

فرادرس

فرا تراژیک کلاس درس
www.faradars.org

اصول طراحی کامپایلرها

مدرس:

منوچهر بابایی

کارشناس ارشد مهندسی کامپیوتر (نرم افزار)

دانشگاه کاشان

درس چهارم: تجزیه بالابه پایین

FaraDars.org

تجزیه بالابه پایین (top-down parsing)

- در تجزیه کننده‌های بالابه پایین برای تشخیص تعلق رشته ورودی به گرامر درخت تجزیه از ریشه (علامت شروع گرامر) به سمت برگها (پایانه‌ها در گرامر) ساخته می شود. در واقع ترتیب ساخت درخت، براساس اشتقاق چپ می باشد.

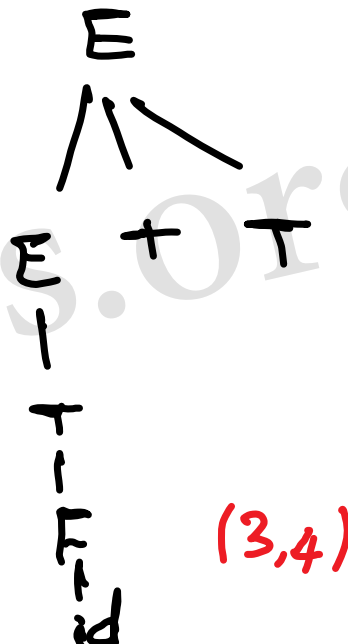
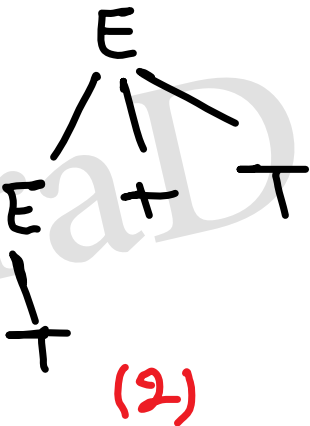
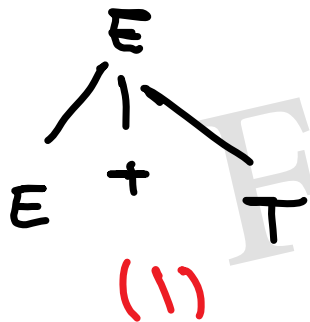
FaraDars.org

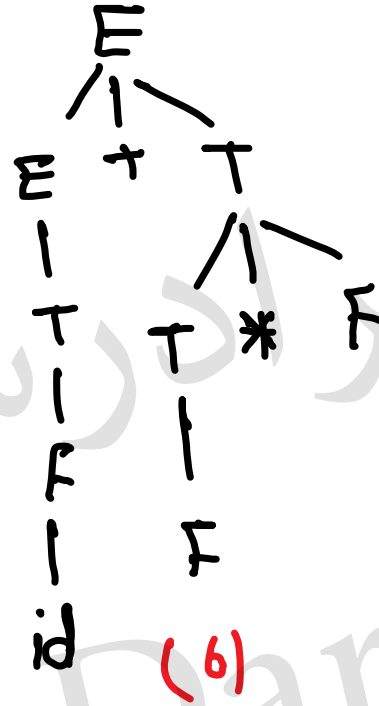
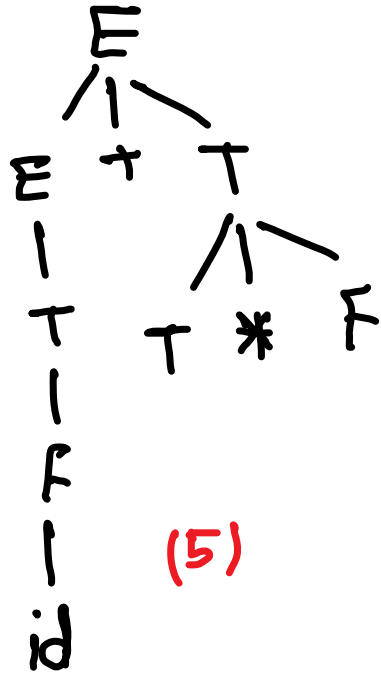
مثال: برای رشته $id+id*id$ براساس گرامر زیر تجزیه کننده بالابه پایین رسم کنید.

