

فرادرس

فرا ترازی یک کلاس درس
www.faradars.org

اصول طراحی کامپایلرها

مدرس:

منوچهر بابایی

کارشناس ارشد مهندسی کامپیوتر (نرم افزار)

دانشگاه کاشان

درس پنجم: تجزیه پایین به بالا

FaraDars.org

$$S \rightarrow aBa$$

$$B \rightarrow bB \mid b$$

$\begin{array}{c} aBa \\ \underline{B} \\ S \end{array}$

تجزیه پایین به بالا

- تجزیه کننده پایین به بالا ساخت درخت تجزیه را از برگها شروع می کند و به سمت ریشه پیش می رود. که برگ ها توکن های ورودی و ریشه نماد شروع گرامر است.
- در این تجزیه کننده در هر مرحله افزایش کاهش رشته ورودی به سمت ریشه، یک زیر رشته خاص که با سمت راست یک قاعده تولید منطبق است، با نماد سمت چپ، آن قاعده جایگزین می گردد.
- ترتیب بکارگیری قواعد، در تجزیه کننده پایین به بالا عکس ترتیب بکارگیری قواعد در اشتقاق راست می باشد.

$$\begin{array}{c} aBa \\ aBa \\ S \end{array}$$

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

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

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

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

$$id + id * id$$

$$id + id * F$$

$$id + F * F$$

$$id + T * F$$

$$id + T$$

$$id + T$$

$$F + T$$

$$T + T$$

$$E + T$$

$$E$$

تجزیه پایین به بالا - اشتقاق راست

$S \rightarrow AB$
 $A \rightarrow aAC \mid a$
 $B \rightarrow bBC \mid b$
 $C \rightarrow c$

اشتقاق راست

S
 AB
 $AbBC$
 $AbBc$
 $Abbc$
 $aACbbc$
 $aAcbbc$
 $aacbbc$

تجزیه پایین به بالا

$aacbbc$
 $aAcbbc$
 $aACbbc$
 $Abbc$
 $AbBc$
 $AbBC$
 AB
 S

دستگیره

- اگر ترتیب کاهش رشته ورودی به سمت نماد شروع گرامر، معکوس سمت راست ترین اشتقاق باشد، آنگاه دنباله ای که در هر مرحله به یک غیر پایانه در سمت چپ کاهش می یابد، دستگیره (Handle) نام دارد

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

$$id + id$$

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

$$F \rightarrow id$$

استفادۀ راست

E
E + T
E + F
E + id
T + id
F + id
id + id

گزینۀ پایین به بالا

id + id
F + id
T + id
E + id
E + F
E + T
E

Handle

id
F
T
id
F
E + T

مثال: دستگیره‌ها را برای تجزیه پایین به بالای رشته abbaaaabc مشخص کنید

$$S \rightarrow aABc$$

$$A \rightarrow bAa \mid ba$$

$$B \rightarrow aBb \mid a$$

اشتقاق راست

S

aABc

aAaBbc

aAaabc

abAaabc

abbaaabc

جزء پایین به بالا

a bbaaabc

ab Aaabc

aA abc

aA aBbc

aABc

S

دستگیره

b a

b A a

a

a B b

a A B c

دستگیره : دنباله‌ای از پایه‌ها و غیر پایه‌ها است که درست راست یکی قاعده تولید وجود دارد، و کاهش آن به سمت چپ قاعده تولید یکی مرحله از مراحل معکوس راست‌ترین اشتقاق است.

FaraDars.org

گرامر مبهم: دو درخت گوناگون
مربوط به یک رشته

$$E \rightarrow E + E$$

$$E \rightarrow E * E$$

$$E \rightarrow (E) | id$$

E

$$E + E$$

$$E + E * E$$

$$E + E * id$$

$$E + id * id$$

$$id + id * id$$

E

$$E * E$$

$$E * id$$

$$E + E * id$$

$$E + id * id$$

$$id + id * id$$

مثال: دستگیره‌ها را برای رشته $id + id * id$ مشخص کنید.

دستیابی

بازیابی

E

$E + E$

$E + E * E$

$E + E * id$

$E + id * id$

$id + id * id$

$id + id * id$

$E + id * id$

$E + E * id$

$E + E * E$

$E + E$

E

id

id

id

$E * E$

$E + E$

گزینه ها

id

id

E + E

id

E * E

گزینه های بین به بالا

id + id * id

E + id * id

E + E * id

E * id

E * E

E

E

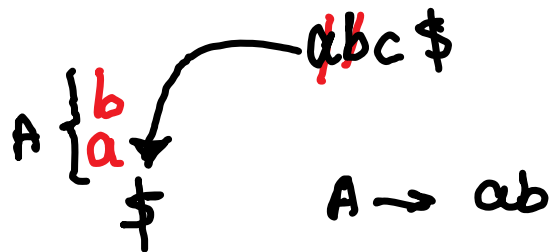
E * E

E * id

E + E * id

E + id * id

id + id * id



تجزیه انتقال- کاهش

○ در تجزیه انتقال-کاهش، یک پشته نمادهای گرامر را نگهداری می کند، و بخشی از رشته ورودی که هنوز تجزیه نشده است در بافر ورودی قرار می گیرد.

○ در روند پویش از چپ به راست در رشته ورودی، تجزیه کننده چندین نماد را به پشته انتقال می دهد. و این عمل تا زمانی که یک دستگیره β در بالای پشته ظاهر شود ادامه می یابد. سپس β با سمت چپ مولد مناسب کاهش می یابد. این عمل تا زمانی که خطایی پیدا شود یا پشته شامل نماد شروع و ورودی تهی شود، ادامه می یابد.

b
a
:
:
\$

ab

○ در تجزیه کننده های انتقال- کاهش چهار عمل وجود دارد: ۱- انتقال ۲- کاهش ۳- پذیرش ۴- خطا

۱- در عمل انتقال، توکن بعدی ورودی به بالای پشته منتقل می شود.

۲- در عمل کاهش، تجزیه کننده می داند که انتهای سمت راست دستگیره در بالای پشته قرار دارد، سپس باید سمت چپ دستگیره را در پشته یافته و تصمیم بگیریم با کدام غیر پایانه دستگیره جایگزین شود.

۳- در عمل پذیرش، تجزیه کننده تکمیل موفقیت آمیز عمل تجزیه را اعلام می کند.

۴- در عمل خطا هنگام خطای نحوی، رویه پوشش خطا فراخوانی می شود.

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

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

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

$\begin{array}{r} id_1 * id_2 \\ \hline E \\ T \\ T * F \\ T * id_2 \\ F * id_2 \\ id_1 * id_2 \end{array}$	$\begin{array}{r} \text{گزینہ} \\ \hline id_1 * id_2 \\ F * id_2 \\ T * id_2 \\ T * F \\ T \\ E \end{array}$
---	--

پشتہ	ورودی	عمل
S	$id_1 * id_p S$	انتقال
$S id_1$	$* id_p S$	$F \rightarrow id$ کاهش
SF	$* id_p S$	$T \rightarrow F$ کاهش
ST	$* id_p S$	انتقال
$ST *$	$id_p S$	انتقال
$ST * id_p$	S	$F \rightarrow id$ کاهش
$ST * F$	S	$T \rightarrow T * F$ کاهش
ST	S	$E \rightarrow T$ کاهش
SE	S	پذیرش

مثال: باتوجه به گرامر زیر، رشته dbca را تجزیه کنید.

$S \rightarrow Aa$

$A \rightarrow Bb \mid BCc$

$B \rightarrow d$

$C \rightarrow b$

dbca
S
Aa
BCca
Bbca
dbca

تجزیه
dbca
Bbca
BCca
Aa
S

لِئَ	درودی	عمل
\$	d bca\$	انتقال
\$d	bca\$	کاهش $B \rightarrow d$
\$B	bca\$	انتقال
\$Bb	ca\$	کاهش $C \rightarrow b$
\$BC	ca\$	انتقال
\$BCc	a\$	کاهش $A \rightarrow BCc$
\$A	a\$	انتقال
\$Aa	\$	کاهش $S \rightarrow Aa$
\$S	\$	Accept

برخوردها در تجزیه‌کننده‌های پایین به بالا

- در خلال عمل تجزیه در تجزیه‌کننده‌های انتقال-کاهش در انتخاب عمل صحیح ممکن است با برخوردهای زیر مواجه شویم:

۱- برخورد انتقال-کاهش

۲- برخورد کاهش-کاهش

برخورد انتقال-کاهش

لرزم مجسم

$S \rightarrow \text{if } E \text{ then } S$

| $\text{if } E \text{ then } S \text{ else } S$

| x

$E \rightarrow y$

if y then if y then x else x

اشتقاق راست:

تجزیه پایین به بالا:

If E then S

If E then if E then S else S

If E then if E then S else x

If E then if E then x else x

If E then if y then x else x

If y then if y then x else x

If y then if y then x else x

If E then if y then x else x

If E then if E then x else x

If E then if E then S else x

If E then if E then S else S

If E then S

S

درودی

شبه

\$
:
.

if y then if y then x else x \$

:

\$if E then if E then S

else x \$

دستگیره

کافی



\$if E then S

انتقال



if E then if E then S else x

- اگر عمل کاهش انجام شود، آنگاه val مربوط به iF اول خواهد بود.
- اگر عمل انتقال انجام شود. آن گاه val مربوط به iF دوم خواهد بود.

FaraDars.org

$A \rightarrow ab|...-$

$B \rightarrow -|ab$

برخورد کاهش-کاهش

○ برخورد کاهش به کاهش به صورت می تواند اتفاق بیفتد:

الف) اگر در بالای پشته یک دستگیره وجود داشته باشد، که سمت راست بیش از یک مولد باشد، آنگاه نمی توان سمت چپ مناسب را برای کاهش انتخاب نمود.

