

قسمت ششم - تجزیه پائین به بالا - تجزیه گره‌های SLR

احمد خادم زاده ([khademzadeh@mshdiau.ac.ir](mailto:khademzadeh@mshdiau.ac.ir))

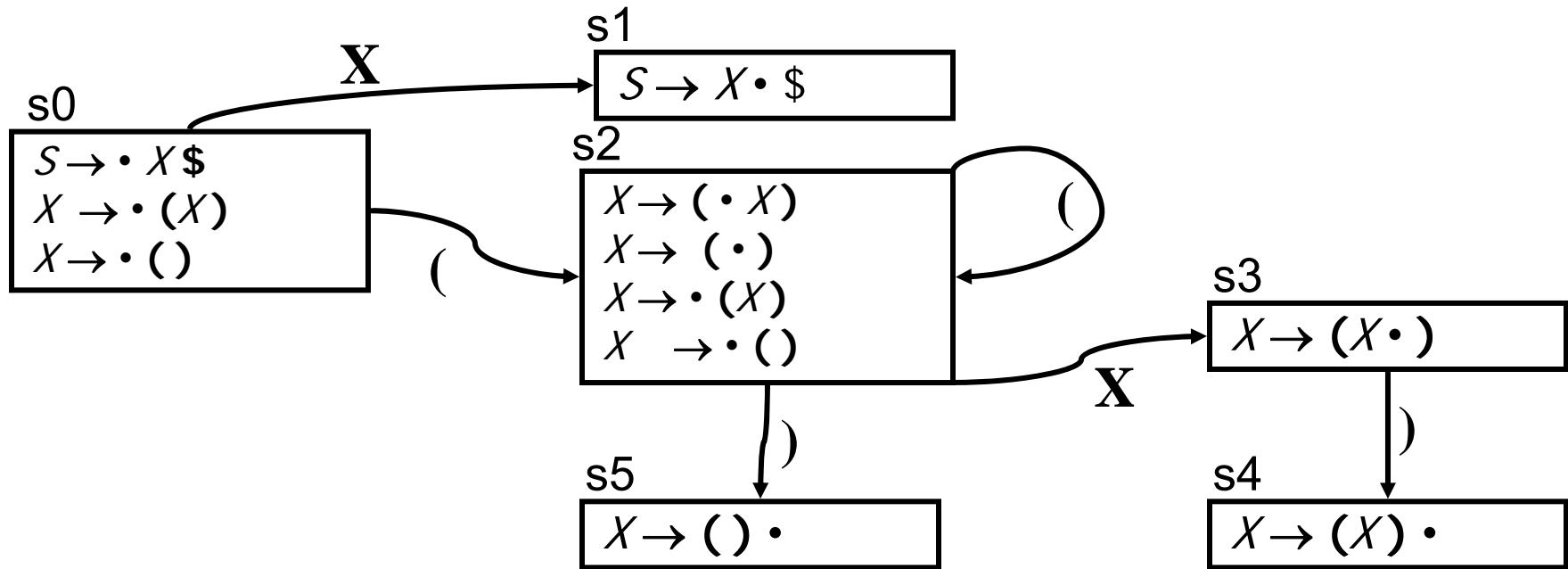
دانشگاه آزاد مشهد

# اصول طراحی کامپایلر

# نحوه ساخت جدول تجزیه (اسلاید تکراری)

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

# مثال دو - گراف انتقال (اسلاید تکراری)



$S \rightarrow X \$$

$X \rightarrow (X)$

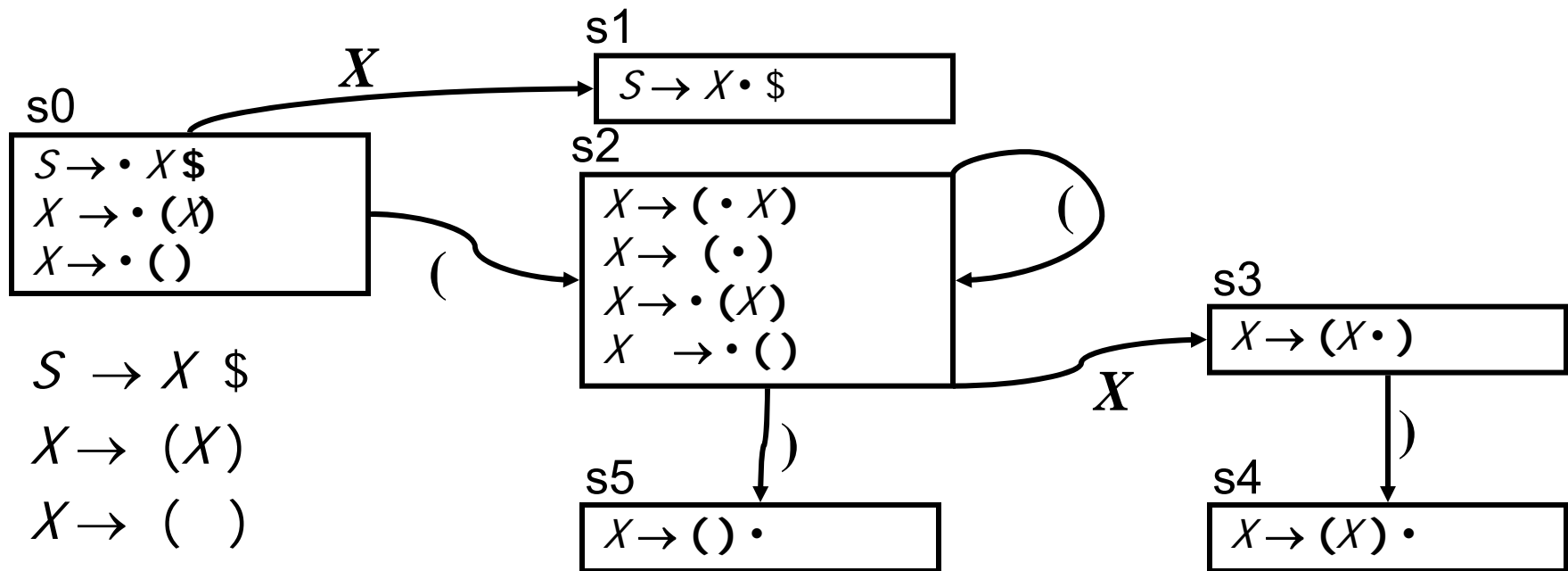
$X \rightarrow ( )$

# ساخت جدول تجزیه (اسلاید تکراری)

- 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.
- اگر در حالت  $i$  ام، پیکربندی  $S \rightarrow S'$  را داشتیم،  $Action[i, \$]$  را پذیرش (Accept) قرار می دهیم.

# مثال دو - جدول تجزیه (اسلاید تکراری)

State	ACTION			Goto
	(	)	\$	
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)	



# مثال

- برای گرامر زیر گراف انتقال را می کشیم.