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid id$$



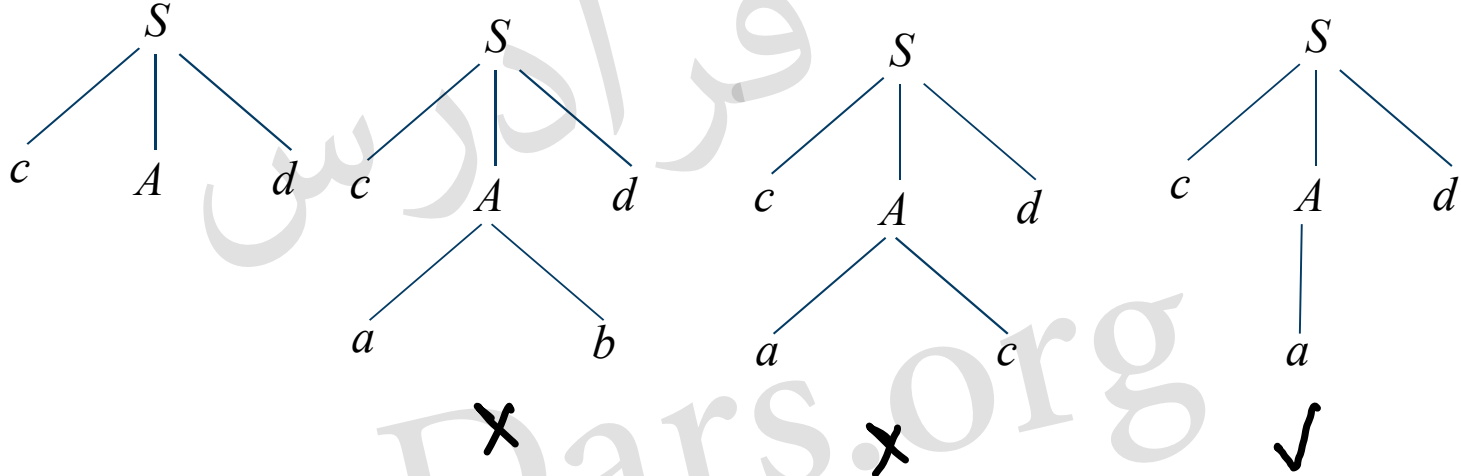


تجزیه پیشگو بازگشتی

$$S \rightarrow cAd$$

$$A \rightarrow ab \mid ac \mid a$$

رشته $\underline{c}ad \rightarrow$



First(α)

○ اگر α یک رشته از نمادهای گرامر باشد، First(α) مجموعه‌ای از پایانه‌ها خواهد بود که رشته‌های مشتق شده با آنها شروع می‌شوند. اگر $\alpha \xrightarrow{*} \lambda$ ، آنگاه λ نیز در First(α) قرار دارد.

$$A \rightarrow aA \mid B$$

$$B \rightarrow \lambda$$

$$\text{First}(A) = \{a, \lambda\}$$

$$A \rightarrow a$$

$$A \rightarrow BCD$$

$$B \rightarrow \lambda$$

$$C \rightarrow \lambda$$

$$D \rightarrow d \mid \lambda$$

نحوه محاسبه $First(x)$:

○ در صورتیکه X یک پایانه باشد $First(x) = x$

○ اگر X یک غیر پایانه و $X \rightarrow Y_1 Y_2 \dots Y_k$ مولدی برای $k \geq 1$ باشد، آنگاه اگر برای هر $a; i$ در مجموعه $First(Y_i)$ قرار

داشته باشد، و λ در تمام مجموعه‌های $First(Y_1) \dots First(Y_{i-1})$ قرار داشته باشد، آنگاه a به مجموعه $First(X)$ اضافه می‌شود.

○ اگر برای تمام مقادیر $j=1, 2, \dots, k$ رشته لاندا در مجموعه $First(Y_j)$ وجود داشته باشد آنگاه λ به $First(X)$ اضافه می‌گردد.

مثال: باتوجه به گرامرهای زیر First، رامحاسبه کنید.

$$A \rightarrow ACa \mid \lambda$$

$$C \rightarrow cC \mid \lambda$$

$$S \rightarrow bSa \mid SAS \mid A$$

$$A \rightarrow aA \mid \lambda$$

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \lambda$$

$$F \rightarrow (E) \mid id$$

$$S \rightarrow ABCbDE$$

$$A \rightarrow aA \mid ABc \mid \lambda$$

$$B \rightarrow b \mid \lambda$$

$$C \rightarrow Sc \mid \lambda$$

$$D \rightarrow d$$

$$E \rightarrow eE \mid \lambda$$

$$A \rightarrow ACa \mid \lambda$$

$$C \rightarrow cC \mid \lambda$$

$$\text{First}(A) = \{\lambda, c, a\}$$

$$\text{First}(C) = \{c, \lambda\}$$

$$S \rightarrow bSa \mid SAS \mid A$$

$$A \rightarrow aA \mid \lambda$$

$$\text{First}(S) = \{b, a, \lambda\}$$

$$\text{First}(A) = \{a, \lambda\}$$

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \lambda$$

$$F \rightarrow (E) \mid id$$

$$\text{First}(E) = \{ (, id \}$$

$$\text{First}(E') = \{ +, \lambda \}$$

$$\text{First}(T) = \text{First}(E) = \text{First}(F)$$

$$\text{First}(T') = \{ *, \lambda \}$$

$$S \rightarrow ABCbDE$$

$$A \rightarrow aA \mid ABc \mid \lambda$$

$$B \rightarrow b \mid \lambda$$

$$C \rightarrow Sc \mid \lambda$$

$$D \rightarrow d$$

$$E \rightarrow eE \mid \lambda$$

$$F(S) = \{a, b, c, \lambda\}$$

$$F(A) = \{a, b, \lambda, c\}$$

$$F(B) = \{b, \lambda\}$$

$$F(C) = \{a, b, c, \lambda\}$$

$$F(D) = \{d\}$$

$$F(E) = \{e, \lambda\}$$

FOLLOW(A)

- برای غیر پایانه A ، $FOLLOW(A)$ برابر است با مجموعه‌ای از پایانه‌ها مانند a که در هر شبه جمله، بلافاصله در سمت راست A قرار می‌گیرند. عبارت دیگر مجموعه‌ای از پایانه‌ها مانند a به شکلی که برای هر α, β یک اشتقاق بصورت $S \xrightarrow{*} \alpha A a \beta$ وجود داشته باشد.
- در صورتیکه A ، آخرین نماد در شبه جمله باشد، آنگاه $\$$ در $FOLLOW(A)$ قرار می‌گیرد.

ab\$

FOLLOW(A)

$$S \rightarrow AaS$$

$$A \rightarrow AB$$

$$B \rightarrow bB \mid \lambda$$

○ در \$ FOLLOW(S) قرار داده می شود که S نماد شروع گرامر و \$ نماد مشخص کننده انتهای سمت راست رشته ورودی است.

