

Syntax Analysis → Tokenlar birleştirir.

CFG $\xrightarrow{\text{LL(1) SLE}}$ Stack Otomata

Si'ye göre Ni Üretimleri

form of si	Production Ni
ϵ	N_i
a	$N_i \rightarrow a$
$S_j S_k$	$N_i \rightarrow N_j N_k$
$S_j S_k$	$N_i \rightarrow N_j$ $N_i \rightarrow N_k$
S_j^+	$N_i \rightarrow N_j N_i$ $N_i \rightarrow$
S_j^+	$N_i \rightarrow N_j N_i$ $N_i \rightarrow N_j$
$S_j^?$	$N_i \rightarrow N_j$ $N_i \rightarrow$

Linear Grammar

$A \rightarrow x B x \rightarrow$ sağda lineer
 $A \rightarrow B x x \rightarrow$ solda lineer
 $S \rightarrow a S A \rightarrow$ lineer
 $S \rightarrow a S A x \rightarrow$ lineer değil

Düzgün Grammar

Düzgün olmalı ya ya sağda ya solda lineer olmalı gerekir.

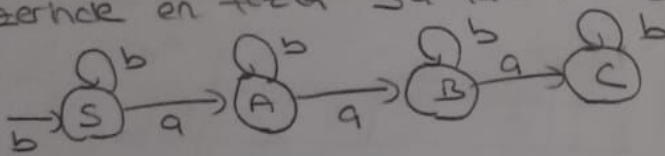
Türetim

$S \rightarrow aAB$

\swarrow
 A ile türetim \rightarrow önce sağ türetim
 \searrow
 B ile \rightarrow önce solda türetim

Grammar örnekleri

1) Üzerinde en fazla 3a ikeren düzgün grameri bul.



$S \rightarrow a A b S a$
 $A \rightarrow b A a B a$
 $B \rightarrow a C b B a$
 $C \rightarrow b C a$

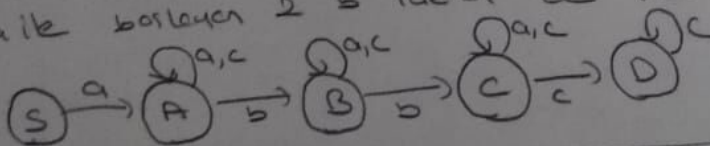
2) Gittir uzunlukta kiler

$S \rightarrow a a S b b S b a S a b S a$

3) $L = \{ a^n + (L a a)^n b^n \} b^{*}$ for $n \geq 1$

$\hookrightarrow S \rightarrow a a S B a$
 $B \rightarrow b B b$

4) ilk başlayan 2 b ikeren cc ile biten



$S \rightarrow a A$
 $A \rightarrow b B a A a A$
 $B \rightarrow b C a B b C B$
 $C \rightarrow a C c C c D$
 $D \rightarrow C$

5) $L = \{ a^n b^m : m \neq n \text{ GF} \}$

$S \rightarrow A S_1 S_2 B$
 $S_1 \rightarrow a S_1 b a$
 $A \rightarrow a A a$

Belirsizlik

→ Gramerde bazı stringler için bir kaç tane farklı syntax ağacı oluyorsa gramer belirsizdir.
 $a \oplus b \oplus c \rightarrow$ soldan sağa değerlendiriliyor \oplus soldan birleşmeli: $(a \oplus b) \oplus c$

Belirsiz Gramerleri Yeniden Yazma

$$\left. \begin{array}{l} E \rightarrow E \oplus E \\ E \rightarrow \text{num} \end{array} \right\} \left. \begin{array}{l} E \rightarrow E \oplus E' \\ E \rightarrow E' \\ E' \rightarrow \text{num} \end{array} \right\} \begin{array}{l} \text{sol} \\ \text{rekürsif} \end{array}$$

$$\left. \begin{array}{l} E \rightarrow E \oplus E' \\ E \rightarrow E' \\ E' \rightarrow \text{num} \end{array} \right\} \begin{array}{l} \text{sağ} \\ \text{rekürsif} \end{array}$$

Birleşme özelliği yoksa (non rekürsif)

$$\left. \begin{array}{l} E \rightarrow E' \oplus E' \\ E \rightarrow E' \\ E' \rightarrow \text{num} \end{array} \right\}$$

$$\left. \begin{array}{l} E \rightarrow E \oplus E' \\ E \rightarrow E' \oplus E \\ E \rightarrow E' \\ E' \rightarrow \text{num} \end{array} \right\} \begin{array}{l} \text{Belirsiz} \\ \text{gramer. (Çünkü eşit öncelikli operatörler} \\ \text{sağ ve soldan birleşiyorlar.)} \end{array}$$

Ayrıştırma

- 1) Yukarıdan aşağıya \rightarrow tahmin ederek $\left. \begin{array}{l} \text{Genişlik öncelikli} \\ \text{Derinlik öncelikli} \end{array} \right\} \begin{array}{l} \text{LL} \rightarrow \text{soldan türetim} \\ \rightarrow \text{deterministik değildir.} \end{array}$
- 2) Aşağıdan yukarıya \rightarrow string ile uyumlu sağ tarafa bakarak $\left. \begin{array}{l} \text{Genişlik öncelikli} \\ \text{Derinlik öncelikli} \end{array} \right\} \begin{array}{l} \text{LR} \rightarrow \text{soldan türetim} \\ \rightarrow \text{deterministik değildir.} \end{array}$

Bir Yukarıdan Aşağıya Ayrıştırma Otomatik Oluşturma

PDA \rightarrow CFG'leri tanıyan makine modelidir.

LL \rightarrow Aşağı itme otomati kullanır.

