

# BIMU4032

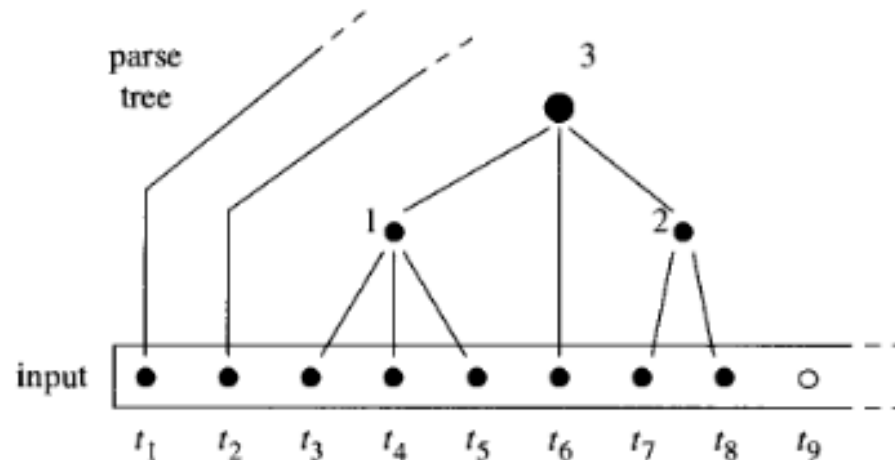
## DERLEYİCİ TASARIMI

# Bu Haftaki Konu Başlıkları

- **Aşağıdan Yukarı Ayırıştırma(Bottom-up Parsing)**
- **LR(k) Ayırıştırmada Kullanılacak PDA Modeli**
- **İşaretli Kurallar**
- **İşaretli Kural Kümesinin Kapanışı**
- **LR(1) Ayırıştırıcı Geçiş Diyagramının Oluşturulması**
- **LR(1) Ayırıştırma Sürecinin Yürütülmesi**

# Aşağıdan Yukarı Ayırıştırma(Bottom-up Parsing)

Bu ayırıştırma yönteminde syntax ağacı önce düğümler sonra kök mantığı ile oluşturulur. Alt ağacın kökü ağacın alt düğümleri oluşturulduktan sonra oluşturulur. Bu yöntemde bir düğüm oluşturulduğu zaman bu düğümün tüm alt düğümleri çoktan oluşturulmuş demektir (mevcuttur ve biliniyordur). Düğüm kendi etiketini de bilir. Ayırıştırıcı, sonra düğümü oluşturur, etiketler ve çocuk düğümlerine bağlar.



A bottom-up parser constructing its first, second, and third nodes.

# Aşağıdan Yukarı Ayırıştırma(Bottom-up Parsing)

Aşağıdan-yukarıya ayırıştırma, ayrıştırılacak ifadeden **(w)** başlanarak bir dizi işlem sonunda gramerin başlangıç simgesi olan **S** elde edilmeye çalışılır. Aşağıdan-yukarıya ayırıştırma, sağdan türetme (**rightmost derivation**) kullanılır. Buna göre, bir ifadenin **(w)** aşağıdan-yukarıya ayrıştırılmasında, bu ifadenin sağdan türetilmesinde yapılan işlemler tersine yapılır.

$$S \Rightarrow a\underline{S} \Rightarrow aA\underline{B} \Rightarrow aAB\underline{B} \Rightarrow aA\underline{B}ba \Rightarrow a\underline{A}baba \Rightarrow aab\underline{A}baba \\ \Rightarrow aababbaba$$
$$aab\underline{a}bbaba \Rightarrow aab\underline{A}baba \Rightarrow aA\underline{b}aba \Rightarrow aAB\underline{b}a \Rightarrow aAB\underline{B} \Rightarrow a\underline{AB} \\ \Rightarrow a\underline{S} \Rightarrow S$$

# LR(k) Ayırıştırırmada Kullanılacak PDA Modeli

LR(k) grameri  $G = \langle V_N, V_T, P, S \rangle$  olsun.

