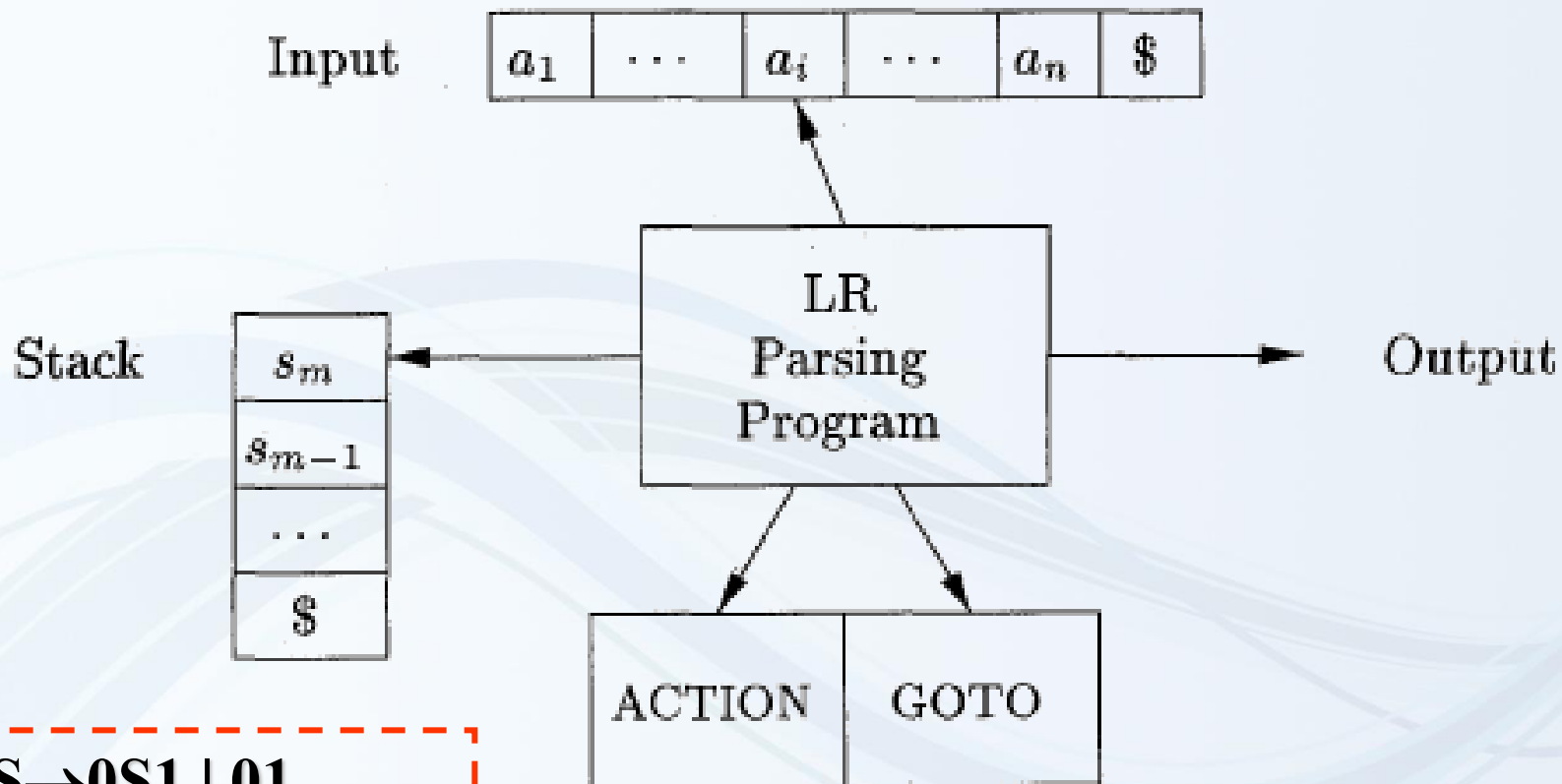


4.6 LR语法分析



$S \rightarrow 0S1 \mid 01$

输入: 0011

移入-归约语法分析(*Shift-reduce parsing*)

- ❖ 自底向上语法分析的一种方法。
- ❖ 使用一个**栈**来保存状态（和文法符号），使用一个**输入缓冲区**来存放将要进行语法分析的其余符号。
- ❖ 句柄在被识别（归约）时，完整地出现在栈顶。

	符号栈	输入
分析开始时	\$	w\$
分析终止时	\$\$	\$

S是文法的开始符，w为待识别的输入串

移入-归约语法分析器的四种动作

1. 移入(*shift*): 将下一个输入符号入栈。
2. 归约(*reduce*): 栈顶已经出现句柄，决定用某个非终结符来替换它。
3. 接受(*accept*): 语法分析过程成功完成。
4. 报错(*error*): 发现并处理一个语法错误。