Aşağı itme Otomati

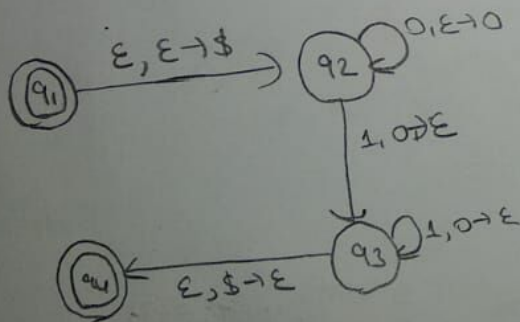
$$Q = \{q_1, q_2, q_3, q_4\},$$

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

$$\Gamma = \{0, \$\},$$

$$F = \{q_1, q_4\}$$

Input	0			1			ϵ		
Stack	0	\$	ϵ	0	\$	ϵ	0	\$	ϵ
q_1									
q_2			$\{q_2, 0\}$			$\{q_3, \epsilon\}$			
q_3						$\{q_3, \epsilon\}$			
q_4									$\{q_4, \epsilon\}$



0 \rightarrow E \rightarrow 0 okleme
 $\epsilon \rightarrow 0 \rightarrow 0$ okleme
 $\epsilon \rightarrow \$ \rightarrow$ başlangıç
 $\$ \rightarrow \epsilon \rightarrow$ bitiş

b \rightarrow c \rightarrow dönüş ve değiştir.
 $\epsilon \rightarrow c \rightarrow$ okla
 $b \rightarrow \epsilon \rightarrow$ çıkar
 $\epsilon \rightarrow \epsilon \rightarrow$ değişiklik yapma

baş \rightarrow (q1)
 $\$ \rightarrow$ başlangıç

D'A'm Tanıdışı Dil

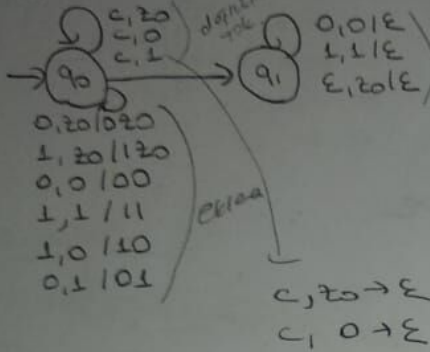
$a, b \rightarrow c \rightarrow$ stacke yazılan

inputtan okunan

stackten okunan

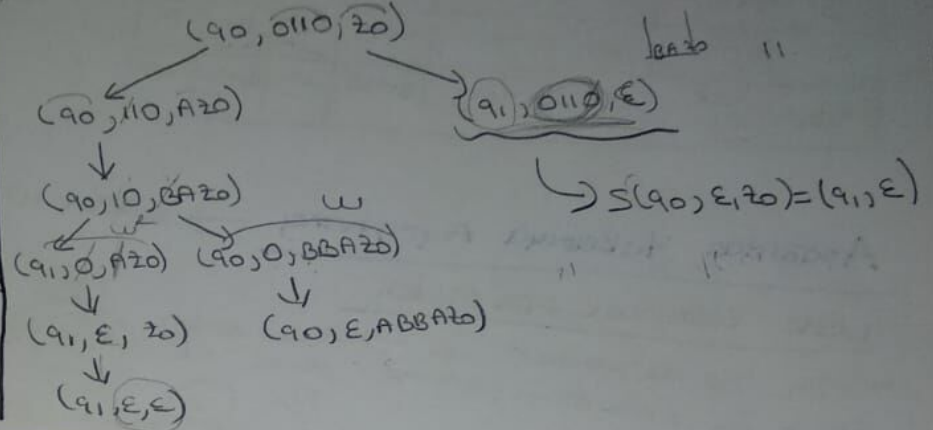
wcwr

$P: S \rightarrow OSO | 1S1 | c$



wcwr

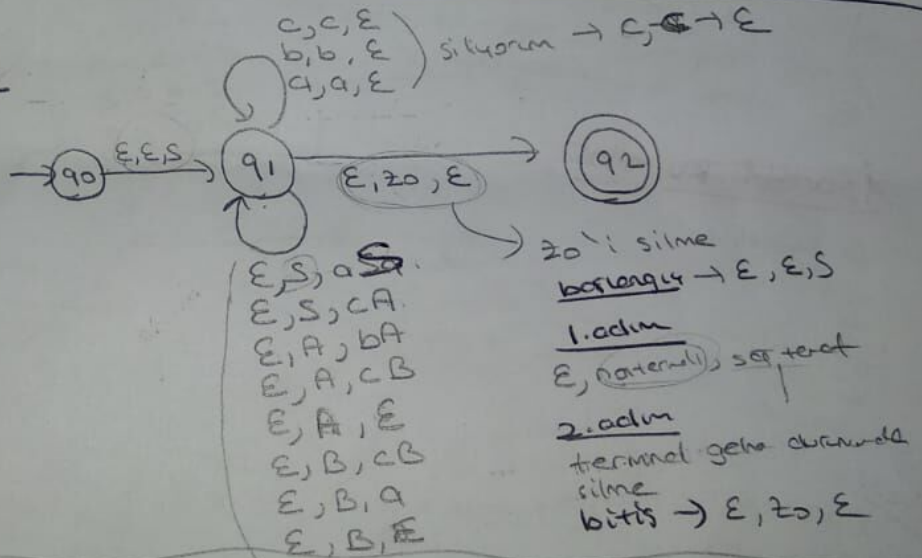
her 0 için A
her 1 için B



LL(k) Ayırtırma

LL(k)'nin PDA modeli

$S \rightarrow aS | cA$
 $A \rightarrow bA | cB | \epsilon$
 $B \rightarrow cB | a | \epsilon$



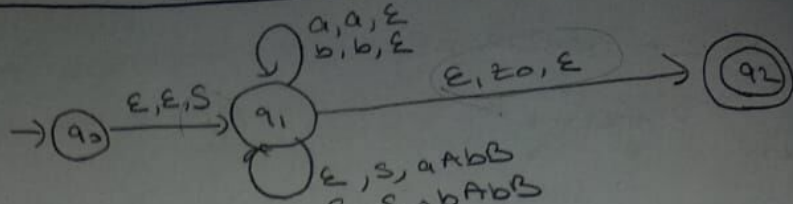
$\Delta = \{(p, e, e), (q, s)\}, T_1$
 $\{(q, e, s), (q, asa)\}, T_2$
 $\{(q, e, s), (q, bsb)\}, T_3$
 $\{(q, e, s), (q, c)\}, T_4$
 $\{(q, a, a), (q, e)\}, T_5$
 $\{(q, b, b), (q, e)\}, T_6$
 $\{(q, c, c), (q, e)\}, T_7$

