

تجزیه پائین به بالا (ادامه)

تجزیه-پائین به بالا-

■ هدف ؟

■ تفاوت با- تجزیه- بالا به- پائین

□ ترتیب ساخت درخت

□ استفاده از قوانین به صورت عکس

■ (عمل کاهش)

□ قوی تر و پیچیده تر

■ یکی- از- رایج- ترین- و- قوی- ترین- شیوه- هلی- تجزیه- پائین- به- بالا-،
تجزیه- انتقالی- کاهش- می- باشد که در- بخش- قبل- مورد- با آن-
آشنا شدیم-.

تجزیه-گره‌های انتقالی-کاهشی (Shift-Reduce)

- شامل یک پشته و یک جدول تجزیه می باشد.
- رشته ورودی را به دو قسمت تقسیم می کند
 - بخش خلاصه (تجزیه) شده
 - بخشی که در پشته قرار گرفته است.
 - بخش تجزیه نشده
 - هنوز در ورودی قرار دارد و خوانده نشده است.
- اعمال-این-تجزیه-گر-شامل-انتقال-کاهش-پذیرش-و-گزیرش-خطا می باشد.

تجزیه گره‌های انتقالی کاهشی- عمل کاهش

■ گرامر زیر را در نظر بگیرید:

S E
E T | E + T
T id | (E)

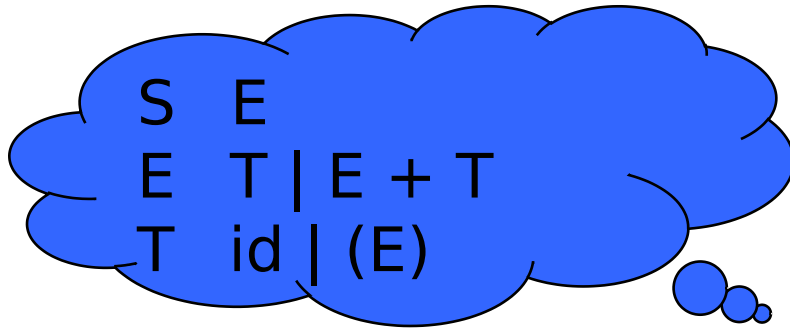
■ اگر پشته حاوی (id باشد، با انجام یک عمل کاهش، می تواند محتویات آن به T تغییر یابد.

■ حالت پذیرش نیز حالت خاصی از کاهش است. کاهش کل محتوای پشته به نماد شروع گرامر.

■ به بخشی از محتوای پشته که در عمل کاهش مورد استفاده قرار گرفته است، دستگیره (handle) گفته می شود.

■ شناخت دستگیره یکی از موارد مشکل مربوط به این تجزیه گر می باشد.

تجزیه-گره‌های-انتقالی-کا-هشی-عمل-انتقال



- گاهی ممکن است دستگیره ای درپشته وجود نداشته باشد. در این حالت باید عمل انتقال را انجام داد.
- مثلا اگر پشته حاوی (و ورودی id+id باشد، باید از بخش خلاصه نشده به بخش خلاصه شده انتقال داد.

تجزیه LR

- نوع خاصی از تجزیه گره‌های انتقالی کاهش می‌دهد.
- اختصار چیست؟
- گرامرهای LR
- رابطه این گرامرها با گرامرهای LL
- مزیت آن بر تجزیه گره‌های LL
- عیب آن نسبت به تجزیه گره‌های LL

تجزیه LR - چگونگی تعیین دستگیره.

- آیا فقط با توجه به نماد بالای پشته این کار انجام می گیرد؟
- حالت (state) چیست؟
- جدول تجزیه LR دارای دو بخش می باشد.
- بخش GOTO جدول چه زمانی کاربرد دارد؟

انواع تجزیه گر LR

- انواع تجزیه گر LR

- LR(k)

- simple LR(k) : SLR(K)

- lookahead LR(k) : LALR(K)

- که در هر یک K تعداد ترم پیش بینی را مشخص می کند. اما بیشتر کار ما با $k=0$ یا $k=1$ است.

- نحوه عمل همه تجزیه گر ها - ما - نند مثال - خیری - است که مشاهده کردیم.

- تفاوت این تجزیه گر ها در اندازه و نحوه ساخت جدول تجزیه است.

