



اصول طراحی کامپیوترها

حسین کارشناس

دانشکده مهندسی کامپیوتر

ترم اول ۹۸ - ۹۷

تجزیه LR

- گرامرهای غیر SLR(1)

- وجود برخورد (انتقال/کاهش یا کاهش/کاهش) در برخی حالتها

- مثال: گرامرهای زیر SLR(1) نیستند

$$\begin{array}{lcl} S & \rightarrow & L = R \mid R \\ L & \rightarrow & *R \mid \text{id} \\ R & \rightarrow & L \end{array}$$

$$\begin{array}{l} S \rightarrow S a S \\ S \rightarrow b \end{array}$$

$$\begin{array}{lcl} S & \rightarrow & A a A b \mid B b B a \\ A & \rightarrow & \epsilon \\ B & \rightarrow & \epsilon \end{array}$$

- نیاز به الگوریتمهای قویتر تجزیه LR

- Canonical LR – CLR

- کلیترین روش برای ساخت جداول تجزیه LR

- Lookahead LR – LALR

- بهینه‌سازی CLR برای تجزیه سریع‌تر

تجزیه LR

- نگهداری اطلاعات بیشتر در موارد برای تصمیم‌گیری در مورد کاهش
- اگر حالتی از ماشین $LR(0)$ حاوی مورد $A \rightarrow \beta$ بوده و ورودی بعدی (a) در مجموعه پسین A باشد ($a \in \text{Follow}(A)$) کاهش با قاعده تولید $A \rightarrow \beta$ فقط وقتی صحیح است که $\alpha A a$ یک پیشوند ممکن باشد
- نگهداری نشانه‌های پایانی که می‌توانند پس از دستگیره β قرار داشته باشند
 - استفاده از موارد $LR(1)$
- موارد $LR(1)$ یک قاعده تولید گرامر
 - یک زوج مؤلفه به صورت $[A \rightarrow \beta.\gamma, a]$
 - $A \rightarrow \beta.\gamma$ یک مورد $LR(0)$ است
 - a یک نشانه پایانی گرامر یا نشانه ویژه انتهای ورودی (\$) است

تجزیه LR

- موارد LR(1) یک قاعده تولید گرامر (ادامه)
- مؤلفه‌های دوم ممکن برای مورد $[A \rightarrow \beta.\gamma, a]$ زیر مجموعه‌ای از مجموعه پسین A ($\text{Follow}(A)$) هستند
- استفاده از مورد $[A \rightarrow \beta.\gamma, a]$ برای تصمیم‌گیری در مورد کاهش
- اگر حالت فعلی ماشین حاوی مورد $[A \rightarrow \beta., a]$ باشد ($\gamma = \epsilon$) و ورودی بعدی a باشد
آنگاه کاهش با $A \rightarrow \beta$
- موارد LR(1) معتبر برای یک پیشوند ممکن
- مورد $[A \rightarrow \beta.\gamma, a]$ برای پیشوند ممکن $\alpha\beta$ معتبر است اگر اشتقاق راست $S' \xRightarrow{*} \alpha A w \Rightarrow \alpha \beta \gamma w$ وجود داشته باشد و داشته باشیم:
 $a \in \{ \text{First}(w) - \epsilon \} \cup \{ \$: \text{nullable}(w) \}$

تجزیه LR

- ماشین LR(1) برای بررسی محتوای پشت به نگاه به ورودی
- محاسبه بسته استاندارد مجموعه موارد LR(1)
- کاملاً مشابه بسته استاندارد مجموعه موارد LR(0) محاسبه می شود
- تعاریف جدید برای عملگرهای بستار و انتقال

```
void items( $G'$ ) {  
    initialize  $C$  to CLOSURE( $\{[S' \rightarrow \cdot S, \$]\}$ );  
    repeat  
        for ( each set of items  $I$  in  $C$  )  
            for ( each grammar symbol  $X$  )  
                if ( GOTO( $I, X$ ) is not empty and not in  $C$  )  
                    add GOTO( $I, X$ ) to  $C$ ;  
    until no new sets of items are added to  $C$ ;  
}
```

تجزیه LR

• بستار یک مجموعه موارد LR(1) (I)

```
SetOfItems CLOSURE(I) {  
    repeat  
        for ( each item  $[A \rightarrow \alpha \cdot B \beta, a]$  in  $I$  )  
            for ( each production  $B \rightarrow \gamma$  in  $G'$  )  
                for ( each terminal  $b$  in FIRST( $\beta a$ ) )  
                    add  $[B \rightarrow \cdot \gamma, b]$  to set  $I$ ;  
    until no more items are added to  $I$ ;  
    return  $I$ ;  
}
```