$$S \rightarrow X \$$$

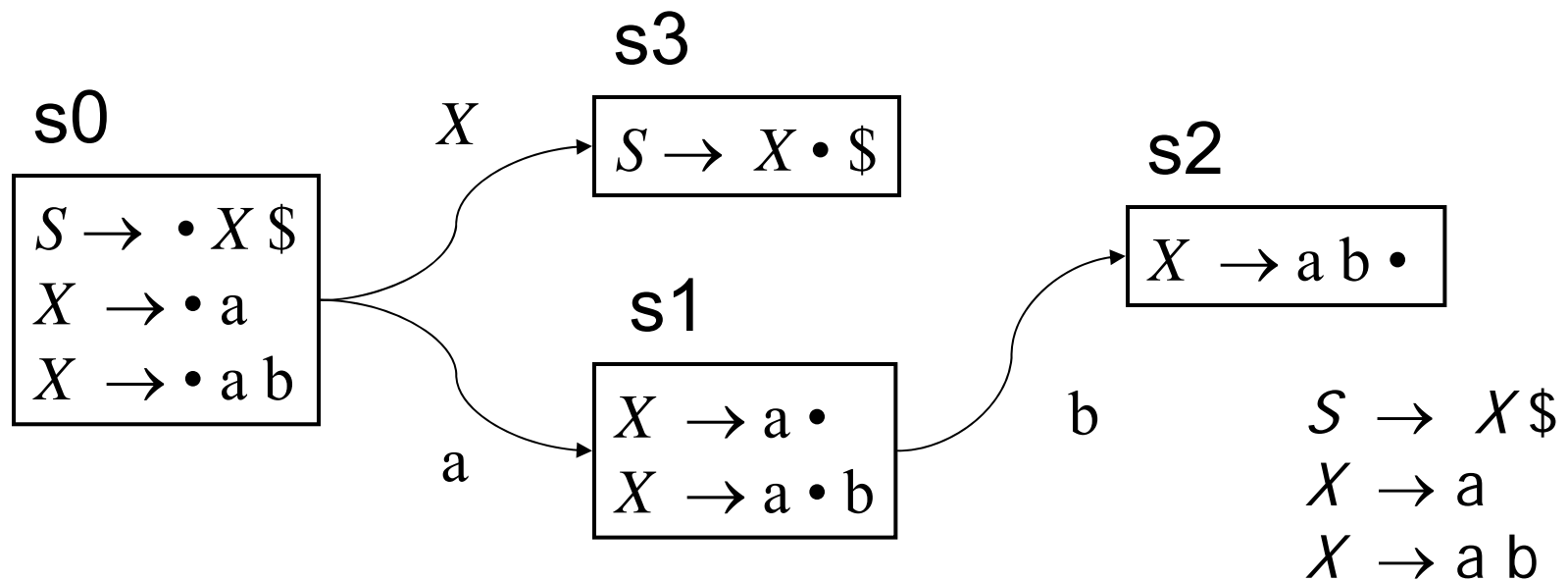
$$X \rightarrow a$$

$$X \rightarrow a b$$

- آیا این گرامر یک گرامر LR(0) است؟ چرا؟

# مثال - ادامه

	ACTION			Goto
State	a	b	\$	X
s0	shift to s1	error	error	goto s3
s1	reduce(2)	<b>S/R Conflict</b>	reduce(2)	
s2	reduce(3)	reduce(3)	reduce(3)	
s3	error	error	accept	



# مشکل تجزیه گره‌های LR(0)

- مشکل این تجزیه گر عدد 0 آن است، یعنی در این تجزیه گر از نشانه پیش بینی استفاده نمی شود.
- امکان برخورد های ذیل در این تجزیه گر وجود دارد:
  - برخورد انتقال-کاهش
  - برخورد کاهش-کاهش
- راه حل ؟
  - استفاده از یک ترم پیش بینی
- از ترم پیش بینی برای انتقال یا کاهش استفاده کنیم ؟
- سه روش مختلف  $SLR(1)$ ،  $LALR(1)$  و  $LR(1)$  از ترم پیش بینی به روش های مختلفی استفاده می کنند.

