

# SÖZDİZİM ÇÖZÜMLEME (2)

1

## Temel Ayırıştırma Yöntemleri

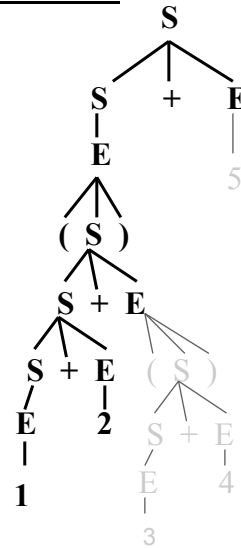
- Ayırıştırıcı bir sözcük katarının L(G) dilinin kuralları ile türetilir olup olmadığını inceler ve bu süreç boyunca sözcük katarına ilişkin bir **ayırıştırma ağacı** oluşturur
- Ayırıştırma yöntemleri ağacın oluşturulma düzenine göre ikiye ayrılır
  - Yukarıdan-aşağıya (top-down)
  - Aşağıdan-yukarıya (bottom-up)

2

## Yukarıdan-aşağıya Ayırıştırma

### Yukarıdan-aşağıya ayırıştırma:

Bu grupta yer alan yöntemler başlangıç simgesi S'ten başlar, S'i ağacın köküne yerleştirir ve sözcük katarını yapraklarda elde edene kadar ağacı derinliğine doğru kuracak olan türetim adımlarını gerçekleştirirler

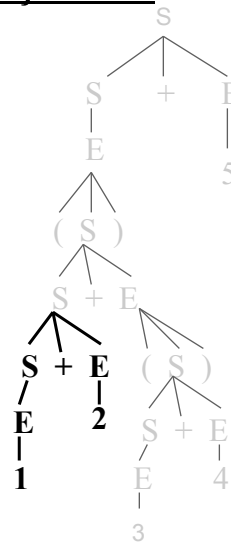


3

## Aşağıdan-yukarıya Ayırıştırma

### Aşağıdan-yukarıya ayırıştırma:

Bu grupta yer alan yöntemler ise sözcüklerden başlayıp, bunları yapraklara yerleştirirler ve bu yaprakları türeten kuralları bulup ara düğümleri oluşturarak, ağacı aşağıdan yukarıya doğru kurarlar. Amaç, köke S başlangıç simgesini yerleştirebilmektir



4

## Yukarıdan-aşağıya Ayırıştırma Yöntemi

Amaç: Bir sözcük katarını üreten  
en-soldan türetimler dizisi oluşturmak

$$\begin{array}{l} S \rightarrow E + S \mid E \\ E \rightarrow \text{sayı} \mid (S) \end{array}$$

Kısmen türetilmiş katar	sonraki sözcük	<u>ayrıştırılmış/ayrıştırılmamış</u>
$\rightarrow E + S$	(	$(1+2+(3+4))+5$
$\rightarrow (S) + S$	1	$(1+2+(3+4))+5$
$\rightarrow (E+S)+S$	1	$(1+2+(3+4))+5$
$\rightarrow (1+S)+S$	2	$(1+2+(3+4))+5$
$\rightarrow (1+E+S)+S$	2	$(1+2+(3+4))+5$
$\rightarrow (1+2+S)+S$	2	$(1+2+(3+4))+5$
$\rightarrow (1+2+E)+S$	(	$(1+2+(3+4))+5$
$\rightarrow (1+2+(S))+S$	3	$(1+2+(3+4))+5$
$\rightarrow (1+2+(E+S))+S$	3	$(1+2+(3+4))+5$
$\rightarrow \dots$		

5

## Yukarıdan-aşağıya Ayırıştırmada Sorun

- Bir sonraki giriş sözcüğüne bakarak hangi türetim kuralının seçileceğine karar verebilmek gerekir

$$\begin{array}{l} S \rightarrow E + S \mid E \\ E \rightarrow \text{sayı} \mid (S) \end{array}$$

Örnek giriş 1: "(1)"  $S \rightarrow E \rightarrow (S) \rightarrow (E) \rightarrow (1)$

Örnek giriş 2: "(1)+2"  $S \rightarrow E+S \rightarrow (S)+S \rightarrow (E)+S$   
 $\rightarrow (1)+E \rightarrow (1)+2$

Örnek 2'de, "E+S" türetiminin seçimine nasıl karar verilir?  
Eğer "E+S" yerine "(S)" seçilmiş olsaydı işlem başarıyla sonuçlanırmıydı?

