



编译原理

第三章 语法分析(8)

中国海洋大学 计算机系 王欣捷

wangxinjie@ouc.edu.cn

上次课回顾

- 3.5 LR分析器

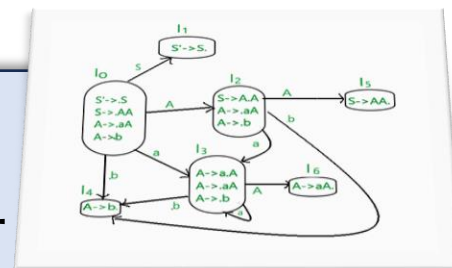
- 3.5.3 构造SLR分析表

- a. 知其所以然：SLR分析表的机制——识别活前缀的DFA
 - b. 一些概念：LR(0)项目、拓广文法、项目集闭包
 - c. 会构造SLR分析表
 - d. 出现冲突的情况

上次课回顾

• SLR分析表构造过程:

- 1. 构造识别活前缀的DFA
 - 1.1 构造**拓广文法**，为产生式编号
 - 1.2 构造**LR(0)项目集**规范族（closure, goto 函数）
 - 1.3 画出**DFA**，所有状态都为接收状态



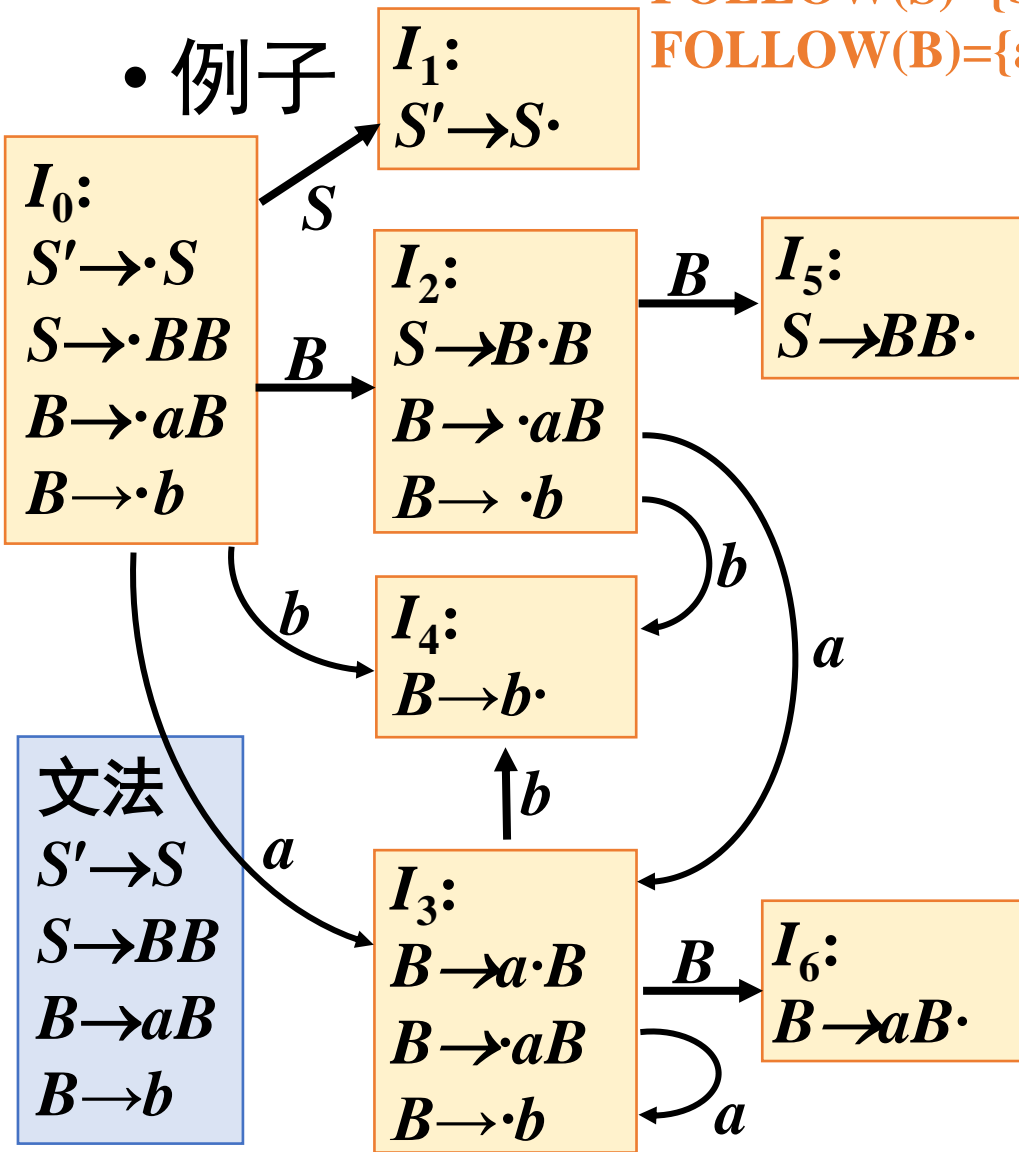
- 2. 构造SLR分析表
 - 2.1 根据项目集族和DFA填**action-goto**表，其中在项目集内遇到归约项目时，按**FOLLOW集合**填归约

ACTION			GOTO	
a	b	\$	A	S
S3	S4		2	1
S3	S4	accept		
S3	S4		5	
R3	R3	R3	6	
		R1		
R2	R2	R2		

上次课回顾

• 例子

$FOLLOW(S) = \{\$ \}$
 $FOLLOW(B) = \{a, b, \$ \}$



文法
 $S' \rightarrow S$
 $S \rightarrow BB$
 $B \rightarrow aB$
 $B \rightarrow b$