# استفاده از ترم پیش بینی فقط برای کاهش

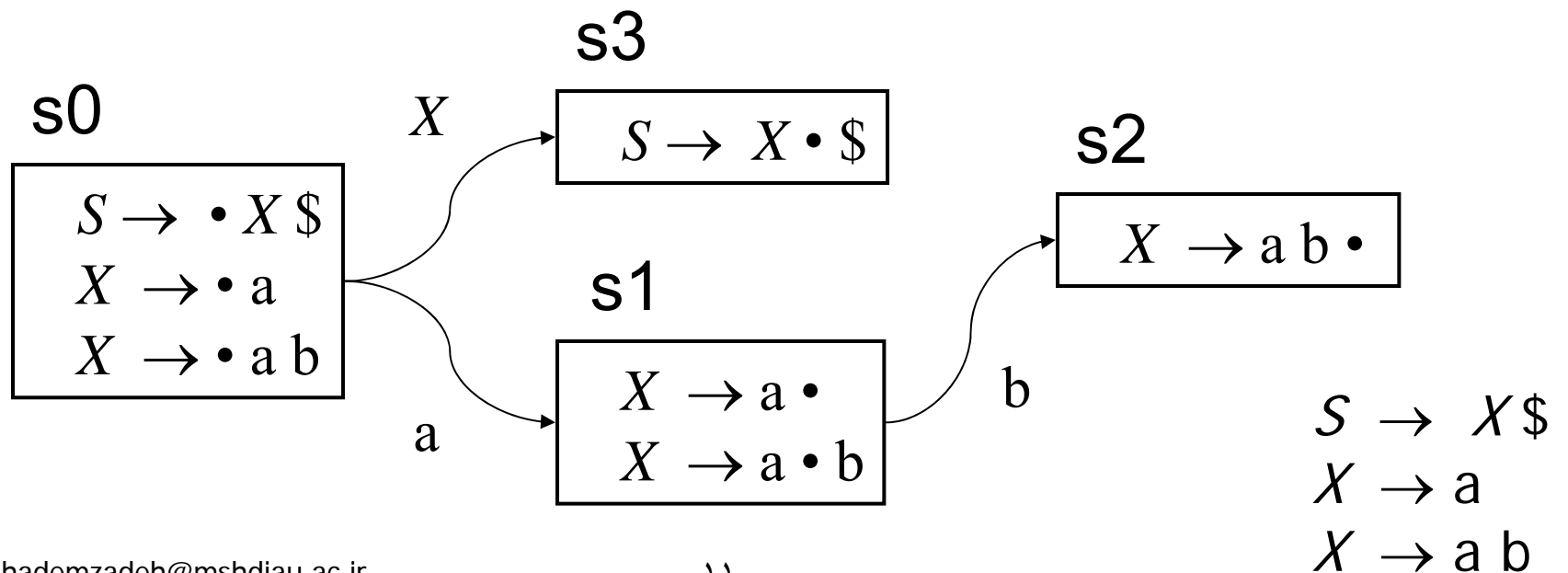
- به این روش  $SLR(1)$  گفته می شود.
- $S$  در این تجزیه گر اختصار چیست؟
- در این تجزیه گر همه چیز مشابه  $LR(0)$  می باشد بجز :
  - اگر در یک حالت آئمی مثل  $A \rightarrow \beta \bullet$  داشته باشیم، فقط در صورتی عمل کاهش را انجام می دهیم، که نشانه بعدی ورودی در ... قرار داشته باشد.
- با داشتن یک نشانه پیش بینی، مثال قبل چه تغییری می کند؟

# ساخت جدول تجزیه (1) SLR

- 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 مثل  $\alpha \rightarrow X$ ، آن حالت یک حالت کاهش (با کمک قاعده  $\alpha \rightarrow X$ ) می باشد. در این صورت فقط برای مجموعه پیرو غیر پایانه یعنی  $X$ ، عمل کاهش را مشخص می کنیم.

# جدول تجزیه جدید

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



# برخورد کاهش-کاهش و $SLR(1)$

- در مثال اخیر دیدیم که  $SLR(1)$  می تواند مشکل برخورد انتقال-کاهش را مرتفع نماید.
- آیا این تجزیه گر در مورد برخورد کاهش-کاهش نیز کاربردی دارد؟
- گرامر زیر را در نظر بگیرید؛ آیا این گرامر  $LR(0)$  است؟ چرا؟

$$E' \rightarrow E$$

$$E \rightarrow E + T \mid T \mid V = E$$

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

$$V \rightarrow id$$

# چه گرامری $SLR(1)$ است؟

- اگر در یک مجموعه آیتمی مثل  $A \rightarrow u \bullet x v$  که  $x$  در آن پایانه است داشتیم؛ و در همان مجموعه آیتم کاملی مثل  $B \rightarrow w \bullet$  داشتیم آنگاه
  - $x$  نباید عضو  $Follow(B)$  باشد. چرا؟
- اگر در یک مجموعه دو آیتم کامل مثل  $A \rightarrow u \bullet$  و  $B \rightarrow v \bullet$  داشتیم، آنگاه
  - $Follow(A) \cap Follow(B)$  باید تهی باشد. چرا؟
- گرامرهای  $LR(0)$  و  $SLR(1)$  چه نسبتی با هم دارند؟

# کاستی (1) SLR

- از تمام اطلاعاتی که از قبل بدست آورده ایم استفاده نمی کند!
- در (1) SLR اگر  $X \rightarrow u$  داشته باشیم، یعنی دستگیره  $u$  را در بالای پشته داریم.
- در چه صورتی عمل کاهش را انجام می دهیم؟
- آیا ما از تمام اطلاعات موجود در پشته استفاده کرده ایم؟
- آیا مسیری از گرامر که موجب رسیدن ما به این حالت شده است را، در نظر گرفته ایم؟
- آیا این امکان وجود دارد که اطلاعاتی که در پائین  $u$  قرار دارند، به ما دستگیره مناسبتری را پیشنهاد دهند؟
- شاید نباید بازای کل مجموعه پیرو  $X$  عمل کاهش را انجام داد.

