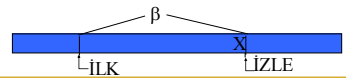


SÖZDİZİM ÇÖZÜMLEME (3)

1

İLK VE İZLE KÜMELERİ

- **İLK(β)**: (β : herhangi terminal/non-terminal simgeleri katarı)
 - β '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



2

İLK Kümesinin Belirlenmesi

İLK (X) kümesinin oluşturulması:

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

3

İLK Kümesi İçin Örnek

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

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

4

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 ϵ dışındaki tüm elemanlarını İZLE(B) kümesine ekle
3. $A \rightarrow \alpha B$ veya $A \rightarrow \alpha B \vee \epsilon \in \text{İLK}(\beta)$ ise, İZLE(A) kümesinde yer alan tüm elemanları İZLE(B) kümesine ekle

5

İZLE Kümesi İçin Örnek

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

$\text{İLK}(S) = \{\text{sayı}, (\}$
 $\text{İLK}(S') = \{\epsilon, +\}$
 $\text{İLK}(E) = \{\text{sayı}, (\}$

1. Kural uygula: $\text{İZLE}(S) = \{\$ \}$
2. Kural uygula: $S \rightarrow ES'$ $\text{İZLE}(E) \neq \{\text{İLK}(S') - \epsilon\} = \{+\}$
 $S' \rightarrow \epsilon \mid +S$ -----
 $E \rightarrow \text{sayı} \mid (S)$ $\text{İZLE}(S) \neq \{\text{İLK}(\epsilon) - \epsilon\} = \{\$, \}$
3. Kural uygula: $S \rightarrow ES'$ $\text{İZLE}(E) \neq \text{İZLE}(S) = \{+, \$\}$
 (çünkü $S' \epsilon$ indirgenebilir)
 $\text{İZLE}(S') \neq \text{İZLE}(S) = \{\$, \}$

6

Ayrıştırma Tablosunun Oluşturulması

$\text{İLK}(S) = \{\text{sayı}, (\}$
 $\text{İLK}(S') = \{\epsilon, +\}$
 $\text{İLK}(E) = \{\text{sayı}, (\}$

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

- Her $X \rightarrow \beta$ türetim kuralı için şu adımları yürüt
 1. $\text{İLK}(\beta)$ kümesinde yer alan her simge için tablonun X satırına " $\rightarrow \beta$ " ekle
 2. Eğer $\epsilon \in \text{İLK}(\beta)$ ise, İZLE(X) kümesinde yer alan her simge için tablonun X satırına " $\rightarrow \beta$ " ekle

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

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

7

Belirsiz Gramerler

Sol rekürsif ve belirsiz gramerler için oluşturulan ayrıştırma tablolarında çelişkili girişlere rastlanabilir (aynı girişe karşı 2 veya daha fazla türetim kuralı)

$S \rightarrow S + S \mid S * S \mid \text{sayı}$

$\text{İLK}(S+S) = \text{İLK}(S*S) = \text{İLK}(\text{sayı}) = \{\text{sayı}\}$

Tablonun "S" satırını "sayı" sütünuna her üç türetim kuralı da yerleştirilmek istenecektir!

8

LL(1) Gramer

- LL(1) gramer: Ayrıştırma tablosunda tüm girişler için tek tanım oluşan gramer. Bir gramerin aşağıdaki özelliklere sahip ise, LL(1)'dir.
- $X \rightarrow \alpha \mid \beta$ iki ayrı türetim kuralı ise:
 - α ve β 'nin türetebildiği katarlar aynı "a" terminali ile başlamazlar.
 - α veya β 'dan yalnızca biri ϵ türetebilir
 - $\beta \rightarrow \epsilon$ ise, $\alpha \notin ZLE(X)$ kümesinde yer alan terminaller ile başlayan katarlar türetemez.