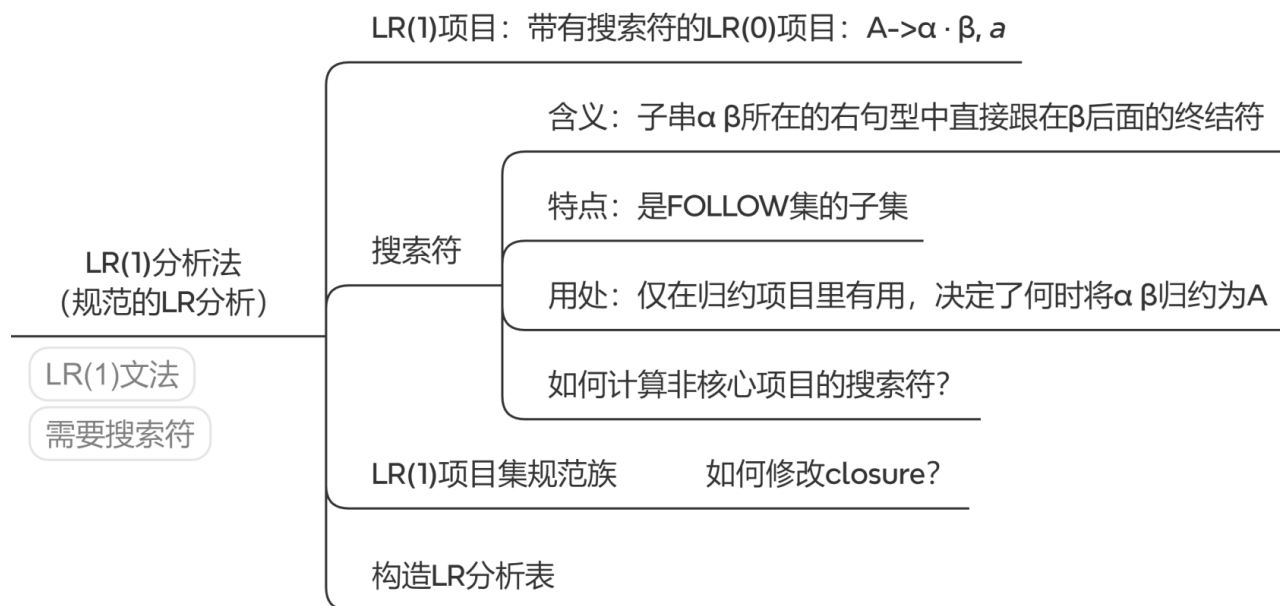
状态	ACTION			GOTO	
	a	b	\$	S	B
0	s3	s4		1	2
1			acc		
2	s3	s4			5
3	s3	s4			6
4	r3	r3	r3		
5			r1		
6	r2	r2	r2		

以串bab为例，
理解DFA识别活前缀的过程

栈	输入	动作
0	b a b \$	移进
0 b 4	a b \$	按 $B \rightarrow b$ 归约
0 B 2	a b \$	移进
0 B 2 a 3	b \$	移进
0 B 2 a 3 b 4	\$	按 $B \rightarrow b$ 归约
0 B 2 a 3 B 6	\$	按 $B \rightarrow aB$ 归约
0 B 2 B 5	\$	按 $S \rightarrow BB$ 归约
0 S 1	\$	接受

本次课内容

- 3.5 LR分析器
 - 3.5.3 构造SLR分析表（续）
 - 3.5.4 构造规范的LR分析表
 - 核心是用搜索符代替FOLLOW集



3.5.3 构造SLR分析表（续）

- 使用SLR分析表的LR分析是SLR分析，能够构造出无冲突的SLR分析表的文法是**SLR文法**。
- SLR分析表中如果出现动作冲突（**移进-归约冲突，归约-归约冲突**），则文法就不是SLR的。
- SLR文法都不是二义的，但是它描述能力有限，有些非二义的文法不能用SLR分析。