State	Unread Input	Stack	Config
P	abbcbbba	ε	-
q	abbcbbba	S	T1
q	abbcbbba	aSa	T2
q	abbcbbba	Sa	T5
q	bbcbba	bSba	T3
q	bbcbba	Sba	T6
q	bbcbba	bSbba	T3
q	bbcbba	Sbba	T6
q	cbba	Sbba	T4
q	cbba	εbba	T4
q	cbba	bba	T2
q	cbba	εba	T6
q	cbba	ε	T5

LL(1)'nin Ayrıştırma Çizelgesinin Oluşturulması

$S \rightarrow aAbB \mid bAbB$
 $A \rightarrow aA \mid a$
 $B \rightarrow bB \mid b$

(\hookrightarrow LL(1))



	aq	Ab	bq	bb	ads	bds	ds
S	$S \rightarrow aAbB$	yon	$S \rightarrow bAbB$	yon	yon	yon	yon
A	$A \rightarrow aA$	yon	yon	yon	$A \rightarrow a$	yon	yon
B	yon	yon	yon	$B \rightarrow bB$	yon	$B \rightarrow b$	yon

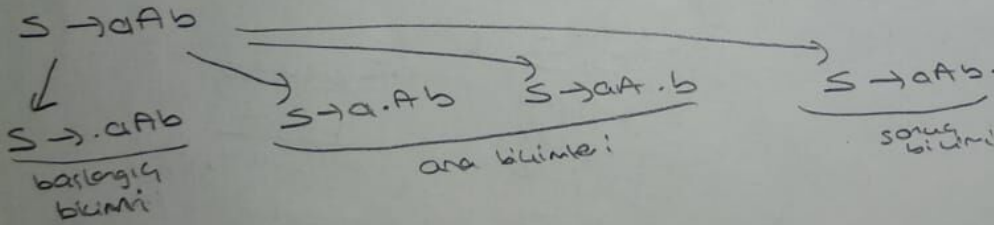
$E, S, aAbB$
 $E, S, bAbB$
 E, A, aA
 E, A, a
 E, B, bB
 E, B, b

Aşağıdan Yukarıya Ayrıştırma \rightarrow sağdan türetme

Lösde kullanarak PDA Modeli

- \rightarrow PDA q_0 durumundadır ve u içinde Z_0 vardır.
- \rightarrow PDA q_0 durumundayken, u sonuna her bir terminal a singe u üzerine ekler. Bu ilene u altına derir.
- \rightarrow PDA q_0 durumundayken, u içindeki a singeler (B) bir yerden yazma kuralının $A \rightarrow B$ sağ tarafına eşit ise, u içindeki a singeler (B) silinir. ve u üzerine A ekler. Bu ilene indirgenme derir.

İşaretili Kurallar



Örnek

$S \rightarrow aSA$
 $A \rightarrow ab \mid B$
 $B \rightarrow Bc \mid B$
 $C \rightarrow Cc \mid c$

$8+8+4$
 \hookrightarrow 20
 her üçü de var

$K = \{ S \rightarrow a.S, C \rightarrow .C \}$

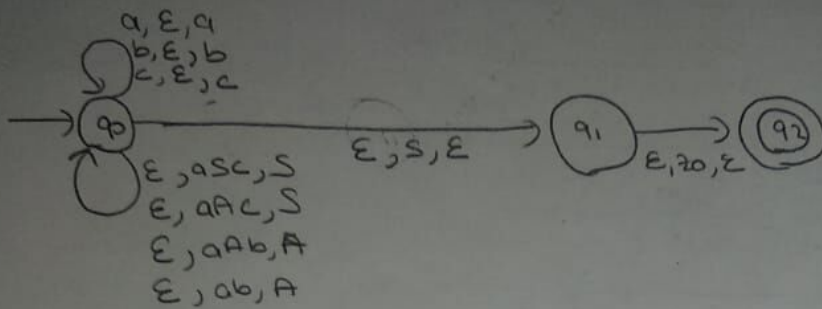
İşaretili kural kümesinin kapalı.

$K^+ = \{ S \rightarrow a.S, E \rightarrow .C, S \rightarrow .aS, S \rightarrow .A, A \rightarrow .AB, A \rightarrow .B, B \rightarrow .Bc, B \rightarrow .B, C \rightarrow .Cc, C \rightarrow .C \}$