ب) اگر در یک انتقال، دو دستگیره در بالای پشته ظاهر شود، آنگاه برخورد کاهش کاهش پیش می آید.

αAb

$S \rightarrow id(PL)$

$S \rightarrow E := E$

$PL \rightarrow PL, P \mid P$

$P \rightarrow id$

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

$EL \rightarrow EL, E \mid E$

پشته

...id(id

ورودی

,id)...

$P(i, z)$

$id(id, id)$

روش تجزیه تقدم عملگر (Operator Precedence Parsing)

- این تجزیه کننده بیشتری برای عبارات محاسباتی مناسب است.
- در این روش پایانه ها بعنوان عملگر در نظر گرفته می شوند.
- در تجزیه کننده از تقدم عملگرها برای تعیین دستگیره استفاده می شود.
- این تجزیه کننده فقط برای گرامرهای عملگر قابلیت استفاده دارد.

گرامرهای عملگر

○ یک گرامر مستقل از متن، گرامر عملگر است در صورتیکه دارای شرایط زیر را دارا باشد:

$$S \rightarrow AB \quad \times$$

الف) قاعده λ نداشته باشد.

ب) در سمت راست هیچ قاعده ای بیش از یک غیر پایانه در کنار هم نباشند.

ج) بین هر دو پایانه حداکثر یک رابطه تقدم وجود داشته باشد.

○ در صورتیکه دو پایانه کنار هم باشند و یا در بین آنها فقط یک غیر پایانه باشد، باهم رابطه دارند.

$$S \rightarrow asb$$

$$S \rightarrow ab$$

$$E \rightarrow E + E \mid E * E$$

$$E + E$$

$$E * E$$

$$E + E * E$$

$$E \rightarrow (E)$$

$$((E))$$

مشخص کردن رابطه بین عملگرها

$$E \rightarrow E + E \mid E * E \mid id$$

$$E + E * E$$

$$E * E + E$$

$+, * \rightarrow$	$+ < . *$	$* . > +$
$id, + \rightarrow$	$id . > +$	$+ < . id$
$+, + \rightarrow$	$+ . > +$	
$*, * \rightarrow$	$* . > *$	

$$), (\times$$

$$((\rightarrow (< . ($$

$$)) \rightarrow) . >)$$

اولویت پایانه‌ها :

۱- روابط شرکت پذیری

۲- تقدم عملگرها

یادداشت‌های سمت راست ←

id	*	+
<.	>.	>.
<.	>.	>.
—	>.	>.

↑
یادداشت‌های سمت چپ

FaraDars.org

$(E + T)$

$E + T \$$

$(())$

جدول تجزیه تقدم عملگر

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$

$+ A ($

$(A +$

$) +$

$) ($

$\rightarrow) 2 (\times$

	$+$	$*$	$($	$)$	id	$\$$
$+$	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$	$< \cdot$	$\cdot >$
$*$	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$	$< \cdot$	$\cdot >$
$($	$< \cdot$	$< \cdot$	$< \cdot$	$\cdot =$	$< \cdot$	$-$
$)$	$\cdot >$	$\cdot >$	$-$	$\cdot >$	$-$	$\cdot >$
id	$\cdot >$	$\cdot >$	$-$	$\cdot >$	$-$	$\cdot >$
$\$$	$< \cdot$	$< \cdot$	$< \cdot$	$-$	$< \cdot$	$\cdot =$

\$ a b c \$

الگوریتم تجزیه تقدم عملگر

\$

- در ابتدای و انتهای رشته ورودی \$ قرار می دهیم
 - بین پایانه ها رابطه را مشخص می کنیم.
 - ابتدا تجزیه کننده علامت \$ را در پشت خالی قرار می دهد، پایانه بالای پشته (u) و توکن ورودی (v) را در نظر بگیریم و با توجه به جدول تجزیه تقدم عملگرها بصورت زیر عمل می کنیم:
- (۱) اگر $v < u$ باشد، آنگاه عمل انتقال انجام می شود، ابتدا رابطه $<$ و بعد از آن v وارد پشته می شود
 - (۲) اگر $u = v$ باشد آنگاه فقط v بالای پشته نوشته می شود.
 - (۳) اگر $u > v$ باشد، آنگاه عمل کاهش انجام می شود، در واقع کل عبارت بین $< .. >$ (دستگیره) حذف می شود.
 - (۴) اگر بین u, v هیچ رابطه ای وجود نداشته باشد آنگاه خطای نحوی اعلام می شود.

مثال: تجزیه تقدم عملگر را برای رشته $id+id*id$ انجام دهید.

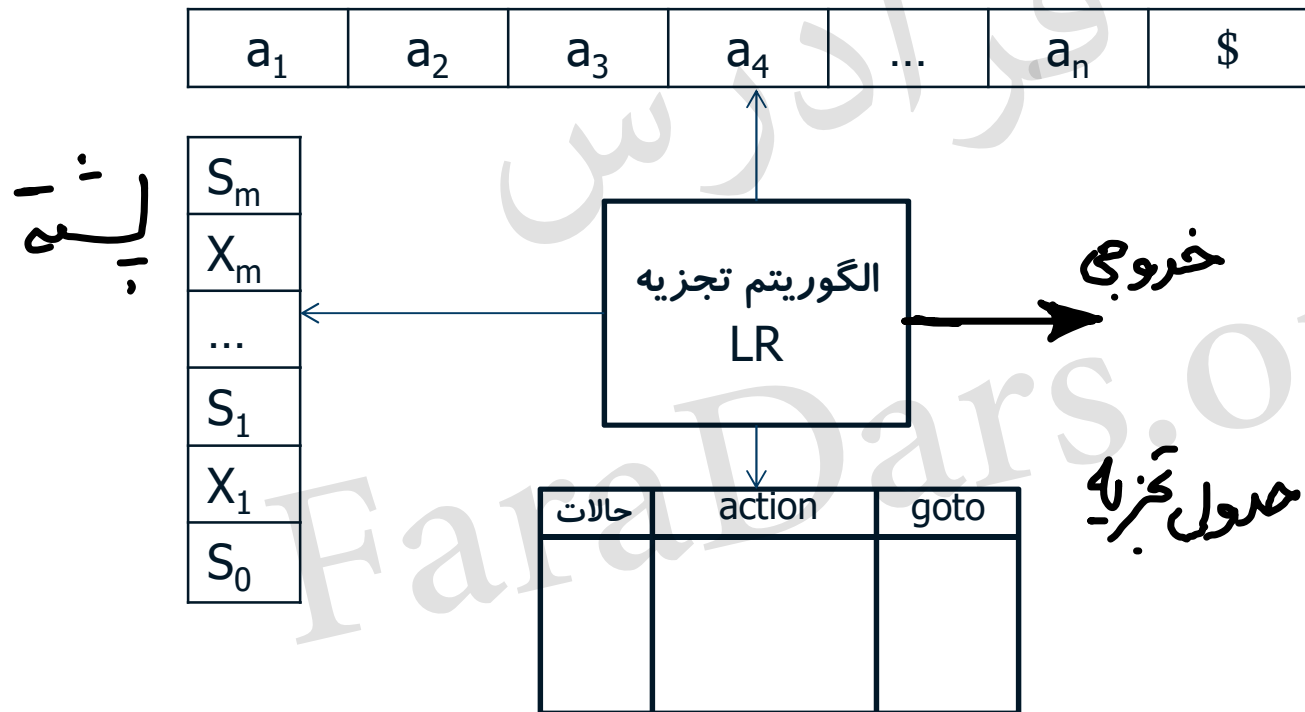
پشته	ورودی	عمل	انتقال
\$	$id+id*id\$$	$\$ < . id$	کاهش
$\$ < . id >$	$+ id * id \$$	$id . > +$	انتقال
$\F	$+ id * id \$$	$\$ < . +$	کاهش
$\$^F < . +$	$id * id \$$	$+ < . id$	انتقال
$\$^F < . + < . id >$	$* id \$$	$id . > *$	کاهش
$\$^F < . +^F$	$* id \$$	$+ < . *$	انتقال

عمل	دوری	پایه
انتقال	$id \$$	$\$ \overset{F}{<} . \overset{F}{<} * \overset{F}{<} .$
کاهش	$id > \$$	$\$ \overset{F}{<} . \overset{F}{<} * \overset{F}{<} id . >$
کاهش	$* . > \$$	$\$ \overset{F}{<} . \overset{F}{<} * \overset{F}{<} . >$
کاهش	$+ . > \$$	$\$ \overset{F}{<} . + \overset{F}{<} .$
پذیرش	$\$$	$\$ \overset{F}{=}$

تجزیه کننده LR(k)

- در این تجزیه کننده L به معنی عمل پویش از سمت چپ به راست است.
- R به این منظور استفاده شده است، که ما داریم اشتقاق راست را بصورت معکوس استفاده می کنیم.
- k تعداد توکن های مورد نیاز برای انتخاب عمل صحیح می باشد.
- از تجزیه کننده های پیشگوی بالا به پایین قوی تر هستند.
- نقطه ضعف این تجزیه کننده ها این است که پیاده سازی دستی آنها سخت است