- تمام موارد موجود در مجموعه موارد I در بستار آن هستند
- b نشان‌دهنده تمام نشانه‌های پایانی است که در یک اشتقاق راست معتبر می‌تواند بعد از نشانه غیرپایانی B بیاید: $S' \xRightarrow{*} \delta A a w \Rightarrow \delta \alpha \underline{B \beta} a w$

تجزیه LR

- انتقال از یک مجموعه موارد LR(1) (I) با نشانه X

```
SetOfItems GOTO(I, X) {  
    initialize J to be the empty set;  
    for ( each item  $[A \rightarrow \alpha \cdot \underline{X} \beta, a]$  in I )  
        add item  $[A \rightarrow \alpha \underline{X} \cdot \beta, a]$  to set J;  
    return CLOSURE(J);  
}
```

- در هنگام انتقال، مؤلفه دوم بدون تغییر با موارد منتقل می‌شود

• ماشین LR(1)

- حالت‌های ماشین مجموعه‌های موارد LR(1) هستند

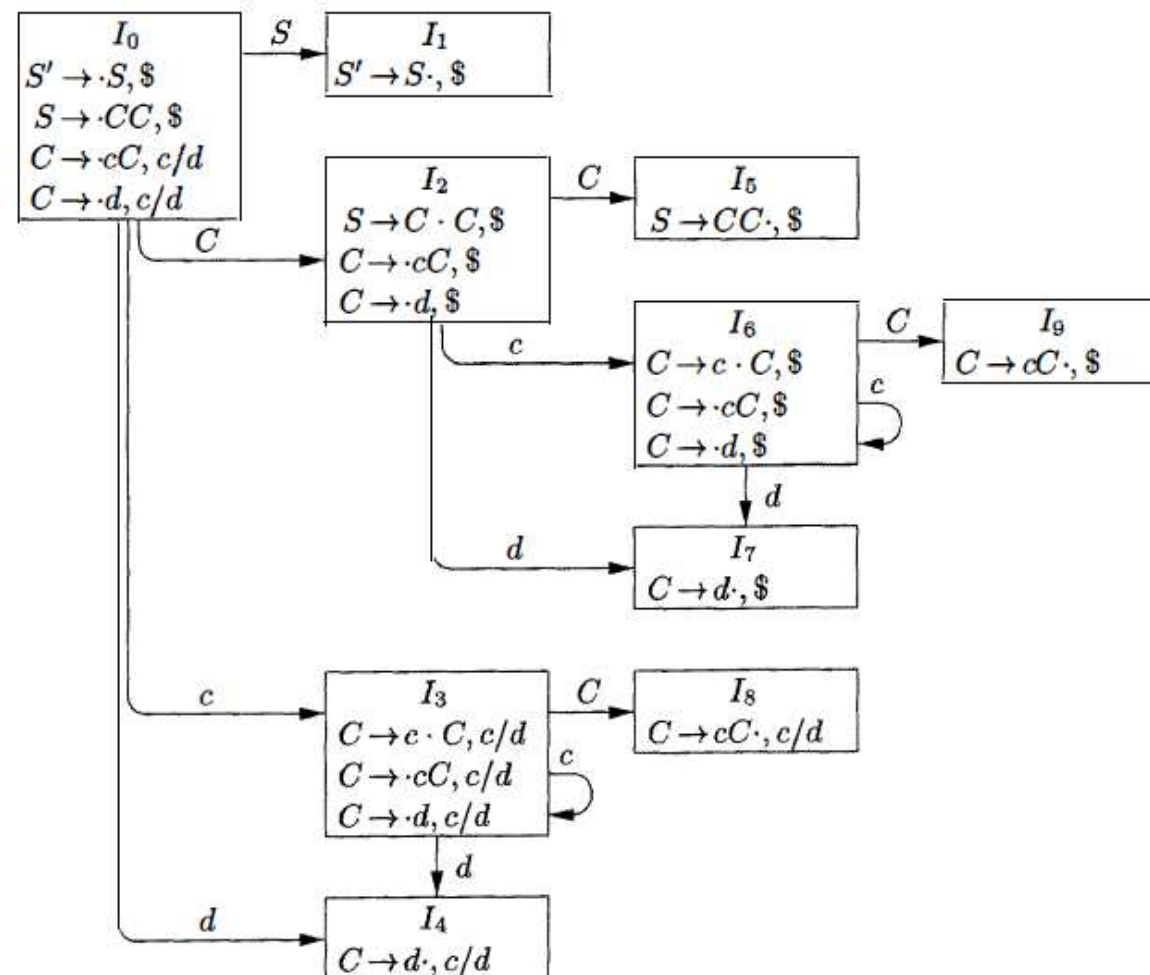
- تمام حالت‌ها نهایی هستند

- حالت اولیه ماشین: $\text{Closure}(\{[S' \rightarrow \cdot S, \$]\})$

- انتقال‌های ماشین با استفاده از عملگر انتقال (Goto) مشخص می‌شوند

تجزیه LR

• مثال: ماشین LR(1) برای گرامر مضاعف روبرو

$$\begin{aligned} S' &\rightarrow S \\ S &\rightarrow CC \\ C &\rightarrow cC \mid d \end{aligned}$$