3.5.3 构造SLR分析表（续）

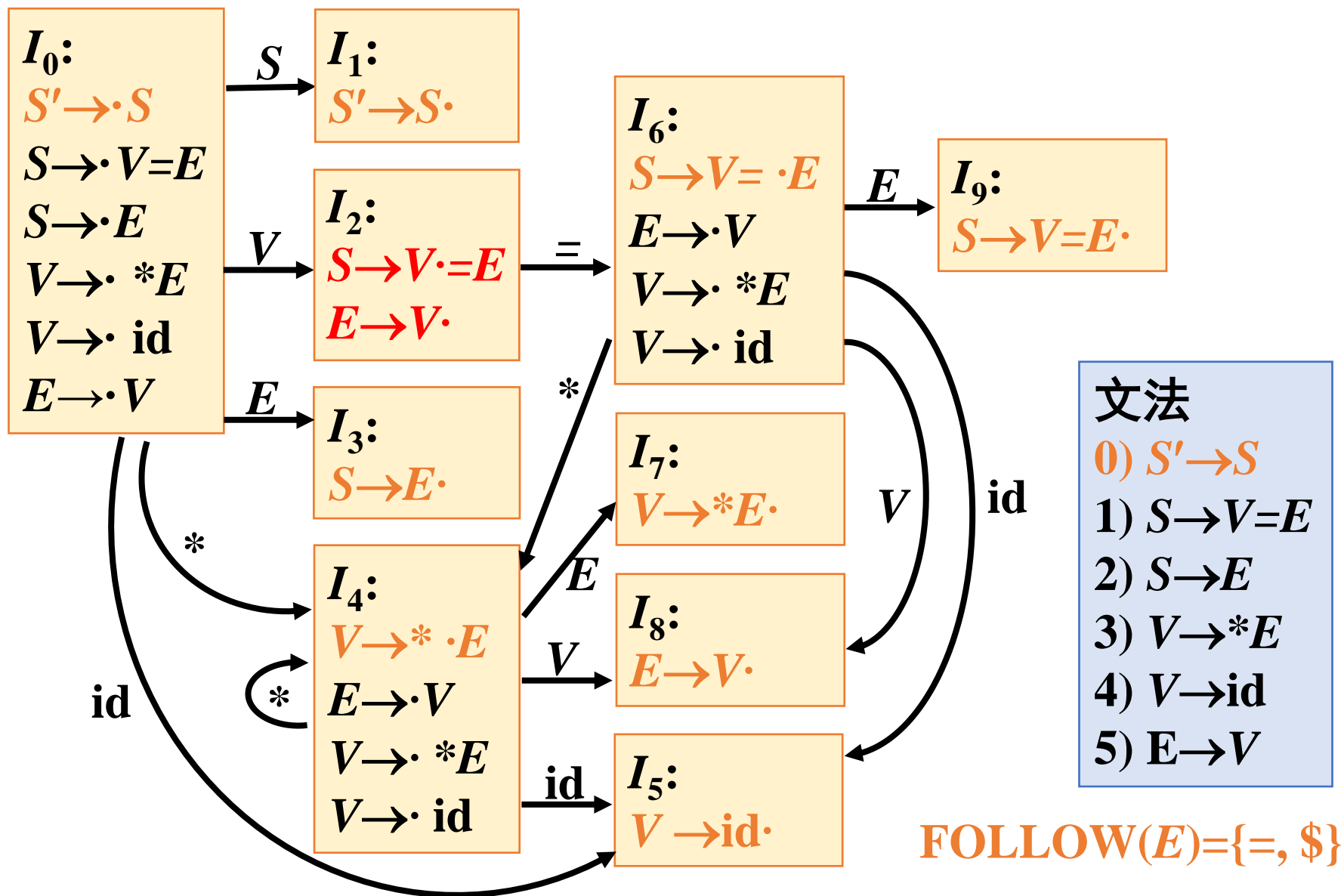
• 例3.29 SLR(1)文法的描述能力有限

$$S \rightarrow V = E$$
$$S \rightarrow E$$
$$V \rightarrow * E$$
$$V \rightarrow \text{id}$$
$$E \rightarrow V$$

小节练习：尝试为例
3.29构造SLR分析表，
大家一起来做吧！



3.5.3 构造SLR分析表 (续)



3.5.3 构造SLR分析表 (续)

I_0 :

- $S' \rightarrow \cdot S$
- $S \rightarrow \cdot V = E$
- $S \rightarrow \cdot E$
- $V \rightarrow \cdot * E$
- $V \rightarrow \cdot \text{id}$
- $E \rightarrow \cdot V$

V

I_2 :

- $S \rightarrow V \cdot = E$
- $E \rightarrow V \cdot$

第一项目使得

$action[2, =] = s6$

第二项目使得

$action[2, =]$ 为按

$E \rightarrow V$ 归约, 因为 $=$ 是 E 的一个后继符

文法

0) $S' \rightarrow S$

1) $S \rightarrow V = E$

2) $S \rightarrow E$

3) $V \rightarrow * E$

4) $V \rightarrow \text{id}$

5) $E \rightarrow V$

$=$ 是 E 的一个后继符: $S \Rightarrow V = E \Rightarrow * E = E$
但是, 实际不存在以 $E = \dots$ 开始的右句型



想一想, 为什么会冲突?

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

3.5.3 构造SLR分析表（续）

- **SLR**文法描述能力有限的原因：
 - 归约时缺乏对上下文的考虑：即归约时只考虑了下一个输入符号是否属于与归约项目 $[A \rightarrow \alpha \cdot]$ 相关联的 **FOLLOW(A)** 集合，而没考虑串 α 所在的右句型的上下文（即在该右句型中 α 能否被归约为 A ）。
 - $b \in \text{FOLLOW}(A)$ 只是归约 α 的一个必要条件，而非充分条件
- 解决方法：规范LR分析（LR(1)分析）

3.5.4 构造规范的LR分析表

- 高德纳 (Donald Ervin Knuth)



3.5.4 构造规范的LR分析表

LR 分
析表

KMP
算法

TEX

科学源自好奇和热爱。

科研道路上，没有所谓的成功学，也没有所谓的秘密武器。只要有了热爱，有了努力，是否成功，其实不重要。因为，做了自己热爱的事情，这个过程就会让人感到心理满足。而且，只有满怀对科研创新的热爱，才能激励自己挑战科学高峰、破解科学之谜，勇敢迈向梦想的远方。



3.5.4 构造规范的LR分析表

- 对于产生式 $A \rightarrow \alpha$ 的归约，在**不同的使用位置**， A 会要求不同的**后继符号**

文法

0) $S' \rightarrow S$

1) $S \rightarrow V=E$

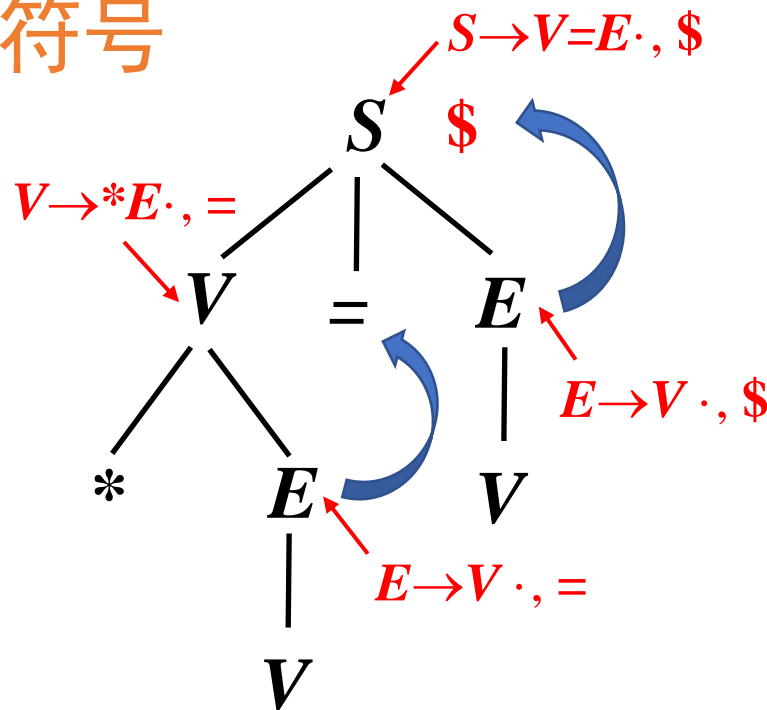
2) $S \rightarrow E$

3) $V \rightarrow *E$

4) $V \rightarrow \text{id}$

5) $E \rightarrow V$

X	$\text{FOLLOW}(X)$
S	$\$$
V	$=, \$$
E	$=, \$$



- 在特定位置， A 的后继符集合是 $\text{FOLLOW}(A)$ 的**子集**

3.5.4 构造规范的LR分析表

- **LR(1)项目:**

- 重新定义项目，让它带上**搜索符**，成为如下形式：

$$[A \rightarrow \alpha \cdot \beta, a]$$

- 搜索符是在子串 $\alpha\beta$ 所在的**右句型**中直接跟在 β **后面的终结符**。
- 搜索符在 β 不为 ϵ 的情况下是没什么用处的，但当 β 为 ϵ 时，它**决定了何时将 $\alpha\beta$ 归约为 A** 。
- **LR(1)**中的1实际上指**搜索符的长度**。
- 这样的 a 的集合总是**FOLLOW(A)的子集**，而且它通常是一个**真子集**。