LR(k)语法分析技术

- ❖ **L**表示从左到右地扫描输入串；**R**表示反向构造一个**最右推导**序列；**k**表示在做决定时，向前看**k**个输入符号。
- ❖ 对于文法**G**，如果存在从左到右扫描的移入-归约分析器，能够在**G**的右句型的句柄出现在栈顶时识别出来，则称**G**是LR的。

LR(0)项目

❖ 简称**项**，由文法G的产生式加一个圆点构成。

➤ $S \rightarrow XY$ 有3个项目： $S \rightarrow \cdot XY$ ， $S \rightarrow X \cdot Y$ 和 $S \rightarrow XY \cdot$

✓ $S \rightarrow \cdot XY$ 表示希望接下来看到一个能从XY推导得到的串

✓ $S \rightarrow X \cdot Y$ 表示已经看到一个可以由X推导得到的串，希望接下来看到一个能从Y推导得到的串

✓ $S \rightarrow XY \cdot$ 表示栈顶看到了XY，可以把它归约为S

➤ 圆点位于产生式最右端的项目为**归约项目**，如果该产生式左端的文法符号为开始符，则为**接受项目**

➤ 圆点后第一个符号为**终结符**时的项目为**移进项目**

➤ 圆点后第一个符号为**非终结符**时的项目为**待约项目**

增广文法（拓广文法）

❖ 文法 $G[S]$ 的增广文法 G' ：

构造方法：引入**新开始符号** S' ，并增加一个**新产生式** $S' \rightarrow S$

➤ **当且仅当**使用产生式 $S' \rightarrow S$ 进行归约时，输入符号串被接受。

LR(0)项目 (例)

对于增广文法:

$S' \rightarrow S$ $S \rightarrow 0S1 \mid 01$

接受项目: $S' \rightarrow S \bullet$

归约项目: $S \rightarrow 0S1 \bullet$ $S \rightarrow 01 \bullet$

移进项目: $S \rightarrow \bullet 0S1$ $S \rightarrow 0S \bullet 1$ $S \rightarrow \bullet 01$ $S \rightarrow 0 \bullet 1$

待约项目: $S' \rightarrow \bullet S$ $S \rightarrow 0 \bullet S1$

LR(0)项目 (例2)

对于增广文法:

$S' \rightarrow S$ $S \rightarrow AS \mid \varepsilon$ $A \rightarrow aA \mid b$

接受项目: $S' \rightarrow S \bullet$

归约项目: $S \rightarrow AS \bullet$ $S \rightarrow \bullet$ $A \rightarrow aA \bullet$ $A \rightarrow b \bullet$

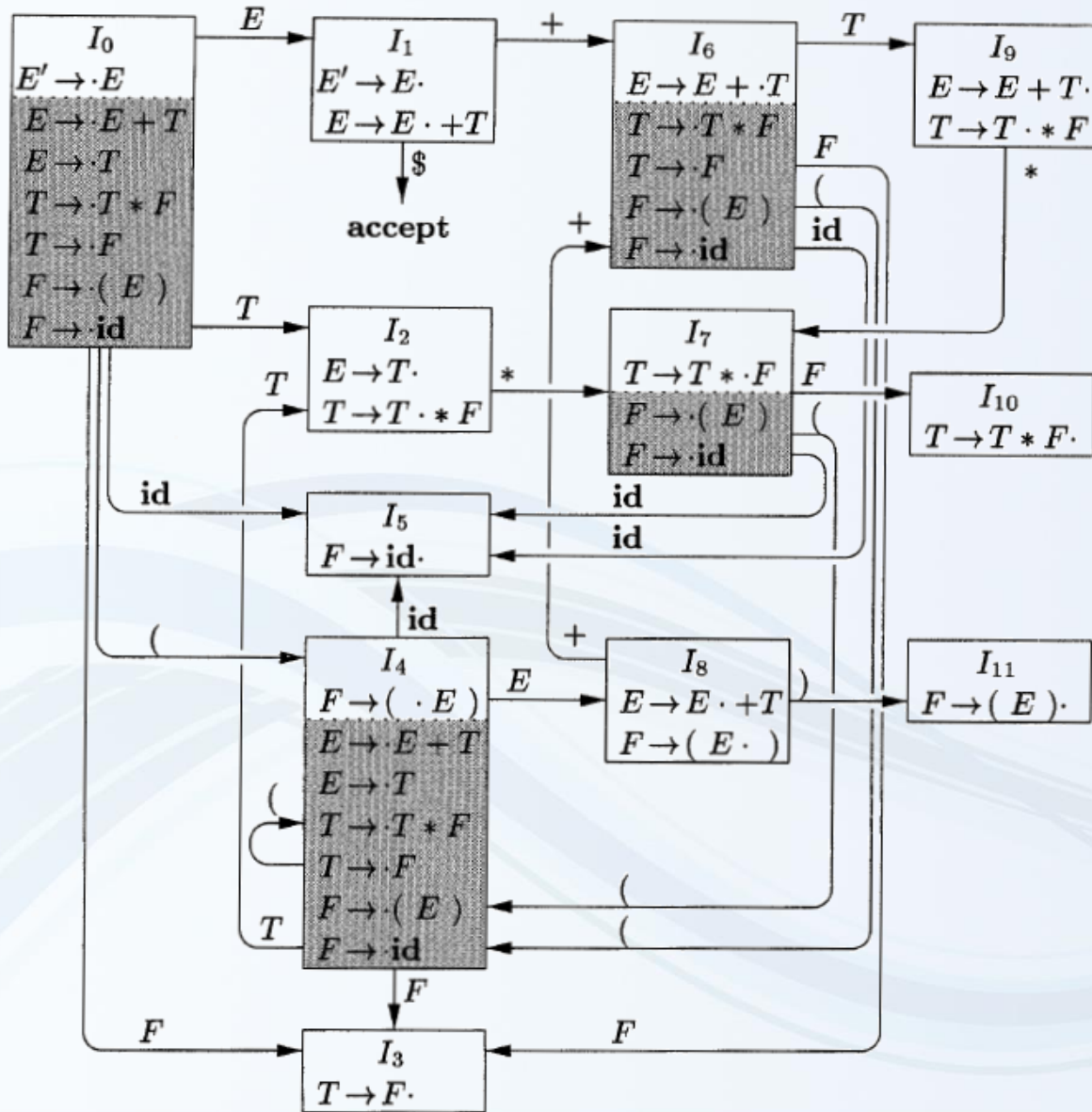
移进项目: $A \rightarrow \bullet aA$ $A \rightarrow \bullet b$