○ اگر مولدی به صورت $S \rightarrow \alpha B \beta$ وجود داشته باشد آنگاه هر چیزی در $FIRST(\beta)$ به جز لاندا به مجموعه FOLLOW(B) اضافه می شود.

بنابراین داریم:

$$FOLLOW(B) = FOLLOW(B) \cup \{FIRST(\beta) - \{\lambda\} \mid A \rightarrow \alpha B \beta \in G\}$$

$$A \rightarrow \alpha BC$$

$$C \rightarrow \lambda$$

FOLLOW(A)

○ اگر مولدی به صورت $A \rightarrow \alpha B$ وجود داشته باشد، یا مولدی به صورت $A \rightarrow \alpha B \beta$ که $FIRST(\beta)$ حاوی λ باشد
 ($\beta \xrightarrow{*} \lambda$)، آنگاه هر چیزی در مجموعه FOLLOW(A) به FOLLOW(B) اضافه می شود.

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \lambda$$

$$F \rightarrow (E) \mid id$$

$$Follow(E) = \{ \$,) \}$$

$$Follow(E') = \{ \$,) \}$$

$$Follow(T) = \{ +, \$,) \}$$

$$Follow(T') = \{ +,), \$ \}$$

$$Follow(F) = \{ *, +,), \$ \}$$

مثال: باتوجه به گرامرهای زیر FOLLOW، رامحاسبه کنید.

$$A \rightarrow BeC$$

$$B \rightarrow aB \mid \lambda$$

$$C \rightarrow bCA \mid c$$

$$S \rightarrow Be$$

$$B \rightarrow ABC \mid Bb \mid \lambda$$

$$A \rightarrow Aa \mid \lambda$$

$$C \rightarrow dAB \mid \lambda$$

$$E \rightarrow TE'$$

$$E' \rightarrow +E \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow T \mid \lambda$$

$$F \rightarrow PF'$$

$$F' \rightarrow *F' \mid \lambda$$

$$P \rightarrow (E) \mid a \mid b \mid c$$

$$A \rightarrow BeC$$

$$B \rightarrow aB \mid \lambda$$

$$C \rightarrow bCA \mid c$$

$$\text{Follow}(A) = \text{Follow}(c)$$

$$\text{Follow}(B) = \{e\}$$

$$\text{Follow}(c) = \text{First}(A) = \{a, e\}$$

$$\hookrightarrow \{a, e, \$\}$$

FaraDars.org

$$S \rightarrow Be$$

$$B \rightarrow ABC \mid Bb \mid \lambda$$

$$A \rightarrow Aa \mid \lambda$$

$$C \rightarrow dAB \mid \lambda$$

$$\text{Follow}(S) = \{\$ \}$$

$$\text{Follow}(B) = \{b, d, e\}$$

$$\text{Follow}(C) = \text{Follow}(B)$$

$$\text{Follow}(A) = \{a, d\}$$

$$E \rightarrow TE'$$

$$E' \rightarrow +E \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow T \mid \lambda$$

$$F \rightarrow PF'$$

$$F' \rightarrow *F' \mid \lambda$$

$$P \rightarrow (E) \mid a \mid b \mid c$$

$$\text{Follow}(E) = \{), \$ \}$$

$$\text{Follow}(E') = \{), \$ \}$$

$$\text{Follow}(T) = \{ +,), \$ \}$$

$$\text{Follow}(T') = \text{Follow}(T)$$

$$\text{Follow}(F) = \{ (, a, b, c, +,), \$ \}$$

$$\text{Follow}(F') = \{ (, a, b, c, +,), \$ \}$$

$$\text{Follow}(P) = \{ *, (, a, b, c, +,), \$ \}$$

تجزیه کننده پیشگو

$$A \rightarrow aB$$

$$B \rightarrow bB \mid \lambda$$

- در این تجزیه کننده برای هر غیر پایانه، یک تابع وجود دارد.
- براساس هر نماد ورودی از رشته، سمت راست یک غیر پایانه انتخاب می شود.

FaraDars.org

ایجاد تجزیه کننده پیشگو برای گرامر مقابل:

$$S \rightarrow A \mid B$$

$$A \rightarrow aA \mid c$$

$$B \rightarrow bB \mid d$$

سه تابع $S()$, $A()$, $B()$ را در نظر می گیریم:

Void S()

```
{  
    if(lookahead=='a' || lookahead=='c')  
        A()  
    else if(lookahead=='b' || lookahead=='d')  
        B()  
    else  
        {cout<<"Error"; }  
}
```

First(A)={a,c}

First(B)={b,d}

a b

Void A()

```
{  
    if(look ahead=='a')  
    { match('a'); A();}  
    else if(look ahead=='c')  
        match('c');  
    else {cout<<"Error";}  
}
```

Void B()

```
{  
    if(look ahead=='b')  
        { match('b'); B();}  
    else if(look ahead=='d')  
        match('d');  
    else {cout<<"Error";}  
}
```

FaraDars.org

در گرامرهای زیر نمی توان تجزیه کننده پیشگو ایجاد کرد:

α_1, α_2

$$S \rightarrow SaA \mid A$$

$$A \rightarrow aA \mid a$$

$$S \rightarrow aS \mid aA$$

$$A \rightarrow aA \mid b$$

```
Void S()
```

```
{
```

```
    S();
```

```
    match('a');
```

```
    A();
```

```
}
```


برخورد First / First:

$$A \rightarrow \alpha_1 | \alpha_2 | \dots | \alpha_n$$

$$\exists i, j; first(\alpha_i) \cap first(\alpha_j) \neq \emptyset \quad i \neq j$$

FaraDars.org

مثال: درگرامرهای زیر برخورد First/First را حذف کنید.

$$S \rightarrow aS \mid aA \mid ab$$

$$A \rightarrow bA \mid \lambda$$

$$\begin{aligned} S &\rightarrow aS' \\ S' &\rightarrow S \mid A \mid b \\ A &\rightarrow bA \mid \lambda \end{aligned}$$

$$A \rightarrow aABb \mid aAB \mid aA$$

$$B \rightarrow bB \mid aB$$

$$\begin{aligned} A &\rightarrow aAA' \\ A' &\rightarrow Bb \mid B \mid \lambda \rightarrow \begin{cases} A' \rightarrow BA'' \mid \lambda \\ A'' \rightarrow b \mid \lambda \end{cases} \\ B &\rightarrow bB \mid aB \end{aligned}$$

تجزیه کننده پیشگویی غیر بازگشتی:

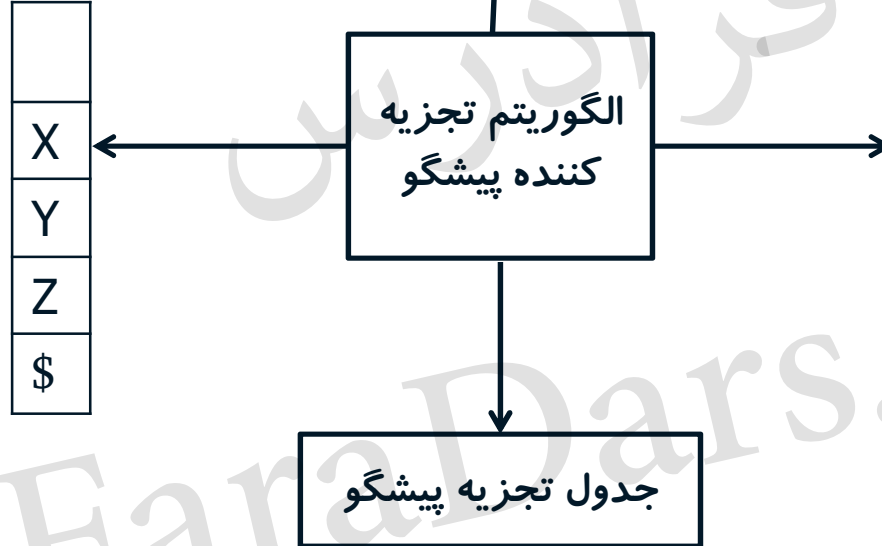
استفاده از ساختارکد و if های متوالی سرعت اجرا را پایین می آورد، لذا برای حل مشکل از جدول تجزیه استفاده می کنیم، در واقع برای هر غیر پایانه، متناسب با هر پایانه ورودی، قاعده تولید مناسب را در جدول ثبت می کنیم. و تجزیه کننده از این جدول برای انتخاب مولد مناسب استفاده می کند.

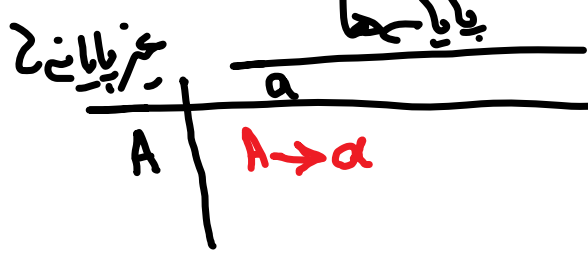
$$S \rightarrow aS \mid bS \mid \lambda$$

$$\begin{array}{cc} a & b \\ S & S \end{array} \quad \begin{array}{l} S \rightarrow aS \\ S \rightarrow bS \end{array}$$

\$طاه

a_1	a_2	a_3	a_4	...	a_n	\$
-------	-------	-------	-------	-----	-------	----





ایجاد جدول تجزیه پیشگو:

$$A \rightarrow aA$$

$$A \rightarrow BC$$

$$C \rightarrow \lambda$$

$$B \rightarrow A\alpha\lambda$$

○ برای هر مولد از گرامر به صورت $A \rightarrow \alpha$ مراحل زیر را انجام می دهیم

①

$$M[A, a] \text{ ----- } A \rightarrow \alpha$$

$$A \rightarrow \alpha \in M[A, FIRST(\alpha) - \lambda]$$

②

○ اگر λ در $FIRST(\alpha)$ باشد، $A \rightarrow \alpha$ را برای هر پایانه b در $FOLLOW(A)$ به $M[A, b]$ اضافه می کنیم.

○ اگر λ در $FIRST(\alpha)$ باشد و $\$$ در $FOLLOW(A)$ باشد، $A \rightarrow \alpha$ را به $M[A, \$]$ اضافه می کنیم.