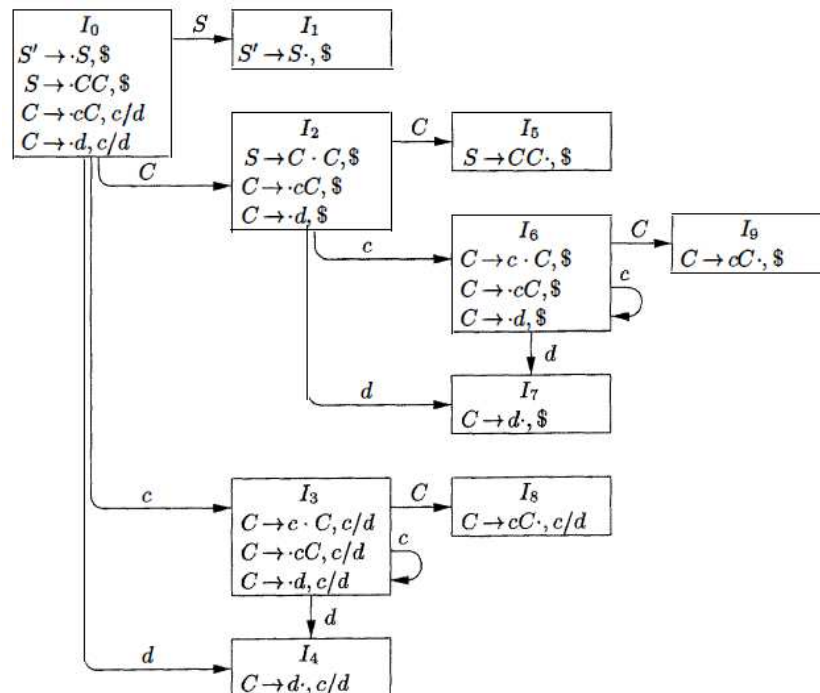
تجزیه LR

- تجزیه CLR با استفاده از ماشین LR(1) و با توجه به ورودی فعلی
- جدول تجزیه CLR
- بخش کنش‌ها ($\text{Action}[s, a]$)
 - اگر حالت s حاوی مورد $[A \rightarrow \alpha. , a]$ باشد ($A \neq S'$) آنگاه:
 $\text{Action}[s, a] = \text{Reduce } A \rightarrow \alpha$
 - اگر حالت s حاوی مورد $[S' \rightarrow S. , \$]$ باشد آنگاه: $\text{Action}[s, \$] = \text{Accept}$
 - اگر حالت s حاوی مورد $[A \rightarrow \alpha.a\beta, b]$ باشد آنگاه: $\text{Action}[s, a] = \text{Shift}$
 - اگر s حالت خطا (حالت بن‌بست) باشد آنگاه: $\text{Action}[s, :] = \text{Error}$
- بخش انتقال ($\text{Goto}[s, X]$)
- پیاده‌سازی مدل انتقال ماشین LR(1)

تجزیه LR

• مثال: جدول تجزیه CLR برای گرامر مضاعف روبرو

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & CC \\ C & \rightarrow & cC \mid d \end{array} \quad \begin{array}{l} (1) \\ (2) \\ (3) \end{array}$$