待约项目: $S' \rightarrow \bullet S$ $S \rightarrow \bullet AS$ $S \rightarrow A \bullet S$ $A \rightarrow a \bullet A$

LR(0)自动机的使用

采用P155图4-31中的自动机分析id*id

stack	symbols	input	action
0	\$	id*id\$	Shift to 5
0 5	\$ id	*id\$	Reduce by $F \rightarrow id$
0 3	\$ F	*id\$	Reduce by $T \rightarrow F$
0 2	\$ T	*id\$	Shift to 7
0 2 7	\$ T *	id\$	Shift to 5
0 2 7 5	\$ T * id	\$	Reduce by $F \rightarrow id$
0 2 7 <u>10</u>	\$ T * F	\$	Reduce by $T \rightarrow T * F$
0 2	\$ T	\$	Reduce by $E \rightarrow T$
0 1	\$ E	\$	accept



项集的闭包

❖项集：项目的集合。

❖项集的闭包CLOSURE

设I是一个项集, 使用以下规则来构建CLOSURE(I) :

1. 首先将I中所有元素加入CLOSURE(I).
2. 若项目 $A \rightarrow \alpha \cdot B \beta \in \text{CLOSURE}(I)$, 且 $B \rightarrow \gamma$ 是一个产生式, 则加项目 $B \rightarrow \cdot \gamma$ 入CLOSURE(I)

反复使用规则2, 直至无新项目可加入

项集的闭包（例）

现有文法：

$$E' \rightarrow E$$

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

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

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

求项集 $\{E' \rightarrow \bullet E\}$ 的闭包

GOTO函数

❖ **GOTO(I, X)**, 其中I是一个项集, X是一个文法符号

➤ 定义了I中**所有**形如 $[A \rightarrow \alpha \cdot X \beta]$ 的项目所**对应的**项目 $[A \rightarrow \alpha X \cdot \beta]$ 的**集合的闭包**。