(1) Ayrıştırıcı Geziş Diyagramının Oluşturulması

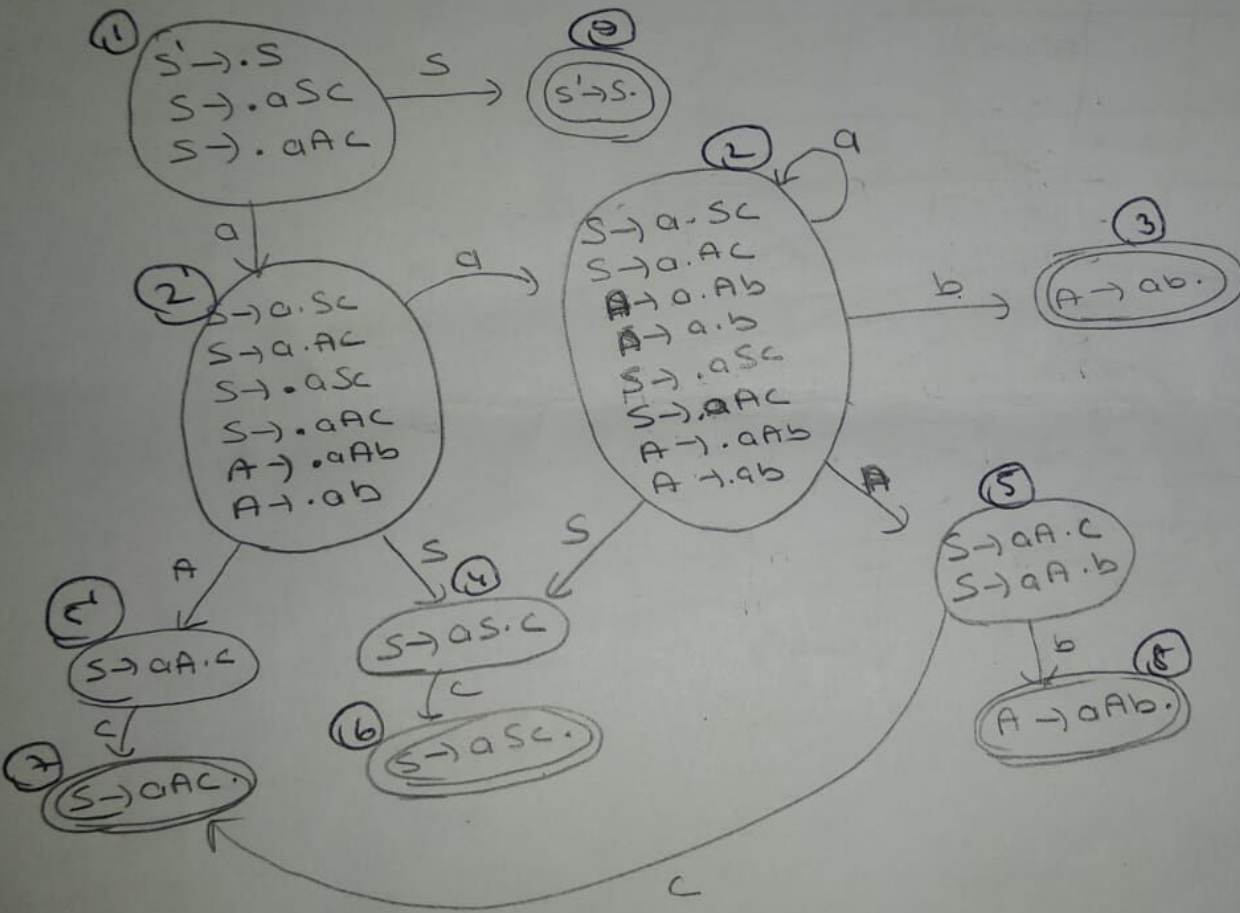
$$\begin{aligned} S &\rightarrow aSc \mid aAc \\ A &\rightarrow aAb \mid ab \end{aligned} \quad L(G) = \{ a^n b^m c^k \mid m, k \geq 1, n = m+k \}$$

1. adım



*** $LL(1)$ 'de terminaler kullanılarak LR(0) state eklenir. Aynı zamanda $LL(1)$ $LR(1)$ a, E, q a, E, q 3.1.6'de son 2'ye değinilir. (bitiş durumu) (E, ϵ, E)

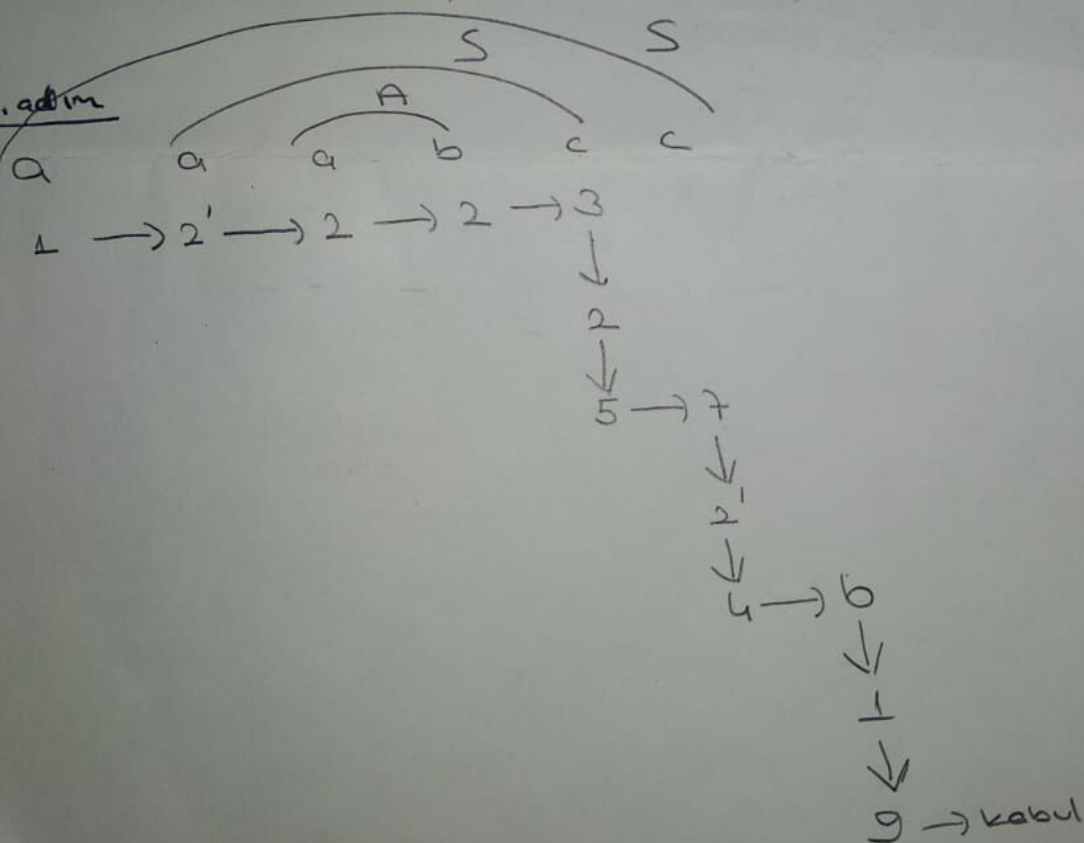
2. adım



3. adm

Durum	a	b	c	S	A	ds
1	kaydır 2'	-	-	9	-	-
2'	kaydır 2	-	-	4	S'	-
2	kaydır 2	kaydır 3	-	4	S	-
3		ab → A				indirgen
4	-	-	kaydır 6	-	-	-
5'	kaydır 7	-	-	-	-	-
5	-	kaydır 8	kaydır 7	-	-	-
6			indirgen aSc → S			indirgen
7			aAc → S			indirgen
8		aAb → A				indirgen
9						Kabul