③

$$A \rightarrow \alpha \in M[A, FOLLOW(A)] \mid \lambda \in FIRST(\alpha)$$

○ اگر با اجرای مراحل بالا مولدی در خانه $M[A, a]$ وجود نداشته باشد، آنگاه خطا اعلام می شود.

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \lambda$$

$$F \rightarrow (E) \mid id$$

جمله تجزیه
بایاها

عربیاینها	<i>id</i>	+	*	()	\$
<i>E</i>	$E \rightarrow TE'$	(1)		$E \rightarrow TE'$	(2)	(3)
<i>E'</i>		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
<i>T</i>	$T \rightarrow FT'$			$T \rightarrow FT'$		
<i>T'</i>		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
<i>F</i>	$F \rightarrow id$			$F \rightarrow (E)$		

$$A \rightarrow CB \mid \lambda$$

$$B \rightarrow cCB \mid \lambda$$

$$C \rightarrow ED$$

$$D \rightarrow aED \mid \lambda$$

$$E \rightarrow b \mid (A)$$

$$F(A) = \{b, c, \lambda\}$$

$$F(B) = \{c, \lambda\}$$

	a	b	c	()	\$
A		$A \rightarrow CB$		$A \rightarrow CB$	$A \rightarrow \lambda$	$A \rightarrow \lambda$
B			$B \rightarrow cCB$		$B \rightarrow \lambda$	$B \rightarrow \lambda$
C			$C \rightarrow ED$		$C \rightarrow ED$	
D	$D \rightarrow aED$	$D \rightarrow \lambda$		$D \rightarrow \lambda$	$D \rightarrow \lambda$	
E		$E \rightarrow b$		$E \rightarrow (A)$		



الگوریتم تجزیه غیر بازگشتی پیشگو

- در صورتیکه رشته w در $L(G)$ باشد، این الگوریتم سمت چپ ترین اشتقاق را برای w برگشت می دهد و در غیر این صورت اعلام خطا می کند.
- در ابتدا علامت $\$$ به انتهای رشته w اضافه می شود و در بافر ورودی قرار می گیرد. و نماد شروع گرامر در بالای پشته و بالای $\$$ قرار می گیرد
- اگر نماد بالای پشته X و توکن جاری را a بنامیم، آنگاه تجزیه کننده، تا هنگامیکه پشته خالی نشده است:
(الف) اگر X پایانه بود و با توکن a برابر بود آنگاه از بالای پشته حذف شده، و a با توکن بعدی مقداردهی می شود.
- (ب) اگر X یک غیر پایانه بود و $M[X, a]$ در جدول تجزیه تهی بود آنگاه اعلام خطا می شود

ج) اگر X برابر یک غیرپایانه بود و $M[X, a]$ برابر قاعده $y_1 y_2 \dots y_k \rightarrow X$ باشد، آنگاه X را از بالای پشته حذف کرده و به جای آن $y_1, y_2, \dots, y_{k-1}, y_k$ را قرار می دهیم.

د) اگر X پایانه بود و با توکن a یکسان نبود اعلام خطا می شود.

FaraDars.org

Id+id*id\$

قاعده	ورودی	پشته	تطبیق داده شده
	$id + id * id\$$	$E\$$	
$E \rightarrow TE'$	$id + id * id\$$	$TE' \$$	id
$T \rightarrow FT'$	$id + id * id\$$	$FT' E' \$$	id
$F \rightarrow id$	$id + id * id\$$	$idT' E' \$$	id
id	$+id * id\$$	$T' E' \$$	
$T' \rightarrow \lambda$	$+id * id\$$	$E' \$$	
$E' \rightarrow +TE'$	$+id * id\$$	$+TE' \$$	
$+$	$id * id\$$		$id +$
$T \rightarrow FT'$	$id * id\$$	$FT' E' \$$	$id +$
$F \rightarrow id$	$id * id\$$	$idT' E' \$$	$id + id$
id	$*id\$$	$T' E' \$$	
$T' \rightarrow *FT'$	$*id\$$	$*FT' E' \$$	

قاعده	ورودی	پشته	تطبیق داده شده
$*$	$id\$$	$FT 'E' \$$	$id + id *$
$F \rightarrow id$	$id\$$	$idT 'E' \$$	$id + id *$
id	$\$$	$T 'E' \$$	$id + id * id$
$T' \rightarrow \lambda$	$\$$	$E' \$$	$id + id * id$
$E' \rightarrow \lambda$	$\$$	$\$$	$id + id * id$

گرامر LL(1)

یک گرامر LL(1) است، اگر برای هر توکن ورودی در اشتقاق چپ به راست، بتوان غیرپایانه بعدی را برای گسترش انتخاب نمود. و در واقع عدد یک نشان دهنده این است، که فقط با دیدن یک توکن کافیت تا قانون تولید بعدی را انتخاب نمود. بطور کلی میتوان گفت که گرامر LL(1) مشخصات زیر را دارد:

۱- فاقد برخورد FIRST/FIRST است.

۲- فاقد برخورد FOLLOW/FIRST باشد.

$$A \rightarrow \alpha / \beta$$

$$\beta \xrightarrow{*} \gamma$$

$$\text{First}(\alpha) \cap \text{Follow}(A) \neq \emptyset$$

$$A \rightarrow aB \mid \lambda$$

$$B \rightarrow Aa$$

A

a

$$A \rightarrow aB$$

$$A \rightarrow \lambda$$

این گرامر بدلیل رجس

Follow / First

LL(1) نیست.