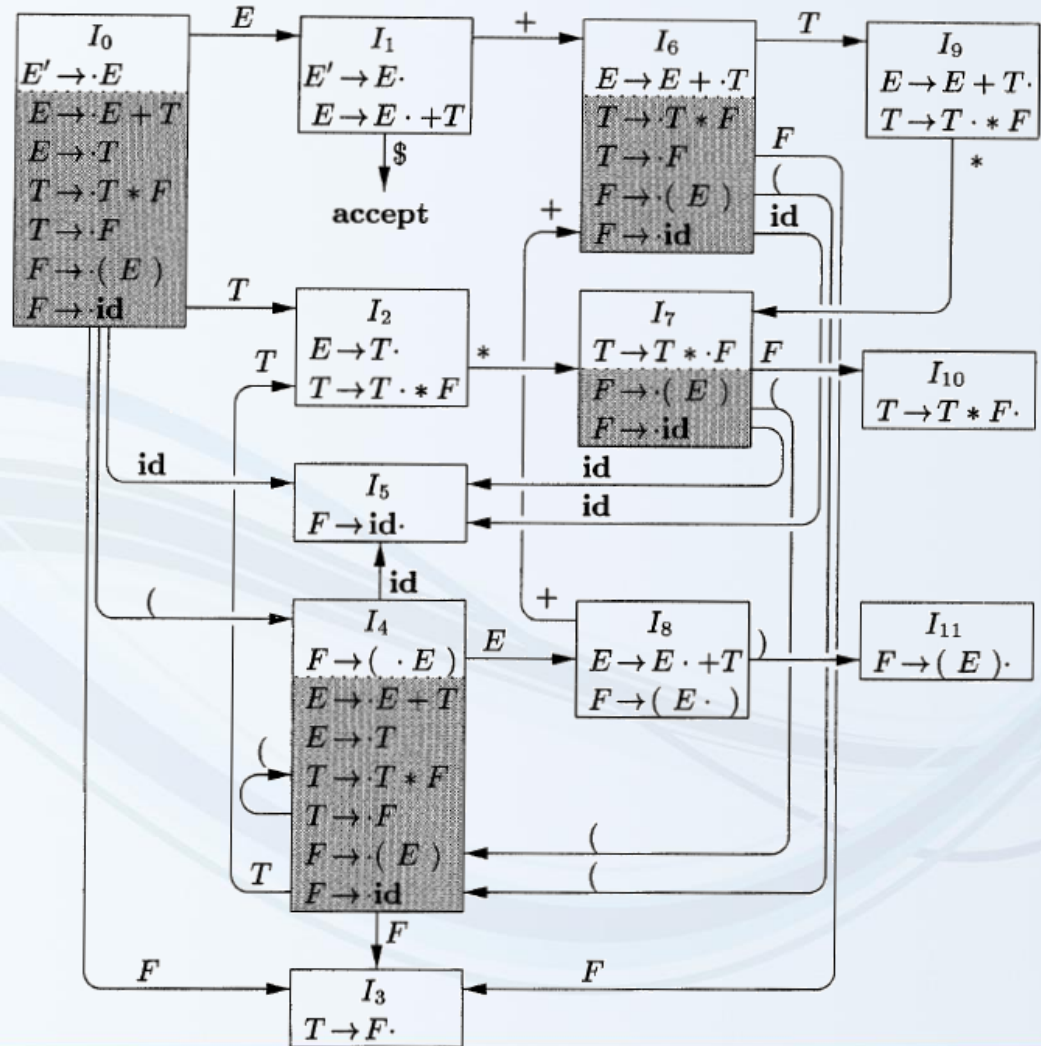
GOTO函数 (例)

GOTO(I₀,E)

GOTO(I₀,())

GOTO(I₁,+)

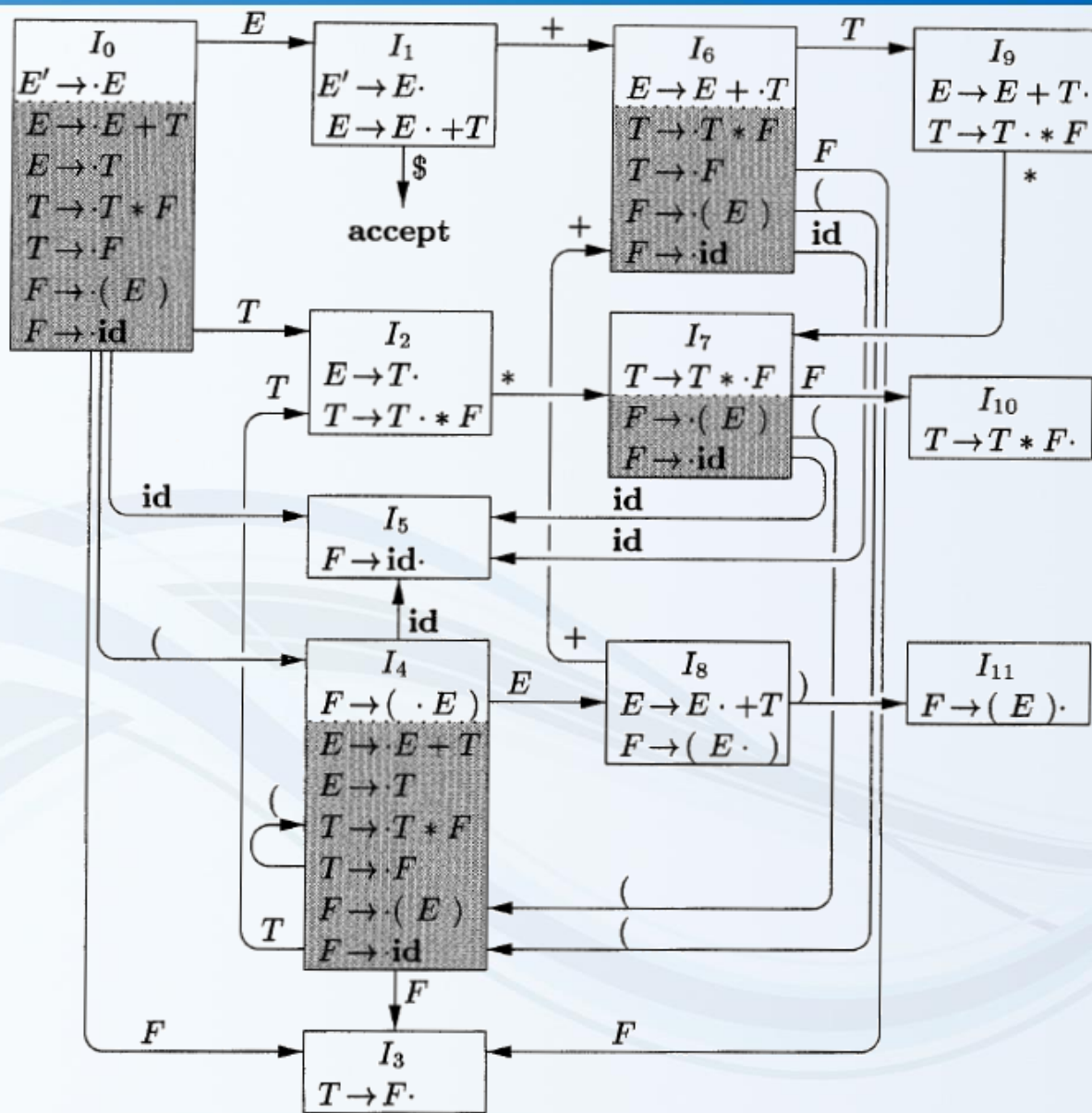
GOTO(I₆,T)