Bu gramer tarafından türetilen dilin ayrıştırılmasında 3 durumlu bir PDA( $\uparrow$ ) kullanılır.

$$M = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, \Phi \rangle$$

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

$$\Sigma = V_T$$

$$\Gamma = V_N \cup V_T \cup \{ Z_0 \}$$

$$F = \{ q_2 \}$$

$$\delta : \quad 1) \quad a \in V_T \quad : \quad \delta(q_0, a, Z_0) = (q_0, a)$$

$$2) \quad (A \Rightarrow \beta) \in P \quad : \quad \delta(q_0, \epsilon, \beta) = (q_0, A)$$

$$3) \quad : \quad \delta(q_0, \epsilon, S) = (q_1, \epsilon)$$

$$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$$

Tanımlanan PDA hem boş yığınla hem de uç durumla tanıyan bir PDA'dır. Doğru bir sözcük için hem yığın boşalmakta, hem de sonlu denetim birimi bir uç duruma ( $q_2$ ) ulaşmaktadır.

# LR(k) Ayırtırmada Kullanılacak PDA Modeli

- Başlangıçta PDA  $q_0$  durumundadır ve yığında  $Z_0$  vardır.
- PDA  $q_0$  durumundayken, şeritten okunan her bir terminal simge (a) yığına eklenir. Bu işleme kaydırma işlemi (**shift operation**) denilir.
- PDA  $q_0$  durumundayken, yığının üstündeki simgeler ( $\beta$ ) bir yeniden yazma kuralının ( $A \rightarrow \beta$ ) sağ tarafına eşit ise, yığından bu simgeler ( $\beta$ ) silinir ve yığına A eklenir. Bu işleme de indirgeme işlemi (**reduction operation**) denilir.
- PDA  $q_0$  durumunda iken, yığının üstünde S varsa,  $\varepsilon$  hareketi ile yığından S silinir ve sonlu denetim birimi  $q_1$  durumuna geçer.
- PDA  $q_1$  durumunda iken, yığının üstünde  $Z_0$  varsa,  $\varepsilon$  hareketi ile yığından  $Z_0$  silinir ve sonlu denetim birimi  $q_2$  durumuna geçer.

Sonuçta giriş dizisinin sonuna ulaşıldığında, yığın boşalmışsa ve sonlu denetim birimi  $q_2$  durumundaysa sözcük tanınır.

# LR(k) Ayırıştırma Kulllanılacak PDA Modeli

Aşağıdan-yukarıya ayırıştırma işlemi yukarıda sunulan PDA modeline göre gerçekleştirildiğinde:

- Kaydırma (shift) işlemi mi yoksa indirgeme (reduction) işlemi mi yapılacağı,
- Eğer indirgeme işlemi yapılacaksa da, olası indirgeme işlemlerinden hangisinin yapılacağı konusunda belirsizlik vardır.

Diğer güçlük ise yığının üstündeki  $n$  simgenin sorgulanması ve bu simgelerin yeniden yazma kurallarından birinin sağ tarafına eşit olduğunun belirlenmesidir. Bütün bu belirsizlik ve güçlükleri aşarak ayırıştırma işlemi deterministik biçimde sürdürebilmek için ayırıştırma diyagramı kullanılır.

# LR(k) Ayırıştırma Kulllanılacak PDA Modeli

Bu PDA modelinde hareket genel olarak aşağıdaki gibi gösterilmektedir:

$$\delta(q_i, s, \beta) = (q_j, \alpha) \quad s \in V_T \cup \{\epsilon\} \quad \alpha, \beta \in \Gamma^*$$

**s**: bir terminal simge ya da  $\epsilon$ . Eğer **s** bir terminal simge ise, hareketin yapılabilmesi için giriş şeridinden okunan simgenin **s** olması gerekir ve hareketin yapılması ile şeritten bir sonraki simge okunur.

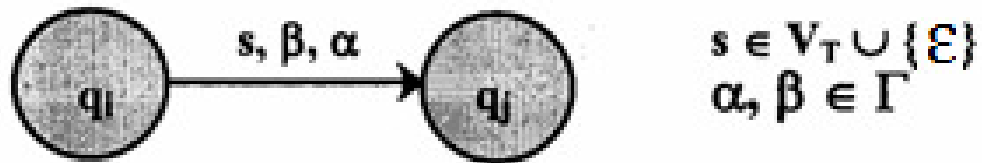
**$\beta$**  : yığın alfabesi simgelerinden oluşan bir alt dizi ya da  $\epsilon$ . Eğer  **$\beta$**  bir dizi ise, hareketin yapılabilmesi için yığının üstünde bu dizinin bulunması gerekir ve hareketin yapılması ile yığından bu alt dizi silinir.