6

## Sorunun Kaynağı Gramerdir

$$S \rightarrow E + S \mid E$$

$$E \rightarrow \text{sayı} \mid (S)$$

- Bu gramer ile, yalnızca **bir tane** “sonraki sözcük” kullanılarak yukarıdan-aşağıya ayrıştırma işlemi yapılamaz → LL(1) grameri değildir!
  - LL(1) = bir tane (1) sonraki sözcük ile, en soldan türetimler kullanılarak (Left-most derivation), soldan-sağa taranabilir(Left-to-right scanning) gramer
  - Yukarıdan-aşağıya ayrıştırmanın mümkün olabilmesi için, grameri yeniden düzenleyip LL(1) hale getirmek gerekir.

7

## LL(1) Gramerine Dönüştürme (1)

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

**Sorun:** A’in hangi seçeneğini tercih edilmesi gerektiğine, ancak ilk ifadeyi ( $\alpha$ ) izleyen sözcüğü gördükten sonra karar verilebilmesi

**Çözüm :** A seçeneklerinin ortak olan bölümünü ( $\alpha$ ) alıp, karar noktasına yeni bir nonterminal ( $A'$ ) ekle.  $A'$  tanımında, türetimlerin geriye kalan bölümlerine yer ver.

$$A \rightarrow \alpha \beta \mid \alpha \gamma \quad \Rightarrow \quad \begin{array}{l} A \rightarrow \alpha A' \\ A' \rightarrow \beta \mid \gamma \end{array}$$

8

## LL(1) Gramerine Dönüştürme (2)

$$S \rightarrow E + S$$

$$S \rightarrow E$$

$$E \rightarrow \text{sayı}$$

$$E \rightarrow (S)$$


$$S \rightarrow ES'$$

$$S' \rightarrow \varepsilon$$

$$S' \rightarrow +S$$

$$E \rightarrow \text{sayı}$$

$$E \rightarrow (S)$$

• Sorun: S'in hangi seçeneğini tercih edilmesi gerektiğine, ancak ilk ifadeyi (E) izleyen sözcüğü gördükten sonra karar verilebilmesi

• Çözüm: S tanımlarının ortak olan bölümünü (E) alıp, karar noktasına yeni bir nonterminal (S') ekle.  
S' tanımında, türetimlerin geriye kalan bölümlerine yer ver.

$$S' \rightarrow +S \text{ ve } S' \rightarrow \varepsilon$$

9

## Yeni Gramer ile Ayrıştırma

$$S \rightarrow ES'$$

$$S' \rightarrow \varepsilon \mid +S$$

$$E \rightarrow \text{sayı} \mid (S)$$

<u>kısmen türetilmiş katar</u>	<u>sonraki sözcük</u>	<u>ayrıştırılmış/ayrıştırılmamış</u>
$\rightarrow ES'$	(	(1+2+(3+4))+5
$\rightarrow (S)S'$	1	(1+2+(3+4))+5
$\rightarrow (ES')S'$	1	(1+2+(3+4))+5
$\rightarrow (1S')S'$	+	(1+2+(3+4))+5
$\rightarrow (1+ES')S'$	2	(1+2+(3+4))+5
$\rightarrow (1+2S')S'$	+	(1+2+(3+4))+5
$\rightarrow (1+2+S)S'$	(	(1+2+(3+4))+5
$\rightarrow (1+2+ES')S'$	(	(1+2+(3+4))+5
$\rightarrow (1+2+(S)S')S'$	3	(1+2+(3+4))+5
$\rightarrow (1+2+(ES')S')S'$	3	(1+2+(3+4))+5
$\rightarrow (1+2+(3S')S')S'$	+	(1+2+(3+4))+5
$\rightarrow (1+2+(3+E)S')S'$	4	(1+2+(3+4))+5
$\rightarrow \dots$		