P155图4-31

规范LR(0)项集族的计算

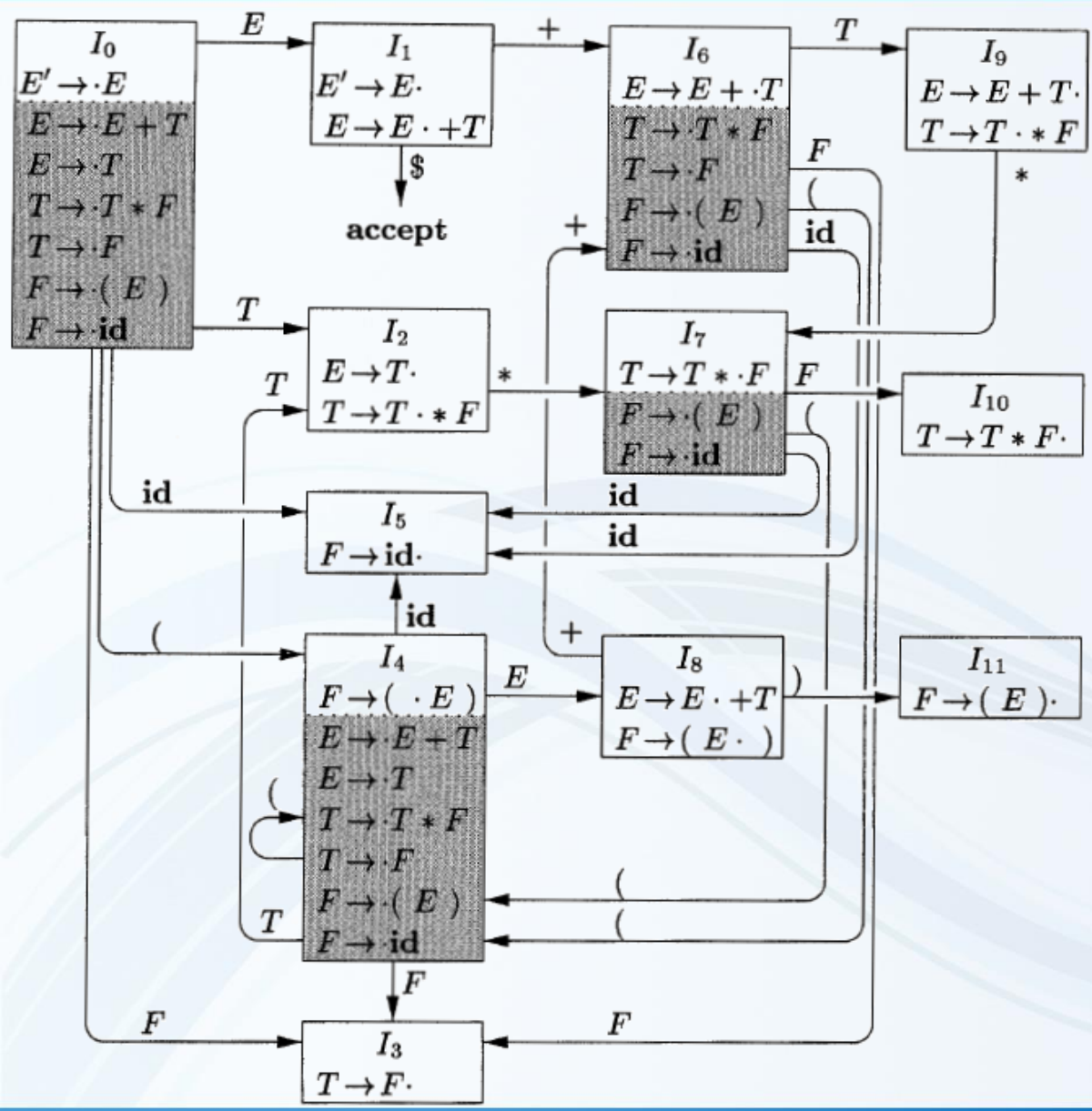
```
void items(G') {  
    C = CLOSURE({[S'→•S]});  
    repeat  
        for (each set of items I in C)  
            for (each grammar symbol X)  
                if (GOTO(I,X) is not empty and not in C)  
                    add GOTO(I,X) to C;  
    until no new sets of items are added to C on a round;  
}
```