9

Sol-Rekürsif Gramerler

- Sol-rekürsif gramerlerle yukarıdan-aşağıya ayrıştırıcılar geliştirilemez çünkü rekürsif çağrılarının ne zaman durdurulacağı bilinemez
- Sol-rekürsif gramerler LL(1) değildir!
 - $S \rightarrow S\alpha$
 - $S \rightarrow \beta$
- Ayrıştırma tablosunun S satırının $ILK(\beta)$ 'ya karşı düşen tüm sütunlarında, her iki türetim kuralına da yer verilmek istenecektir

10

Sol-rekürsif Özelliğin Giderilmesi

- $X \rightarrow X\alpha \mid \beta$ (β X nonterminali ile başlamaz) ise, bu türetim kuralını aşağıdaki kural çiftiyle değiştir

$$X \rightarrow X\alpha \mid \beta \quad \longrightarrow \quad \begin{array}{l} X \rightarrow \beta X' \\ X' \rightarrow \alpha X' \mid \epsilon \end{array}$$

- Genel olarak:

$$\begin{array}{l} A \rightarrow X\alpha_1 \mid \dots \mid X\alpha_m \\ X \rightarrow \beta_1 \mid \dots \mid \beta_n \end{array} \quad \longrightarrow \quad \begin{array}{l} X \rightarrow \beta_1 X' \mid \dots \mid \beta_n X' \\ X' \rightarrow \alpha_1 X' \mid \dots \mid \alpha_m X' \mid \epsilon \end{array}$$

11

Örnek Sol-rekürsif gramer

Verilen grameri sol-rekürsif özelliğini giderecek şekilde dönüştürün

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

- Yeni E' nonterminalini tanımlayarak ilk türetim kuralından sol-rekürsif yapıyı kaldır
 $E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
- Aynı adımları ikinci türetim kuralına da uygula
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
- Son türetim kuralı rekürsif değildir, işleme gerek yok

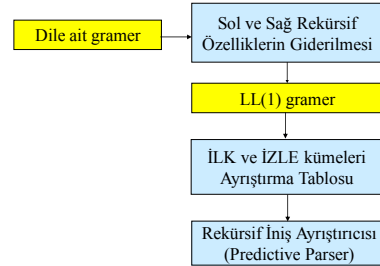
12

LL(1) Gramer Yaratmak İçin

- Sol-rekürsif gramer ile başla
 - $S \rightarrow S + E$
 - $S \rightarrow E$
- Ve sol-rekürsif özelliği giderecek algoritmayı uygula
 - $S \rightarrow ES'$
 - $S' \rightarrow +ES' \mid \varepsilon$
- Sağ-rekürsif gramer ile başla
 - $S \rightarrow E + S$
 - $S \rightarrow E$
- Ve seçeneklerin ortak bölümlerini kaldıracak algoritmayı uygula
 - $S \rightarrow ES'$
 - $S' \rightarrow +S \mid \varepsilon$

13

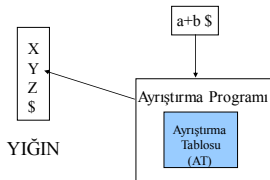
Yukarıdan Aşağıya Ayrıştırıcı



14

Rekürsif İniş Ayrıştırıcısı

- Bir yığın kullanarak gerçekleştirme



- Giriş sözcükleri \$ simgesi ile sonlandırılır
- Yığında ilk durumda \$ ve başlangıç simgesi S bulunmalıdır

15

Ayrıştırma Programı

- Ayrıştırma programı yığında yer alan X simgesi ve a giriş simgesi (terminal) ile ayrıştırma tablosuna yönelerek ayrıştırma hareketlerini denetler
- **Ayrıştırma Algoritması:**
 1. $X=a=\$$ ise, başarı durumu ilan et ve sonlan
 2. $X=a<>\$$ ise, X'i yığından çek ve giriş katarı üzerinde bir sonraki simgeye ilerle
 3. X bir nonterminal ve $AT[X,a]$ girişinde $X \rightarrow UVW$ gibi bir türetim yer alıyor ise, yığından X'i çek ve yığına W V U (U en üstte) simgelerini ekle. Eğer söz konusu giriş boş ise bir hata mesajı üret

16

Ötele-İndirge Ayrıştırması (shift-reduce)

- Ayrıştırma işlemi bir dizi “öteleme” ve “indirgeme” adımlarından oluşur
- Ayrıştırıcı durumu: Terminal ve non-terminaller içeren bir yığınla gösterilir
- Yığın ve giriş bir sonraki türetim adımını belirler

türetim adımı	yığın	taranmamış giriş
$(1+2+(3+4))+5 \leftarrow$		$(1+2+(3+4))+5$
$(E+2+(3+4))+5 \leftarrow$	(E	$+2+(3+4))+5$
$(S+2+(3+4))+5 \leftarrow$	(S	$+2+(3+4))+5$
$(S+E+(3+4))+5 \leftarrow$	(S+E	$+ (3+4))+5$
...		