تجزیه گر LR(0)

- ساده ترین و ضعیف ترین تجزیه گر LR است.
- از دیدگاه تئوریک اهمیت دارد و کاربرد عملی ندارد.
- از صفر ترم پیش بینی استفاده می کند.

تجزیه گر- LR(1)

- با افزودن یک ترم پیش بینی به تجزیه گر LR(0) این نوع تجزیه گر را داریم.
- قدرت آن خیلی بیشتر از LR(0) می شود.
- فقط برای عدد کمی از گرامرها می توان تجزیه گر LR(0) تولید کرد، در حالی که برای اغلب گرامرهای غیر مبهم می توان این نوع تجزیه گر را تولید کرد.
- عیب استفاده از یک ترم پیش بینی، بزرگ شدن بیش از حد جدول تجزیه است. (حدود چند صد حالت در LR(0) و حدود چندین هزار حالت در LR(1))

چگونگی ساخت جدول تجزیه - مقدمه

■ پیکربندی یا Configuration

□ مثال

A •XYZ

A X•YZ

A XY•Z

•A XYZ

■ نماد ورودی بعدی از رشته ورودی چه می تواند باشد؟

■ و-جود- نقطه- در- انتهای یک پیکربندی نشانه چیست؟

■ و-جود- نقطه- در- وسط یک پیکربندی نشانه چیست؟

چگونگی ساخت جدول تجزیه - مقدمه (2)

مثلا

A $X \bullet YZ$

به چه معنی است ؟
اگر قواعدی به فرم

Y $u | w$

داشته باشیم، آیا می توان گفت که تمام این سه پیکربندی، نشانگر یک حالت می باشند؟

بنابر این به سه پیکربندی زیر یک حالت (مجموعه پیکربندی) گفته می شود و به این عمل بستار گفته می شود.

A $X \bullet YZ$

Y $\bullet u$

Y $\bullet w$

چگونگی ساخت جدول تجزیه - مقدمه (3)

- جدول تجزیه ما برای هر مجموعه پیکربندی، یک حالت خواهد داشت.
- می توانیم ساخت این بستارها را به شکل ساخت یک ماشین خودکار محدود انجام دهیم.

نحوه ساخت یک مجموعه پیکربندی (item)

■ از یک پیکر بندی مثل نمونه زیر شروع می کنیم:

$$A \quad X_1 \dots X_i \bullet X_{i+1} \dots X_j$$

■ اگر نقطه پیش از یک غیرپایانه قرار گرفته باشد، پیکربندی های دیگری هم به آن مجموعه اضافه می شود. مثلاً در مثال فوق اگر قاعده دیگری در گرامر مثل

$$X_{i+1} \quad Y_1 \dots Y_g \mid Z_1 \dots Z_h$$

■ داشته باشیم، دو پیکر بندی زیر را به این مجموعه می افزایم:

$$X_{i+1} \bullet Y_1 \dots Y_g$$

$$X_{i+1} \bullet Z_1 \dots Z_h$$

تا چه زمان موازید دیگری نیز به مجموعه پیکر بندی اضافه می شود؟

دستورالعمل ساخت یک مجموعه پیکربندی

- برای ساخت یک مجموعه پیکربندی با شروع از پیکربندی $A \bullet \underline{u}$ بستار را بصورت زیر انجام می دهیم:
 - $A \bullet \underline{u}$ را به بستار اضافه می کنیم.
 - اگر نماد ابتدای \underline{u} یک پایانه است، کار ما تمام شده است.
 - اگر \underline{u} با یک غیرپایانه مثل B آغاز می شود، تمام قواعدی را که B در سمت چپ آنها قرار دارد را به مجموعه می افزائیم؛ و در نقطه را در سمت چپ ترین مکان RHS آنها قرار می دهیم:
$$B \bullet \underline{v}$$
 - مراحل 2 و 3 را برای قواعدی که جدیداً به مجموعه اضافه شده است، تکرار می کنیم.