**$\alpha$**  : yığın alfabesi simgelerinden oluşan bir alt dizi ya da  $\epsilon$ . Eğer  **$\alpha$**  bir alt dizi ise, hareketin yapılması ile yığına  **$\alpha$**  eklenir. Eğer  **$\alpha$**  boş dizi ( $\epsilon$ ) ise, hareketin yapılması ile yığına hiçbir şey eklenmez.



# LR(k) Ayırıştırma Kulllanılacak PDA Modeli

Bu üç parametre **read**, **pop** ve **push** parametreleri olarak anılır.



Bu üç parametreden biri ya da ikisi olabilir. (read,  $s$ ), (pop,  $\beta$ ) ve (push,  $\alpha$ )'nın anlamları aşağıdaki gibidir:

**read:** hareketin yapılabilmesi için şeritten okunan son simgenin  $s$  olması gerekir ve hareketin yapılması ile şeritten bir sonraki simge okunur.  $s = \epsilon$  ise şeritten son okunan simge ne olursa olsun hareket yapılır ve hareketin yapılması ile şeritten bir sonraki simge okunmaz.

**pop:** hareketin yapılabilmesi için yığının üstünde  $\beta$ 'nin bulunması gerekir ve hareketin yapılması ile yığının üstünden  $\beta$  silinir.  $\beta = \epsilon$  ise yığının üstünde ne olursa olsun hareket yapılır ve hareketin yapılması ile yığından hiçbir şey silinmez.

# LR(k) Ayırtırmada Kullanılacak PDA Modeli

Eğer  $\beta$  birden çok simgeden oluşuyorsa, yığının üstünde  $\beta$  aranırken,  $\beta'$  nin en sağdaki simgesi yığının en üstünde aranmalıdır. Örneğin  $\beta = aAB$  ise ve yığını en üstündeki üç simge sağdaki gibiyse:

<b>B</b>
<b>A</b>
<b>a</b>
<b>:</b>
<b>:</b>

yığının üstünde  $\beta$  vardır.

<b>a</b>
<b>A</b>
<b>B</b>
<b>:</b>
<b>:</b>

**push:** hareketin yapılması ile yığına  $\alpha$  eklenir.  $\alpha = \epsilon$  ise hareketin yapılması ile yığına hiçbir şey eklenmez.

Eğer  $\alpha$  birden çok simgeden oluşuyorsa,  $\alpha$  yığına eklendiğinde,  $\alpha$ 'nın en soldaki simgesi en üste gelir. Örneğin  $\alpha = aAB$  ise,  $\alpha$ 'nın eklenmesinden sonra yığının görünümü soldaki gibi olur.

# İşaretli Kurallar

Bir CFG'nin yeniden yazma kuralları  $A \Rightarrow \beta$  biçimindedir. Bu gösterimde  $\beta$  terminal simge ve gramer değişkenlerinden oluşan bir dizidir. Bu dizinin herhangi bir yerine “.” işareti konularak oluşturulan kurala **işaretli kural** denir. Örneğin  $(S \Rightarrow aAb)$  yeniden yazma kuralına bakılırsa, bu kuraldan 4 işaretli kural oluşturulabilir:

$$S \Rightarrow .aAb$$

$$S \Rightarrow a.Ab$$

$$S \Rightarrow aA.b$$

$$S \Rightarrow aAb.$$

Bu işaretli kurallardan ilki “**başlangıç biçimini**”, sonraki iki tanesi “**ara biçimleri**”, sonuncusu ise “**sonuç biçimini**” oluşturmaktadır.

# İşaretli Kural Kümesinin Kapanışı

**K** belirli sayıda işaretli kuraldan oluşan bir dizi olsun. Bu dizinin kapanışını (**K**<sup>+</sup>) elde etmek için, dizideki kurallardan, “.” dan hemen sonra bir değişkenin (**A**) yer aldığı kurallar belirlenir ve bu değişkenlere ilişkin işaretli kuralların ilk biçimleri diziye eklenir. Yeni eklenen kurallarda “.” dan hemen sonra bir değişken (**A**) yer alıyorsa, aynı işlemler, diziye yeni kural eklenmesi bitene kadar tekrarlanır.