مثال: کدام گرامر $LL(1)$ است؟

$$S \rightarrow as \mid A \mid \lambda$$

$$A \rightarrow bSc$$

$$A \rightarrow BCD \mid b \mid \lambda$$

$$B \rightarrow CC \mid eB \mid dC$$

$$C \rightarrow c \mid \lambda$$

$$D \rightarrow dD \mid \lambda$$

$$\checkmark S \rightarrow as \mid A \mid \lambda$$

$$\checkmark A \rightarrow bSc$$

$$\text{First}(as) = a$$

$$\text{First}(A) = b$$

\rightarrow

$$\text{Follow}(s) = c$$

$$\rightarrow \text{Follow}(s) \cap \text{First}(s) = \emptyset$$

این گرامر $LL(1)$ است.

این گرامر

- ✗ $A \rightarrow BCD \mid b \mid \lambda$
- ✗ $B \rightarrow CC \mid eB \mid dC$
- ✗ $C \rightarrow c \mid \lambda$
- $D \rightarrow dD \mid \lambda$

گرامر $LL(1)$ نیست

$$\text{First}(BCD) = \{e, d, c, \lambda\}$$

- در گرامر یک قاعده به شکل $S \rightarrow \alpha \mid \beta$ داریم که α ، β ، λ را تولید می کنند پس گرامر

شرط $LL(1)$ بودن را ندارد

$$\text{First}(CC) = \{c, \lambda\}$$

$$c \in \text{Follow}(B)$$

$$\rightarrow \text{First}(B) \cap \text{Follow}(B) \neq \emptyset$$

مثال: برای گرامر زیر جدول تجزیه پیشگوی بازگشتی تشکیل دهید؟

$$A \rightarrow aB.Cb$$

$$B \rightarrow CBb \mid \lambda$$

$$C \rightarrow c \mid \lambda$$

این گرامر LL(1) است

A a b c $\$$

$A \rightarrow aBCb$

B

$B \rightarrow CBb$
 $B \rightarrow \lambda$

$B \rightarrow CBb$
 $B \rightarrow \lambda$

C

$C \rightarrow \lambda$

$C \rightarrow c$
 $C \rightarrow \lambda$

گرامر مبهم :

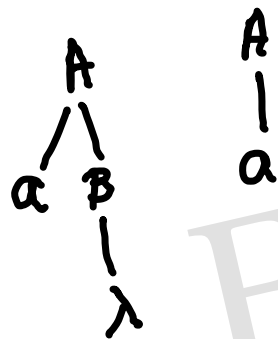
$$A \rightarrow aB \mid a$$

$$B \rightarrow \lambda$$

مثال: در مورد گزاره های زیر اظهار نظر کنید.

✓ الف) گرامرهای مبهم، $LL(1)$ نیستند.

✓ ب) گرامرهای دارای بازگشتی چپ $LL(1)$ نیستند.



نتیجه: هر گرامر که $LL(1)$ باشد مبهم نیست.

$$A \rightarrow Aa|b$$

$$b \in \text{First}(Aa)$$

$$\text{First}(b) = b$$

$$A \rightarrow A\alpha|\beta$$

$$\text{First}(\beta) \in \text{First}^+(A\alpha)$$

$$\rightarrow \text{First}(\beta) \cap \text{First}(A\alpha) \neq \emptyset$$

مثلاً $LL(1)$ بودن را نقض می کند.

$$A \rightarrow A\alpha \mid \lambda \quad ?$$

$$\text{First}(A) = \text{First}(\alpha)$$

$$\text{Follow}(A) = \text{First}(\alpha)$$

در این قاعده تولید با بازگشتی هیچ برخورد
وجود ندارد لذا $(1) LL$ است.

FaraDars.org

ایجاد گرامر $LL(1)$

با اعمال زیر میتوان گرامر غیر $LL(1)$ را به $LL(1)$ تبدیل نمود:

۱- حذف بازگشتی چپ

۲- فاکتورگیری چپ

۳- رفع ابهام از گرامر مبهم

۴- نوشتن گرامر جدید برای زبان مربوط به گرامر غیر $LL(1)$

مثال: گرامرهای زیر را به گرامر LL(1) تبدیل کنید؟

$$A \rightarrow BCD \mid \lambda$$

$$B \rightarrow Aa \mid b$$

$$C \rightarrow c \mid \lambda$$

$$D \rightarrow Cd$$

$$A \rightarrow aA \mid B$$

$$B \rightarrow bB \mid a$$

$$S \rightarrow S * S \mid S + S \mid x$$

$$A \rightarrow BCD \mid \lambda$$

$$B \rightarrow Aa \mid b$$

$$C \rightarrow c \mid \lambda$$

$$D \rightarrow Cd$$

BCD

AaCD

BCDAaCD

bcdacd

$$A \rightarrow \overbrace{Aa}^{\alpha} \overbrace{CD}^{\beta_1} \mid \overbrace{\lambda}^{\beta_2}$$