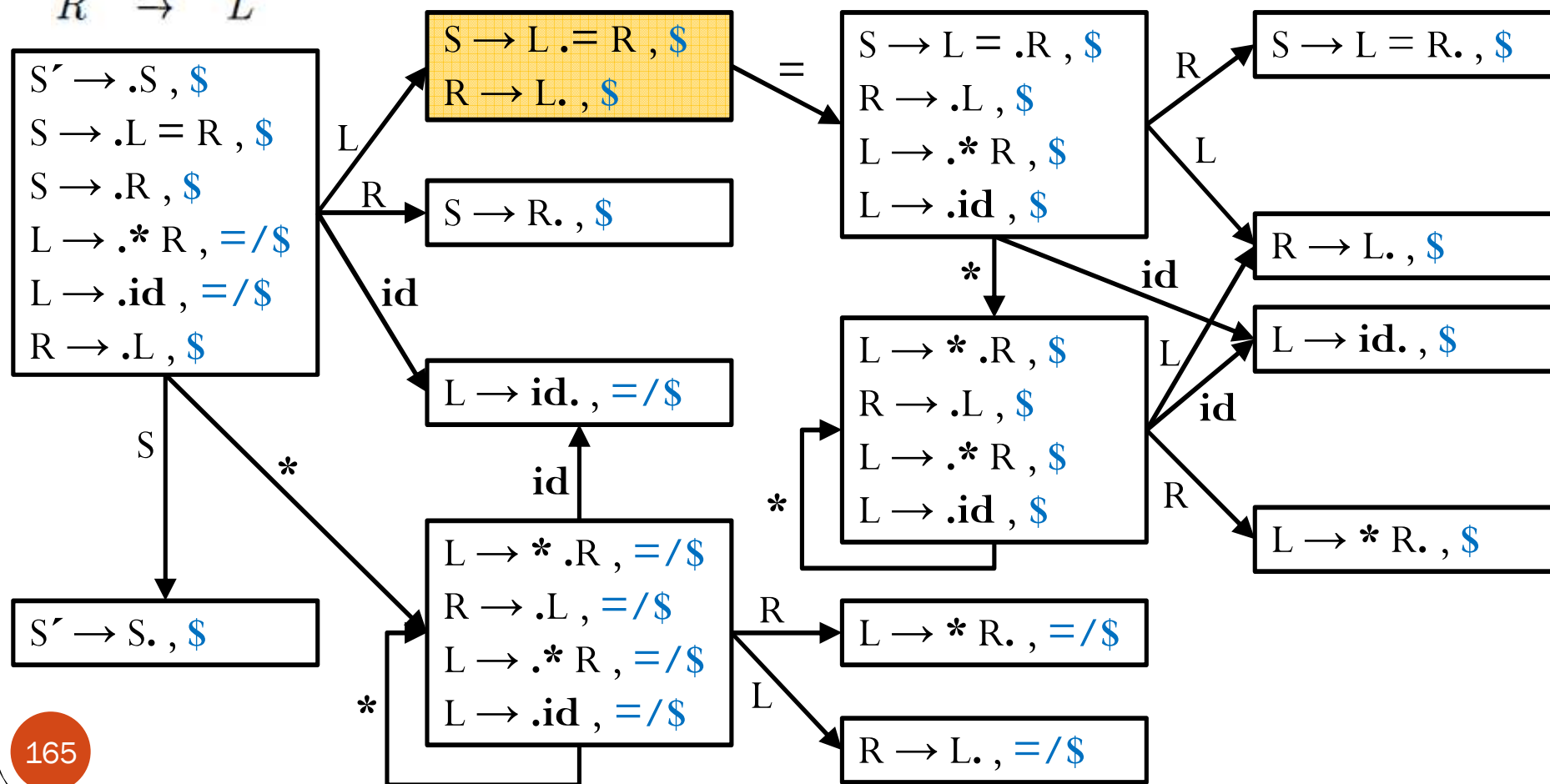
STATE	ACTION			GOTO	
	c	d	\$	S	C
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

• سوال: با توجه به جدول تجزیه بالا رشته $ccdd\$$ چگونه تجزیه می شود؟

تجزیه LR

• مثال: ماشین LR(1) برای گرامر روبرو

$S \rightarrow L = R \mid R$
 $L \rightarrow *R \mid \text{id}$
 $R \rightarrow L$



تجزیه LR

• گرامر $LR(1)$

- گرامری که در هر خانه از بخش کنش‌های جدول CLR ساخته شده برای آن حداکثر یک کنش وجود داشته باشد
- نباید برخوردی بین کنش‌ها وجود داشته باشد

• هر گرامر $SLR(1)$ یک گرامر $LR(1)$ است

- تفکیک حالت‌ها در تجزیه CLR با توجه به ورودی
- برای یک گرامر $SLR(1)$ تعداد حالت‌های تجزیه‌گر CLR به مراتب بیشتر از تجزیه‌گر SLR ساخته شده برای آن گرامر است
- امکان وجود چندین مجموعه موارد $LR(1)$ برای هر مجموعه موارد $LR(0)$
- تفاوت این مجموعه‌های موارد فقط در مؤلفه‌های دوم آنهاست

تجزیه LR

- مثال: برای گرامر یک زبان برنامه‌نویسی متداول مانند C
 - تجزیه‌گر SLR دارای حدود چند صد حالت است
 - توصیف برخی از ساختارهای برنامه‌نویسی با گرامر SLR به راحتی ممکن نیست
 - تجزیه‌گر CLR دارای حدود چند هزار حالت است
- اهمیت کاهش تعداد حالت‌های تجزیه
 - کاهش حافظه مورد نیاز برای پیاده‌سازی تجزیه‌گر
 - کاهش زمان تجزیه برای برنامه‌های ورودی
- تجزیه‌گر LALR