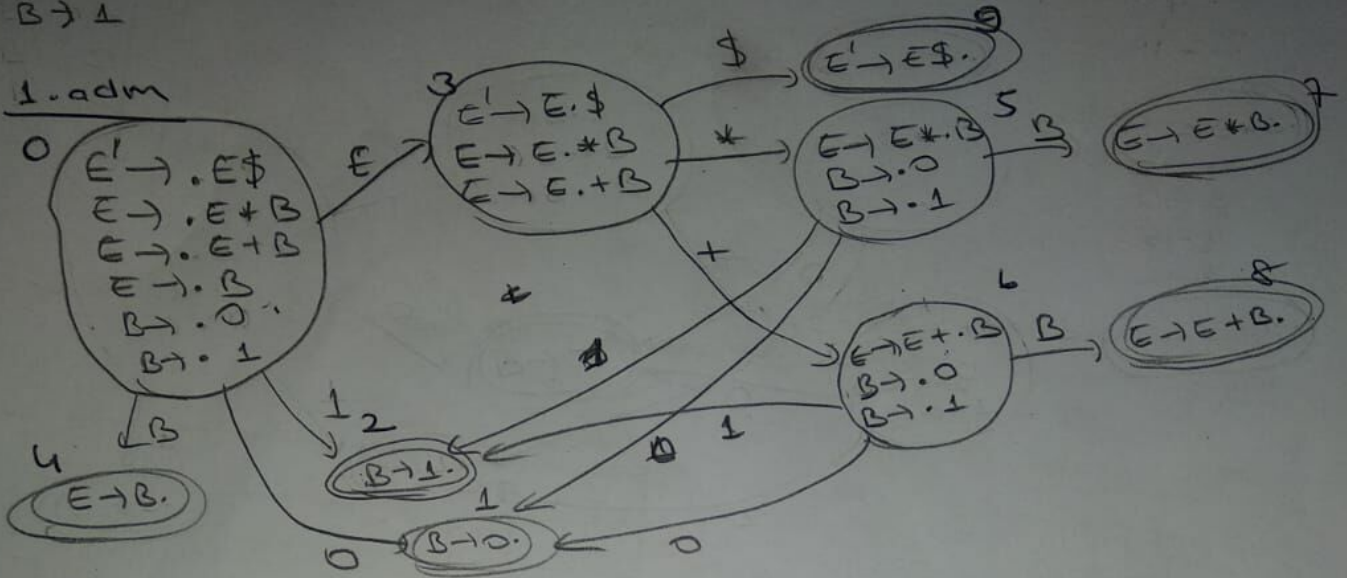
4. adm



LR(0)

$E \rightarrow E * B$
 $E \rightarrow E + B$
 $E \rightarrow B$
 $B \rightarrow 0$
 $B \rightarrow 1$

1. adm

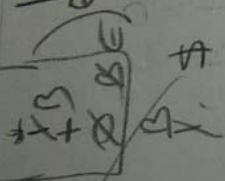


2. adm

	+	*	0	1	B	E	\$	ds
0	-	-	k1	k2	4	3	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	k6	k5	-	-	-	-	9	-
4	-	-	-	-	-	-	-	-
5	-	-	k4	k2	7	-	-	-
6	-	-	k1	k2	8	-	-	-
7	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-

$E \rightarrow E * B$
 $E \rightarrow E + B$

3. adm



$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$

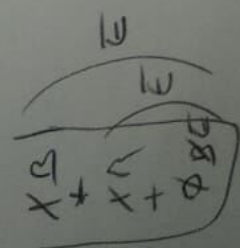
$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$

$0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$



final 2018

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

$V = \{A, B\}$

$\Sigma = \{a, b, c\}$

P:

$S \rightarrow qABB$

$A \rightarrow aAc$

$A \rightarrow \lambda$

$B \rightarrow bB$

$B \rightarrow c$

Yanda tanımlı olan CFG için LL(1) ayarlarında kullanılabcek olan PDA'ı geçişleri düzenleyiniz ve "aaccb" katarı için geçişleri ve her adımdaki işi gösteriniz.

PDA için ilk adım: $((p, \lambda, \lambda), (q, S))$
 $aaccb$ katarı için geçişler $((p, aaccb)) \vdash ((q, aaccb, S)) \vdash ((q, ?, ?)) \vdash \dots$