# İşaretli Kural Kümesinin Kapanışı

Örneğin bir gramerin yeniden yazma kuralları:

$$S \Rightarrow aS \mid A$$

$$A \Rightarrow Ab \mid B$$

$$B \Rightarrow Bc \mid C$$

$$C \Rightarrow Cc \mid c$$

olsun. Gramerin yeniden yazma kuralının 20 biçimi vardır. Bunlardan 8'i başlangıç biçimi, 8'i sonuç biçimi, 4'ü de ara biçimdir.

# İşaretili Kural Kümesinin Kapanışı

$S \Rightarrow aS \mid A$       Gramer için,

$$A \Rightarrow Ab \mid B$$
$$\mathbf{B} \Rightarrow \mathbf{B_c} \mid \mathbf{C}$$
$$\mathbf{C} \Rightarrow \mathbf{C}\mathbf{c} \mid \mathbf{c}$$

**$K = \{ S \Rightarrow a.S \quad C \Rightarrow .c \}$**  İşaretli kural kümesinin kapanışı aşağıdaki şekildedir.

$$\mathbf{K}^+ = \{ \begin{array}{l} \mathbf{S} \Rightarrow \mathbf{a.S} \\ \mathbf{C} \Rightarrow \mathbf{.c} \\ \mathbf{S} \Rightarrow \mathbf{.aS} \\ \mathbf{S} \Rightarrow \mathbf{.A} \\ \mathbf{A} \Rightarrow \mathbf{.Ab} \\ \mathbf{A} \Rightarrow \mathbf{.B} \\ \mathbf{B} \Rightarrow \mathbf{.Bc} \\ \mathbf{B} \Rightarrow \mathbf{.C} \\ \mathbf{C} \Rightarrow \mathbf{.Cc} \\ \mathbf{C} \Rightarrow \mathbf{.c} \end{array} \}$$

# LR(1) Ayırıştırıcı Geçiş Diyagramının Oluşturulması

Bir LR(1) grameri verildiğinde bu gramer için LR(1) ayırıştırıcının geçiş diyagramı şu şekilde oluşturulur.

- Gramere yeni bir değişken ( $S'$ ) ve yeni bir kural ( $S' \rightarrow S$ ) eklenir. Bu kuralın işaretli başlangıç birimi olan ( $S' \rightarrow S$ )'nin kapanışı geçiş diyagramındaki 1 numaralı düğümü oluşturur.
- Geçiş diyagramındaki her düğüm için bu düğümde yer alan işaretli kurallarda “.” nın sağında yer alan ( $A \rightarrow \dots \bullet X \dots$ ) her simgeye ( $X$ : terminal ya da nonterminal simge) karşılık bir geçiş oluşturulur.  $X$  simgesi ile başlayan geçişin ucundaki düğümde, “.” nın  $X$ 'in sağına geçmesi ile elde edilen ( $A \rightarrow \dots X \bullet \dots$ ) işaretli kural kapanışı yer alacaktır. Eğer elde edilecek işaretli kurallar dizisinin kapanışı mevcut düğümlerden birinde yer alıyorsa  $X$  geçişi bu düğüme yapılır. Aksi takdirde geçiş diyagramına yeni bir düğüm eklenir. Düğümün içerisine ilgili işaretli kurallar dizisinin kapanışı yazılır ve  $X$  geçişi bu yeni eklenen düğümde son bulur.
- Eğer bir düğümdeki işaretli kural bir sonuç biçimi ( $A \rightarrow \dots X \bullet \dots$ ) ise, bu kural düğümdeki tek kural olmalıdır. Bu durumda bu düğüm bir uç düğüm olur ve çift çember ile gösterilir.
- Geçiş diyagramı, diyagrama eklenen her yeni düğümden başlayan geçişler tamamlanıncaya kadar sürdürülür.