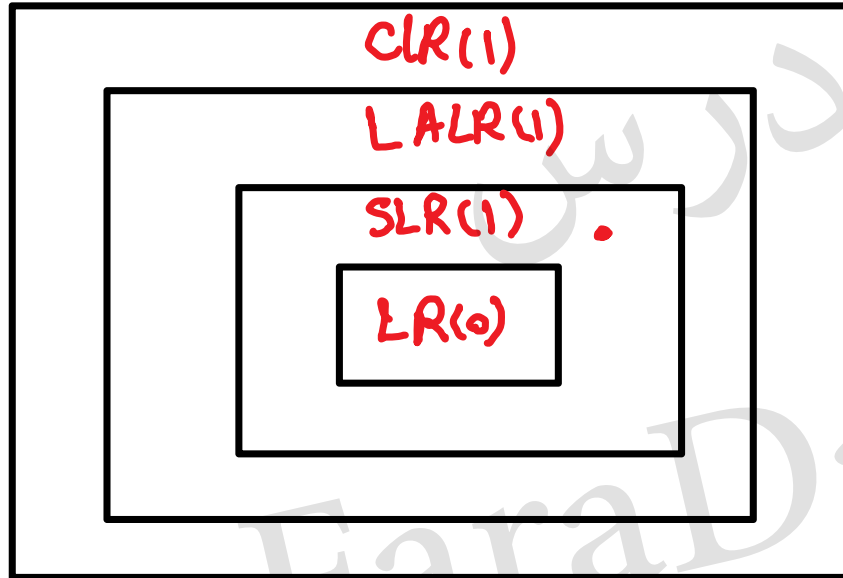
ساختار تجزیه کننده های LR



قسمت های مختلف تجزیه کننده

- ورودی: رشته ورودی برای تجزیه. به انتهای رشته ورودی \$ اضافه می شود، و تجزیه کننده با مشاهده \$ تشخیص می دهد که رشته پایان یافته است.
- پشته: محتوای پشته بصورت $S_0 X_1 S_1 \dots X_m S_m$ است، که در آن S_i نشان دهنده حالات و X_i نشاندهنده نمادهای گرامر است.
- جدول تجزیه: جدول تجزیه از دو بخش `goto action` است، قسمت `action` برای انتخاب عمل صحیح، و قسمت `goto` برای حالت بعدی را مشخص می کند.
- الگوریتم تجزیه کننده LR: این الگوریتم براساس توکن جاری رشته ورودی و حالت بالای پشته و محتوای جدول تجزیه، مرحله بعدی را تعیین می کند.

انواع تجزیه کننده های LR



- تجزیه کننده LR(0)
- تجزیه کننده SLR(1)
- تجزیه کننده LALR(1)
- تجزیه کننده CLR(1)

abc

$A \rightarrow ab$

ساخت جدول تجزیه به روش LR(0)

$S \rightarrow A \cdot$

○ قلم LR: یک قاعده تولید است که نقطه ای در سمت راست آن قرار دارد. $S \rightarrow \alpha \cdot \beta$

○ موقعیت نقطه در قلم LR مشخص کننده این است که α در بالای پشته قرار دارد و انتظار می رود که β نیز در ادامه مشاهده شود. تا بتوان کاهش به S را انجام داد

○ برای قاعده $S \rightarrow XYZ$ اقسام LR(0) بصورت زیر است:

$S \rightarrow \cdot XYZ$

$S \rightarrow X \cdot YZ$

$S \rightarrow XY \cdot Z$

$S \rightarrow XYZ \cdot$

○ اگر قاعده $S \rightarrow \lambda$ را داشته باشیم، تنها یک قلم بصورت $S \rightarrow \cdot$ را تولید می کند.

تابع بستار (Closure): اگر I مجموعه ای از قلم ها برای گرامر G باشد، آنگاه $Closure(I)$ مجموعه قلم های ایجاد شده از I با استفاده از دو قاعده زیر است:

۱- هر قلم در I به $Closure(I)$ اضافه می شود.

۲- اگر $A \rightarrow X.YZ$ در $Closure(I)$ و $Y \rightarrow \alpha$ در گرامر باشد، آنگاه $Y \rightarrow \alpha$ به بستار I اضافه می شود.