21

Ötele ve İndirge İşlemleri

- Ötele:** sonraki sözcüğü yığına yerleştir

yığın	giriş	işlem
($1+2+(3+4))+5$	ötele 1
(1	$+2+(3+4))+5$	

- İndirge:** $X \rightarrow \beta$ türetim kuralı için, yığında yer alan β simgelerini X ile yer değiştir (pop β , push X)

yığın	giriş	işlem
(S+E	$+(3+4))+5$	indirge $S \rightarrow S+E$
(S	$+(3+4))+5$	

22

Ötele-İndirge Ayrıştırması

türetim adımı	yığın	giriş kafarı	işlem
$(1+2+(3+4))+5$		$(1+2+(3+4))+5$	ötele
$(1+2+(3+4))+5$	($1+2+(3+4))+5$	ötele
$(1+2+(3+4))+5$	(1	$+2+(3+4))+5$	indirge $E \rightarrow \text{sayı}$
$(E+2+(3+4))+5$	(E	$+2+(3+4))+5$	indirge $S \rightarrow E$
$(S+2+(3+4))+5$	(S	$+2+(3+4))+5$	ötele
$(S+2+(3+4))+5$	(S+	$2+(3+4))+5$	ötele
$(S+2+(3+4))+5$	(S+2	$+(3+4))+5$	indirge $E \rightarrow \text{sayı}$
$(S+E+(3+4))+5$	(S+E	$+(3+4))+5$	indirge $S \rightarrow S+E$
$(S+(3+4))+5$	(S	$+(3+4))+5$	ötele
$(S+(3+4))+5$	(S+	$+(3+4))+5$	ötele
$(S+(3+4))+5$	(S+($3+4))+5$	ötele
$(S+(3+4))+5$	(S+(3	$+4))+5$	indirge $E \rightarrow \text{sayı}$
...			

23

Çözülmesi Gereken Sorunlar

- Hangi işlemin uygulanacağına (ötele veya indirge) nasıl karar verilecek ve indirgeme hangi türetim kuralına uygulanacak
- Sorun:
 - Bazı durumlarda indirgeme işlemi yapılabilir fakat yapılmamalıdır
 - Bazı durumlarda ise, indirgeme işlemi farklı şekillerde yapılabilir, nasıl karar verilmeli

24

LR Ayırıştırıcısı

- Kullanılan yöntem
 - Bir ayırıştırma durumları kümesinden yararlanır
 - Simgeler ve durumlar içeren bir yığın kullanır
 - Örnek yığın: "1 (6 S 10 + 5" (mavi renk => durum numarası)
(siyah renk => simge)
 - Ayırıştırma Tablosu kullanarak
 - Uygulanacak bir sonraki işlemi (ötele/indirge) belirler
 - Bir sonraki durumu belirler
- Sonuç: Ayırıştırıcı adımları tablodan tam olarak belirlenir

25

LR Ayırıştırma Tablosu

	Terminaller (C)	non-terminaller
durum (S)	Sonraki işlem Ve durum	Sonraki durum
	işlem tablosu	durum tablosu

- Algoritma: O anda geçerli olan S durumu ve C giriş terminaline karşı düşen tablo girişine bak
 - Eğer $\text{Tablo}[S,C] = \text{ötele}(S')$ ise **öteleme işlemi**:
 - **push(C), push(S')**
 - Eğer $\text{Tablo}[S,C] = X \rightarrow \alpha$ ise **indirgeme işlemi**:
 - **pop(2*| α |), S' = top(), push(X), push(Tablo[S',X])**

26

LR Ayırıştırma Tablosu Örneği

Amaç: Bu tabloyu bir algoritma ile üretmek

	Giriş terminalleri					Non-terminaller	
	()	d	.	\$	S	L
Durum	1	ø3		ø2		4	
	2	S→d	S→d	S→d	S→d		
	3	ø3		ø2		7	5
	4						
	5		ø6		ø8		
	6	S→(L)	S→(L)	S→(L)	S→(L)		
	7	L→S	L→S	L→S	L→S		
	8	ø3		ø2		9	
	9	L→L,S	L→L,S	L→L,S	L→L,S		

ø: ötele

27