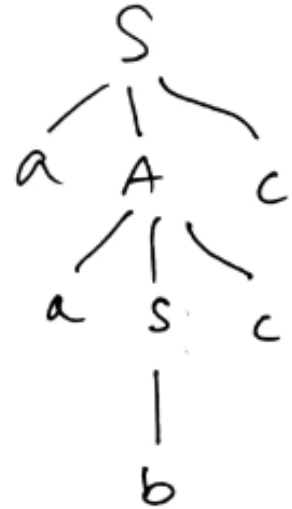
ScC20

bccz → s → b

CC 20

C 20

20



# von LL(3) grammatica

$$P \rightarrow b D S e$$

$$D \rightarrow d \vee D \mid \cancel{d} d \vee$$

$$S \rightarrow u \vee S \mid u \vee$$

<u>P</u>					
D					
S					

("P" a kezdőszimbólum)

Hogyan lehet eldönteni, hogy az adott szöveg LL(3) nyelvi-e?

?



At lehet alaktéri  
LL(1) grammatika

LL(3):

$$P \rightarrow bDSe$$

$$D \rightarrow dvD \mid dv$$

$$S \rightarrow uvS \mid uv$$

LL(1):

$$P \rightarrow bDSe$$

$$D \rightarrow dvE$$

$$E \rightarrow \varepsilon \mid D$$

$$S \rightarrow uvF$$

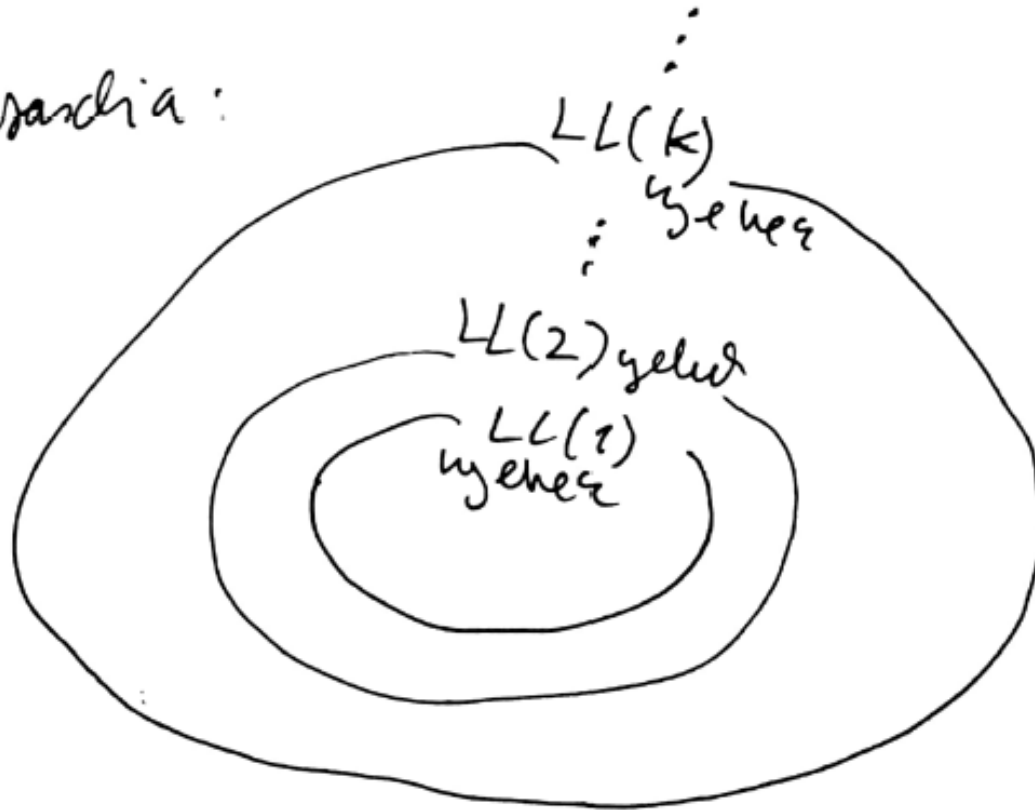
$$F \rightarrow \varepsilon \mid S$$

	$b$	$d$	$v$	$u$	$e$
$P$	$P \rightarrow bDSE$	-	-	-	-
$D$	-	$D \rightarrow dvE$	-	-	-
$E$	-	$E \rightarrow D$	-	$E \rightarrow \varepsilon$	-
$S$	-	-	-	$S \rightarrow uvF$	-
$F$	-	-	-	$F \rightarrow S$	$F \rightarrow \varepsilon$

A 'lalari' hoto' unider  $LL(k)$   
grammatica  $LL(1) - e'$  ?