$$S \rightarrow A$$

$$A \rightarrow BC$$

$$A \rightarrow a$$

$$B \rightarrow bB$$

$$S \rightarrow .A$$

$$A \rightarrow .BC$$

$$A \rightarrow .a$$

$$B \rightarrow .bB$$

$$E' \rightarrow E$$

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

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

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

مثال: باتوجه به گرامر زیر $Closure(\{S \rightarrow .E\})$ را محاسبه کنید

$$I = \{ S \rightarrow .E \}$$

$$\frac{I}{S \rightarrow .E}$$

$$E \rightarrow .E + T$$

$$E \rightarrow .T$$

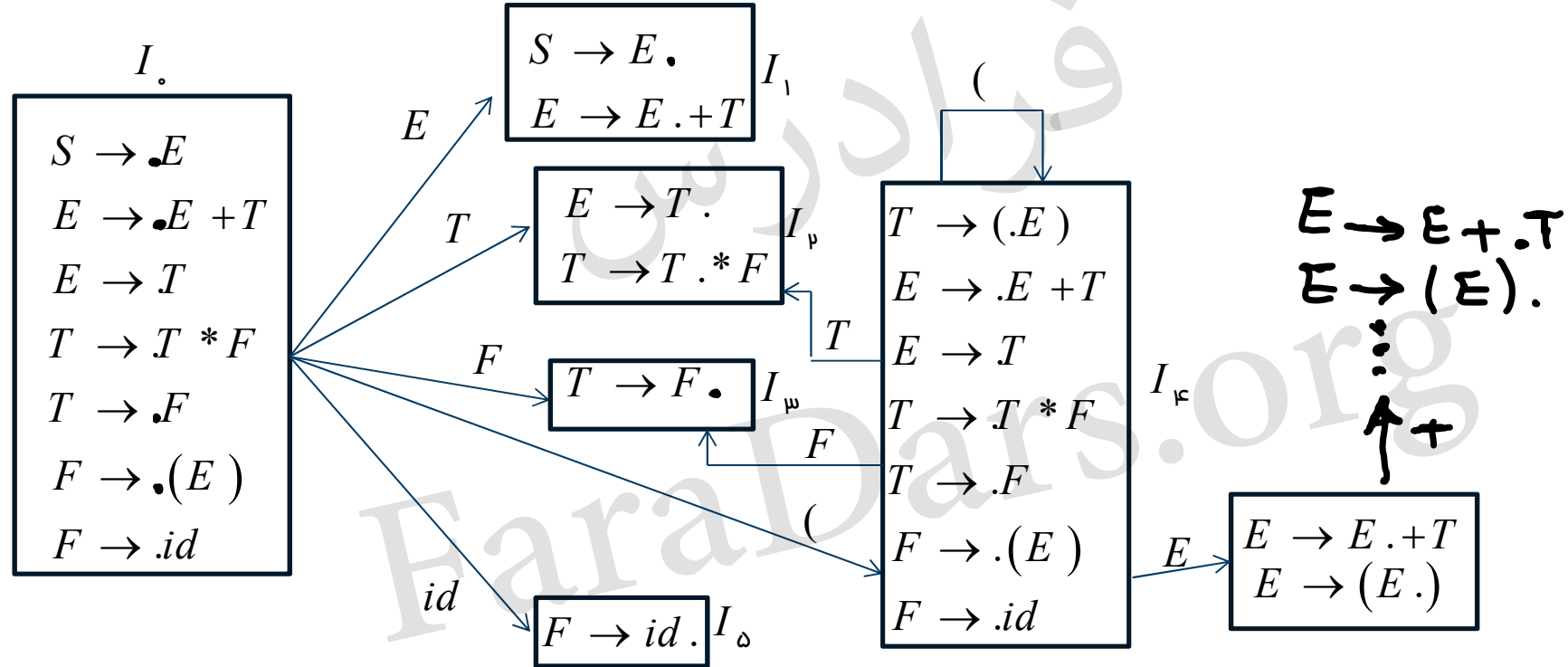
$$T \rightarrow .T * F$$

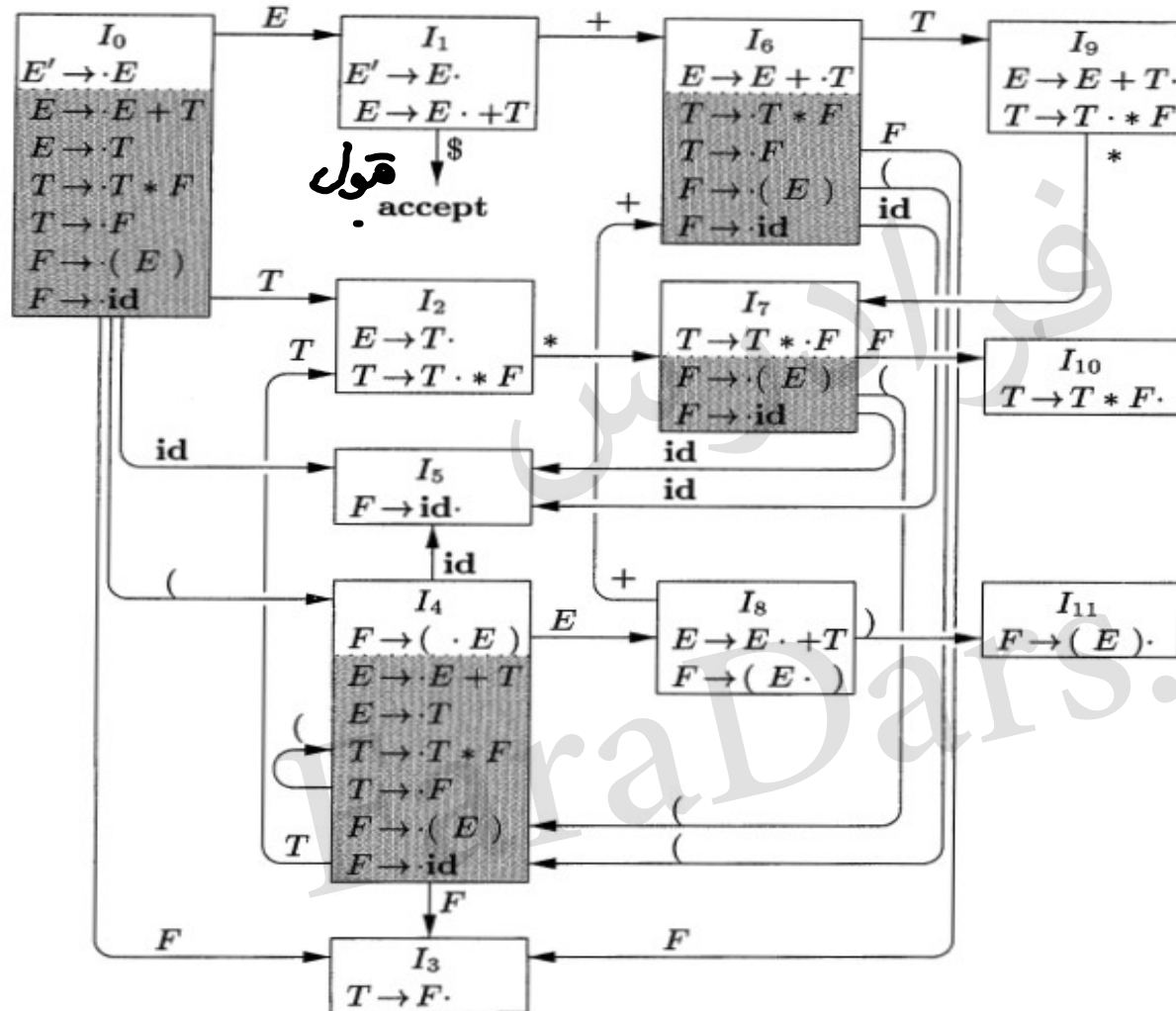
$$T \rightarrow .F$$

$$F \rightarrow .(E)$$

$$F \rightarrow .id$$

رسم نمودار انتقال





ایجاد جدول تجزیه به روش LR(0)

State	action		goto
	id	یا یانه ها \$	غیر یانه ها
0	✗		
1			
2			
.			
.			
"			

1,2 $S \rightarrow AB|C$

3,4 $A \rightarrow aA | a$

5,6 $B \rightarrow bB | b$

7 $C \rightarrow c$

ایجاد جدول تجزیه به روش LR(0)...

-ابتدا قواعد گرامر را شماره گذاری می کنیم.

بخش action:

○ انتقال: اگر در ماشین انتقال حالت، از حالت I_i با پایانه a به حالت I_j انتقال حالت داشتیم، آنگاه:

در بخش action، سطر i و زیرستون مربوط به پایانه a ، S_j قرار می دهیم. که به معنای انتقال (Shift) می باشد.

○ کاهش: اگر در حالت I_i علامت نقطه در انتهای قلم LR(0) به شکل $A \rightarrow \alpha$ باشد، در این حالت یک دستگیره وجود دارد که باید کاهش یابد. آنگاه:

در بخش action، سطر i و زیرستون تمامی پایانه ها، r_k قرار می دهیم. که به معنای کاهش می باشد. و k شماره قاعده تولید در گرامر می باشد.

○ درحالتی که قلم LR(0)، شامل قاعده افزوده $S \bullet \rightarrow S'$ باشد، در سطر آن و ستون مربوط به علامت \$، قبول (Accept) قرار می دهیم.

○ خانه های خالی بخش action نشان دهنده خطای نحوی هستند.

بخش goto:

○ اگر در حالت I_i بایک غیرپایانه A به حالت I_j انتقال داشته باشیم، آنگاه:

در بخش goto و زیر ستون غیرپایانه A و در سطر i؛ شماره j را قرار می دهیم.

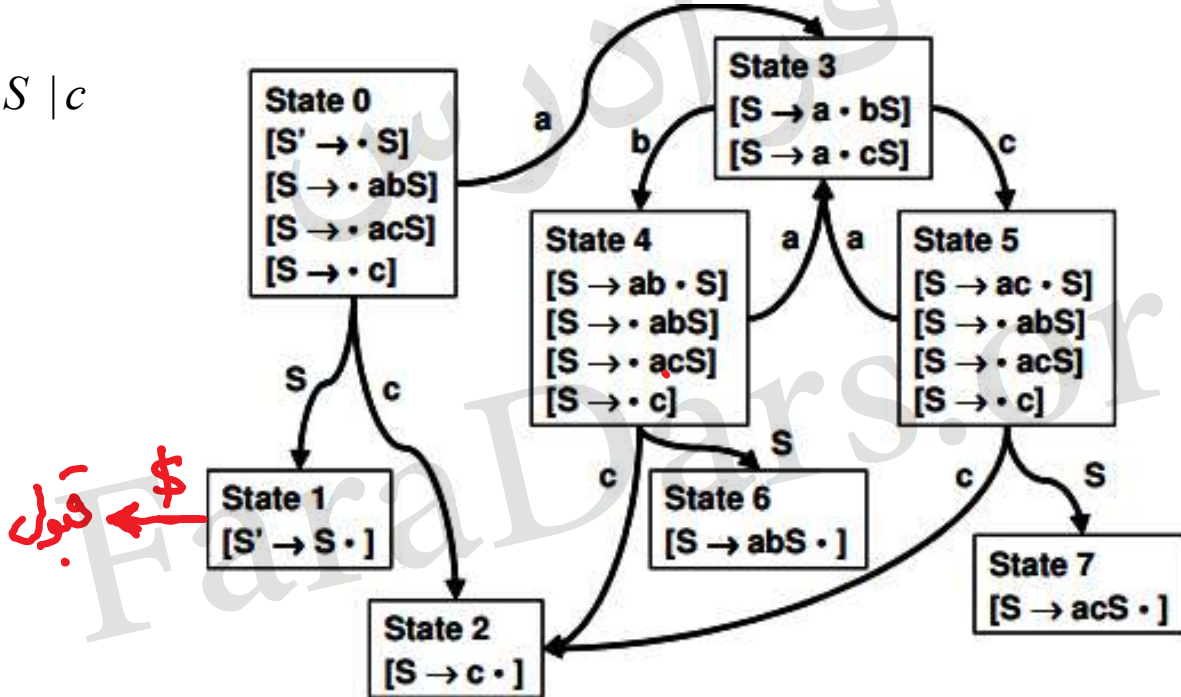
مثال: جدول تجزیه LR(0) را برای گرامر زیر مشخص کنید؟

$S' \rightarrow abS \mid acS \mid c$

1 $S' \rightarrow S$
 $S \rightarrow abS$

2 $S \rightarrow acS$

3 $S \rightarrow c$



حالات	action				goto
	a	b	c	\$	S
0	s_3		s_2		1
1				acc	
2	r_3	r_3	r_3	r_3	
3		s_4	s_5		
4	s_3		s_2		6
5	s_3		s_2		7
6	r_1	r_1	r_1	r_1	
7	r_2	r_2	r_2	r_2	

جعلی کنیزہ
 $R(0)$

○ برخورد ها در جدول تجزیه گرامرهای LR :

الف) برخورد انتقال - کاهش (S/r): اگر حالتی در ماشین انتقال حالت وجود داشته باشد که هم شامل دستگیره و هم شامل انتقال باشد آنگاه این برخورد بوجود می آید.

ب) برخورد کاهش - کاهش (r/r): اگر حالتی در ماشین انتقال حالت وجود داشته باشد، که شامل بیش از یک دستگیره باشد، آنگاه برخورد کاهش - کاهش وجود دارد.