↓

$$A \rightarrow b CDA' \mid A'$$

$$A' \rightarrow a CDA' \mid \lambda$$

$$C \rightarrow c \mid \lambda$$

$$D \rightarrow Cd$$

$$A \rightarrow bcDA' \mid bDA' \mid A'$$

$$A' \rightarrow acDA' \mid aDA' \mid \lambda$$

$$\rightarrow D \rightarrow cd \mid d$$

$$A \rightarrow bB \mid A'$$

$$B \rightarrow cDA' \mid DA'$$

$$A' \rightarrow aB \mid \lambda$$

$$D \rightarrow cd \mid d$$



$$A \rightarrow bB \mid A'$$

$$B \rightarrow ccdA' \mid cdA' \mid dA'$$

$$A' \rightarrow aB \mid \lambda$$

$$A \rightarrow bB \mid A'$$

$$B \rightarrow cC \mid dA'$$

$$C \rightarrow cdA' \mid dA'$$

$$A' \rightarrow aB \mid \lambda$$

$$A \rightarrow bB \mid A'$$

$$B \rightarrow cC \mid dA'$$

$$C \rightarrow cdA' \mid dA'$$

$$A' \rightarrow \epsilon \mid \lambda$$

...
b c d a c d \$

\$A

\$Bb

\$B

\$Cc

\$C

\$A'd

\$A'

\$Ba

\$B

\$Cc

\$C

\$A'd

\$A'

\$

	a	b	c	d	\$
A	$A \rightarrow A'$	$A \rightarrow bB$			$A \rightarrow A'$
B			$B \rightarrow cC$	$B \rightarrow dA'$	
C			$C \rightarrow cdA'$	$C \rightarrow dA'$	
A'	$A' \rightarrow \epsilon$			$A' \rightarrow \lambda$	

$$A \rightarrow aA \mid B$$

$$B \rightarrow bB \mid a$$

$$A \rightarrow aA \mid bB \mid a$$

$$B \rightarrow bB \mid a$$

$a\$$
 $\$A'a$
 $\$A'$
 $\$$

$$A \rightarrow aA' \mid bB$$

$$A' \rightarrow A \mid \lambda$$

$$B \rightarrow bB \mid a$$

A

$$A \rightarrow aA'$$

$$A \rightarrow bB$$

A'

$$A \rightarrow aA'$$

$$A \rightarrow bB$$

$$A' \rightarrow \lambda$$

B

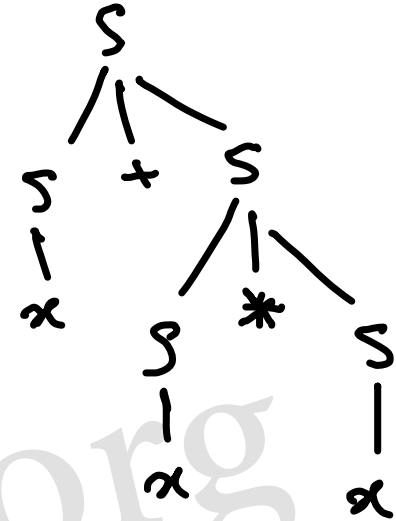
$$B \rightarrow a$$

$$B \rightarrow bB$$

$x + x * x$

$S \rightarrow S * S \mid S + S \mid x$

کدام صحیح



$$A \rightarrow A + B \mid B$$

Red annotations: A red arrow points from the first 'A' to the '+' sign. A red arc labeled α is above the '+ B' part. A red β is above the 'B' part.

$$B \rightarrow B * C \mid C$$

Red annotation: A red arrow points from the first 'B' to the '*' sign.

$$C \rightarrow \epsilon$$

LL(1) سیت

$$A \rightarrow BA'$$

$$A' \rightarrow +BA' \mid \lambda$$

$$B \rightarrow CB'$$

$$B' \rightarrow *CB' \mid \lambda$$

$$C \rightarrow \epsilon$$

LL(1)

لرآمد

	+	*	x	\$
A			$A \rightarrow BA'$	
A'	$A' \rightarrow +BA'$		$A' \rightarrow \lambda$	
B			$B \rightarrow CB'$	
B'	$B' \rightarrow \lambda$	$B' \rightarrow *CB'$	$B' \rightarrow \lambda$	
C			$C \rightarrow x$	

x	+	x	*	x	\$
\$A					\$A'B'
\$A'B					\$A'B'C*
\$A'B'C					\$A'B'C
\$A'B'x					\$A'B'x
\$A'B'					\$A'B'
\$A'					\$A'
\$A'B+					\$
\$A'B					
\$A'B'C					
\$A'B'x					

رفع خطا در تجزیه کننده

تحلیلگر نحوی پس از برخورد با خطا بصورت زیر عمل می کند:

۱-توقف تجزیه: در این روش تجزیه کننده، عمل تجزیه را متوقف کرده و پیام خطا صادر می کند. در واقع

در این روش برای هر خطا عمل تجزیه متوقف می شود.

۲-پوشش خطا: در این روش تجزیه کننده، از روی خطا عبور کرده و تمامی خطاها را کشف می کند. برای پوشش

خطا روش های زیر وجود دارد:

(الف) خطا در محل وقوع آن اصلاح شود، به طوریکه تجزیه کننده به عمل تجزیه ادامه بدهد.

(ب) توکن هایی از ورودی نادیده گرفته می شوند، تا به یک ساختار صحیح نحوی برسد و تجزیه کننده بتواند از روی آن

عبور کند.

$x = x + y$;

$z + + ;$

استراتژی های پوشش خطا

۱- پوشش Panic-Mode: در هنگام تشخیص خطا، در هر مرحله از یک نماد صرف نظر می کنند تا زمانی که به یک هماهنگ کننده برسد. هماهنگ کننده ها در هر زبان معمولاً خاتمه دهنده ها مانند $\}$ یا $\}$ هستند.