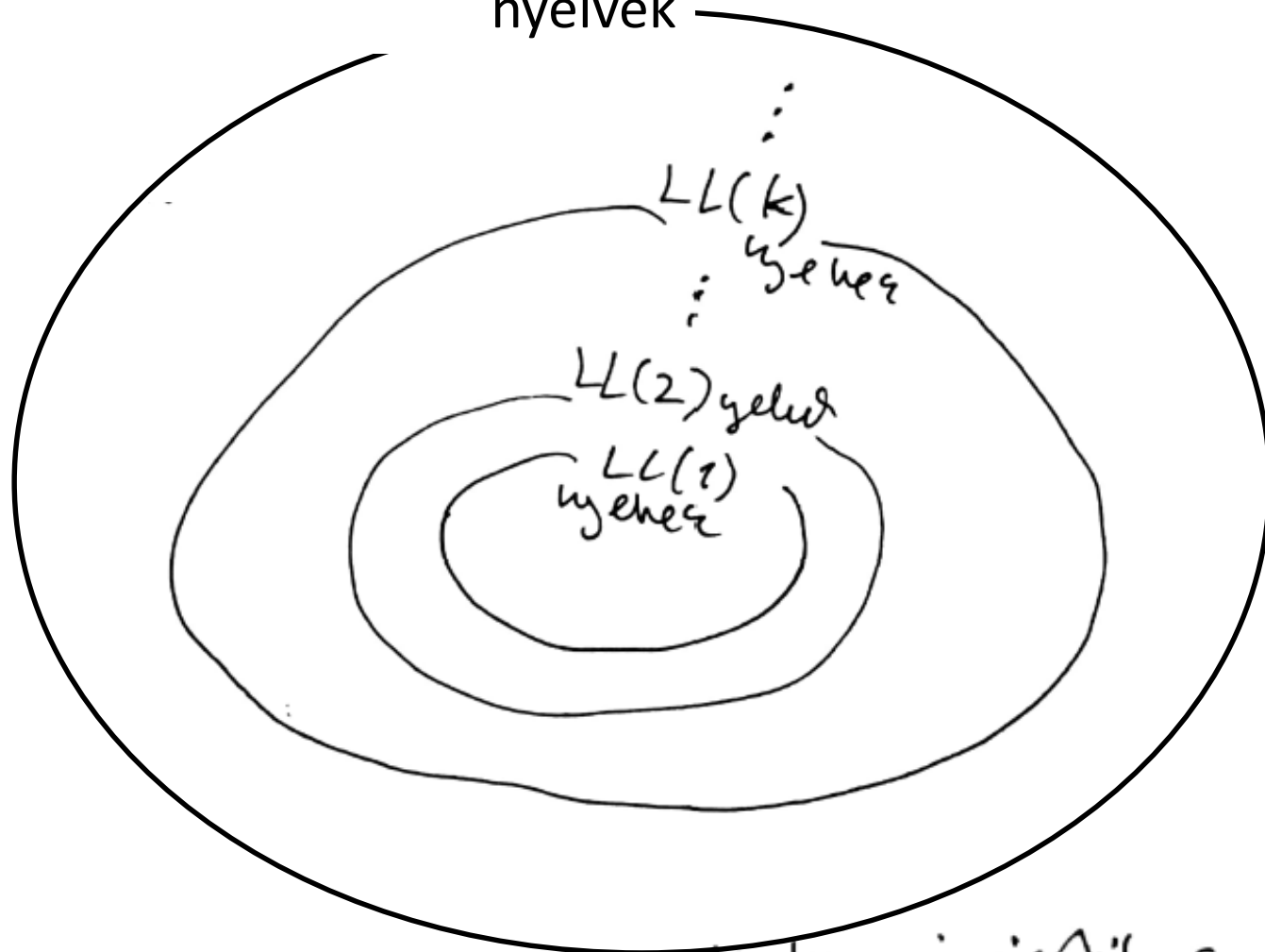
Nem.

Mylr - hierarchia:  
 $k$  reind



(Mian, ugh,  $LL(e)$  ugher?)

determinisztikus környezetfüggetlen  
nyelvek



$LL(\epsilon)$   
nyelvek



determinisztikus  
körny.-független  
nyelvek

# MA

- Veremautomaták, környezetfüggetlen nyelvek elfogadása veremautomatával
- Determinisztikus veremautomaták, determinisztikus környezetfüggetlen nyelvek
- Szintaktikai elemzés/parsing veremautomatával
- Felülről lefelé/top-down elemzés, LL(k) grammatikák





# A könyvekben

- J. Martin: 5.1 – 5.3 fejezet, 164 - 181. oldal
- Dömösi et al.: 7.5 fejezet, 156 – 160. oldal (*az üres veremmel elfogadó veremautomata nem kell*); 51. tétel a 161. oldalon; 163 – 174. oldal