$S' \rightarrow S$

$S \rightarrow Xa$

$X \rightarrow Yb$

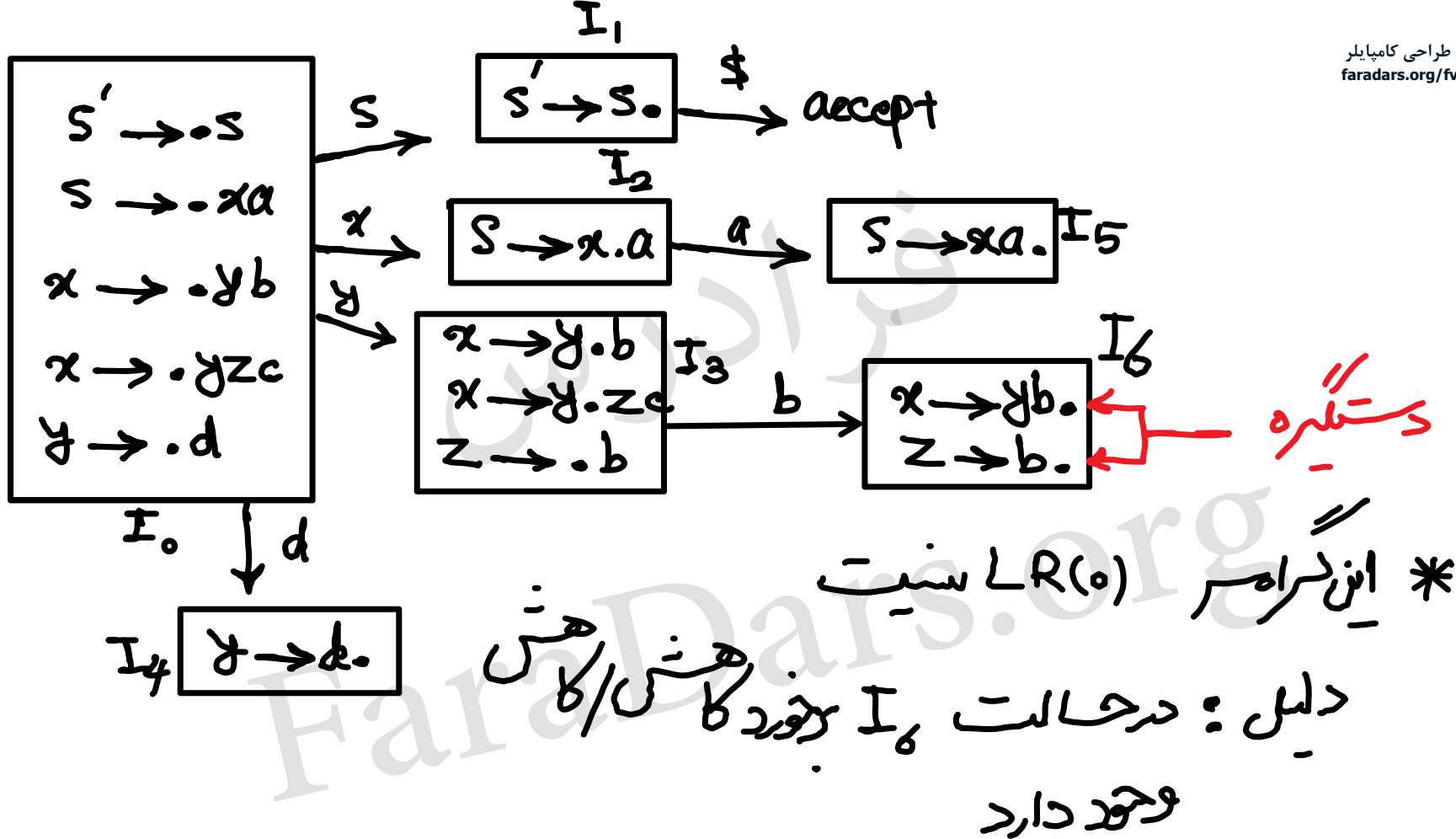
$X \rightarrow YZc$

$Y \rightarrow d$

$Z \rightarrow b$

مثال: آیا گرامر زیر LR(0) است؟

FaraDars.org



ایجاد جدول تجزیه به روش SLR(1)

<i>State</i>	<i>action</i>	<i>goto</i>
	پایانه‌ها	غیرپایانه‌ها

ایجاد جدول تجزیه به روش SLR(1)...

-ابتدا قواعد گرامر را شماره گذاری می کنیم.

بخش **action**:

○ انتقال: اگر در ماشین انتقال حالت، از حالت I_i با پایانه a به حالت I_j انتقال حالت داشتیم، آنگاه:

در بخش **action**، سطر i و زیرستون مربوط به پایانه a ، S_j قرار می دهیم. که به معنای انتقال (Shift) می باشد.

○ کاهش: اگر در حالت I_i علامت نقطه در انتهای قاعده به شکل $A \rightarrow \alpha \bullet$ باشد، در این حالت یک دستگیره وجود دارد که باید کاهش یابد. آنگاه:

در بخش **action**، سطر i و زیرستون پایانه هایی که در Follow(A) قرار دارند، r_k قرار می دهیم. که به معنای کاهش می باشد، و k شماره قاعده تولید در گرامر می باشد.

○ درحالتی که قلم LR(0)، شامل قاعده افزوده $S' \rightarrow S \bullet$ باشد، در سطر آن و ستون مربوط به علامت \$، قبول (Accept) قرار می دهیم.

○ خانه های خالی بخش action نشان دهنده خطای نحوی هستند.

بخش goto:

○ اگر در حالت I_i بایک غیرپایانه A به حالت I_j انتقال داشته باشیم، آنگاه:

در بخش goto و زیر ستون غیرپایانه A و در سطر i؛ شماره j را قرار می دهیم.

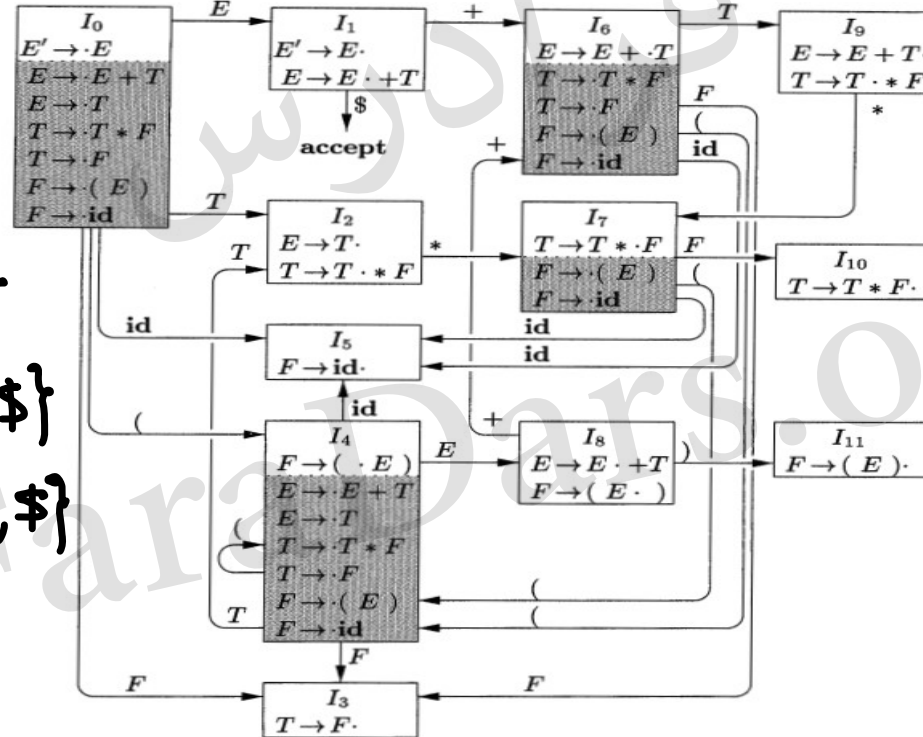
مثال: جدول تجزیه SLR(1) برای گرامر زیر مشخص کنید

$$\begin{array}{l} 1 \quad 2 \\ E \rightarrow E + T \mid T \\ 3 \quad 4 \\ T \rightarrow T * F \mid F \\ 5 \quad 6 \\ F \rightarrow (E) \mid id \end{array}$$

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

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

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



حالات

Action					
id	+	*	()	\$
0	S ₅		S ₄		
1		S ₆			acc
2	r ₂	S ₇		r ₂	r ₂
3	r ₄	r ₄		r ₄	r ₄
4	S ₅		S ₄		
5		r ₆	r ₆	r ₆	r ₆
6	S ₅		S ₄		
7	S ₅		S ₄		
8		S ₆		S ₁₁	
9	r ₁	S ₇		r ₁	r ₁
10	r ₃	r ₃		r ₃	r ₃
11	r ₅	r ₅		r ₅	r ₅

Goto		
E	T	F
1	2	3

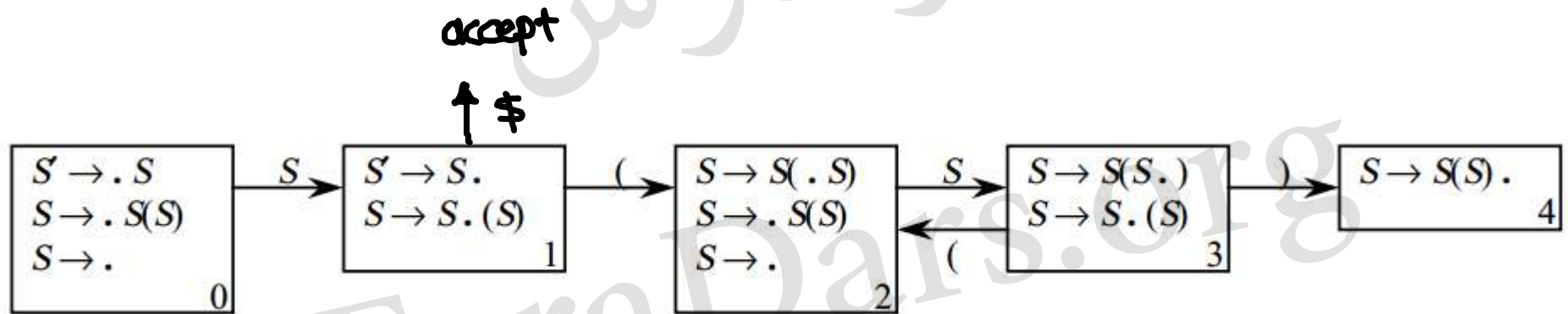
8 2 3

9 3
10

مثال: جدول تجزیه (1)SLR را برای گرامر زیر رسم کنید؟

$$S' \rightarrow S$$

$$S \rightarrow S(S) | \lambda$$



هشتم انتقال حالت

state	Action			Goto
	()	\$	S
0	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	1
1	s2		accept	
2	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	3
3	s2	s4		
4	$r S \rightarrow S (S)$	$r S \rightarrow S (S)$	$r S \rightarrow S (S)$	

○ برخورد ها در جدول تجزیه کننده SLR(1):

الف) برخورد انتقال - کاهش (S/r): اگر یک حالت داشته باشیم که بصورت زیر باشد:

در صورتیکه Follow(A) و First(γ) اشتراکشان تهی نباشد آنگاه برخورد بوجود می آید.

$$\begin{array}{l} A \rightarrow \alpha \bullet \\ B \rightarrow \beta \bullet \gamma \end{array}$$