تجزیه LR

- کاهش تعداد حالت‌های تجزیه با ادغام مجموعه‌های موارد LR(1)
- ادغام مجموعه‌های مواردی که دارای مؤلفه‌های اول (هسته) یکسان هستند
 - هسته (core) یک مجموعه موارد LR(1) همان مجموعه موارد LR(0) متناظر با آن است
- مثال: ادغام دو مجموعه موارد LR(1)

$C \rightarrow d\cdot, \$$

→

$C \rightarrow d\cdot, c/d/\$$

$C \rightarrow d\cdot, c/d$
- آیا ادغام مجموعه موارد LR(1) باعث ایجاد برخورد می‌شود؟
 - باعث برخورد انتقال/کاهش نمی‌شود
 - تصمیم‌گیری در مورد انتقال با ادغام مجموعه‌ها فرقی نخواهد کرد
 - انتقال از یک مجموعه موارد (I) با یک نشانه (X) فقط به هسته آن بستگی دارد
 - وجود این برخورد پس از ادغام به معنی وجود آن قبل از ادغام در یکی از حالت‌ها است

تجزیه LR

- امکان برخورد کاهش/کاهش با ادغام مجموعه موارد LR(1)
- مثال: گرامر زیر یک گرامر LR(1) است
- جدول تجزیه LR(1) دارای برخورد نیست

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & a A d \mid b B d \mid a B e \mid b A e \\ A & \rightarrow & c \\ B & \rightarrow & c \end{array}$$

- با ادغام دو مجموعه موارد دارای هسته یکسان برخورد کاهش/کاهش ایجاد می شود

$$\begin{array}{l} \{[A \rightarrow c\cdot, d], [B \rightarrow c\cdot, e]\} \\ \{[A \rightarrow c\cdot, e], [B \rightarrow c\cdot, d]\} \end{array}$$



$$\begin{array}{l} A \rightarrow c\cdot, d/e \\ B \rightarrow c\cdot, d/e \end{array}$$

- چنین گرامری LALR(1) نیست

تجزیه LR

• روش تجزیه LALR

- سعی در بهینه‌سازی تجزیه‌گر CLR با کاهش تعداد حالت‌ها
- برای هر گرامر تعداد حالت‌های تجزیه‌گرهای LALR و SLR برابر است
- محاسبه حالت‌های تجزیه‌گر LALR
- برابر با حالت‌های بدست آمده از ادغام مجموعه‌های موارد $LR(1)$ با هسته یکسان

• جدول تجزیه LALR

- بخش کنش‌ها ($Actions[s, a]$)
- کاملاً مشابه بخش کنش‌ها در جدول تجزیه CLR ساخته می‌شود
- بخش انتقال ($Goto[s, X]$)
- پیاده‌سازی مدل انتقال ماشین $LR(1)$ با حالت‌های ادغام شده

تجزیه LR

- عملگر انتقال برای مجموعه موارد ادغام شده (J) با نشانه X
- اگر $J = I_1 \cap I_2 \cap \dots \cap I_k$ باشد آنگاه انتقال از J با نشانه X برابر است با:

$$\text{Goto}(J, X) = \text{Goto}(I_1, X) \cup \text{Goto}(I_2, X) \cup \dots \cup \text{Goto}(I_k, X)$$

گرامر LALR

- بدون برخورد در بخش کنش‌های جدول LALR ساخته شده برای آن

مثال: جدول تجزیه LALR برای گرامر مضاعف روبرو

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & C C \\ C & \rightarrow & c C \mid d \end{array}$$

