

第三章：语法分析

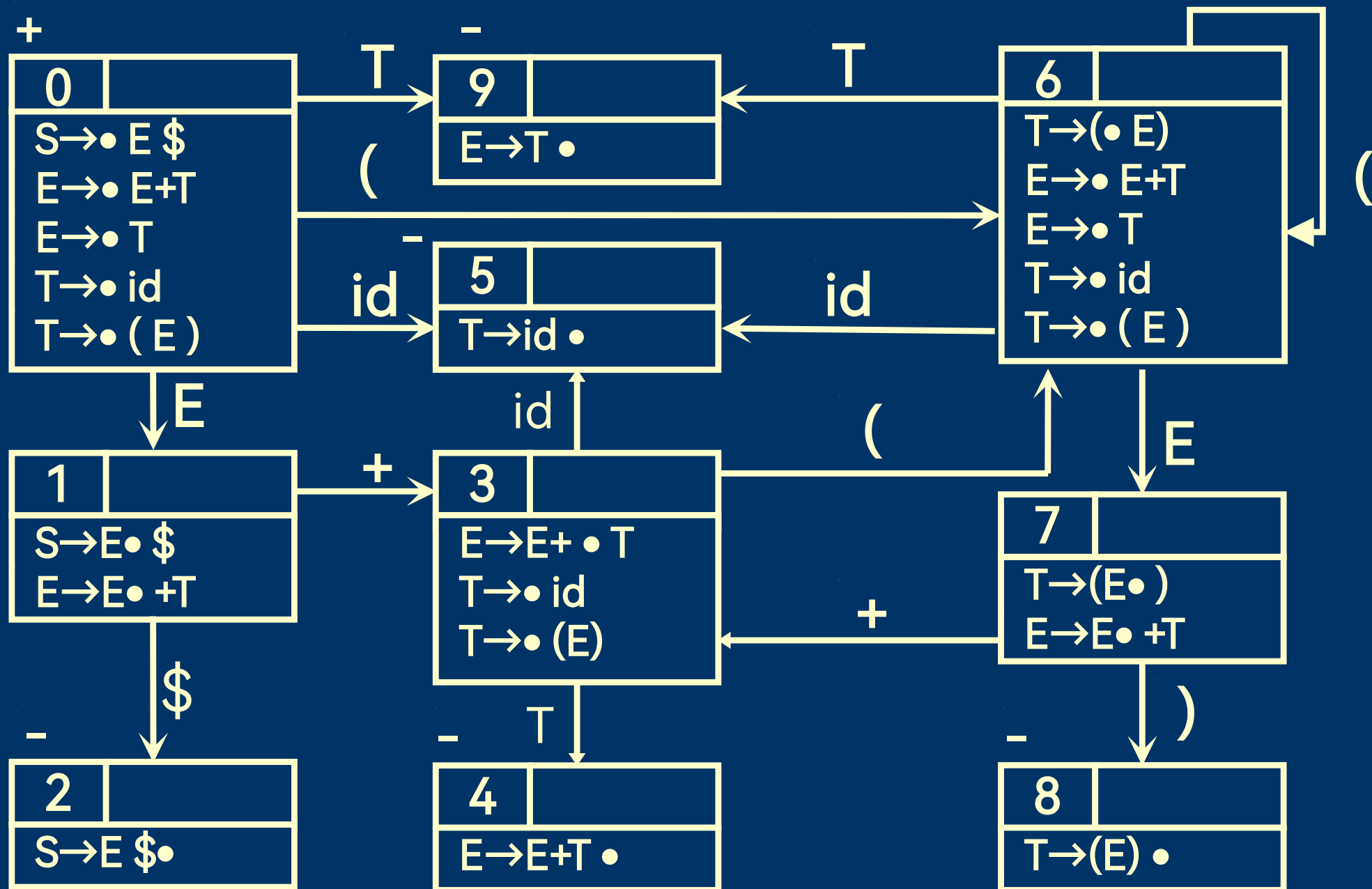
LR(0)语法分析
SLR1(1)方法



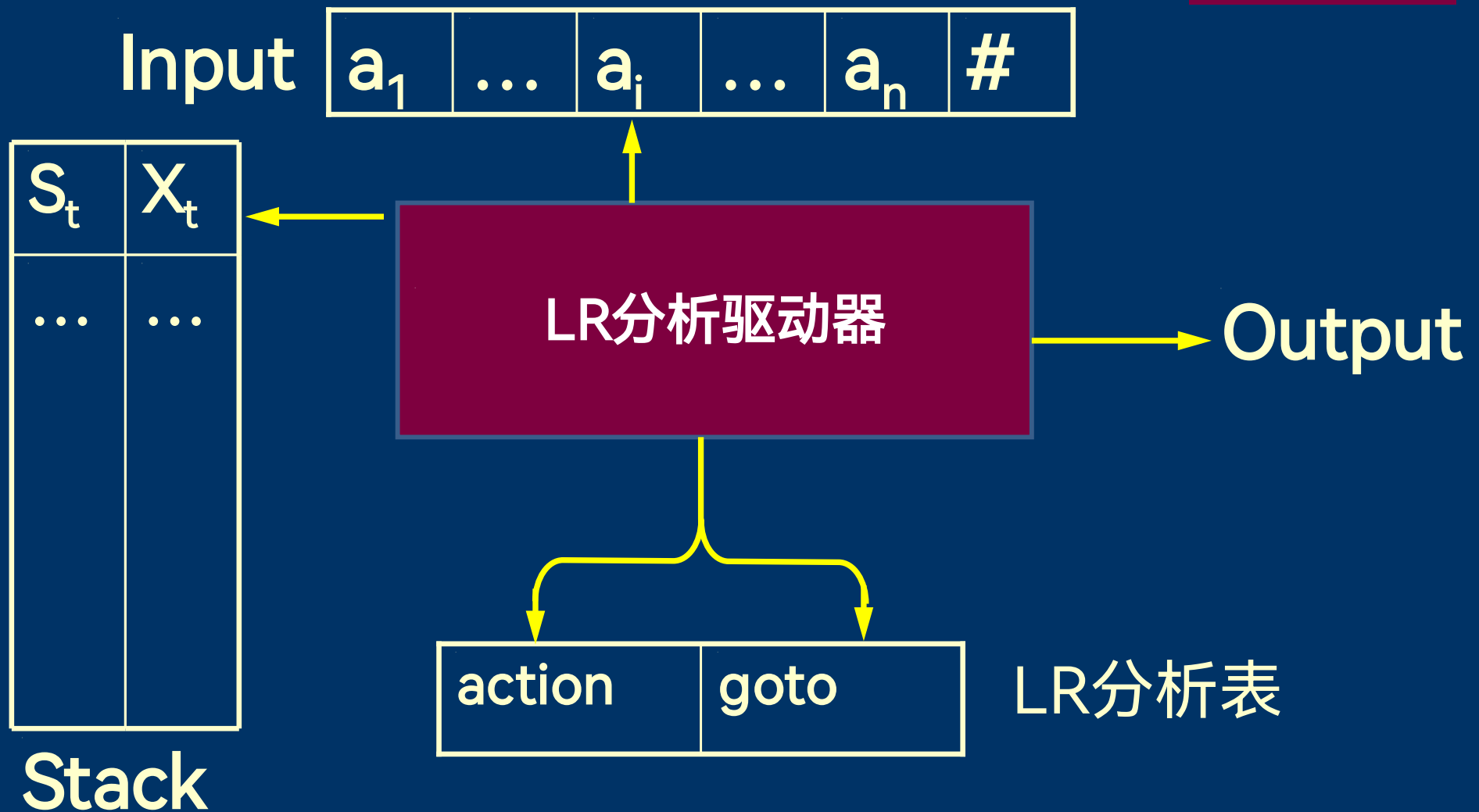
例子

设有文法G(S):

$$S \rightarrow E \$$$
$$E \rightarrow E + T$$
$$E \rightarrow T$$
$$T \rightarrow \text{id} | (E)$$