P155图4-31

LR语法分析表

- ❖ 语法分析动作函数ACTION有两个参数：状态 i ，当前输入 a (**终结符或 $\$$**)
 - 移入(s_j): 移入状态 j ，同时移入符号 a
 - 归约(r_j): 假设**第 j 条规则**为 $A \rightarrow \beta$ ，则将栈顶的 β 归约为 A 。
 - 接受：接受输入串，完成语法分析。
 - 报错。发现一个错误并执行纠正动作。
- ❖ GOTO函数：有两个参数：状态 i 和当前的**非终结符 A** 。
若 $GOTO(I_i, A) = I_j$ ，则在当前状态为 i ，并且归约得到非终结符 A 时，移入与 A 对应的状态 j 。



FOLLOW(E)={+,),\\$}
FOLLOW(T)={+,*),\\$}
FOLLOW(F)={+,*),\\$}

- (0) $E' \rightarrow E$
- (1) $E \rightarrow E + T$
- (2) $E \rightarrow T$
- (3) $T \rightarrow T * F$
- (4) $T \rightarrow F$
- (5) $F \rightarrow (E)$
- (6) $F \rightarrow id$

SLR(1)语法分析表

state	ACTION						GOTO		
	id	+	*	()	\$	E	T	F
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6		s11					
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

SLR(1)语法分析表的构建算法

1. 为增广文法 G' 构建LR(0)项目集族 $C = \{I_0, I_1, \dots, I_n\}$

2. 每个项集 I_i 确定状态 i 时的动作:

(a) If $[A \rightarrow \alpha \bullet a \beta]$ is in I_i , and $GOTO(I_i, a) = I_j$, then $ACTION[i, a] = s_j$ (表示 "shift j ")
这里 a 是终结符.

(b) If $[A \rightarrow \alpha \bullet]$ is in I_i , 则 $ACTION[i, a] = r_m$ (表示 "reduce $A \rightarrow \alpha$ ", m 是该规则的编号)。这里 $a \in FOLLOW(A)$, 并且 $A \neq S'$

(c) If $[S' \rightarrow S \bullet]$ is in I_i , 则 $ACTION[i, \$]$ 为 "accept"

如果存在冲突项目, 则称**文法非SLR(1)**.

3. 对于非终结符 A , 如果 $GOTO(I_i, A) = I_j$, 则在表中 $GOTO[i, A] = j$.

4. 所有**未定义项目**对应**出错**。

5. 状态0对应于项目 $S' \rightarrow \bullet S$ 的闭包。

LR(0)分析表

❖ 如果基于某文法构建的**所有**LR(0)项集，要么只有移入项目，要么只有**一个**归约项目或接受项目，则可以构建LR(0)分析表，并且：

- 该文法称为LR(0)文法
- 在分析时，不需要看输入即可确定动作
- 分析表中，归约项目无需考虑后继

LR(0)分析（例）

对文法： $E \rightarrow aA \mid bB$ $A \rightarrow cA \mid d$ $B \rightarrow cB \mid d$ ，试构造LR(0)分析表，如果是LR(0)文法，试分析bccd