10

## Tahmine Dayalı Ayırıştırma

### ■ LL(1) gramer:

- Belirli bir non-terminal için, bir sonraki sözcük (lookahead symbol) uygulanacak olan türetim kuralını **kesinlikle** belirler
- Yukarıdan - aşağıya ayırıştırma → **“Tahmine Dayalı Ayırıştırma”** (Predictive Parsing) adını alır
- Doğru tahminleri yönlendiren bir ayırıştırma tablosu yardımıyla adımlar yürütülür.
- Tablonun her satırı bir non-terminale, her sütünü da bir terminal sözcüğe karşı düşer:  
non-terminal x terminal → türetim kuralları

11

## Tablo ile Ayırıştırma

$$S \rightarrow ES'$$

$$S' \rightarrow \varepsilon \mid +S$$

$$E \rightarrow \text{sayı} \mid (S)$$

kısmen türetilmiş katar

sonraki sözcük **ayırıştırılmış/ayırıştırılmamış**

→ES'	(	(1+2+(3+4))+5
→(S)S'	1	(1+2+(3+4))+5
→(ES')S'	1	(1+2+(3+4))+5
→(1S')S'	+	(1+2+(3+4))+5
→(1+S)S'	2	(1+2+(3+4))+5
→(1+ES')S'	2	(1+2+(3+4))+5
→(1+2S')S'	+	(1+2+(3+4))+5

Ayırıştırma Tablosu

	sayı	+	(	)	\$
S	→ ES'		→ ES'		
S'		→ +S		→ ε	→ ε
E	→ sayı		→ (S)		

12

## Ayrıştırıcının Gerçeklenmesi

- Tablo kolaylıkla bir **“Rekürsif İniş Ayrıştırıcısı”** na (Recursive Descent Parser) dönüştürülebilir
- Her biri ayrı bir nonterminali ayrıştırmak üzere, 3 fonksiyona gerek olacaktır:  
ayrıştır\_S(), ayrıştır\_S'(), and ayrıştır\_E()

	num	+	(	)	\$
S	→ ES'		→ ES'		
S'		→ +S		→ ε	→ ε
E	→ num		→ (S)		

13

## Rekürsif-İniş Ayrıştırıcısı

```

void ayrıştır_S() {
    switch (sözcük) {
        case sayı: ayrıştır_E(); ayrıştır_S'(); return;
        case '(': ayrıştır_E(); ayrıştır_S'(); return;
        default: AyrıştırmaHatası();
    }
}

```

sonraki sözcük

	sayı	+	(	)	\$
<b>S</b>	→ <b>ES'</b>		→ <b>ES'</b>		
S'		→ +S		→ ε	→ ε
E	→ sayı		→ (S)		

14

## Rekürsif-İniş Ayırıştırıcısı(2)

```
void ayırıştır_S'() {
    switch (sözcük) {
        case '+': sözcük = input.read(); ayırıştır_S(); return;
        case '(': return;
        case EOF: return;
        default: AyırıştırmaHatası();
    }
}
```

	sayı	+	(	)	\$
S	→ ES'		→ ES'		
S'		→ +S		→ ε	→ ε
E	→ sayı		→ (S)		