انتقال حالت بین مجموعه-پیکربندی ها-

- اطلاعات دیگر مورد نیاز برای ساخت جدول تجزیه انتقال حالت بین مجموعه پیکربندی ها می باشد.
- به این منظور مجموعه مابعد (successor) را تعریف می کنیم.
- اگر یک مجموعه پیکربندی مثل C و یک نماد از گرامر مثل X را در نظر بگیریم، تابع مابعد، مجموعه پیکربندی بعدی را که پس از شناخت یک نماد گرامر باید به آن برویم را، مشخص می کند: $C' = \text{successor}(C, X)$
- چگونه-مجموعه C' را بدست آوریم؟

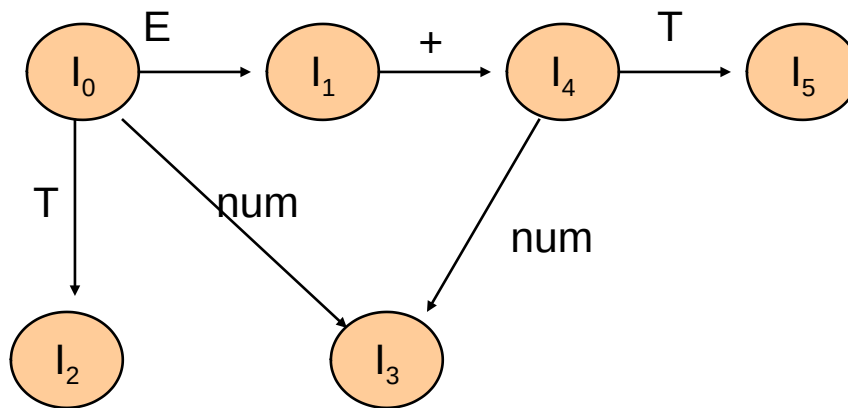
نحوه- ساخت جدول تجزیه- (1)

- قدم اول : ساخت یک DFA (نمودار انتقال)
- قدم دوم : ساخت جدول تجزیه با کمک DFA

محاسبه مجموعه پیکربندی ها (item)

- برای ساخت جدول تجزیه (هر دو بخش Action و Goto) باید مجموعه پیکربندی ها و توابع ما بعد را محاسبه کنیم.
- هدف ما کاهش کل رشته ورودی است، به این منظور باید در یک قاعده نماد شروع گرامر را در سمت راست قاعده داشته باشیم. به همین منظور گرامر را تقویت (augment) می کنیم. و قاعده $S \rightarrow S'$ را به گرامر می افزائیم.
- برای محاسبه پیکربندی ها و توابع مابعد از کجا شروع کنیم؟

ساخت‌جزیه-گر. (مثال 1)



E	E+T:1
E	T:2
T	num:3

0:S E
 1:E E+T
 E T:2
 T num:3

ساخت تجزیه گر. (ادامه مثل 1)

I0: S .E
 E .E+T
 E .T
 T .num

I3: goto(I0,num)
 T num.

reduce by rule 3

I1: goto (I0,E)
 S E.
 E E. +T

accept

I4: goto (I1,+)
 E E +.T
 T .num

I5: goto (I4,T)
 E E +T.

reduce by rule 1

I2:goto (I0,T)
 E T.