ب) برخورد کاهش - کاهش (r/r): اگر یک حالت داشته باشیم که بصورت زیر باشد:

در صورتیکه Follow(A) و Follow(B) اشتراکشان تهی نباشد آنگاه برخورد بوجود می آید.

$$\begin{array}{l} A \rightarrow \alpha \bullet \\ B \rightarrow \beta \bullet \end{array}$$

مثال: آیا گرامر زیر $SLR(1)$ است؟

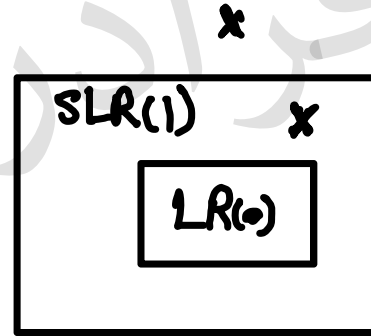
$$S \rightarrow Xa$$

$$X \rightarrow Yb$$

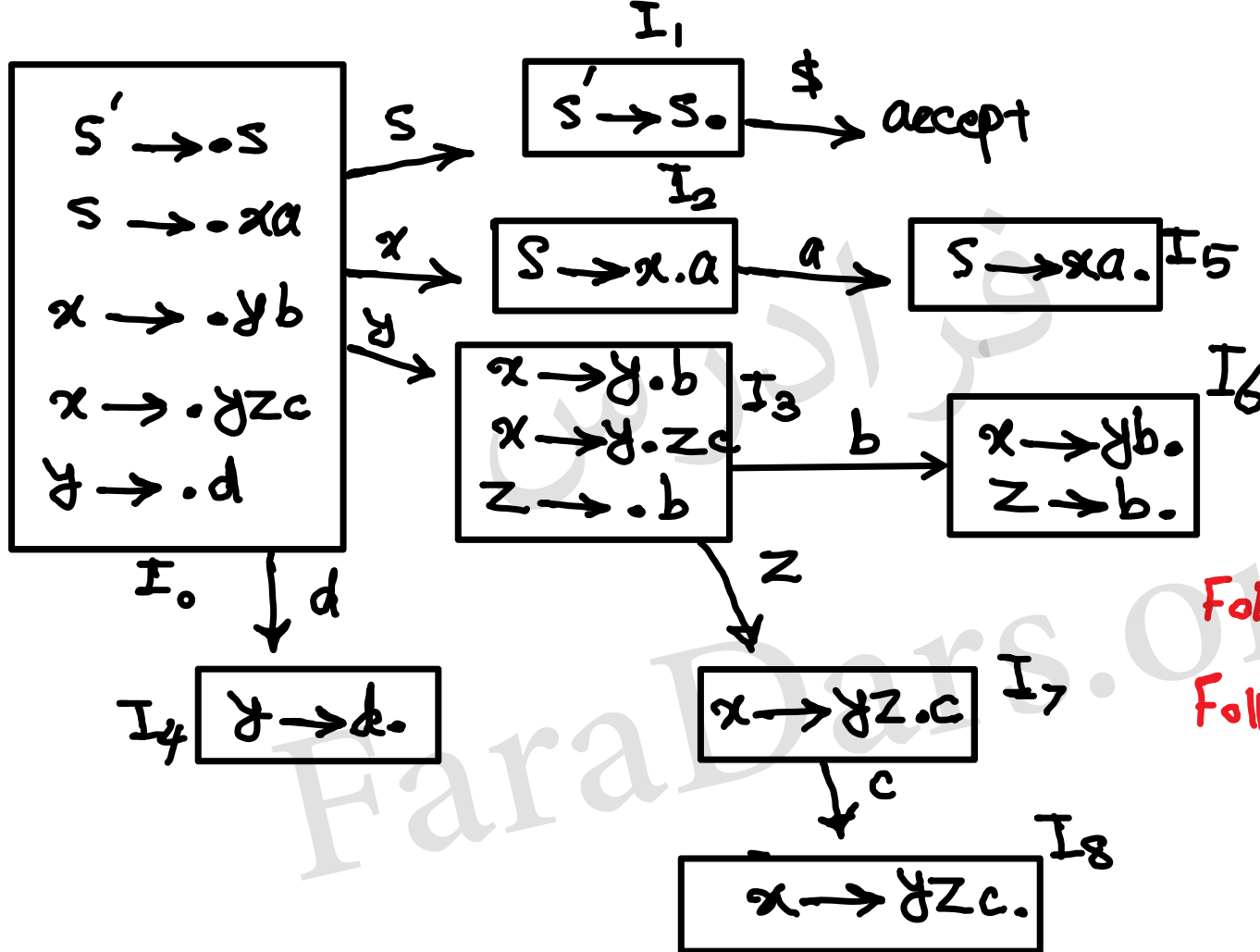
$$X \rightarrow YZc$$

$$Y \rightarrow d$$

$$Z \rightarrow b$$



گرامر با توجه به ماشین انتقال حالت اساید بعدی فاقد برگزید است پس $SLR(1)$ است.



Follow(x) = {a}

Follow(z) = {c}

الگوریتم تجزیه LR(1)

○ ابتدا در انتهای رشته ورودی علامت $\$$ را قرار می‌دهیم. و حالت صفر در پشته خالی قرار می‌گیرد.

○ اگر توکن جاری را a و بالای پشته را با s نام‌گذاری کنیم، آنگاه الگوریتم تجزیه LR با توجه به جدول تجزیه اعمال زیر را انجام می‌دهد:

الف) اگر $action[s, a]$ برابر یک انتقال S_i باشد، آنگاه ابتدا a را به بالای پشته اضافه کرده و سپس i را به بالای پشته اضافه می‌کند.

ب) اگر $action[s, a]$ برابر یک کاهش r_k باشد، آنگاه عمل کاهش انجام می‌شود. و دستگیره از بالای پشته حذف شده و سپس در صورتیکه بالای پشته x باشد، آنگاه در بالای پشته سمت چپ قاعده تولید شماره قرار می‌گیرد و بعد از آن $goto[x, A]$ قرار می‌گیرد، که A همان سمت چپ قاعده تولید شماره k است. در واقع به اندازه دو برابر طول دستگیره از بالای پشته

- ج) اگر $action[s,a]$ برابر $accept$ باشد، آنگاه عمل تجزیه با موفقیت انجام شده است.
- ب) اگر $action[s,a]$ خالی باشد، آنگاه خطای نحوی اتفاق می افتد.

FaraDars.org

مثال: رشته $(())$ را با توجه به گرامر زیر تجزیه کنید؟

$$S \rightarrow S (S) \mid \lambda$$

state	Action			Goto S
	()	\$	
0	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	1
1	s2		accept	
2	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	$r S \rightarrow \epsilon$	3
3	s2	s4		
4	$r S \rightarrow S (S)$	$r S \rightarrow S (S)$	$r S \rightarrow S (S)$	

یستم

$S \rightarrow \lambda$

$goto(0, S)$

stack	input	action
\$0	(())\$	r $S \rightarrow \epsilon$
\$0S1	(())\$	s2
\$0S1(2	()\$	r $S \rightarrow \epsilon$
\$0S1(2S3	()\$	s2
\$0S1(2S3(2)\$	r $S \rightarrow \epsilon$
\$0S1(2S3(2S3)\$	s4
\$0S1(2S3(2S3)4)\$	r $S \rightarrow S (S)$
\$0S1(2S3)\$	s2
\$0S1(2S3(2)\$	r $S \rightarrow \epsilon$
\$0S1(2S3(2S3)\$	s4
\$0S1(2S3(2S3)4)\$	r $S \rightarrow S (S)$
\$0S1(2S3)\$	s4
\$0S1(2S3)4	\$	r $S \rightarrow S (S)$
\$0S1	\$	accept

درستگیر

مشکلات روش های LR(0) و SLR(1)

○ در روش LR(0) اگر نقطه در انتهای سمت راست یک قلم LR(0) قرار بگیرد ($A \rightarrow \alpha.$) آنگاه در برای تمامی پایانه ها، بدون توجه به نماد جاری و قاعده کاهش داده شده، عمل کاهش را انجام می دهد. در حالیکه خیلی از این پایانه ها برای ادامه تجزیه امیدوار کننده نیستند.

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

$$A \rightarrow d .$$

مشکلات روش های LR(0) و SLR(1)...

- در روش SLR(1) اگر نقطه در انتهای سمت راست یک قلم LR(0) قرار بگیرد ($A \rightarrow \alpha$) آنگاه برای تمامی پایانه ها، که در $\text{Follow}(A)$ قرار دارند، عمل کاهش انجام می شود. این در حالیست که برای تمامی پایانه هایی که عمل کاهش قبل از آنها انجام شده است، ممکن است برای بعضی ادامه تجزیه امیدوارکننده نباشد.

$$S \rightarrow \dots A a \dots \mid \dots A b \dots \mid \dots A c \dots$$

$$A \rightarrow d .$$

مثالی از کاهش، نادرست در تجزیه کننده SLR(1)

$$S' \rightarrow S$$

$$S \rightarrow L = R \mid R$$

$$L \rightarrow *R$$

$$R \rightarrow L$$

$$L \rightarrow id$$

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

$$I_0$$

$$I_1$$

$$S' \rightarrow \bullet S$$

$$S \rightarrow L \bullet = R$$

$$S \rightarrow \bullet L = R$$

$$R \rightarrow L \bullet$$

$$S \rightarrow \bullet R$$

$$L \rightarrow \bullet * R$$

$$L \rightarrow \bullet id$$

$$R \rightarrow \bullet L$$

$$R = \underline{\quad}$$

ایجاد جدول تجزیه به روش (1) CLR...