15

## Rekürsif-İniş Ayırıştırıcısı(3)

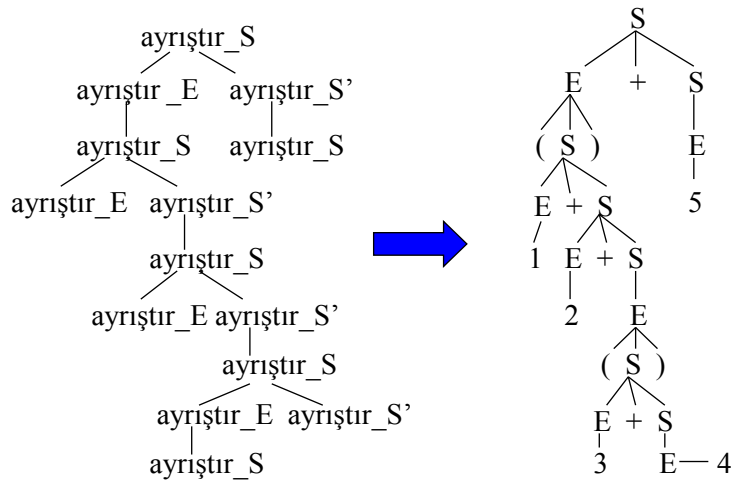
```
void ayırıştır_E() {
    switch (sözcük) {
        case sayı: sözcük = input.read(); return;
        case '(': sözcük = input.read(); ayırıştır_S();
            if (sözcük != ')') AyırıştırmaHatası();
            sözcük = input.read(); return;
        default: AyırıştırmaHatası();
    }
}
```

	sayı	+	(	)	\$
S	→ ES'		→ ES'		
S'		→ +S		→ ε	→ ε
E	→ sayı		→ (S)		

16



## Çağrı Ağacı = Ayırıştırma Ağacı



17

## Ayırıştırma Tablosunun Oluşturulması

Verilen gramerden ayırıştırma tablosunu otomatik olarak üretecek bir algoritmaya gereksinim vardır.

$S \rightarrow ES'$   
 $S' \rightarrow \epsilon \mid +S$   
 $E \rightarrow \text{sayı} \mid (S)$

algoritma  
  
 ??

	sayı	+	(	)	\$
S	ES'		ES'		
S'		+S		$\epsilon$	$\epsilon$
E	sayı		(S)		

18

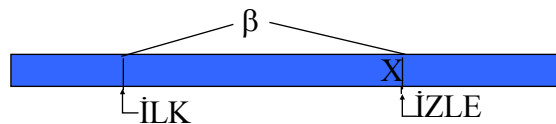
## Ayrıştırma Tablosunun Oluşturulması

- Ayrıştırıcının oluşturulabilmesi için:
  - Bir non-terminal simgeyi izleyebilecek her sonraki sözcüğün **en fazla bir** türetim kuralıyla indirgenebilir olması gerekir
  - İki yardımcı kavram yararlı olacaktır:  
**İLK** ve **İZLE** (FIRST ve FOLLOW) kümeleri

19

## İLK VE İZLE KÜMELERİ

- **İLK( $\beta$ )**: ( $\beta$ : herhangi terminal/non-terminal simgeleri katarı)
  - $\beta$ 'nin **her farklı açılımında ilk simge** olarak yer alabilecek olan tüm terminal simgeler kümesi
- **İZLE(X)**: (X: bir non-terminal)
  - X'in türetimini **izleyebilecek** olan tüm giriş simgeleri kümesi



20

## Ayrıştırma Tablosu Girişleri

- $X \rightarrow \beta$  türetimini ele alalım:
- Tablonun X satırını seçip,  $\text{İLK}(\beta)$  kümesinde yer alan her simgenin gösterildiği her sütuna " $\rightarrow \beta$ " ekle
- Eğer  $\beta$ 'dan boş katar ( $\epsilon$ ) türetilebilir ise,  $\text{İZLE}(\beta)$  kümesinde yer alan her simgenin gösterildiği sütuna da " $\rightarrow$ "  $\beta$  ekle
- Sonuç tabloda çelişen girişler oluşmaz ise gramer  $\text{LL}(1)$ 'dir

$S \rightarrow ES'$   
 $S' \rightarrow \epsilon \mid +S$   
 $E \rightarrow \text{sayı} \mid (S)$

	sayı	+	(	)	\$
S	ES'		ES'		
S'		+S		$\epsilon$	$\epsilon$
E	sayı		(S)		