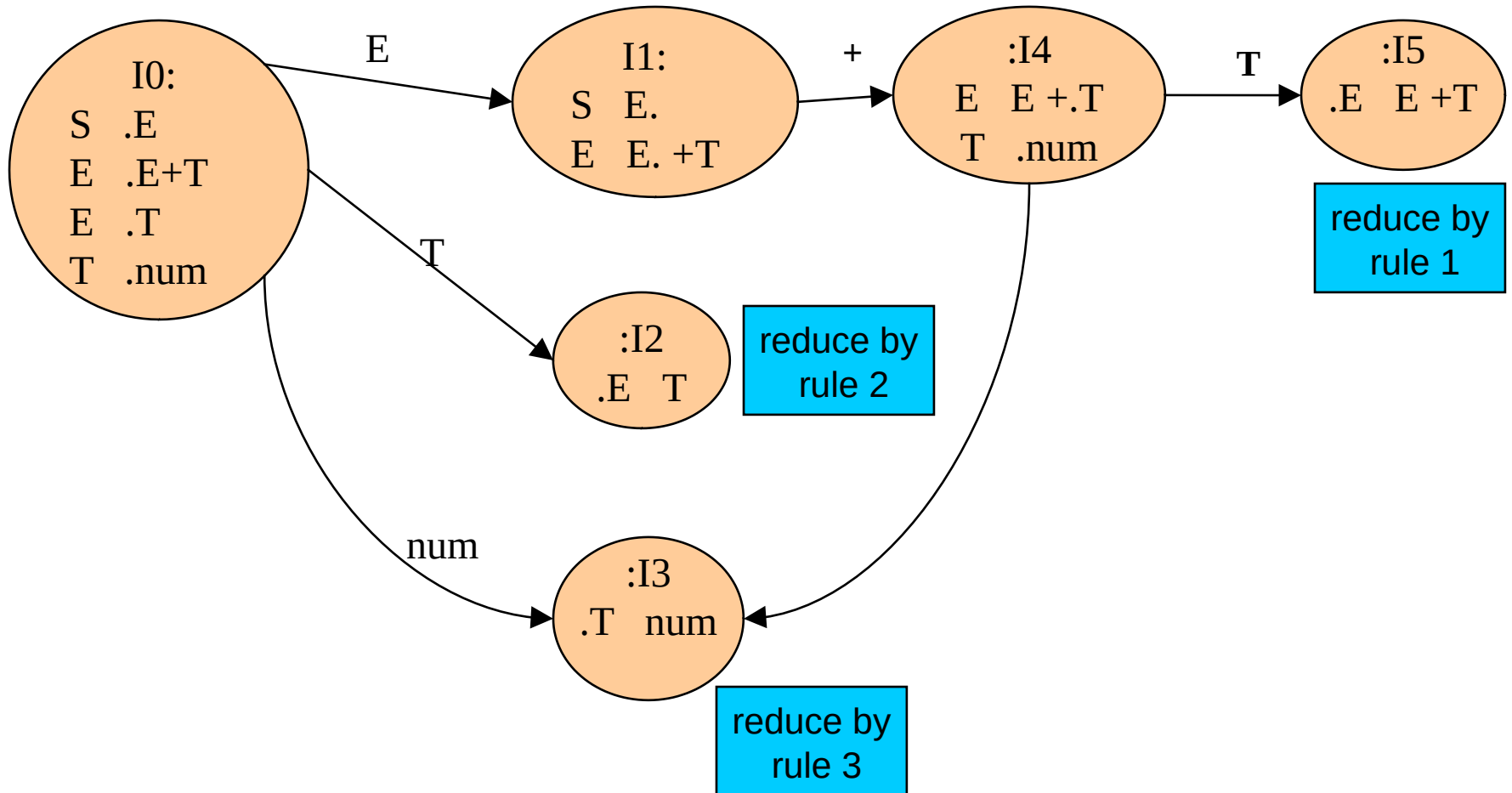
I3: goto(I4,num)

reduce by rule 2

follow	Var
\$	S
\$,+	E
\$,+	T

ساخت تجزیه گر- نمودار- انتقال- (ادامه مثال- 1)

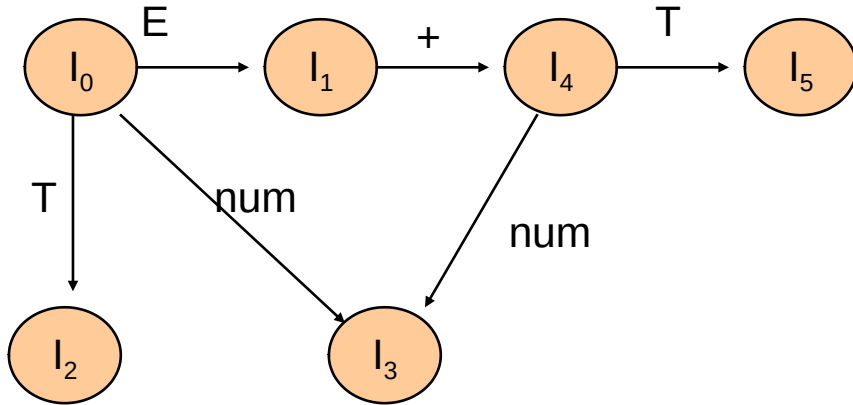
\$ Accept if



ساخت تجزیه گر- جدول-

تجزیه

(ادامه مثال- 1)



0:S E
1:E E+T
E T:2
T num:3

follow	Var
\$	S
\$,+	E
\$,+	T

to go		action			LA
T	E	\$	num	+	state
2	1	-	S3	-	0
-	-	acc	-	S4	1
-	-	r2	-	r2	2
		r3	-	r3	3
5	-	-	S3	-	4
		r1	-	r1	5