۲- پوشش Phrase-Level: در این روش، در مواجهه با خطا، تصحیح موضعی انجام می شود، یعنی با درج یا حذف یک یا چند نماد از بالای پشته و یا از یک یا چند توکن از رشته ورودی، امکان ادامه کار تجزیه کننده وجود دارد.

۳- پوشش Error-Production: در این روش به گرامر علاوه بر قواعد اصلی یکسری قواعد اضافه می شود، که ساختارهای مولد خطا را تولید می کنند. لذا اگر برای تجزیه یک عبارت از توکن ها، از یک یا چند مولد خطا استفاده شود. تجزیه کننده می تواند خطا را کشف کرده و به تجزیه ادامه دهد.

نحوه پوشش خطا در تجزیه کننده پیشگوی غیربازگشتی

یکی از روش ها، پوشش Panic-mode است، در این روش توکن ورودی تا رسیدن به یک توکن از مجموعه هماهنگ کننده، نادیده گرفته می شوند، که اساس کار بدین صورت است:

۱- اگر پایانه a بالای پشته قرار داشته باشد که با توکن جاری مغایرت دارد، آنگاه خطا پیش می آید، آنگاه به منظور پوشش خطا، پایانه a از بالای پشته حذف می شود. که در انتها در پیغام خطای فقدان a به برنامه نویس داده می شود.

$C \rightarrow c\beta$

abc
xx

۲- اگر بالای پشته یک غیرپایانه α بود و در جدول تجزیه خانه متناظر با این غیرپایانه و توکن جاری خالی بود، آنگاه:

الف) نمادهای $FIRST(\alpha)$ را به عنوان هماهنگ کننده A در نظریه گیریم، چون ممکن است امکان ادامه تجزیه بر اساس α ، اگر یک نماد از $FIRST(\alpha)$ در ورودی ظاهر شود، وجود خواهد داشت.

ب) نمادهای $FOLLOW(\alpha)$ را به عنوان هماهنگ کننده α در نظریه گیریم، اگر توکن‌ها تا زمان دیده شدن یک عضو از $FOLLOW(\alpha)$ ، نادیده گرفته شوند و سپس α از بالای پشته حذف گردد، آنگاه انتظار می‌رود که تجزیه ادامه یابد.

$A \rightarrow abA \mid B$

$B \rightarrow cB \mid \lambda$

$a.b.a.c.c.\$$

$\$A$

$\$Aba$

$\$Ab$

$\$A$

$\$Aba$

$\$Ab_x$

$\$A$

$\$B$

$\$Bc$

$\$B$

$\$Bc$

$\$B$

$\$$

	a	b	c	\$
A	$A \rightarrow abA$		$A \rightarrow B$	$A \rightarrow B$
B			$B \rightarrow cB$	$B \rightarrow \lambda$

$$A \rightarrow aBc$$

$$B \rightarrow bB \mid \lambda$$

abbaac
...xx.

	a	b	c
A	$A \rightarrow aBc$		
B	—	$B \rightarrow bB$	$B \rightarrow \lambda$

\$A	\$cBb
\$cBa	\$cB
\$cB	\$c
\$cBb	\$
\$cB	

آیا این استراتژی «بخش ب» بهینه است؟

```
i=2*j ;  
if(x==y)  
    i++;
```

FaraDars.org

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \lambda$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \lambda$$

$$F \rightarrow (E) \mid id$$

جمله تجزیه
پایاها

عربیاینها	id	$+$	$*$	$($	$)$	$\$$
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow \lambda$	$E' \rightarrow \lambda$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow \lambda$	$T' \rightarrow *FT'$		$T' \rightarrow \lambda$	$T' \rightarrow \lambda$
F	$F \rightarrow id$			$F \rightarrow (E)$		

) id * + id \$
x

\$E

\$E'T

\$E'T'F

\$E'T'

\$E'T'F*

pop(f) → \$E'T'F

\$E'T'

\$E'

\$E'T+

\$E'T

\$E'T'F

\$E'T'id

\$E'T'

\$E'

\$

مثال: آیا گرامر زیر LL(1) است؟

$$X \rightarrow Y a Y b \mid Z b Z a$$

$$Y \rightarrow a \mid \lambda$$

$$Z \rightarrow b \mid \lambda$$

$$\text{First}(Y a Y b) = \{a\}$$

$$\text{First}(Z b Z a) = \{b\}$$

برخورد First/Follow وجود دارد.

$$\begin{cases} a \in \text{First}(Y) \\ a \in \text{Follow}(Y) \end{cases} \rightarrow \text{First}(Y) \cap \text{Follow}(Y) \neq \emptyset$$

گرامر LL(1) نیست

مثال: برای گر امر زیر جدول تجزیه رسم کنید؟

$$S \rightarrow [SX] | a$$

$$X \rightarrow +SY | Yb | \lambda$$

$$Y \rightarrow -SXc | \lambda$$

FaraDars.org

یا یا نه ها
 غیر یا یا نه ها
 $S \rightarrow [X]$
 $S \rightarrow a$
 $X \rightarrow \lambda$
 $X \rightarrow +SY$
 $X \rightarrow Yb$
 $X \rightarrow \lambda$
 $Y \rightarrow \lambda$
 $Y \rightarrow -SXc$
 $Y \rightarrow \lambda$

FaraDars.org

این اسلاید ها بر مبنای نکات مطرح شده در فرادرس
«آموزش طراحی کامپایلر»
تهیه شده است.

برای کسب اطلاعات بیشتر در مورد این آموزش به لینک زیر مراجعه نمایید
faradars.org/fvsft104