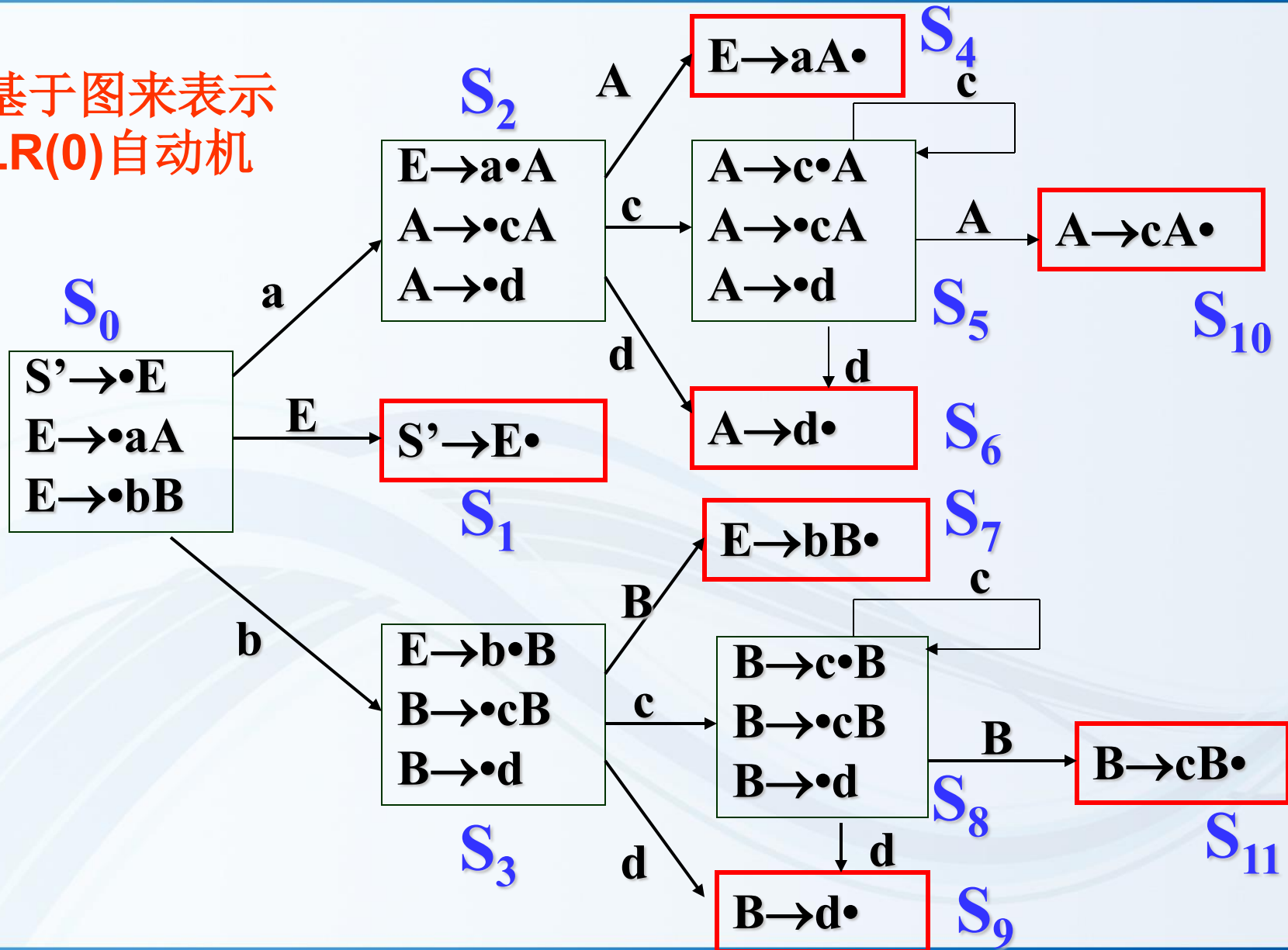
解：引入新的开始符 S' 和规则 $S' \rightarrow E$ ，得增广文法：

(0) $S' \rightarrow E$ (1) $E \rightarrow aA$ (2) $E \rightarrow bB$ (3) $A \rightarrow cA$

(4) $A \rightarrow d$ (5) $B \rightarrow cB$ (6) $B \rightarrow d$

LR分析第一步：构建增广文法并对规则编号!!!

基于图来表示
LR(0)自动机



LR(0)分析 (例)

表格表示的LR(0)自动机

(0) $S' \rightarrow E$

(1) $E \rightarrow aA$

(2) $E \rightarrow bB$

(3) $A \rightarrow cA$

(4) $A \rightarrow d$

(5) $B \rightarrow cB$

(6) $B \rightarrow d$

状态	项目集	后继符号	后继状态
S_0	$\{S' \rightarrow \bullet E$ $E \rightarrow \bullet aA$ $E \rightarrow \bullet bB\}$	E a b	S_1 S_2 S_3
S_1	$\{S' \rightarrow E \bullet\}$	$\#S' \rightarrow E$	S_{12}
S_2	$\{E \rightarrow a \bullet A$ $A \rightarrow \bullet cA$ $A \rightarrow \bullet d\}$	A c d	S_4 S_5 S_6
S_3	$\{E \rightarrow b \bullet B$ $B \rightarrow \bullet cB$ $B \rightarrow \bullet d\}$	B c d	S_7 S_8 S_9

LR(0)分析 (例)

(0) $S' \rightarrow E$

(1) $E \rightarrow aA$

(2) $E \rightarrow bB$

(3) $A \rightarrow cA$

(4) $A \rightarrow d$

(5) $B \rightarrow cB$

(6) $B \rightarrow d$

状态	项目集	后继符号	后继状态
S_4	$\{E \rightarrow aA \bullet\}$	$\#E \rightarrow aA$	S_{12}
S_5	$\{A \rightarrow c \bullet A$	A	S_{10}
	$A \rightarrow \bullet cA$ $A \rightarrow \bullet d\}$	c d	S_5 S_6
S_6	$\{A \rightarrow d \bullet\}$	$\#A \rightarrow d$	S_{12}
S_7	$\{E \rightarrow bB \bullet\}$	$\#E \rightarrow bB$	S_{12}
S_8	$\{B \rightarrow c \bullet B$	B	S_{11}
	$B \rightarrow \bullet cB$	c	S_8
	$B \rightarrow \bullet d\}$	d	S_9
S_9	$\{B \rightarrow d \bullet\}$	$\#B \rightarrow d$	S_{12}
S_{10}	$\{A \rightarrow cA \bullet\}$	$\#A \rightarrow cA$	S_{12}
S_{11}	$\{B \rightarrow cB \bullet\}$	$\#B \rightarrow cB$	S_{12}