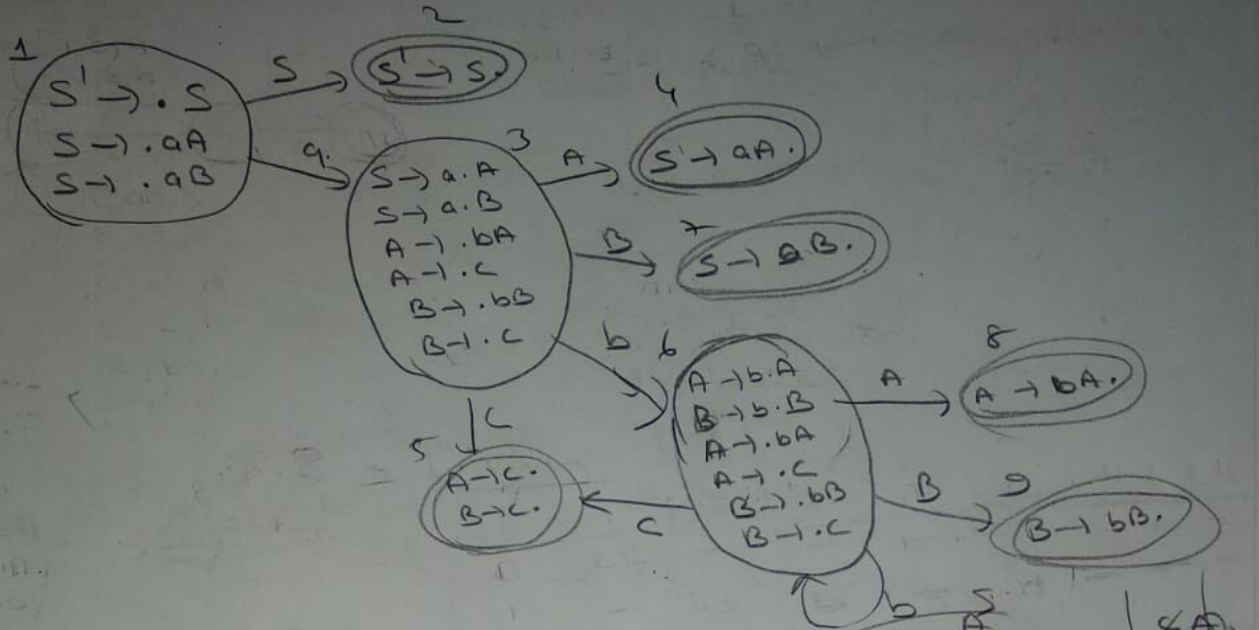
PDA geçişleri

$\Delta = \{((p, \lambda, \lambda), (q, S)),$
 $((q, \lambda, S), (q, aABb)),$ Ekleme
 $((q, \lambda, A), (q, aAc)),$
 $((q, \lambda, A), (q, \lambda)),$
 $((q, \lambda, B), (q, bB)),$
 $((q, \lambda, B), (q, c)),$
 $((q, a, a), (q, \lambda)),$ Çıkarma
 $((q, b, b), (q, \lambda)),$
 $((q, c, c), (q, \lambda)),$
 $((q, \lambda, \$), (q, \lambda))\}$

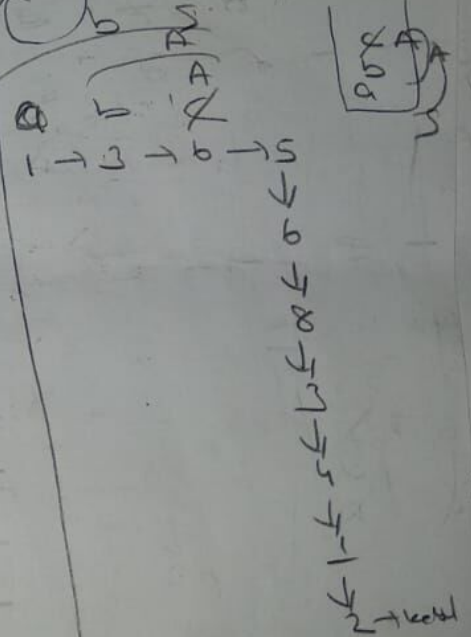
state	okunmuş input	çıkış
p	aaccb	λ
q	aaccb	S
q	aaccb	$qABb$
q	accb	ABb
q	accb	$qACBb$
q	ccb	ACBb
q	cb	qBb
q	cb	Bb
q	b	qcb
q	b	b
q	λ	\$

$S \rightarrow aA|aB$
 $A \rightarrow bA|c$
 $B \rightarrow bB|c$

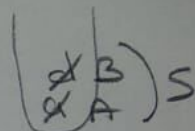
Gramer (LR(0)) ayrıştırıcı için diyagramı ve
 elde edilen diyagramı kullanarak, ayrıştırma tablosu
 oluşturuz. "abc" katarını, tablodaki geçişlerle, cırtaki
 yukarı ayrıştırma yapınız. (Ayrıca...)



	a	b	c	A	B	S	\$	dis
1	k3	-	-	-	-	2	-	-
2	-	-	-	-	-	-	-	kabul
3	-	k6	k5	4	7	-	-	-
4				$S \rightarrow aA$				
5				$A \rightarrow c \quad B \rightarrow c$				
6	-	k6	k5	8	9	-	-	-
7				$S \rightarrow aB$				
8				$A \rightarrow bA$				
9				$B \rightarrow bB$				



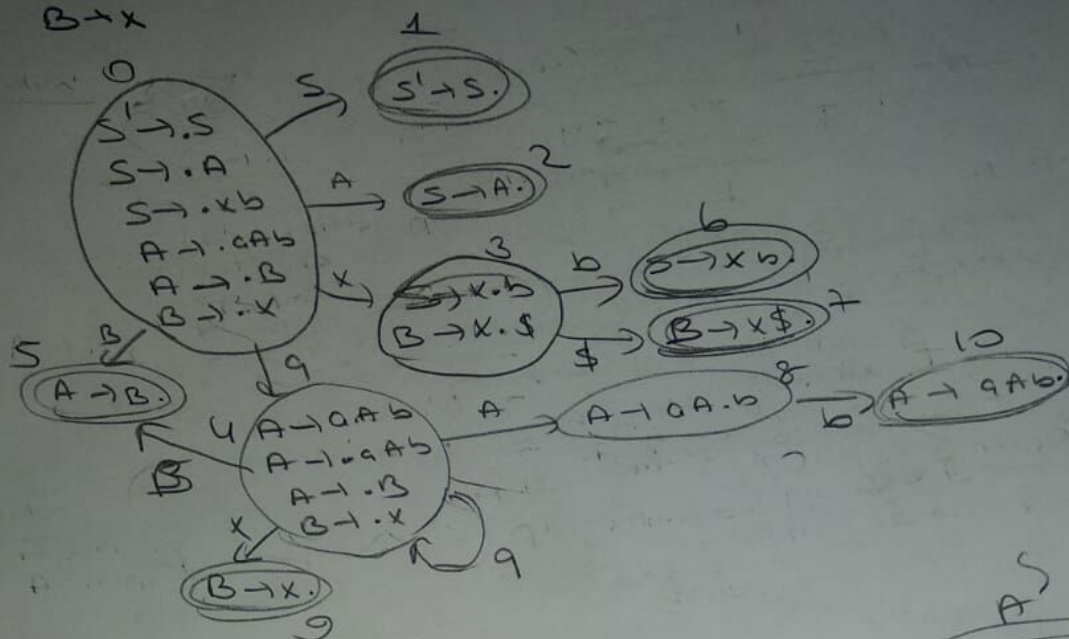
Yonolaki gramer ian LRL) ayritirici gei diler
olusturur.