# مثال – کاستی $SLR(1)$

• گرامر زیر را در نظر بگیرید:

$$0) S' \rightarrow S$$

$$1) S \rightarrow XX$$

$$2) X \rightarrow aX$$

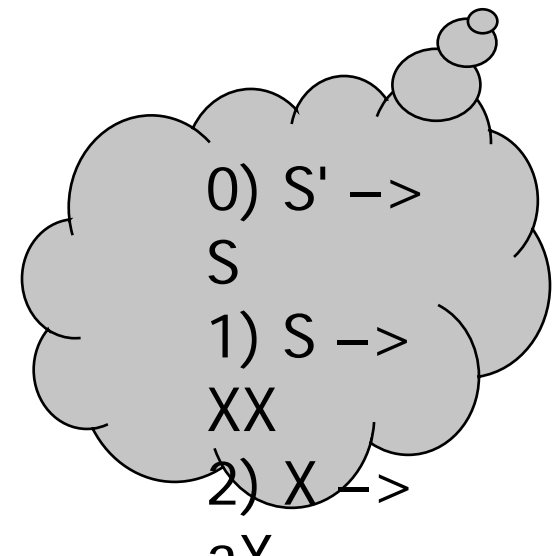
$$3) X \rightarrow b$$

• سعی می کنیم برای این گرامر تجزیه  $SLR(1)$  را انجام دهیم.

# مثال – کاستی SLR(1) (ادامه)

<b>I0:</b> $S' \rightarrow \bullet S$	I1	<b>I3:</b> $X \rightarrow a \bullet X$	I6
$S \rightarrow \bullet XX$	I2	$X \rightarrow \bullet aX$	I3
$X \rightarrow \bullet aX$	I3	$X \rightarrow \bullet b$	I4
$X \rightarrow \bullet b$	I4	<b>I4:</b> $X \rightarrow b \bullet$	Red3
<b>I1:</b> $S' \rightarrow S \bullet$	Acc	<b>I5:</b> $S \rightarrow XX \bullet$	Red1
<b>I2:</b> $S \rightarrow X \bullet X$	I5	<b>I6:</b> $X \rightarrow aX \bullet$	Red2
$X \rightarrow \bullet aX$	I3		
$X \rightarrow \bullet b$	I4		

رشته ای مثال بنزید که توسط این گرامر تولید می شود،  
اما توسط این تجزیه گر، پذیرفته نمی شود.



# کاستی (1) SLR

- از تمام اطلاعاتی که از قبل بدست آورده ایم استفاده نمی کند!
- در (1) SLR اگر  $X \rightarrow u$  داشته باشیم، یعنی دستگیره  $u$  را در بالای پشته داریم.
- در چه صورتی عمل کاهش را انجام می دهیم؟
- آیا ما از تمام اطلاعات موجود در پشته استفاده کرده ایم؟
- آیا مسیری از گرامر که موجب رسیدن ما به این حالت شده است را، در نظر گرفته ایم؟
- آیا این امکان وجود دارد که اطلاعاتی که در پائین  $u$  قرار دارند، به ما دستگیره مناسبتری را پیشنهاد دهند؟
- شاید نباید بازای کل مجموعه پیرو  $X$  عمل کاهش را انجام داد.

# ایدهٔ اصلی در LR(1)

- حالات موجود در LR(0) را با توجه به ترم پیش بینی، می شکند. یعنی ممکن است یک حالت به چند حالت تبدیل شود.
- عمل کاهش را با توجه به آیتم و ترم پیش بینی انجام می دهد.
- عمل کاهش را به ازای کل مجموعهٔ پیرو انجام نمی دهد.