- مجموعه موارد قابل ادغام

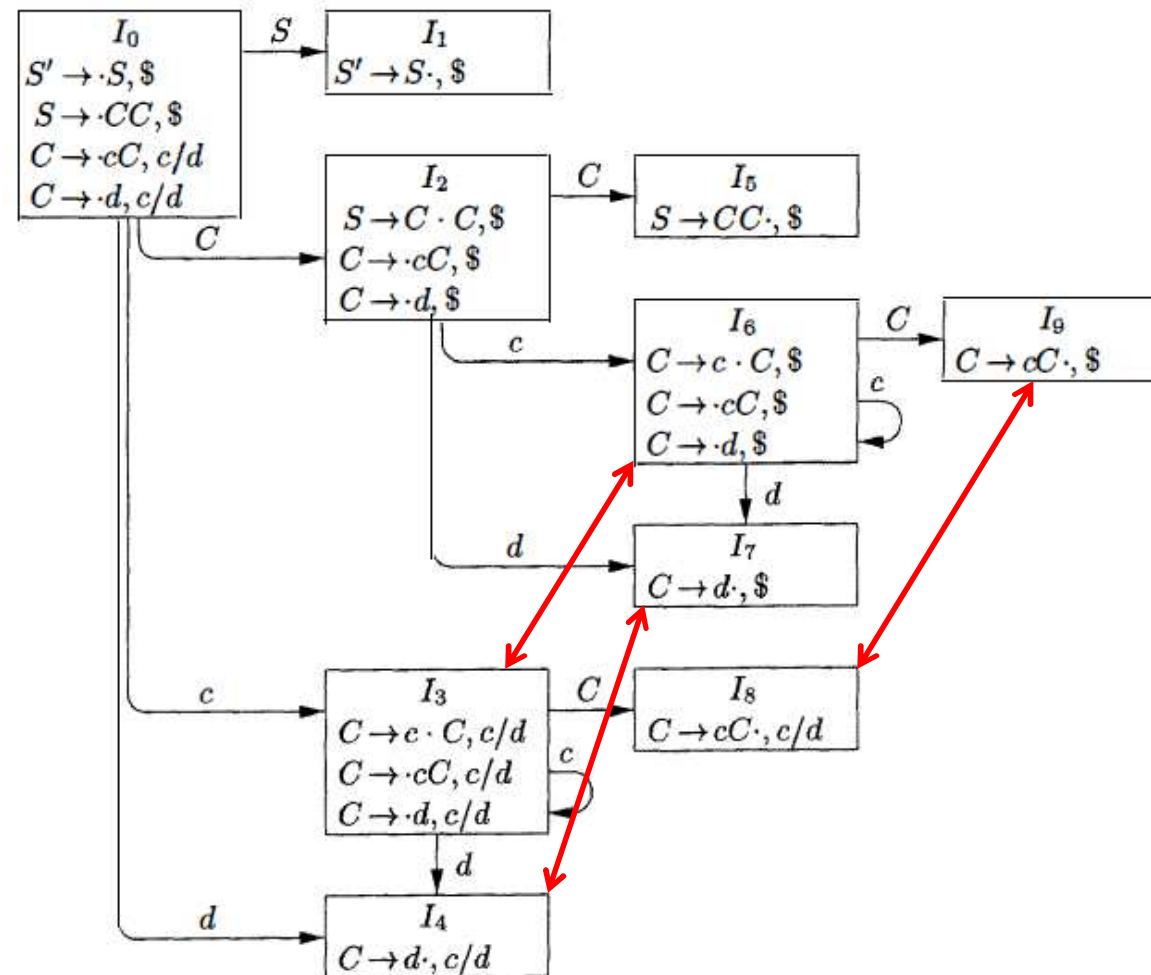
$$\begin{array}{l} I_{36}: \quad C \rightarrow c \cdot C, \quad c/d/\$ \\ \quad \quad C \rightarrow \cdot c C, \quad c/d/\$ \\ \quad \quad C \rightarrow \cdot d, \quad c/d/\$ \end{array}$$

$$I_{47}: \quad C \rightarrow d \cdot, \quad c/d/\$$$

$$I_{89}: \quad C \rightarrow c C \cdot, \quad c/d/\$$$

تجزیه LR

• مثال: جدول تجزیه LALR

$$\begin{array}{lcl} S' & \rightarrow & S \\ S & \rightarrow & CC \\ C & \rightarrow & cC \mid d \end{array}$$


تجزیه LR

$$\begin{aligned}
 S' &\rightarrow S \\
 S &\rightarrow CC \\
 C &\rightarrow cC \mid d
 \end{aligned}$$

(1) (2) (3)

STATE	ACTION			GOTO	
	c	d	\$	S	C
0	s36	s47		1	2
1			acc		
2	s36	s47			5
36	s36	s47			89
47	r3	r3	r3		
5			r1		
89	r2	r2	r2		