LR(1)项目 $[A \rightarrow \alpha \cdot \beta, a]$ 中搜索符 a 的含义是:

- ☐ A 子串 $\alpha\beta$ 所在的右句型中直接跟在 α 后面的终结符
- ☒ B 子串 $\alpha\beta$ 所在的右句型中直接跟在 β 后面的终结符
- ☐ C 非终结符 A 的后继符号集合
- ☐ D 非终结符 A 的开始符号集合

提交

在LR(1)项目 $[A \rightarrow \alpha \cdot \beta, a]$ 中，关于搜索符说法错误的是？

- ☐ A 若 β 不为空，则搜索符无用
- ☒ B 若 β 为空，则搜索符无用
- ☐ C 若 α 不为空，则搜索符无用
- ☐ D 若 α 为空，则搜索符无用

提交

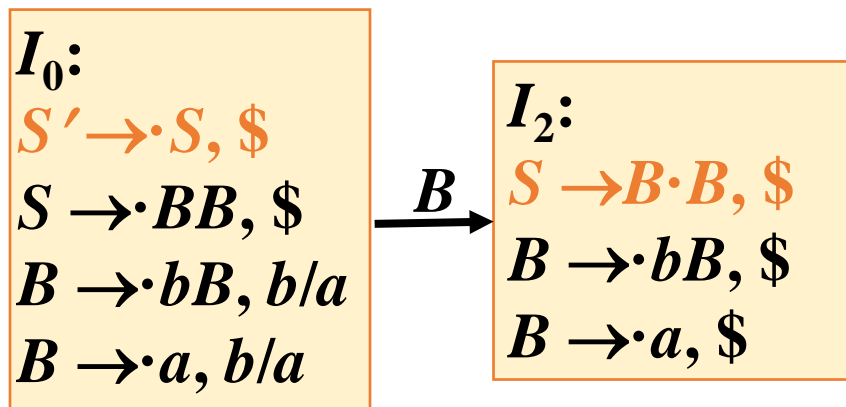
3.5.4 构造规范的LR分析表

- 初始状态的搜索符？

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

- 若已知状态I中项目的搜索符，如何计算 $\text{goto}(I, X)$ 中状态X的搜索符？

X新得到的核心项目与I中原项目有相同搜索符。



3.5.4 构造规范的LR分析表

- 如何计算非核心项目的搜索符?

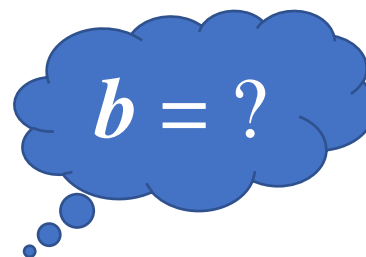
$[A \rightarrow \alpha \cdot B\beta, \mathbf{a}]$

有 $[B \rightarrow \gamma] \in P$



$[B \rightarrow \cdot \gamma, b]$

$b \in \text{FIRST}(\beta a)$



$I_0:$

$S' \rightarrow \cdot S, \$$

$S \rightarrow \cdot BB, \$$

$B \rightarrow \cdot bB, b/a$

$B \rightarrow \cdot a, b/a$

注意, b 不可能为 ϵ , 因为 a 不为 ϵ

对于核心LR(1)项目[$A \rightarrow \alpha \cdot B, a$], 计算其非核心项目[$B \rightarrow \cdot \gamma, b$]时, 搜索符 b 的值是?

- ☐ A α
- ☐ B γ
- ☐ C B
- ☒ D a

提交

对于核心LR(1)项目[$A \rightarrow \cdot B + A, a$], 计算其非核心项目[$B \rightarrow \cdot C * B, b$]时, 搜索符 b 的值是?

- ☐ A a
- ☐ B B
- ☒ C $+$
- ☐ D $*$

提交

3.5.4 构造规范的LR分析表

- 构造规范LR分析表和构造SLR分析表过程基本相同，不同点为：
 - 构造LR(1)项目集规范族：修改 $\text{closure}(I)$ 和 $\text{goto}(I, X)$ 函数，使其带上搜索符。
 - 填写action-goto表时，按照搜索符归约，而不是按FOLLOW集合归约。

3.5.4 构造规范的LR分析表

- 构造规范LR分析表步骤：
 - 构造识别活前缀的DFA
 - 拓广文法并为产生式编号（文法中加入 $S' \rightarrow S$ ）
 - 构造LR(1)项目集规范族。
 - 根据LR(1)项目集规范族和识别活前缀的DFA构造action-goto表（LR分析表）。

3.5.4 构造规范的LR分析表

例3.32：为下面文法构造规范**LR**分析表

$$S \rightarrow BB$$

$$B \rightarrow bB \mid a$$

3.5.4 构造规范的LR分析表

- 第一步：构造识别活前缀的**DFA**

- 拓广文法并为产生式编号

$$(0) S' \rightarrow S$$

$$(1) S \rightarrow BB$$

$$(2) B \rightarrow bB$$

$$(3) B \rightarrow a$$

3.5.4 构造规范的LR分析表

- 第一步：构造识别活前缀的**DFA**
 - 拓广文法并为产生式编号
 - 构造**LR(1)**项目集规范族
 - 初始项目集： $\{[S' \rightarrow \cdot S, \$]\}$
 - 计算 $\text{closure}(\{[S' \rightarrow \cdot S, \$]\})$ 得到第一个项目集 I_0
 - 计算 $\text{goto}(I_0, X)$ 得到其他项目集（状态）