# آیتم ها در LR(1)

- آیتم ها شامل اطلاعات زیر هستند:
  - قواعد تولید
  - موقعیت نقطه در سمت راست قواعد
  - ترم پیش بینی
- شکل آیتم ها بصورت  $[A \rightarrow \alpha \bullet \beta, t]$  است که
  - $A \rightarrow \alpha \beta$  قاعده مورد نظر
  - $A \rightarrow \alpha \bullet \beta$  - مکان نقطه در قاعده
  - ترم پیش بینی -  $t$  نشانگر یک پایانه یا نماد انتهای رشته (\$) می باشد.
  - ممکن است بیش از یک ترم پیش بینی هم داشته باشیم که آنها را با نماد / از هم جدا می کنیم. مثلا

$$A \rightarrow u \bullet, a/b/c$$

# معنی یک آیتم LR(1)

- آیتم  $[A \rightarrow \alpha \bullet \beta, t]$  به این معنی است که
  - تجزیه گر  $\alpha$  را تجزیه کرده است،
  - اگر  $\beta$  را تجزیه کند، نماد ورودی بعدی  $t$  خواهد بود.
  - سپس تجزیه گر باید با توجه به قاعده  $A \rightarrow \alpha \beta$  باید عمل کاهش را انجام دهد.
- معنی آیتم  $A \rightarrow X_1 \dots X_j \bullet, a$  نیز این می باشد که عمل کاهش فقط در صورتی امکان پذیر است که نشانه ورودی بعدی  $a$  باشد.
  - در تجزیه گر SLR(1) چگونه بود؟
- آیا هر عنصری از پایانه ها می تواند نشانه پیش بینی یک آیتم LR(1) باشد؟

# اندازه جدول تجزیه در LR(1)

- در LR(1) می توان مجموعه های کاملاً مشابهی داشت که فقط در ترم پیش بینی متفاوتند.
- این دلیلی بر بزرگ شدن بیش از حد جدول تجزیه می باشد.

# ساخت یک تجزیه گر LR(1)

- تعیین بستار (Closure) و مابعد (Goto) برای آیتم های LR(1)
- ساخت DFA
- ساخت جدول تجزیه

# الگوریتم بستار

## Closure(I)

repeat

for all items  $[A \rightarrow \alpha \bullet X \beta \quad c]$  in I

for any production  $X \rightarrow \gamma$

for any  $d \in \text{First}(\beta c)$

$$I = I \cup \{ [X \rightarrow \bullet \gamma \quad d] \}$$

until I does not change

# الگوریتم بستار

**Goto(I, X)**

$J = \{ \}$

for any item  $[A \rightarrow \alpha \bullet X \beta \quad c]$  in I

$J = J \cup \{[A \rightarrow \alpha X \bullet \beta \quad c]\}$

return Closure(J)

# ساخت DFA

- با آیتم  $[S' \rightarrow \bullet S \$, \quad |]$  شروع می کنیم.
- بستار این آیتم را نوشته و یک حالت را می سازیم.
- از حالات موجود یکی (مثلا  $In$ ) را انتخاب می کنیم.
- بازای هر آیتم  $[A \rightarrow \alpha \bullet X \beta, \quad c]$  در  $In$ 
  - مجموعه  $Goto(In, X)$  را حساب می کنیم.
  - اگر  $Goto(In, X)$  از قبل وجود نداشت، آن را می سازیم.
  - لبه  $X$  را از  $In$  به  $Goto(In, X)$  وصل می کنیم.
- قدم قبلی را تا زمان تولید حالات جدید تکرار می کنیم.

# ساخت جدول تجزیه

- انتقال به یک حالت دیگر با یک پایانه به معنی انتقال به آن حالت است.
- انتقال به یک حالت دیگر با یک غیر پایانه، به معنی `Goto` به آن حالت است.
- اگر آیتمی مثل  $[A \rightarrow \alpha \bullet, a]$  را در یک حالت داشتیم، در خانه با عنوان `a` در این حالت، عمل کاهش با این قاعده را قرار می دهیم.

# مثال

- I0:**  $S' \rightarrow \bullet S, \$$   
 $S \rightarrow \bullet XX, \$$   
 $X \rightarrow \bullet aX, a/b$   
 $X \rightarrow \bullet b, a/b$
- I1:**  $S' \rightarrow S\bullet, \$$
- I2:**  $S \rightarrow X\bullet X, \$$   
 $X \rightarrow \bullet aX, \$$   
 $X \rightarrow \bullet b, \$$
- I3:**  $X \rightarrow a\bullet X, a/b$   
 $X \rightarrow \bullet aX, a/b$   
 $X \rightarrow \bullet b, a/b$