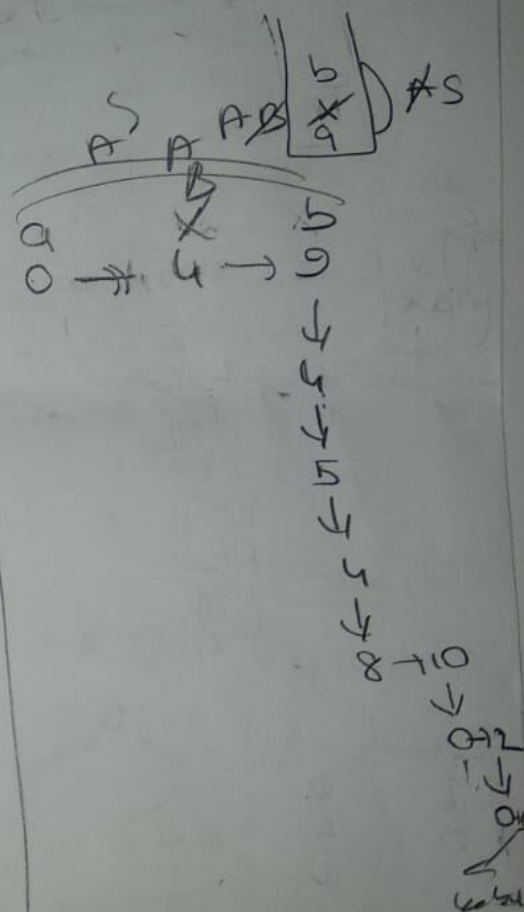
$\begin{array}{c} \overline{A \quad B} \\ A \quad B \\ 1 \rightarrow 6 \\ \downarrow \\ 1 \\ \downarrow \\ 3 \rightarrow 5 \\ \downarrow \\ 3 \\ \downarrow \\ 5 \\ \downarrow \\ 5 \\ \downarrow \\ 1 \\ \downarrow \\ 2 \rightarrow \text{kebul} \end{array}$

- 1) $S \rightarrow A$
 $S \rightarrow xb$
 $A \rightarrow aAb$
 $A \rightarrow B$
 $B \rightarrow x$

LR(1) geçiş diyagramını ve elde edilen diyagramı kullanarak ayrıştırma tablosunu oluşturun. $w = axb$ katarını tablodan faydalanarak, aşağıdaki yukarıya ayrıştırma yapınız.



	a	b	x	A	B	S	\$	ds
0	k4	-	k3	2	5	1	-	-
1	-	-	-	-	-	-	-	-
2	-	-	-	-	-	-	-	-
3	-	k6	-	-	-	-	-	-
4	k4	-	k9	8	5	-	-	-
5	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-
8	-	10	-	-	-	-	-	-
9	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-



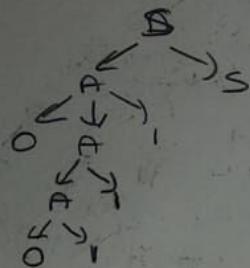
→ Gramer 1:
 $S \rightarrow AS \rightarrow \lambda$
 $A \rightarrow AI \mid OAI \mid OI$

Gramer 2:
 $S \rightarrow ASI$
 $A \rightarrow OAI \mid B$
 $B \rightarrow BII \mid OI$

Yandaki 2 grameri kullanarak 00111 diğeri soldan türetiniz. ve türetim ağacını çiziniz. Gramerlerde belirsizlik var mıdır, gösteriniz? Belirsiz gramerin tanımını veriniz.

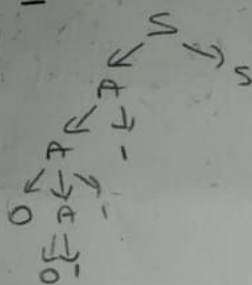
Gramer 1

1) $S \rightarrow \underline{A}S \rightarrow O\underline{A}IS \rightarrow \cancel{OAI}IS \rightarrow 00111S \rightarrow 00111$



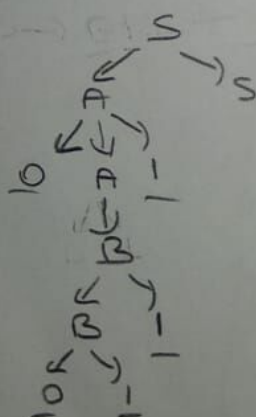
belirsiz bir gramerdir.

2) $S \rightarrow \underline{A}S \rightarrow \underline{A}IS \rightarrow O\underline{A}IIS \rightarrow 00111S \rightarrow 00111$



Gramer 2

1) $S \rightarrow \underline{A}S \rightarrow O\underline{A}IS \rightarrow O\underline{B}IS \rightarrow O\underline{B}IIS \rightarrow 00111S \rightarrow 00111$



soldan belirsiz değildir.

* Bir gramer stringler için birden fazla farklı sentex ağacı oluşturur ise gramer belirsiz olarak adlandır. Gramer sadece string kimeleleri tanımlar, belirsizlik söz konusu değil, fakat, gramer stringi yapısını yansıtmaz. Bir gramer belirsiz olabilir, fakat farklı metadokümanlar, belirsiz bir grameri tanımlar. Bir stringi üretmek için gramer belirsiz olabilir, fakat gramer belirsiz değildir.