# LR(1) Ayırıştırıcı Geçiş Diyagramının Oluşturulması

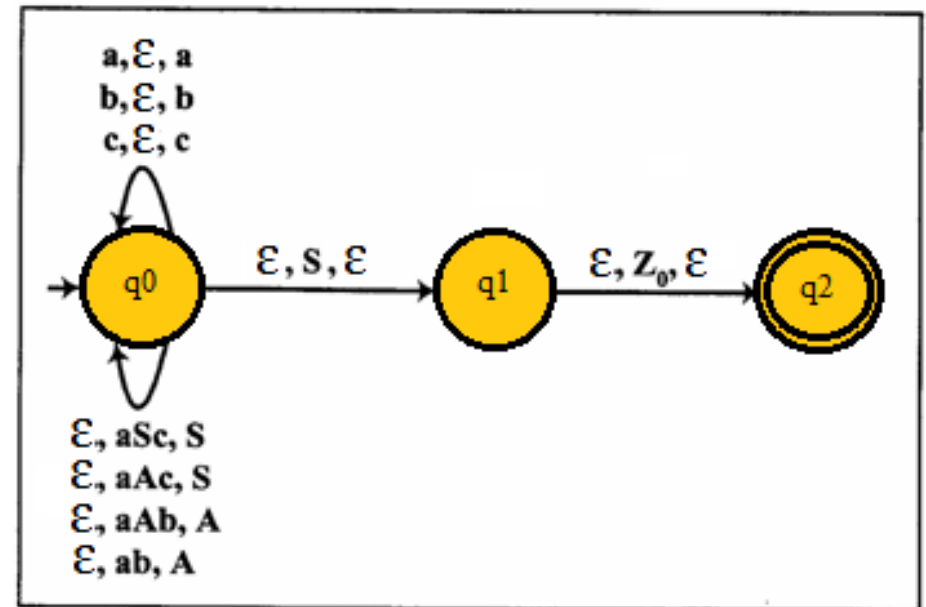
## Örnek 1

$S \rightarrow aSc \mid aAc$

$A \rightarrow aAb \mid ab$