ساخت تجزیه گر- (مثال- 2)

■ برای گرامر تقویت شده زیر این کار را انجام می دهیم:

- 0) S E
- 1) E E + T
- 2) E T
- 3) T (E)
- 4) T id

ساخت تجزیه گر (ادامه مثال 2)

I0: $S \rightarrow \bullet E$ I1
 $E \rightarrow \bullet E+T$ I1
 $E \rightarrow \bullet T$ I2
 $T \rightarrow \bullet (E)$ I3
 $T \rightarrow \bullet id$ I4

I1: goto(I0,E)
 $S \rightarrow E\bullet$ Accept
 $E \rightarrow E\bullet+T$ I5

I2: goto(I0,T)=(I3,T)
 $E \rightarrow T\bullet$ Reduce 2

I3: goto(I0,())=(I3,())
 $T \rightarrow (\bullet E)$ I6
 $E \rightarrow \bullet E+T$ I6
 $E \rightarrow \bullet T$ I2
 $T \rightarrow \bullet (E)$ I3
 $T \rightarrow \bullet id$ I4

I4: goto(I0,id)=(I3,id)
 $T \rightarrow id\bullet$ Reduce 4

I5: goto(I1,+)
 $E \rightarrow E+\bullet T$ I7
 $T \rightarrow \bullet (E)$ I3
 $T \rightarrow \bullet id$ I4

I6: goto(I3,E)
 $T \rightarrow (E\bullet)$ I8
 $E \rightarrow E\bullet+T$ I5

I7: goto(I5,T)
 $E \rightarrow E+T\bullet$ Reduce 1

I8: goto(I6,))
 $T \rightarrow (E)\bullet$ Reduce 3

0) $E' \rightarrow E$
 1) $E \rightarrow E+T$
 2) $E \rightarrow T$
 3) $T \rightarrow (E)$
 4) $T \rightarrow id$

ساخت تجزیه گر- جدول تجزیه LR(0) (ادامه مثال 2)

State	Action					Goto	
	id	+)	(\$	E	T
0	s4		s3			1	2
1		s5			accept		
2	r2	r2	r2	r2	r2		
3	s4		s3			6	2
4	r4	r4	r4	r4	r4		
5	s4		s3				7
6		s5		s8			
7	r1	r1	r1	r1	r1		
8	r3	r3	r3	r3	r3		

ساخت تجزیه گر (ادامه مثال 2)

روش- SLR(1) با توجه به مجموعه Follow

follow	Var
\$	S
\$(+,)	E
\$(+,)	T

- 0) E' E
- 1) E E + T
- 2) E T
- 3) T (E)
- 4) T id

State	Action					Goto	
	id	+)	(\$	E	T
0	s4		s3			1	2
1		s5			accept		
2		r2		r2	r2		
3	s4		s3			6	2
4		r4		r4	r4		
5	s4		s3				7
6		s5		s8			
7		r1		r1	r1		
8		r3		r3	r3		

S E
E T | E + T
T id | (E)

ساخت تجزیه گر (ادامه مثال 2)

PARSE STACK	Automata stack	REMAINING INPUT	table	PARSER ACTION
	0	(id + id)\$	s3	Shift
(03	id + id)\$	s4	Shift
(id	034	+ id)\$	r4	Reduce: T → id, pop(1 item)
(T	032	+ id)\$	r2	Reduce: E → T, pop(1 item)
(E	036	+ id)\$	s5	Shift
(E +	0365	id)\$	s4	Shift
(E + id	03654) \$	r4	Reduce: T → id, pop(1 item)
(E + T	03657) \$	r1	Reduce: E → E + T, pop(3 item)
(E	036) \$	s8	Shift
(E)	0358	\$	r3	Reduce: T → (E), pop(3 item)
T	02	\$	r2	Reduce: E → T, pop(1 item)
E	01	\$	accept	Accept

نمودار-انتقال

- آیا ترتیب تولید و شماره گذاری مجموعه ها اهمیتی دارد؟
- برای اینکه مجموعه پیکربندی ها را به صورت مناسب تری نمایش دهیم، می توانیم از نمودار انتقال (یا نمودار goto) استفاده کنیم.