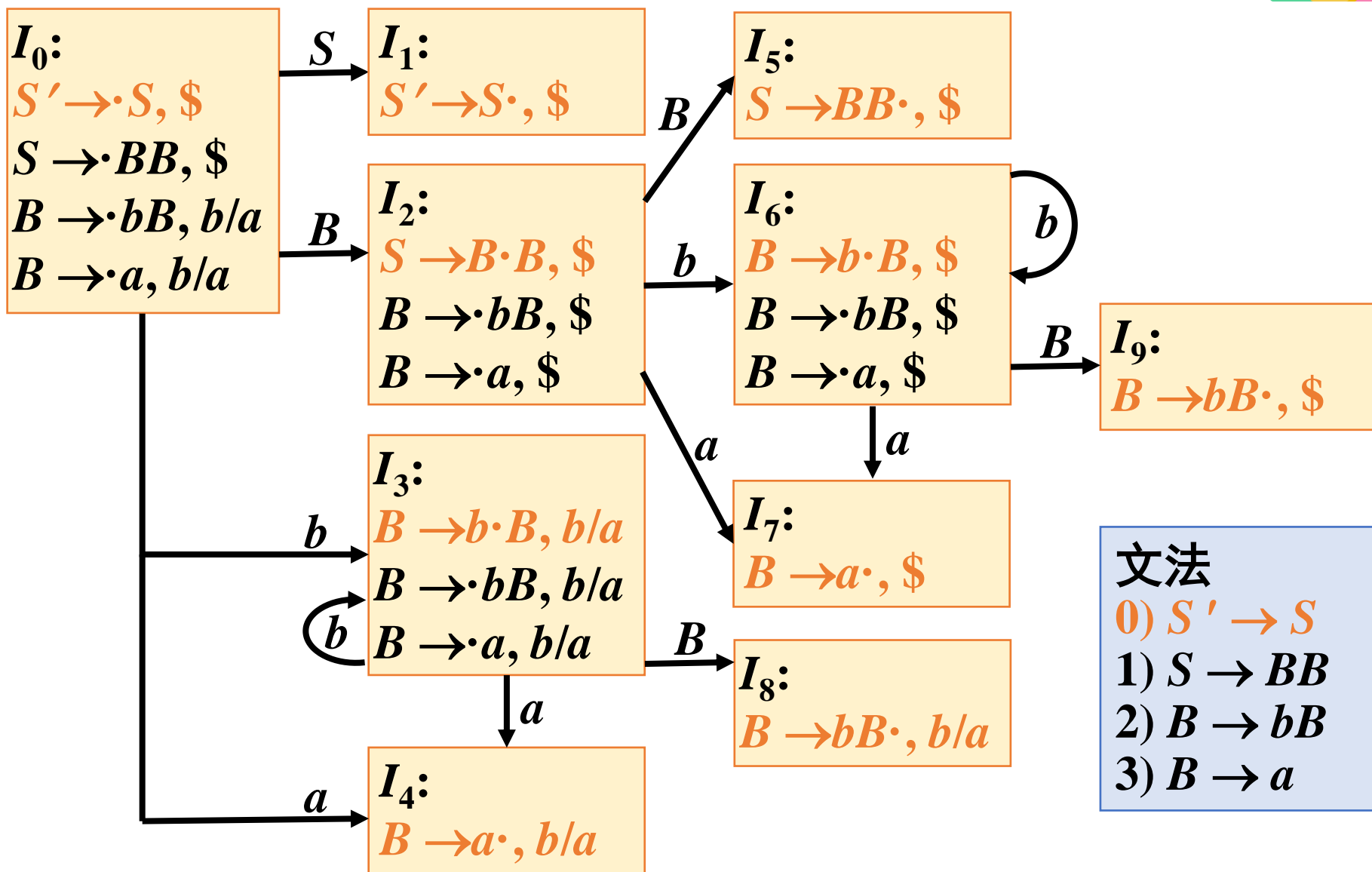
3.5.4 构造规范的LR分析表

- 第一步：构造识别活前缀的DFA
 - 拓广文法并为产生式编号
 - 构造LR(1)项目集规范族
 - 初始项目集： $\{[S' \rightarrow \cdot S, \$]\}$
 - 计算closure($\{[S' \rightarrow \cdot S, \$]\}$)得到第一个项目集 I_0
 - 对closure(I)的修改：若项目 $[A \rightarrow \alpha \cdot B\beta, a]$ 在 I 中，对每一条形式为 $B \rightarrow \gamma$ 的产生式，令 $b = \text{FIRST}(\beta a)$ ，把项目 $[B \rightarrow \cdot \gamma, b]$ 加入闭包。（注意， b 不可能为 ϵ ，因为 a 不为 ϵ ）

3.5.4 构造规范的LR分析表

- 第一步：构造识别活前缀的DFA
 - 拓广文法并为产生式编号
 - 构造LR(1)项目集规范族
 - 初始项目集： $\{[S' \rightarrow \cdot S, \$]\}$
 - 计算closure($\{[S' \rightarrow \cdot S, \$]\}$)得到第一个项目集 I_0
 - 计算goto(I_0, X)得到其他项目集（状态）
 - 注意计算goto(I, X)时，新得到的核心项目与原项目有相同搜索符。
 - 用goto(I_i, X)计算从任意状态 I_i 开始，向栈中压入一个文法符号 X 所得到的状态（项目集）。直到没有新的状态（项目集）产生为止。

3.5.4 构造规范的LR分析表



3.5.4 构造规范的LR分析表

- 第一步：构造识别活前缀的**DFA**
- 第二步：构造**action-goto**表（**LR分析表**）

3.5.4 构造规范的LR分析表

- 第一步：构造识别活前缀的DFA
- 第二步：构造action-goto表（LR分析表）
 - 每个项目集对应了action-goto表中一个状态。
 - 项目集中的项目分四类：
 - 接受项目 $[S' \rightarrow S \cdot, \$]$: $\text{action}[i, \$] = \text{acc}$
 - 移进项目 $[A \rightarrow \alpha \cdot a\beta, b]$: $\text{action}[i, a] = \text{sj}$
 - 归约项目 $[A \rightarrow \alpha \cdot, a]$: $\text{action}[i, a] = \text{rj}$,
 - 待归约项目 $[A \rightarrow \alpha \cdot B\beta, b]$: $\text{goto}[i, B] = j$

3.5.4 构造规范的LR分析表



状态	动 作		转 移	
	a	b	\$	<i>S</i> <i>B</i>
0	s4	s3		1 2
1	acc			
2	s7	s6		5
3	s4	s3		8
4	r3	r3		

3.5.4 构造规范的LR分析表

状态	动 作			转 移	
	a	b	\$	<i>S</i>	<i>B</i>
5	r1				
6	s7	s6			9
7	r3				
8	r2	r2			
9	r2				

3.5.4 构造规范的LR分析表

- 算法3.5: LR(1)项目集的 *closure*() 函数

```
SetOfItems CLOSURE ( I ) {  
    repeat  
        for ( I 中的每个项  $[A \rightarrow \alpha \cdot B\beta, \mathbf{a}]$  )  
            for ( G 的每个产生式  $B \rightarrow \gamma$  )  
                for ( FIRST( $\beta\mathbf{a}$ ) 中的每个符号  $\mathbf{b}$  )  
                    将  $[B \rightarrow \cdot \gamma, \mathbf{b}]$  加入 I 中;  
    until 在某一轮中没有新的项被加入到 I 中;  
    return I;  
}
```