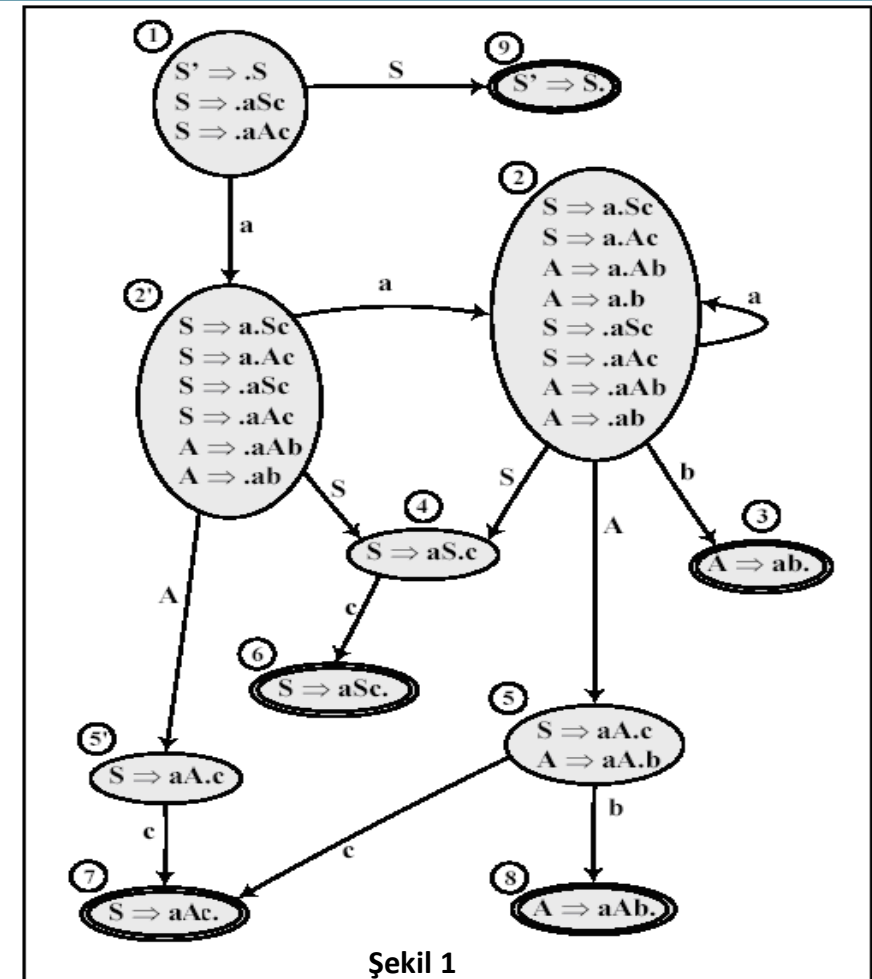
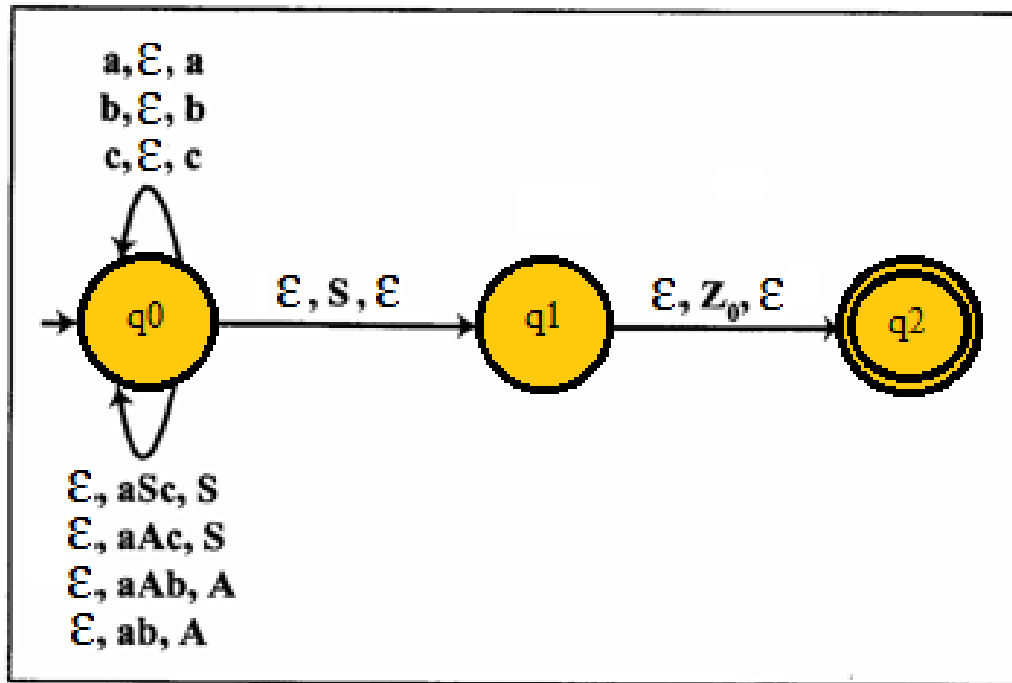
$G_1$  bir CFG'dir. Bu gramerin türettiği dili aşağıdaki gibi tanımlayabiliriz.