**I4:**  $X \rightarrow b\bullet, a/b$

**I5:**  $S \rightarrow XX\bullet, \$$

**I6:**  $X \rightarrow a\bullet X, \$$

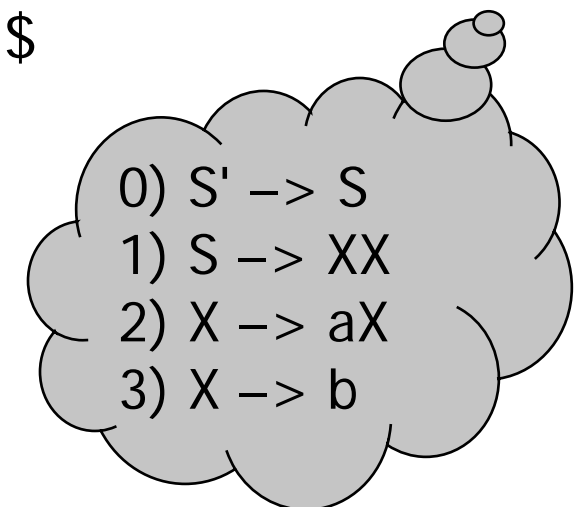
$X \rightarrow \bullet aX, \$$

$X \rightarrow \bullet b, \$$

**I7:**  $X \rightarrow b\bullet, \$$

**I8:**  $X \rightarrow aX\bullet, a/b$

**I9:**  $X \rightarrow aX\bullet, \$$



# جدول تجزیه

State on top of stack	Action			Goto	
	a	b	\$	S	X
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		



# مثال دیگر از SLR(1)

- گرامر زیر را در نظر بگیرید:

$$S' \rightarrow S$$

$$S \rightarrow L = R$$

$$S \rightarrow R$$

$$L \rightarrow *R$$

$$L \rightarrow \text{id}$$

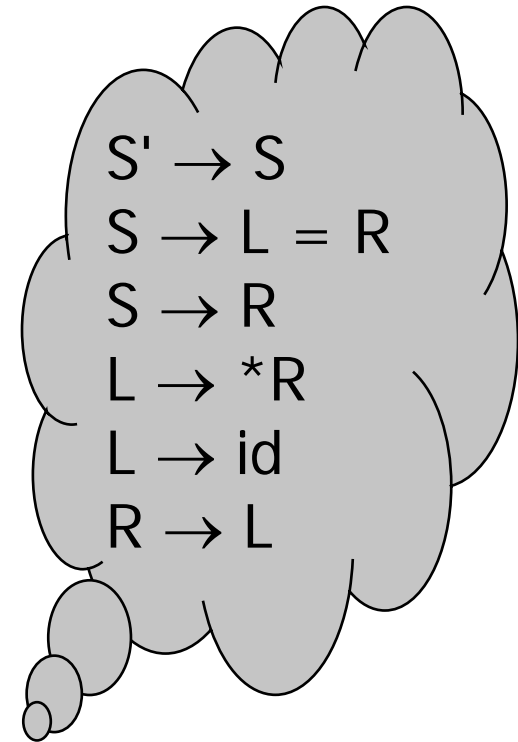
$$R \rightarrow L$$

- این گرامر مربوط به دستور انتساب می باشد.
- برای این گرامر

# مثال دیگر از SLR(1) (ادامه)

10:  $S' \rightarrow \bullet S$   
 $S \rightarrow \bullet L = R$   
 $S \rightarrow \bullet R$   
 $L \rightarrow \bullet * R$   
 $L \rightarrow \bullet id$   
 $R \rightarrow \bullet L$   
11:  $S' \rightarrow S \bullet$   
12:  $S \rightarrow L \bullet = R$   
 $R \rightarrow L \bullet$   
13:  $S \rightarrow R \bullet$   
14:  $L \rightarrow * \bullet R$   
 $R \rightarrow \bullet L$   
 $L \rightarrow \bullet * R$   
 $L \rightarrow \bullet id$

15:  $L \rightarrow id \bullet$   
16:  $S \rightarrow L = \bullet R$   
 $R \rightarrow \bullet L$   
 $L \rightarrow \bullet * R$   
 $L \rightarrow \bullet id$   
17:  $L \rightarrow * R \bullet$   
18:  $R \rightarrow L \bullet$   
19:  $S \rightarrow L = R \bullet$



فرض کنید می خواهیم رشته  $id=id$  را تجزیه کنیم.