<i>State</i>	<i>action</i>	<i>goto</i>
	پایانه‌ها	غیرپایانه‌ها

ایجاد جدول تجزیه به روش (1)CLR...

قلم (1)LR: شامل دو بخش است که بصورت $[A \rightarrow \alpha.\beta, a]$ می باشد. که در آن a یک پایانه یا $\$$ است، که در واقع مولفه پیش بینی می باشد.

محاسبه بستار: برای هر قلم بصورت $[A \rightarrow \alpha.B\beta, a]$ که در آن علامت نقطه قبل از B قرار دارد، آنگاه به ازای هر قاعده $B \rightarrow \gamma$ و هر ترمینال b که در $\text{First}(\beta a)$ قرار دارد قلم جدید $[B \rightarrow \gamma, b]$ به مجموعه اضافه می کنیم

$$\alpha.B\beta \rightarrow \alpha B.\beta$$

مجموعه های قلم LR(1) برای گرامر مقابل:

$S' \rightarrow S$

$S \rightarrow CC$

$C \rightarrow cC \mid d$

$[S' \rightarrow \cdot S, \$]$

$[S \rightarrow \cdot CC, \$]$

$[C \rightarrow \cdot cC, c \mid d]$

$[C \rightarrow \cdot d, c \mid d]$

$[S' \rightarrow S \cdot \$]$

$[S \rightarrow C \cdot C, \$]$

$[C \rightarrow \cdot cC, \$]$

$[C \rightarrow \cdot d, \$]$

$[C \rightarrow d \cdot, \$]$

$[S \rightarrow CC \cdot, \$]$

$[C \rightarrow c \cdot C, \$]$

$[C \rightarrow \cdot cC, \$]$

$[C \rightarrow \cdot d, \$]$

$[S \rightarrow cC \cdot, \$]$

$[C \rightarrow c \cdot C, c \mid d]$

$[C \rightarrow \cdot cC, c \mid d]$

$[C \rightarrow \cdot d, c \mid d]$

$[C \rightarrow cC \cdot, c \mid d]$

$[C \rightarrow d \cdot, c \mid d]$

$S \rightarrow \cdot CC, \$$

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

$S' \rightarrow S$

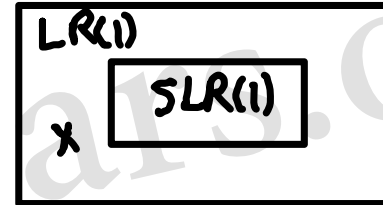
$S \rightarrow aAd \mid bBd \mid aBe \mid bAe$

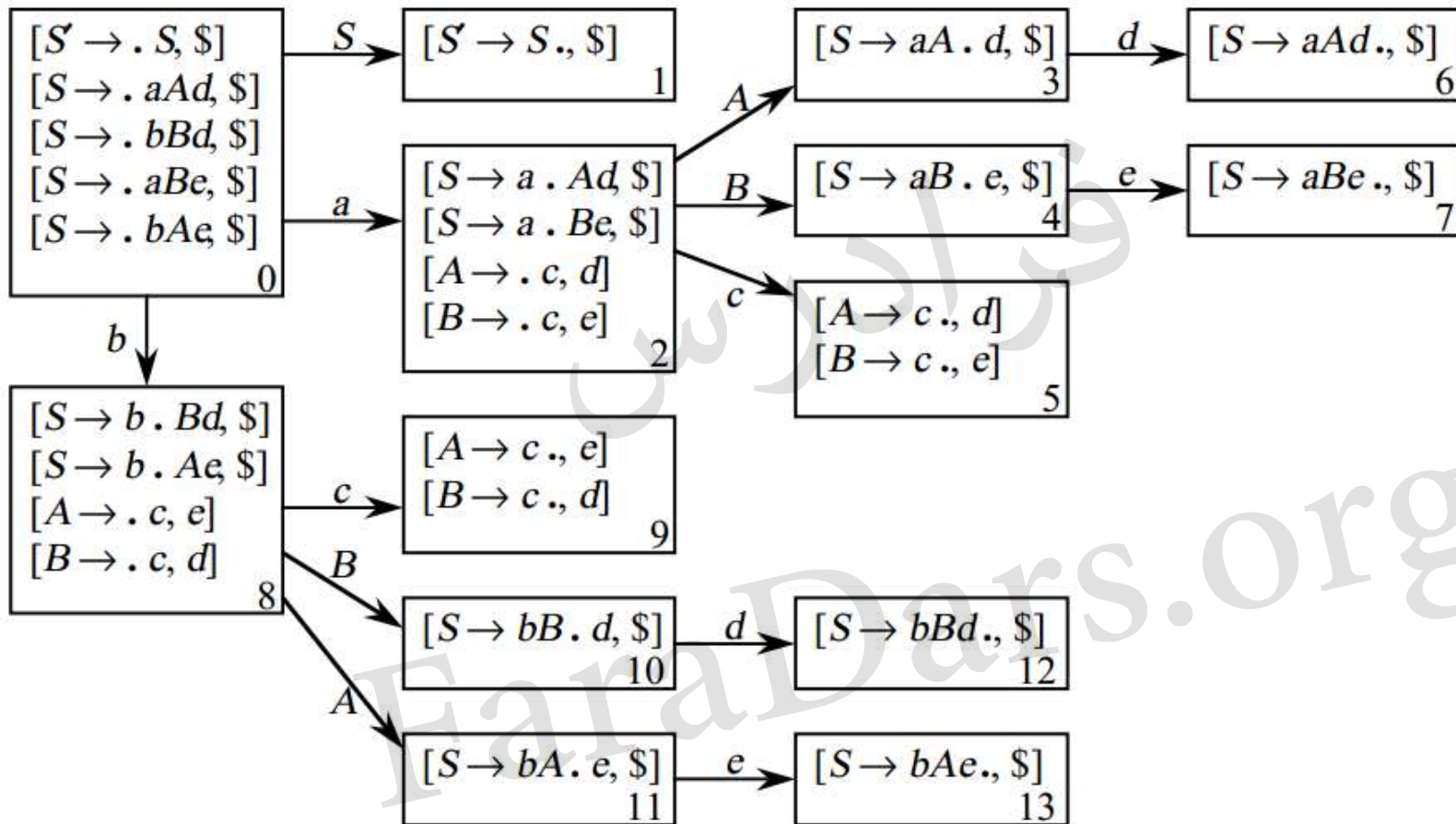
$A \rightarrow c$

$B \rightarrow c$

مثال: برای گرامر زیر ماشین انتقال حالت LR(1) رسم کنید؟

این گرامر LR(0), SLR(1) نیست





ایجاد جدول تجزیه به روش (1)CLR...

-ابتدا قواعد گرامر را شماره گذاری می کنیم.

بخش **action**:

○ انتقال: اگر در ماشین انتقال حالت، از حالت I_i با پایانه a به حالت I_j انتقال حالت داشتیم، آنگاه:

در بخش **action**، سطر i و زیرستون مربوط به پایانه a ، S_j قرار می دهیم. که به معنای انتقال (Shift) می باشد.

○ کاهش: اگر در حالت I_i ، قلم (1)LR به شکل $[A \rightarrow \alpha, a]$ داشته باشیم، که A قاعده شروع گرامر افزوده نباشد، آنگاه در سطر

i از بخش **action** و زیرپایانه/پایانه های a ، r_k قرار می دهیم، که در آن k شماره قاعده $A \rightarrow \alpha$ است.

○ اگر قلم $[S' \rightarrow S, \$]$ در حالت I_i وجود داشته باشد، آنگاه در سطر i و زیرستون $\$$ ، قبول (accept) قرار می دهیم.