$$L(G_1) = \{a^n b^m c^k \mid m, k \geq 1, n = m + k\}$$





# LR(1) Ayırıştırıcı Geçiş Diyagramının Oluşturulması



Şekil 1

# LR(1) Ayırıştırma Sürecinin Yürütülmesi

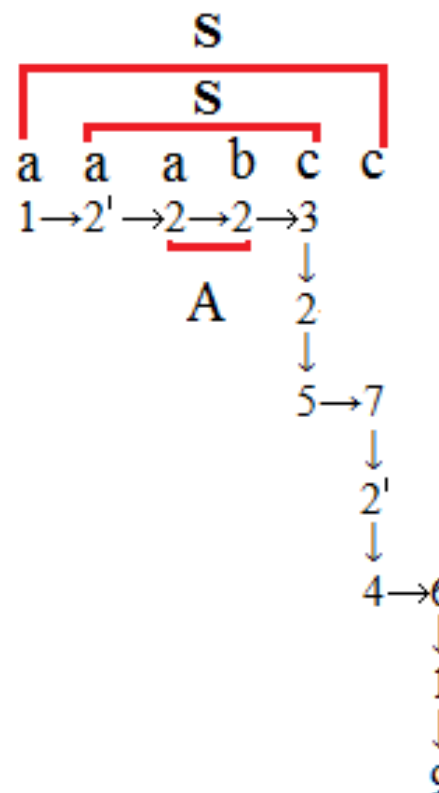
LR(1) ayırıştırma işlemi, aşağıdan-yukarıya ayırıştırmaya uygun olarak oluşturulan PDA( $\uparrow$ ) modeli ile LR(1) ayırıştırma çizelgesi kullanılarak gerçekleştirilir. LR(1) ayırıştırma çizelgesi ile, PDA( $\uparrow$ ) modelinin taşıdığı belirsizlik (kaydırma mı yoksa indirgeme mi yapılacak? İndirgeme yapılacaksa hangi kurala göre yapılacak?) çözülür ve ayırıştırma işlemi deterministik bir işleme dönüştürülür. Her kaydırma işleminden sonra hangi duruma gidileceği ayırıştırma çizelgesinde yer almaktadır. Her indirgeme işleminden sonra hangi duruma gidileceği ise ayırıştırma çizelgesinde yer almaz. Bunun ayırıştırma sırasında dinamik biçimde bulunması gerekir. Bunun için de, kaydırma işlemlerinde, yığına bir simge eklenirken, kaydırma işleminin hangi durumda yapıldığı bilgisinin de yığına eklenmesi gerekir. Ayırıştırma sürecinin nasıl yürütüldüğünü ve her indirgeme işleminden sonra hangi duruma gidileceğinin nasıl bulunduğu bir örnek üzerinde açıklanacaktır.

# LR(1) Ayırıştırma Sürecinin Yürütülmesi

Örnek olarak  $G_1$  grameri tarafından türetilmiş olan  $w = \mathbf{aaabcc}$  ifadesi için ayırıştırma işlemlerinde Şekil 1'deki PDA( $\uparrow$ ) modeli ile Tablo 1'de yer alan ayırıştırma çizelgesi kullanılır.

Dur.	a	b	c	S	A	ds
1	Kaydır, 2'	-	-	9	-	-
2'	Kaydır, 2	-	-	4	5'	-
2	Kaydır, 2	Kaydır, 3	-	4	5	-
3	-	İndirge ( $ab \Rightarrow A$ )	-	-	-	İndirge ( $ab \Rightarrow A$ )
4	-	-	Kaydır, 6	-	-	-
5'	-	-	Kaydır, 7	-	-	-
5	-	Kaydır, 8	Kaydır, 7	-	-	-
6	-	-	İndirge ( $aSc \Rightarrow S$ )	-	-	İndirge ( $aSc \Rightarrow S$ )
7	-	-	İndirge ( $aAc \Rightarrow S$ )	-	-	İndirge ( $aAc \Rightarrow S$ )
8	-	İndirge ( $aAb \Rightarrow A$ )	-	-	-	İndirge ( $aAb \Rightarrow A$ )
9	-	-	-	-	-	Kabul et

Tablo 1



# LR(1) Ayırıştırma Sürecinin Yürütülmesi

1. Başlangıçta PDA  $q_0$  durumundadır, yığında  $Z_0$  simgesi vardır, okuma kafası  $w$ 'nin ilk simgesi **(a)** üzerindedir. Ayırıştırıcı ise 1 numaralı durumdadır (ayırıştırma çizelgesinin 1. satırı).
2. İlk işlem olarak  $w$ 'nin ilk simgesi **(a)** yığına kaydırılır. Kaydırma işlemi 1 numaralı durumda yapıldığı için, yığına **a** simgesinin yanına durum numarası olarak da **1** yazılır. Kaydırma işleminden sonra ayırıştırıcı 2' numaralı duruma gider.
3. İkinci yapılan işlem  $w$ 'nin ikinci simgesinin **(a)** yığına kaydırılmasıdır. Kaydırma işlemi 2' numaralı durumda yapıldığı için, yığına **a** simgesinin yanına durum numarası olarak 2' yazılır. Kaydırma işleminden sonra ayırıştırıcı 2 numaralı duruma gider.
4. Üçüncü işlem olarak  $w$ 'nin üçüncü simgesi **(a)** yığına kaydırılır. Kaydırma işlemi 2 numaralı durumda yapıldığı için, yığına **a** simgesinin yanına durum numarası olarak 2 yazılır. Kaydırma işleminden sonra ayırıştırıcı 2 numaralı durumda kalır.
5. Dördüncü işlem olarak  $w$ 'nin dördüncü simgesi **(b)** yığına kaydırılır. Kaydırma işlemi 2 numaralı durumda yapıldığı için, yığına **b** simgesinin yanına durum numarası olarak da 2 yazılır. Kaydırma işleminden sonra ayırıştırıcı 3 numaralı duruma gider.