LR(0)分析 (例)

栈	符号	输入	动作
0	\$	bccd\$	移入
0 3	\$b	ccd\$	移入
0 3 8	\$bc	cd\$	移入
0 3 8 8	\$bcc	d\$	移入
0 3 8 8 9	\$bccd	\$	归约/B→d
0 3 8 8 <u>11</u>	\$bccB	\$	归约/B→cB
0 3 8 <u>11</u>	\$bcB	\$	归约/B→cB
0 3 7	\$bB	\$	归约/E→bB
0 1	\$E	\$	接受

SLR(1)分析 (例)

文法 $G[S]$: $S \rightarrow AB$ $A \rightarrow aBa | \epsilon$ $B \rightarrow bAb | \epsilon$

1. 该文法是SLR(1)的吗?
2. 若是,请构造它的分析表。
3. 给出输入串baab\$的分析过程。

解: 1、引入新开始符 S' 和规则 $S' \rightarrow S$, 得增广文法 $G[S']$

将规则编号为: (0) $S' \rightarrow S$ (1) $S \rightarrow AB$ (2) $A \rightarrow aBa$ (3) $A \rightarrow \epsilon$
(4) $B \rightarrow bAb$ (5) $B \rightarrow \epsilon$

$FOLLOW(S) = FOLLOW(S') = \{\$ \}$

$FOLLOW(A) = (FIRST(B) - \{\epsilon\}) \cup FOLLOW(S) \cup \{b\} = \{b, \$ \}$

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

SLR(1)分析 (例)

状态	项目集	后继符号	后继状态
S_0	$\{S' \rightarrow \bullet S$ $S \rightarrow \bullet AB$ $A \rightarrow \bullet aBa$ $A \rightarrow \bullet\}$	S A a $b \$ \quad \#A \rightarrow \epsilon$	S_1 S_2 S_3 S_{10}
S_1	$\{S' \rightarrow S \bullet\}$	$\$ \quad \#S' \rightarrow S$	S_{10}
S_2	$\{S \rightarrow A \bullet B$ $B \rightarrow \bullet bAb$ $B \rightarrow \bullet\}$	B b $a \$ \quad \#B \rightarrow \epsilon$	S_4 S_5 S_{10}
S_3	$\{A \rightarrow a \bullet Ba$ $B \rightarrow \bullet bAb$ $B \rightarrow \bullet\}$	B b $a \$ \quad \#B \rightarrow \epsilon$	S_6 S_5 S_{10}
S_4	$\{S \rightarrow AB \bullet\}$	$\$ \quad \#S \rightarrow AB$	S_{10}

FOLLOW(S)={\\$}

FOLLOW(A)={b,\\$}

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

状态	项目集	后继符号	后继状态
S_5	$\{B \rightarrow b \bullet Ab$ $A \rightarrow \bullet aBa$ $A \rightarrow \bullet\}$	A a $b \$ \quad \#A \rightarrow \epsilon$	S_7 S_3 S_{10}
S_6	$\{A \rightarrow aB \bullet a\}$	a	S_8
S_7	$\{B \rightarrow bA \bullet b\}$	b	S_9
S_8	$\{A \rightarrow aBa \bullet\}$	$b \$ \quad \#A \rightarrow aBa$	S_{10}
S_9	$\{B \rightarrow bAb \bullet\}$	$a \$ \quad \#B \rightarrow bAb$	S_{10}

SLR(1)分析 (例)

分析表无冲突项目，故
本文法为**SLR (1)** 文法

(0) $S' \rightarrow S$

(1) $S \rightarrow AB$

(2) $A \rightarrow aBa$

(3) $A \rightarrow \epsilon$

(4) $B \rightarrow bAb$

(5) $B \rightarrow \epsilon$

状态	ACTION			GOTO		
	a	b	\$	S	A	B
0	S ₃	r ₃	r ₃	1	2	
1			acc			
2	r ₅	S ₅	r ₅			4
3	r ₅	S ₅	r ₅			6
4			r ₁			
5	S ₃	r ₃	r ₃		7	
6	S ₈					
7		S ₉				
8		r ₂	r ₂			
9	r ₄		r ₄			

SLR(1)分析 (例)

栈	符号	输入	动作
0	\$	baab\$	归约/ $A \rightarrow \epsilon$
0 2	\$ A	baab\$	移入
0 2 5	\$ A b	aab\$	移入
0 2 5 3	\$ A b a	ab\$	归约/ $B \rightarrow \epsilon$
0 2 5 3 6	\$ A b a B	ab\$	移入
0 2 5 3 6 8	\$ A b a B a	b\$	归约/ $A \rightarrow aBa$
0 2 5 7	\$ A b A	b\$	移入
0 2 5 7 9	\$ A b A b	\$	归约/ $B \rightarrow bAb$
0 2 4	\$ A B	\$	归约/ $S \rightarrow AB$
0 1	\$ S	\$	接受