3.5.4 构造规范的LR分析表

- 算法3.5: LR(1)项目集的 $goto()$ 函数

```
SetOfItems GOTO (  $I$ ,  $X$  ) {  
    将  $J$  初始化为空集;  
    for (  $I$  中的每个项  $[A \rightarrow \alpha \cdot X \beta, a]$  )  
        将项  $[A \rightarrow \alpha X \cdot \beta, a]$  加入到集合  $J$  中;  
    return CLOSURE (  $J$  );  
}
```

3.5.4 构造规范的LR分析表

• 算法3.5：构造LR(1)项目集规范族的函数

```
void items(  $G'$  ) {  
     $C = \{ \text{CLOSURE} (\{ [ S' \rightarrow \cdot S, \$ ] \} ) \};$   
    repeat  
        for (  $C$  中的每个项集  $I$  )  
            for( 每个文法符号  $X$  )  
                if (  $\text{GOTO} ( I, X )$  非空且不在  $C$  中 )  
                    将  $\text{GOTO} ( I, X )$  加入  $C$  中;  
    until 在某一轮中没有新的项集被加入到  $C$  中;  
}
```

3.5.4 构造规范的LR分析表

- 注意：构造LR(1)项目集规范族时，不要遗漏搜索符。例如，给如下文法构造LR(1)项目集 I_0 时：

$$E' \rightarrow E$$

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

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

$$F \rightarrow (E) \mid \text{id}$$



$I_0:$

$$E' \rightarrow \cdot E, \$$$

$$E \rightarrow \cdot E + T, \$/+$$

$$E \rightarrow \cdot T, \$/+$$

$$T \rightarrow \cdot T * F, \$/+/*$$

$$T \rightarrow \cdot F, \$/+/*$$

$$F \rightarrow \cdot (E), \$/+/*$$

$$F \rightarrow \cdot \text{id}, \$/+/*$$

3.5.4 构造规范的LR分析表

- 算法3.6 构造规范的LR分析表算法

- 方法:

注意与SLR分析表的区别

- (1) 构造 G' 的LR(1)项目集规范族 $C = \{ I_0, I_1, \dots, I_n \}$

- (2) 令 I_i 对应状态 i 。状态 i 的action按照下面的方法决定:

(a) **if** $[A \rightarrow \alpha \cdot a \beta, b] \in I_i$ and $GOTO(I_i, a) = I_j$ **then** $ACTION[i, a] = sj$

(b) **if** $[A \rightarrow \alpha \cdot, a] \in I_i$ 且 $A \neq S'$ **then** $ACTION[i, a] = rj$ (j 是产生式 $A \rightarrow \alpha$ 的编号)

(c) **if** $[S' \rightarrow S \cdot, \$] \in I_i$ **then** $ACTION[i, \$] = acc$

- (3) 状态 i 的goto按照下面的方法决定:

(a) **if** $[A \rightarrow \alpha \cdot B \beta, b] \in I_i$ and $GOTO(I_i, B) = I_j$ **then** $GOTO[i, B] = j$

- (4) 没有定义的所有条目都设置为error

- (5) 初始状态是包含 $[S' \rightarrow \cdot S]$ 的项目集对应的状态

3.5.4 构造规范的LR分析表

- 使用规范的**LR(1)**分析表的**LR**分析是规范的**LR(1)**分析，能够构造出无冲突（即表项无多重定义）的**LR(1)**分析表的文法是**LR(1)文法**。
- **LR(1)**分析表中如果出现表项有多重定义的情况（即出现了动作冲突）则文法就不是**LR(1)**的。
- 所有**SLR**文法都是**LR(1)**文法；有些文法不是**SLR**文法，但是是**LR(1)**文法。