21

## İLK Kümesinin Belirlenmesi

**İLK (X) kümesinin oluşturulması:**

1. X bir terminal ise,  $\text{İLK}(X) = \{X\}$
2. X bir non-terminal ve  $X \rightarrow a\alpha$  ise, a terminalini  $\text{İLK}(X)$  kümesine ekle.
3.  $X \rightarrow \epsilon$  ise,  $\epsilon$   $\text{İLK}(X)$  kümesine ekle
4. X bir non-terminal ve  $X \rightarrow Y_1Y_2\dots Y_k$  ise, eğer  $a \in \text{İLK}(Y_i)$  ve  $\epsilon \in \text{İLK}(Y_j)$ ,  $j = 1\dots i-1$  ise, a terminalini  $\text{İLK}(X)$  kümesine ekle ( $Y_1\dots Y_{i-1}$  katarının  $\epsilon$  indirgenebilme durumu)
5.  $\epsilon \in \text{İLK}(Y_1Y_2\dots Y_k)$  ise  $\epsilon$   $\text{İLK}(X)$  kümesine ekle

22

## İLK Kümesi İçin Örnek

$$\begin{aligned} S &\rightarrow ES' \\ S' &\rightarrow \varepsilon \mid + S \\ E &\rightarrow \text{sayı} \mid (S) \end{aligned}$$

1. Kural uygula:  $\text{İLK}(\text{sayı}) = \{\text{sayı}\}$ ,  $\text{İLK}(+) = \{+\}$ , vs.
3. Kural uygula:  $\text{İLK}(S') = \{\varepsilon\}$
4. Kural uygula:  $\text{İLK}(S) = \text{İLK}(E) = \{\}$   
 $\text{İLK}(S') = \text{İLK}('+' ) + \{\varepsilon\} = \{\varepsilon, +\}$   
 $\text{İLK}(E) = \text{İLK}(\text{sayı}) + \text{İLK}('(' ) = \{\text{sayı}, (\}$
4. Kural tekrar:  $\text{İLK}(S) = \text{İLK}(E) = \{\text{sayı}, (\}$   
 $\text{İLK}(S') = \{\varepsilon, +\}$   
 $\text{İLK}(E) = \{\text{sayı}, (\}$

23

## İZLE Kümesinin Belirlenmesi

İZLE (X) kümesinin oluşturulması:

1. S başlangıç simgesi ise, \$ simgesini İZLE(S) kümesine ekle
2.  $A \rightarrow \alpha B \beta$  ise,  $\text{İLK}(\beta)$  kümesinin  $\varepsilon$  dışındaki tüm elemanlarını İZLE(B) kümesine ekle
3.  $A \rightarrow \alpha B$  veya  $A \rightarrow \alpha B \beta$  ve  $\varepsilon \in \text{İLK}(\beta)$  ise, İZLE(A) kümesinde yer alan tüm elemanları İZLE(B) kümesine ekle

24

## İZLE Kümesi İçin Örnek

$$S \rightarrow ES'$$

$$S' \rightarrow \varepsilon \mid +S$$

$$E \rightarrow \text{sayı} \mid (S)$$

$$\text{İLK}(S) = \{\text{sayı}, ( \}$$

$$\text{İLK}(S') = \{\varepsilon, + \}$$

$$\text{İLK}(E) = \{\text{sayı}, ( \}$$

1. Kural uygula:  $\text{İZLE}(S) = \{\$ \}$

2. Kural uygula:  $S \rightarrow ES'$   $\text{İZLE}(E) += \{\text{İLK}(S') - \varepsilon\} = \{+\}$

$$S' \rightarrow \varepsilon \mid +S \text{ -----}$$

$$E \rightarrow \text{sayı} \mid (S) \quad \text{İZLE}(S) += \{\text{İLK}('(') - \varepsilon\} = \{ \$, ) \}$$

3. Kural uygula:  $S \rightarrow ES'$   $\text{İZLE}(E) += \text{İZLE}(S) = \{+, \$, )\}$

(çünkü  $S'$   $\varepsilon$  indirgenebilir)

$$\text{İZLE}(S') += \text{İZLE}(S) = \{ \$, ) \}$$