$V = (V, \Sigma, P, S)$
 $V = (S, X, Y)$
 $\Sigma = \{+, *, (,)\}$
 $P:$
 $S \rightarrow S + X \mid X$
 $X \rightarrow X * Y \mid Y$
 $Y \rightarrow (S) \mid id$

yanda tanımlı olan CFG'nin LL(1)
 ayırtırmada kullanılabacak olan PDA'ları
 geliştiririz. ve $id + id + id + \dots$ cümlesi
 için geçerli ve her adımdaki işi
 gösterir.

PDA'nın ilk adımları: $((P, \lambda, \lambda), (q, S))$

Okunmuş input	Yığın
$id + id * id$	λ
$id + id * id$	S
$id + id * id$	$S + X$
$id + id * id$	$X + X$
$id + id * id$	$Y + X$
$id + id * id$	$id + X$
$+ id * id$	$+ X$
$id * id$	X
$id * id$	$X + Y$
$id * id$	$Y + Y$
$id * id$	$X * Y$
$* id$	$* Y$
id	Y
id	id
λ	$\$$

PDA geçişleri
 $\Delta = \{((P, \lambda, \lambda), (q, S)), T_1$
 $((q, \lambda, S), (q, S+X)) T_2$
 $((q, \lambda, S), (q, X)) T_3$
 $((q, \lambda, X), (q, X*Y)) T_4$
 $((q, \lambda, X), (q, Y)) T_5$
 $((q, \lambda, Y), (q, S)) T_6$
 $((q, \lambda, Y), (q, id)) T_7$
 $((q, +, +), (q, \lambda)) T_8$
 $((q, *, *), (q, \lambda)) T_9$
 $((q, (, (, (q, \lambda)) T_{10}$
 $((q,),), (q, \lambda)) T_{11}$
 $((q, id, id), (q, \lambda)) T_{12}$
 $((q, \lambda, \$), (q, \lambda)) T_{13}$

$((P, \lambda, S), (q, S+X)) \rightarrow$ Stackteki S 'ye $S+X$ yaz.
 $q(\lambda, S) \rightarrow S+X$
 ↓
 drum boşluğu baş

İlk eklemeyi yapılıyor.
 Sonra stackten terminali gelme durumu çıkarma yapılır.

* Terminal gelme $((q, a, a), (q, \lambda))$ a gelirse $a \rightarrow \lambda$ yaz demek silme.

* Nonterminal gelme $((q, \lambda, S), (q, X))$ \rightarrow stackte S 'ye X yaz demek.

* Bitiş $((q, \lambda, \$), (q, \lambda))$
 λ gelirse stackteki $\$$ 'ye λ koy.

Diagram illustrating the LR(0) items and transitions for the grammar:

Grammar Rules:

- $S \rightarrow aAd, \$$
- $S \rightarrow bBd, \$$
- $S \rightarrow aBe, \$$
- $S \rightarrow bAe, \$$
- $B \rightarrow c, e$
- $A \rightarrow c, d$

LR(0) Items and Transitions:

- Item 1: $S' \rightarrow \cdot S \$$
 - Transition on S to Item 2: $S \rightarrow S \cdot \$$
- Item 2: $S \rightarrow S \cdot \$$
 - Transition on $\$$ to Item 15: $S \rightarrow S \$ \cdot$
- Item 3: $S \rightarrow a \cdot Be, \$$
 - Transition on B to Item 6: $S \rightarrow aB \cdot e, \$$
 - Transition on A to Item 5: $S \rightarrow aA \cdot d, \$$
 - Transition on c to Item 7: $B \rightarrow c \cdot, e$
- Item 4: $S \rightarrow b \cdot Bd, \$$
 - Transition on B to Item 9: $S \rightarrow bB \cdot d, \$$
 - Transition on A to Item 8: $S \rightarrow bA \cdot e, \$$
 - Transition on c to Item 10: $B \rightarrow c \cdot, d$
- Item 5: $S \rightarrow aA \cdot d, \$$
 - Transition on d to Item 11: $S \rightarrow aAd \cdot, \$$
- Item 6: $S \rightarrow aB \cdot e, \$$
 - Transition on e to Item 12: $S \rightarrow aBe \cdot, \$$
- Item 7: $B \rightarrow c \cdot, e$
 - Transition on e to Item 14: $B \rightarrow c e \cdot, \$$
- Item 8: $S \rightarrow bA \cdot e, \$$
 - Transition on e to Item 13: $S \rightarrow bAe \cdot, \$$
- Item 9: $S \rightarrow bB \cdot d, \$$
 - Transition on d to Item 14: $S \rightarrow bBd \cdot, \$$
- Item 10: $B \rightarrow c \cdot, d$
 - Transition on d to Item 14: $B \rightarrow c d \cdot, \$$
- Item 11: $S \rightarrow aAd \cdot, \$$
- Item 12: $S \rightarrow aBe \cdot, \$$
- Item 13: $S \rightarrow bAe \cdot, \$$
- Item 14: $B \rightarrow c e \cdot, \$$
- Item 15: $S \rightarrow S \$ \cdot$

$B \rightarrow c, d$
 $A \rightarrow c, e$

$$\begin{array}{c} \text{S} \\ \text{A} \\ a \quad \cancel{d} \quad d \\ 1 \rightarrow 3 \rightarrow 7 \end{array}$$

	a	b	c	d	e	A	B	S	\$	ds
1	k3	k4	-	-	-	-	-	2	-	-
2	-	-	-	-	-	-	-	-	15	-
3	-	-	k7	-	-	5	6	-	-	-
4	-	-	k10	-	-	8	9	-	-	-
5	-	-	-	k11	-	-	-	-	-	-
6	-	-	-	-	k12	-	-	-	-	-
7	B → C		A → C							
8	-	-	-	-	k13	-	-	-	-	-
9	-	-	-	k14	-	-	-	-	-	-
10	B → C, A → C									
11	S → aAd									
12	S → aBe									
13	S → bAe									
14	S → bBd									

U.S.A.S

kebu