3.5.4 构造规范的LR分析表

- 例3.29 SLR无法分析但规范LR可分析的例子：

$S \rightarrow V = E$

$S \rightarrow E$

$V \rightarrow * E$

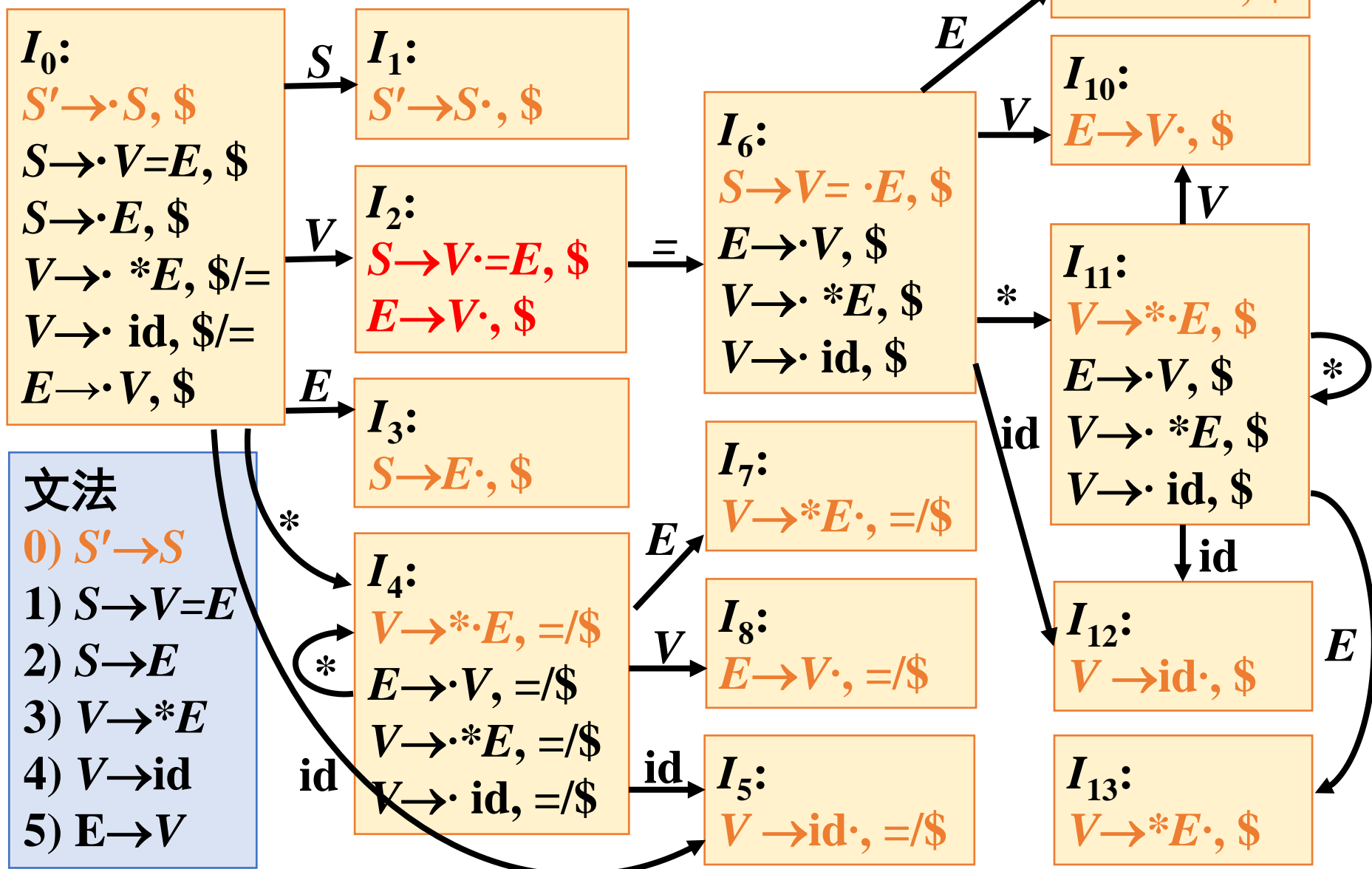
$V \rightarrow \text{id}$

$E \rightarrow V$

我们在本节课开始时
已经构造过此SLR分
析表了，里面有冲突。



3.5.3 构造规范的LR分析表



3.5.4 构造规范的LR分析表

文法

0) $S' \rightarrow S$

1) $S \rightarrow V=E$

2) $S \rightarrow E$

3) $V \rightarrow *E$

4) $V \rightarrow \text{id}$

5) $E \rightarrow V$

状态	ACTION				GOTO		
	*	id	=	\$	S	V	E
0	s4	s5			1	2	3
1				acc			
2			s6	r5			
3				r2			
4	s4	s5				8	7
5			r4	r4			
6	s11	s12				10	9
7			r3	r3			
8			r5	r5			
9				r1			
10				r5			
11	s11	s12				10	13
12				r4			
13				r3			

$I_2:$

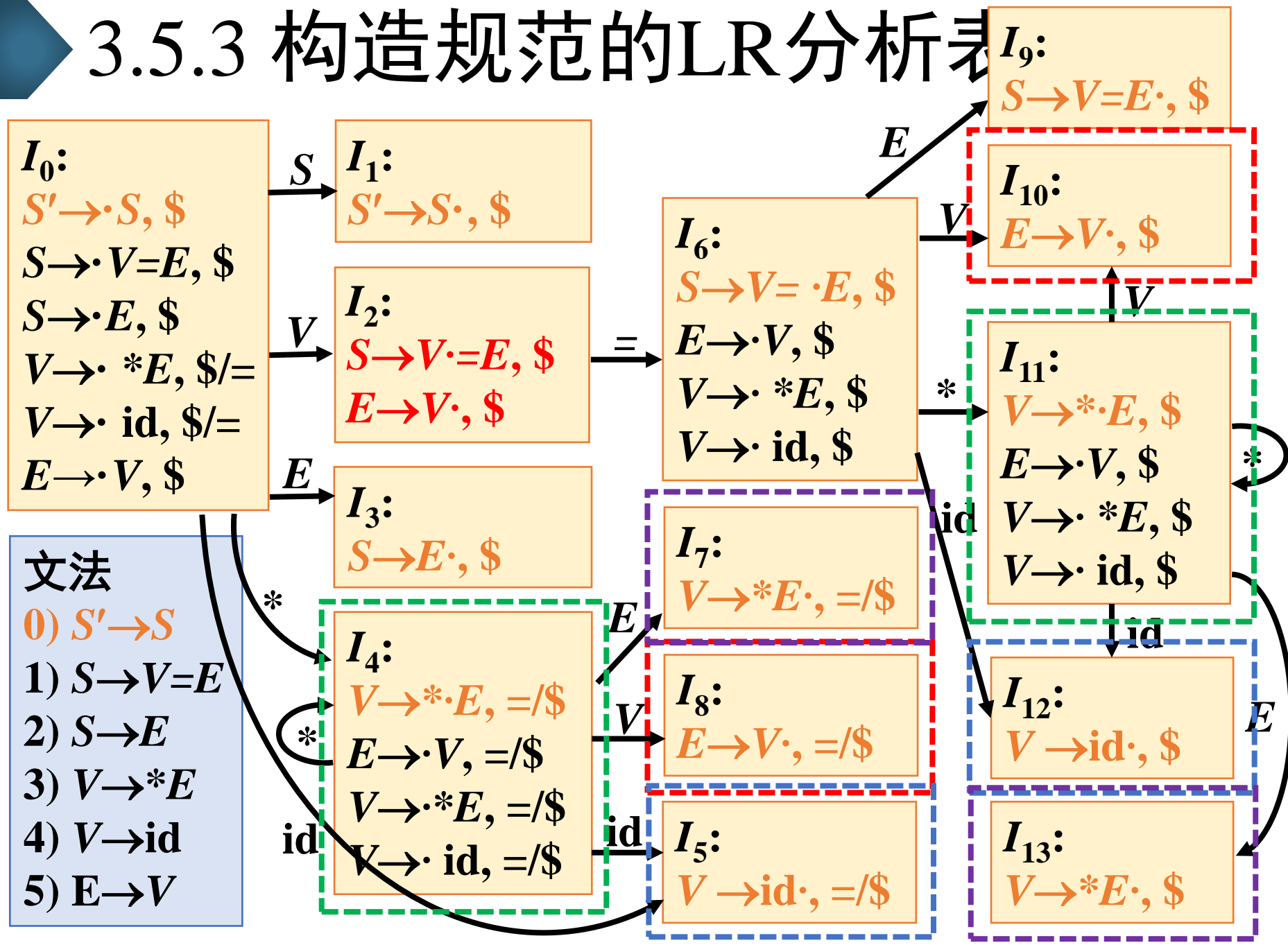
$S \rightarrow V \cdot = E, \$$
 $E \rightarrow V \cdot, \$$