جدول تجزیه LALR

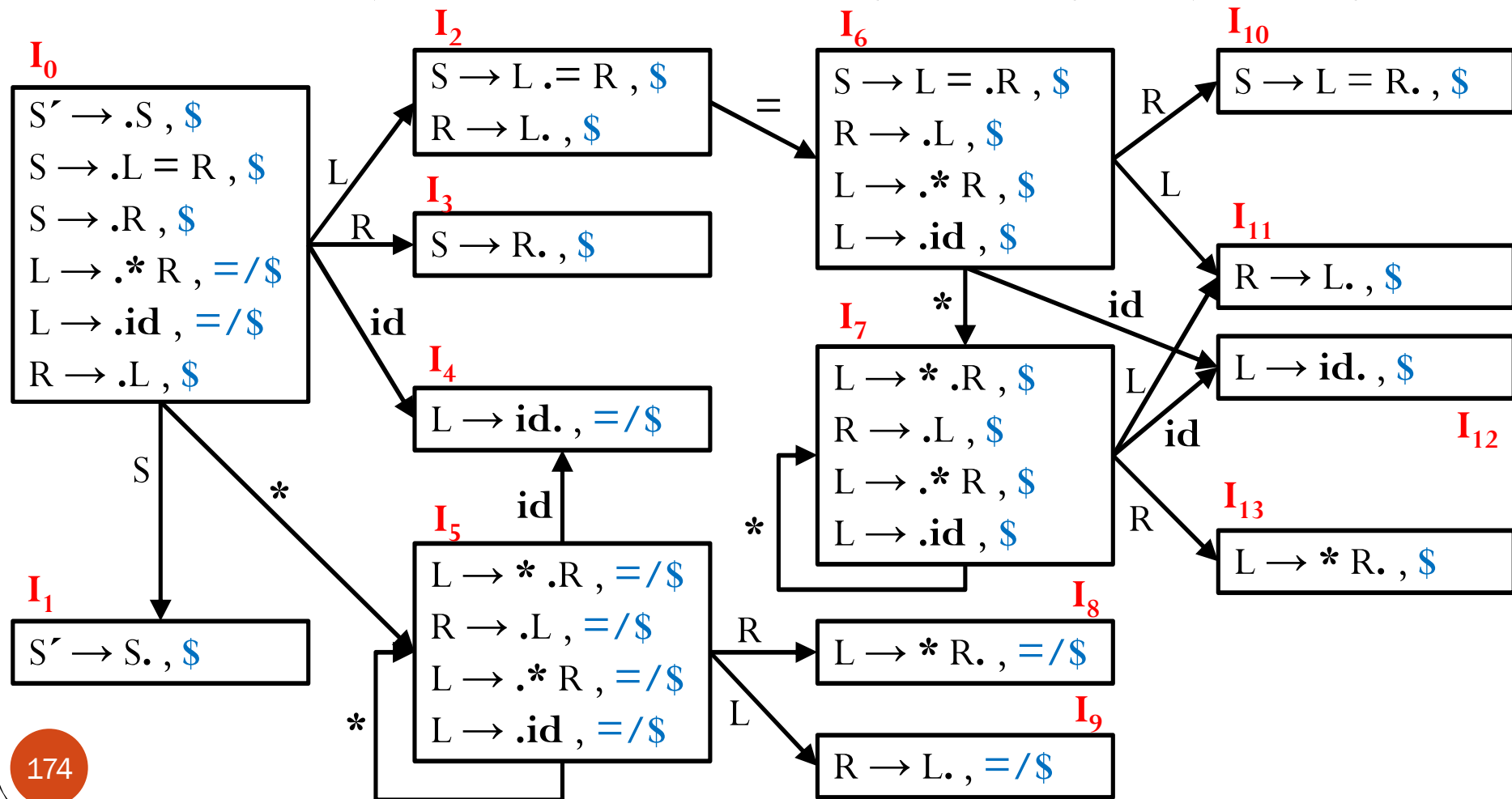
• مثال: جدول تجزیه LALR

STATE	ACTION			GOTO	
	c	d	\$	S	C
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		

جدول تجزیه CLR

تجزیه LR

• سوال: کدام مجموعه‌های موارد LR(1) زیر قابل ادغام هستند؟



تجزیه LR

- عملکرد یکسان تجزیه CLR و LALR برای ورودی‌های صحیح
 - قرارگیری حالت‌های معادل در پشته
 - کنش‌های انتقال و کاهش با ترتیب یکسان انجام می‌شوند
 - مثال: عملکرد دو تجزیه‌گر برای جمله ورودی $cdcd\$$

- شناسایی با تأخیر خطا در تجزیه LALR برای ورودی‌های غلط
 - ممکن است بجای اعلام خطا تعدادی کاهش قبل از کشف خطا صورت بگیرد
 - در نهایت خطا پیش از انتقال نشانه ورودی جدید شناسایی می‌شود
 - مثال: عملکرد دو تجزیه‌گر برای جمله ورودی $cd\$$