مثال: برای گرامر زیر جدول تجزیه LR(1) رسم کنید؟

$S \rightarrow aA d \mid bB d \mid aB e \mid bA e$

$A \rightarrow c$

$B \rightarrow c$

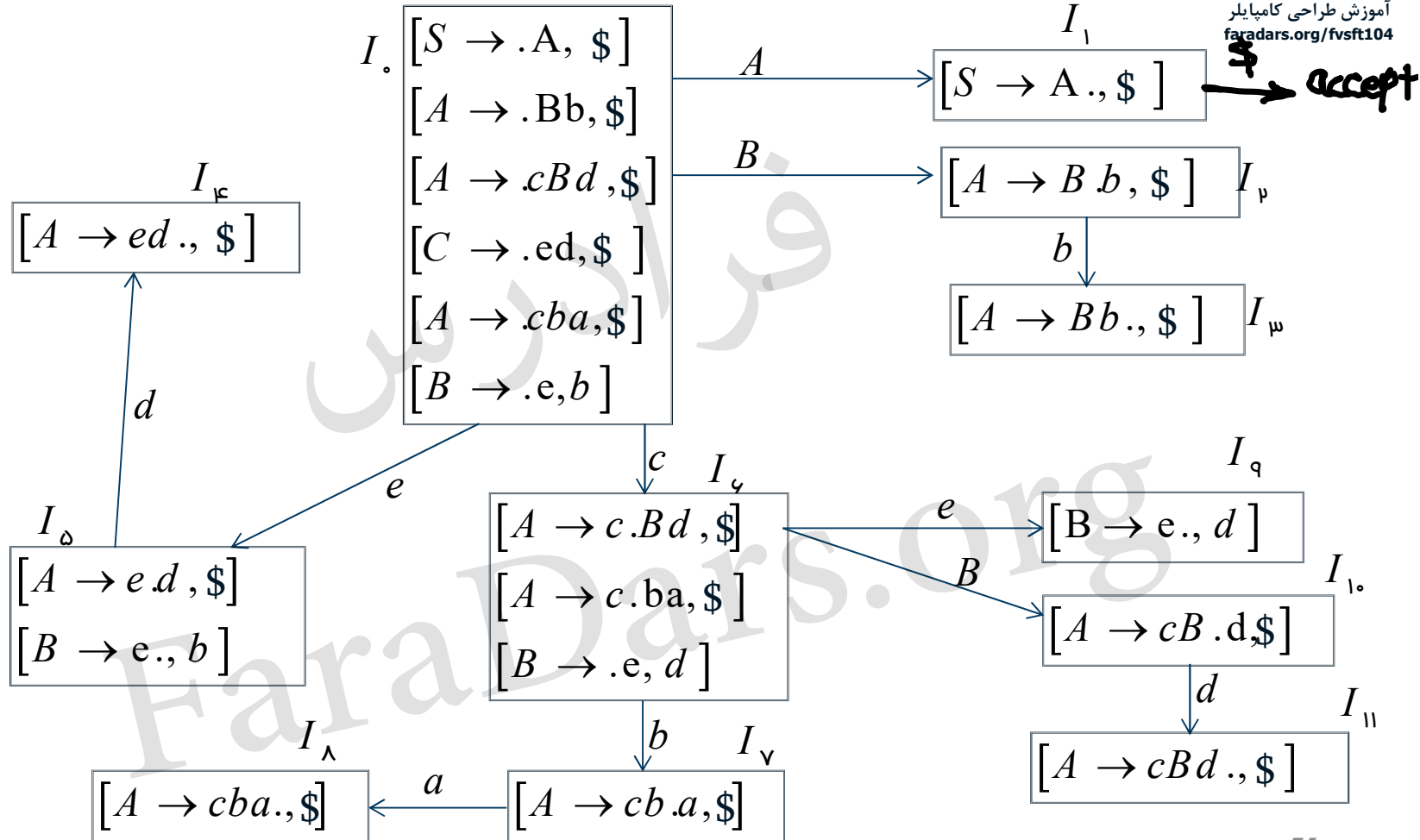
	action						Goto		
حالات	a	b	c	d	e	\$	S	A	B
0	S_2	S_8					1		
1						acc			
2			S_5					3	4
3				S_6					
4					S_7				
5				r_5	r_6				
6						r_1			
7						r_3			
8			S_9					11	10
9				r_6	r_5				
10				S_{12}					
11					S_{13}				
12						r_2			
13						r_4			

مثال: برای گرامر زیر جدول تجزیه LR(1) رسم کنید؟

$S \rightarrow A$

$A \rightarrow B \underset{1}{b} \mid c \underset{2}{B} d \mid e \underset{3}{d} \mid c \underset{4}{b} a$

$B \rightarrow e \underset{5}{}$



	action						Goto	
حالات	a	b	c	d	e	\$	A	B
0			S_6		S_5		1	2
1						acc		
2		S_3						
3						r_1		
4						r_3		
5		r_5		S_4				
6		S_7			S_9			10
7	S_8							
8						r_4		
9				r_5				
10				S_{11}				
11						r_2		

○ برخورد‌ها در جدول تجزیه کننده LR(1):

الف) برخورد انتقال-کاهش (S/r): اگر یک حالت داشته باشیم که بصورت زیر باشد:
در صورتیکه $First(\gamma) \cap a \neq \emptyset$ آنگاه برخورد بوجود می‌آید.

$$\begin{array}{l} A \rightarrow \alpha \bullet, a \\ B \rightarrow \beta \bullet \gamma, b \end{array}$$

ب) برخورد کاهش-کاهش (r/r): اگر یک حالت داشته باشیم که بصورت زیر باشد:
در صورتیکه $b \cap a \neq \emptyset$ آنگاه برخورد بوجود می‌آید.

$$\begin{array}{l} A \rightarrow \alpha \bullet, a \\ B \rightarrow \beta \bullet, b \end{array}$$

ایجاد جدول تجزیه به روش LALR(1)

- ابتدا باید ماشین انتقال حالت LR(1) را مشخص کنیم
- حالت ها، با هسته یکسان را با هم ادغام می کنیم.
- در ادغام هسته را می نویسیم و برای بخش دوم، از اجتماع مجموعه پایانه های پیشگو استفاده می کنیم.
- حالت های ترکیب شده را در یک حالت قرار می دهیم.

مثال: نشان دهید گرامر زیر LR(1) است، اما LALR(1) نیست؟

(تمرین کتاب AHO)

$$S \rightarrow Aa \mid bAc \mid Bc \mid bBa$$

$$A \rightarrow d$$

$$B \rightarrow d$$

FaraDars.org

گرامرهای مبهم در تجزیه LR

- گرامرهای مبهم LR نیستند، و نمیتوان بصورت مستقیم برای آنها جدول تجزیه LR ایجاد کرد.
- برای تجزیه LR یک گرامر مبهم دو روش وجود دارد:

۱- تبدیل گرامر مبهم به گرامر غیرمبهم و سپس ایجاد جدول تجزیه

۲- ایجاد جدول تجزیه LR برای گرامر مبهم و سپس حذف اعمال نادرست در جدول تجزیه

- به چند دلیل استفاده از روش دوم بهتر است:

۱- تعداد قواعد یک گرامر مبهم از تعداد قواعد گرامر غیرمبهم معادلش کمتر است، و این باعث

می شود که جدول تجزیه کوچکتری نیاز باشد.

۲- بعضی از گرامرهای مبهم ساختار زبان را ساده تر و قابل فهم تر بیان می کنند.

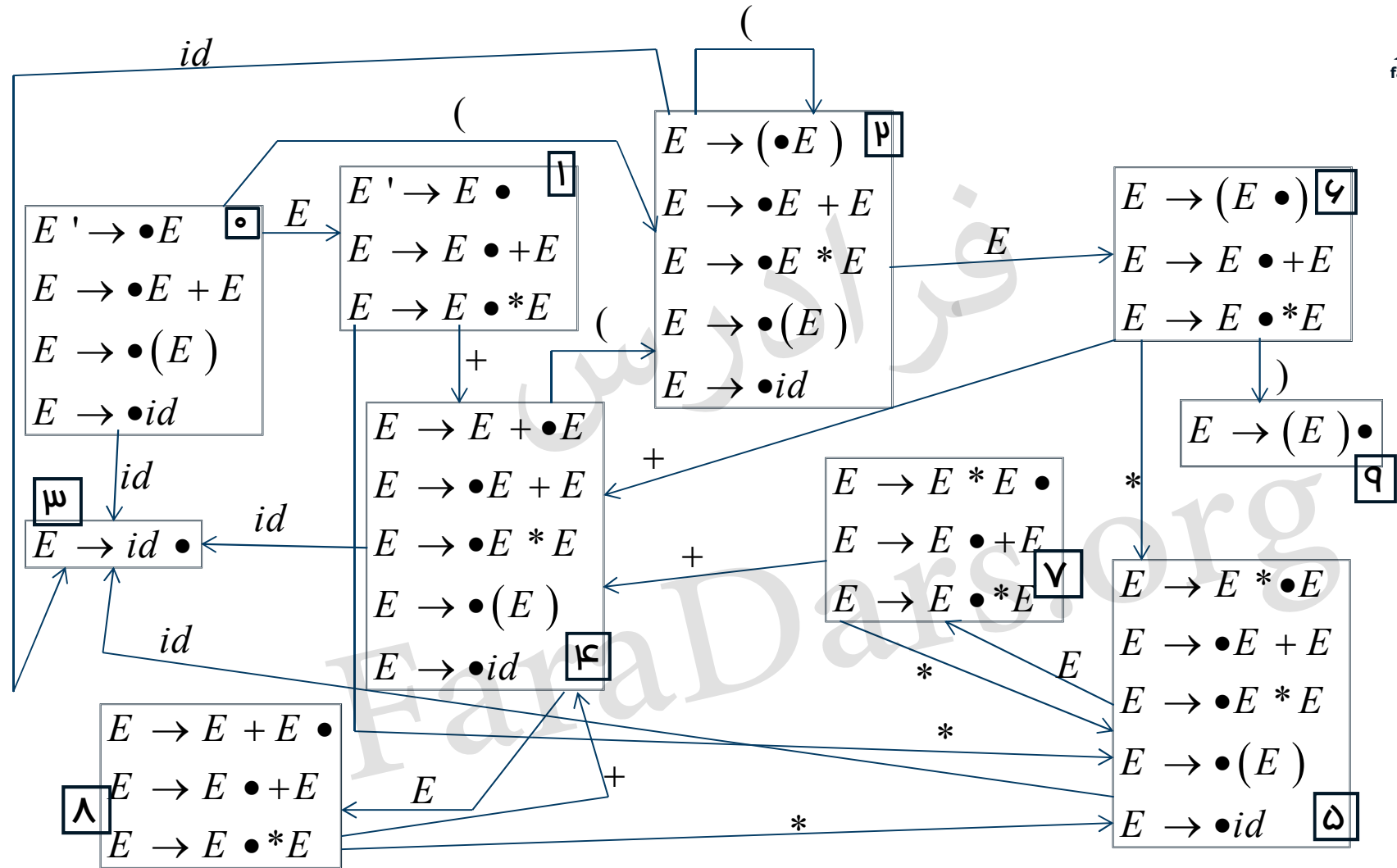
جدول تجزیه برای یک گرامر مبهم

$$E \rightarrow E + T \mid E * E \mid (E) \mid id$$

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

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

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



	action						Goto
حالات	id	+	*	()	\$	E
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							

این اسلایدها 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