\Rightarrow

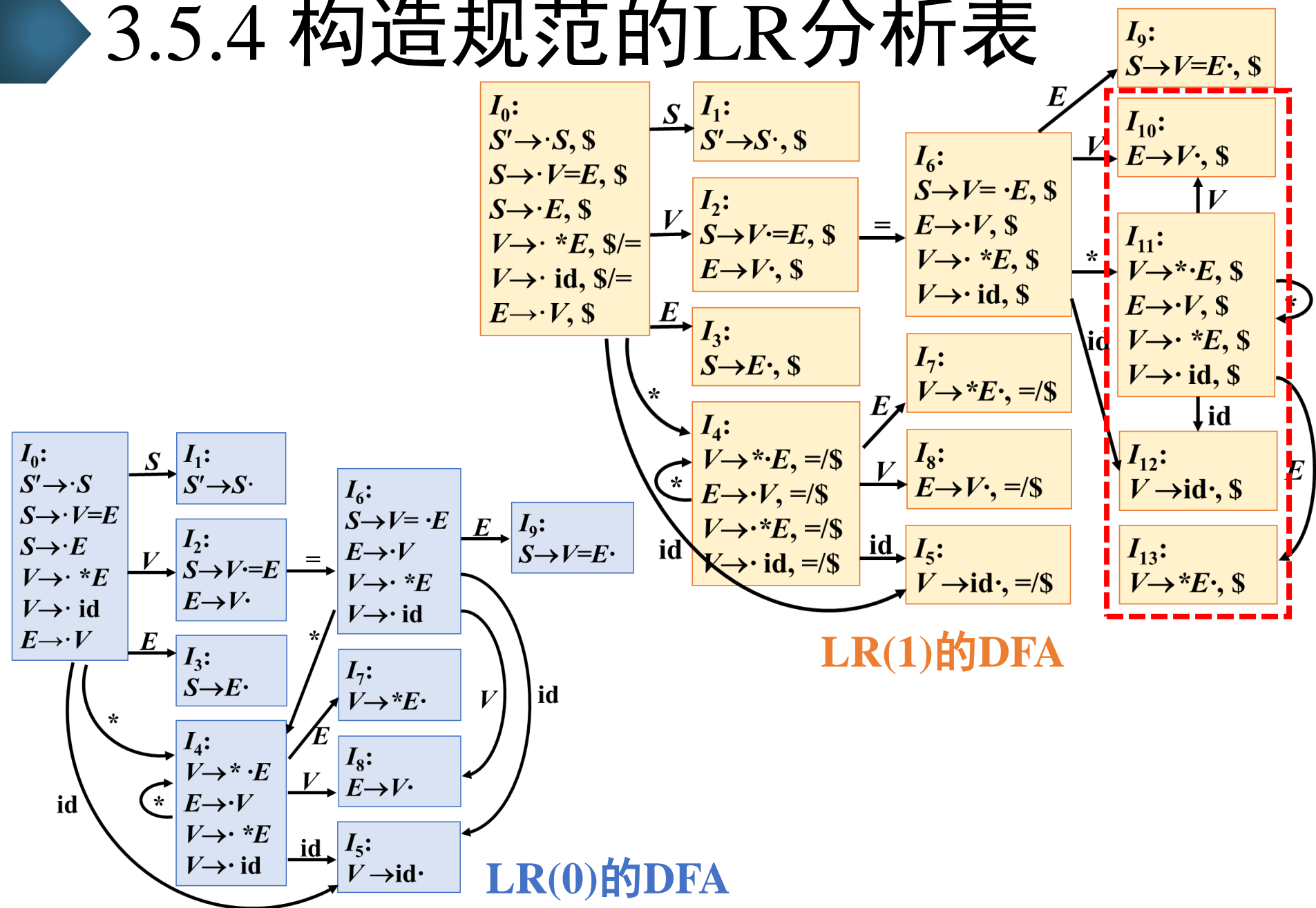
$I_6:$

$S \rightarrow V = \cdot E, \$$
 $E \rightarrow \cdot V, \$$
 $V \rightarrow \cdot * E, \$$
 $V \rightarrow \cdot \text{id}, \$$

3.5.3 构造规范的LR分析表



3.5.4 构造规范的LR分析表



3.5.4 构造规范的LR分析表

- 规范LR分析法的问题：
 - 状态数庞大，构造难度很大。
- 缓解的办法：
 - LALR分析法

3.5.5 构造LALR分析表

- **LALR (LookAhead LR) 分析表特点:**
 - **LALR和SLR的分析表有同样多的状态, 而规范LR分析表要大得多**
 - **LALR的分析能力介于SLR和规范LR之间**
 - **实际的编译器经常使用LALR分析法**

3.5.5 构造LALR分析表

- 同心的LR(1)项目集：
略去搜索符后它们是相同的集合
- LALR分析在规范的LR分析的基础上合并同心项目集
- 合并同心集后的项目集族叫做LALR(1)项目集族。
- 注意：一般而言
 - 一个心对应文法的一个LR(0)项目集
 - LR(1)文法可能使多个项目集同心

本节小结

• 3.5 LR分析器

• 3.5.3 构造SLR分析表（续）

• 3.5.4 构造规范的LR分析表

LR(1)项目：带有搜索符的LR(0)项目： $A \rightarrow \alpha \cdot \beta, a$

LR(1)分析法
(规范的LR分析)

LR(1)文法

需要搜索符

搜索符

含义：子串 $\alpha \beta$ 所在的右句型中直接跟在 β 后面的终结符

特点：是FOLLOW集的子集

用处：仅在归约项目里有用，决定了何时将 $\alpha \beta$ 归约为A

如何计算非核心项目的搜索符？

LR(1)项目集规范族

如何修改closure？

构造LR分析表

作业4——提前布置

- 1、P105/3.21(a)
- 2、构造下面文法的SLR分析表，看看哪里有冲突，以说明该文法不是SLR文法。并构造该文法的规范LR分析表。

$$S \rightarrow V = E$$
$$S \rightarrow E$$
$$V \rightarrow * E$$
$$V \rightarrow \text{id}$$
$$E \rightarrow V$$

- 3、P105/3.24
- 4、P106/3.31
- 5、P107/3.35

作业4的截止时间为**5月9日**，即第3章-习题课之前哦