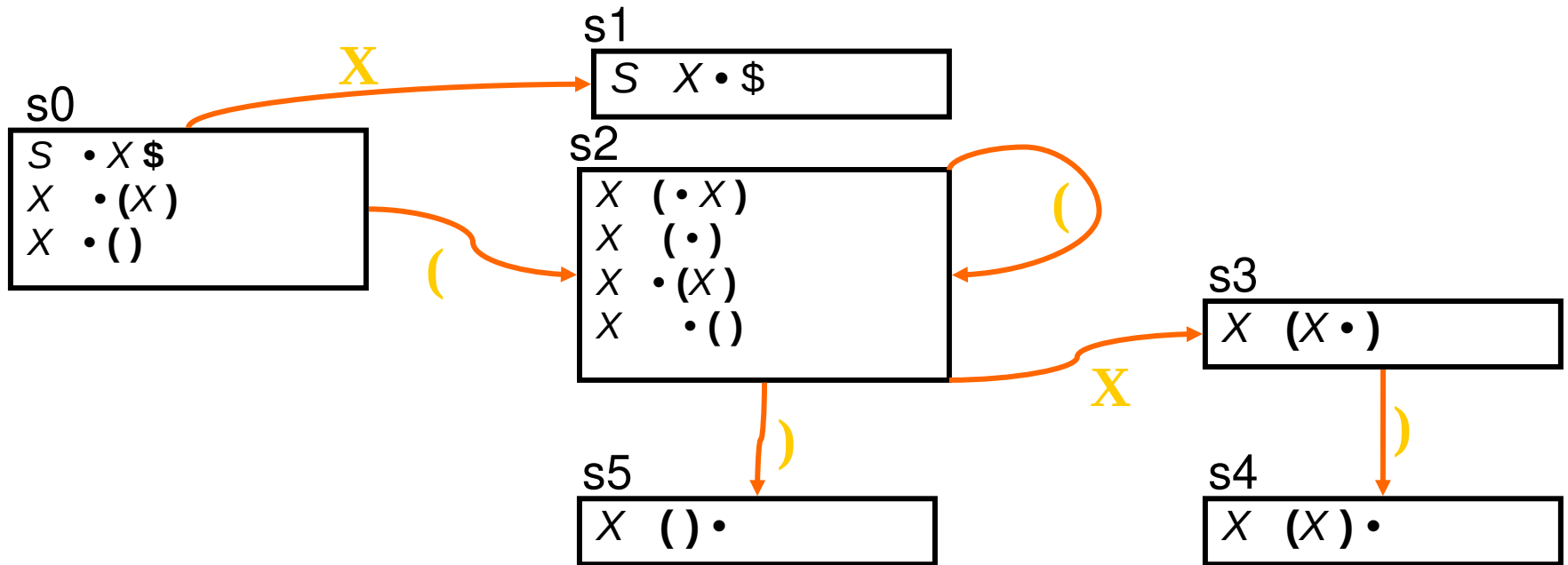
نحوه- ساخت جدول تجزیه-

- قدم اول : ساخت یک DFA (نمودار انتقال)
□ که این کار را انجام دادیم.
- قدم دوم : ساخت جدول تجزیه با کمک DFA

ساخت جدول تجزیه

- بازای هر حالت
 - انتقال به یک حالت دیگر بازای یک پایانه، معادل با Shift به آن حالت می باشد.
 - انتقال به یک حالت دیگر بازای یک غیر پایانه، معادل با Goto به آن حالت می باشد.
 - اگر در یک مجموعه، پیکربندی داشتیم که نقطه در آخر آن قرار گرفته بود، آن حالت یک حالت کاهش (با کمک قاعده مورد نظر) می باشد. در این صورت بازای تمام پایانه ها عمل کاهش را مشخص می کنیم LR(0).
 - اگر در حالت i ام، پیکربندی $S \bullet S'$ را داشتیم، $Action[i, \$]$ را پذیرش (Accept) قرار می دهیم.

مثال 3 - گراف انتقال



$S \quad X \quad \$$
 $X \quad (X)$
 $X \quad ()$

مثال 3 - جدول تجزیه

	ACTION			Goto
State	()	\$	X
s0	shift to s2	error	error	goto s1
s1	error	error	accept	
s2	shift to s2	shift to s5	error	goto s3
s3	error	shift to s4	error	
s4	reduce (2)	reduce (2)	reduce (2)	
s5	reduce (3)	reduce (3)	reduce (3)	