تجزیه LR

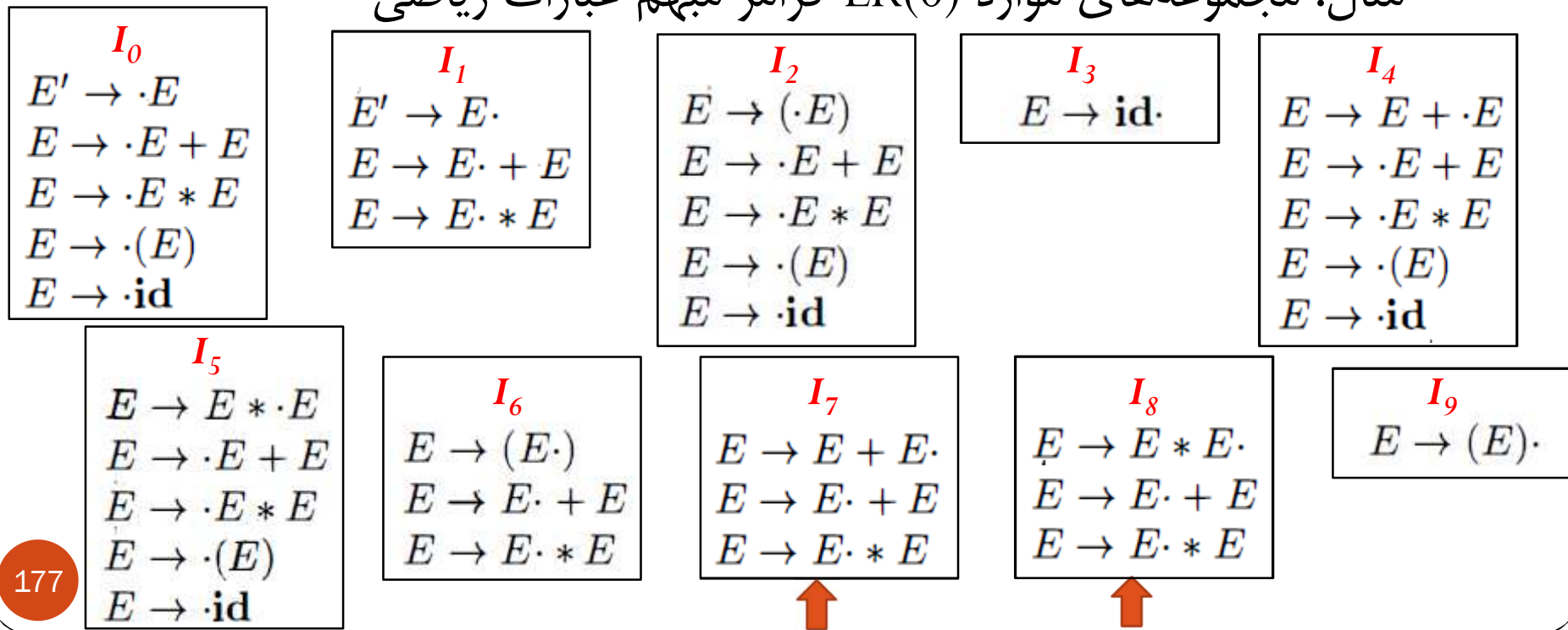
- بکارگیری گرامرهای مبهم
- مشخص کردن برخی ساختارهای متداول زبان‌های برنامه‌نویسی با گرامرهای مبهم موجزتر و قابل فهم‌تر است
- سرعت تجزیه با استفاده از گرامرهای مبهم می‌تواند سریع‌تر از گرامرهای غیرمبهم معادل باشد
- جلوگیری از کاهش با استفاده از قواعد تولید تکی

مثال • $E \rightarrow E + E \mid E * E \mid (E) \mid \text{id}$

$$\begin{array}{lcl} E & \rightarrow & E + T \mid T \\ T & \rightarrow & T * F \mid F \\ F & \rightarrow & (E) \mid \text{id} \end{array}$$

تجزیه LR

- بکارگیری گرامرهای مبهم (ادامه)
- استفاده از قوانین ابهام‌زدا و اعمال مستقیم آنها در جداول تجزیه
- اولویت (precedence) و شرکت‌پذیری (associativity)
- مثال: مجموعه‌های موارد LR(0) گرامر مبهم عبارات ریاضی



تجزیه LR

- مثال: جدول تجزیه SLR برای گرامر مبهم عبارات ریاضی

$$E \rightarrow E + E \mid E * E \mid (E) \mid \text{id}$$

(1)
(2)
(3)
(4)

STATE	ACTION						GOTO
	id	+	*	()	\$	
0	s3			s2			1
1		s4	s5			acc	
2	s3			s2			6
3		r4	r4		r4	r4	
4	s3			s2			7
5	s3			s2			8
6		s4	s5		s9		
7		r1	s5		r1	r1	
8		r2	r2		r2	r2	
9		r3	r3		r3	r3	

I_7

$E \rightarrow E + E \cdot$
 $E \rightarrow E \cdot + E$
 $E \rightarrow E \cdot * E$

I_8

$E \rightarrow E * E \cdot$
 $E \rightarrow E \cdot + E$
 $E \rightarrow E \cdot * E$

Grammar Relationships

Unambiguous Grammars

Ambiguous Grammars