# LR(1) Ayırıştırma Sürecinin Yürütülmesi

6. Ayırıştırıcı 3 numaralı durumda bulunduğu, ileri bakış simgesi de  $w$ 'nin beşinci simgesi ( $c$ ) olduğu için, beşinci olarak yapılacak işlem ( $ab \Rightarrow A$ ) indirgeme işlemidir. Bunun için yığından  $ab$  çıkarılır ve yığına  $A$  eklenir. Yığından  $ab$  çıkarıldığında 2 numaralı duruma dönmüş olur. Çünkü  $ab$ 'nin ilk simgesi olan  $a$  yığına 2 numaralı durumda eklenmiştir. Ayırıştırma çizelgesine göre 2 numaralı durumda  $A$  işlendiği için de ayırıştırıcı 5 numaralı duruma geçer.
7. İndirgeme işleminde giriş şeridi değişmediği için ileri bakış simgesi ( $w$ 'nin beşinci simgesi olan  $c$ ) de değişmemiştir. Ayırıştırıcı da 5 numaralı durumda bulunduğu için altıncı işlem olarak  $w$ 'nin beşinci simgesi ( $c$ ) yığına kaydırılır. Kaydırma işlemi 5 numaralı durumda yapıldığı için, yığına  $c$  simgesinin yanına durum numarası olarak da 5 yazılır. Kaydırma işleminden sonra ayırıştırıcı 7 numaralı duruma gider.
8. Ayırıştırıcı 7 numaralı durumda bulunduğu, ileri bakış simgesi de  $w$ 'nin son simgesi ( $c$ ) olduğu için, yedinci olarak yapılacak işlem ( $aAc \Rightarrow S$ ) indirgeme işlemidir. Bunun için yığından  $aAc$  çıkarılır ve yığına  $S$  eklenir. Yığından  $aAc$  çıkarıldığında 2' numaralı duruma dönmüş olur. Çünkü  $aSc$ 'nin ilk simgesi olan  $a$  yığına 2' numaralı durumda eklenmiştir. Ayırıştırma çizelgesine göre 2' numaralı durumda  $S$  işlendiği için de ayırıştırıcı 4 numaralı duruma geçer.

# LR(1) Ayırıştırma Sürecinin Yürütülmesi

9. İndirgeme işleminde giriş şeridi değişmediği için ileri bakış simgesi ( $w$ 'nin son simgesi olan  $c$ ) de değişmemiştir. Ayırıştırıcı da 4 numaralı durumda bulunduğu için sekizinci işlem olarak  $w$ 'nin son simgesi ( $c$ ) yığına kaydırılır. Kaydırma işlemi 4 numaralı durumda yapıldığı için, yığına  $c$  simgesinin yanına durum numarası olarak da 4 yazılır. Kaydırma işleminden sonra ayırıştırıcı 6 numaralı duruma gider.
10. Ayırıştırıcı 6 numaralı durumda bulunduğu, ileri bakış simgesi de  $ds$  (dizi sonu) olduğu için, dokuzuncu olarak yapılacak işlem ( $aSc \ S$ ) indirgeme işlemidir. Bunun için yığından  $aSc$  çıkarılır ve yığına  $S$  eklenir. Yığından  $aSc$  çıkarıldığında 1 numaralı duruma dönmüş olur. Çünkü  $aSc$ 'nin ilk simgesi olan  $a$  yığına 1 numaralı durumda eklenmiştir. Ayırıştırma çizelgesine göre 1 numaralı durumda  $S$  işlendiği için de ayırıştırıcı 9 numaralı duruma geçer.
11. SAyırıştırıcı 9 numaralı durumda bulunduğu, ileri bakış simgesi de  $ds$  olduğu için ayırıştırma süreci biter ve ( $w = aaabcc$ ) tümcesi kabul edilir.

# LR(1) Ayırıştırma Sürecinin Yürütülmesi

## Örnek 2

$S \Rightarrow aSa \mid bSb \mid c$  bir CFG'dir.