1.1 LR分析模型



1.2.1 LR分析表

□ Action表:

状态 $\times(V_T \cup \{\#\}) \rightarrow$ 动作

动作包括: Shift / Reduce / Accept / Error (移入、归约、成功、失败)

□ Goto表:

状态 $\times V_N \rightarrow$ 状态|error

1.2.2 LR(0)分析表的构造

假设 IS_k 为LR(0)项目集,

action矩阵:

1. 若 $A \rightarrow \alpha \bullet a \beta \in IS_k$, 且 $GO(IS_k, a) = IS_i$, $a \in V_T$, 则 $action(IS_k, a) = S_i$, 表示移入动作。
2. 若 $A \rightarrow \alpha \bullet \in IS_k$, 则对任意 $a \in V_T \cup \{\#\}$, 令 $action(IS_k, a) = R_j$, 其中产生式 $A \rightarrow \alpha$ 的编号为j, 表示用编号为j的产生式进行归约。
3. 若 $Z \rightarrow \alpha \bullet \in IS_k$, 且Z为拓广产生式的左部非终极符(文法的开始符), 则 $action(IS_k, \#) = \text{Accept}$ 。
4. 其它情形, 则 $\text{Error}(n)$, 表示出错标志, 也可不填。

goto矩阵:

若 $GO(IS_k, A) = IS_i$, $A \in V_N$, 则 $goto(IS_k, A) = i$ 。

1.2.3

LR(0)分析表的构造例子

[1]S → E \$

$$[2]E \rightarrow E + T$$
$$[3]E \rightarrow T$$

$[4]T \rightarrow \text{id}$

$$[5]T \rightarrow (E)$$
[illegible]

1.3 LR分析表提供的信息

合法性检查信息 $[A \rightarrow \alpha \bullet a \beta]$

移入/归约信息 $[A \rightarrow \alpha \bullet a \beta][A \rightarrow \pi \bullet]$

移入/归约后的转向状态信息

设当前格局是：

#	X_1	X_2	...	X_k	...	X_t
S_{i0}	S_{i1}	S_{i2}	...	S_{ik}	...	S_{it}

$$a_i a_{i+1} \dots a_n \#$$

移入动作：设 S_{it} 的 a_i 输入边所指向的状态为 S^*

#	X_1	X_2	...	X_k	...	X_t	a_i
S_{i0}	S_{i1}	S_{i2}	...	S_{ik}	...	S_{it}	S^*

归约动作：设按 $A \rightarrow X_{k+1} X_{k+2} \dots X_t$ 进行归约，则首先归约为A

#	X_1	X_2	...	X_k	A.	X_t
S_{i0}	S_{i1}	S_{i2}	...	S_{ik}	...	S_{it}

S_{ik} 的A输出边所指向的状态设为 S^* ，则格局变为：（输入流不变）

#	X_1	X_2	...	X_k	A
S_{i0}	S_{i1}	S_{i2}	...	S_{ik}	S^*

1.4 LR驱动程序

状态栈、符号栈和输入流的开始格局为： $(\#S_1, \#, a_1a_2\dots a_n\#)$

移入：若当前格局为 $(\#S_1S_2\dots S_n, \#X_1X_2\dots X_n, a_ia_{i+1}\dots a_n\#)$ ，且 $\text{Action}(S_n, a_i) = S_j$ ， $a_i \in V_T$ ，则 a_i 入符号栈，第 j 个状态 S_j 入状态栈。即移入后的格局变为： $(\#S_1S_2\dots S_n S_j, \#X_1X_2\dots X_n a_i, a_{i+1}\dots a_n\#)$

1.4 LR驱动程序

归约: 若当前格局为 $(\#S_1S_2\dots S_n, \#X_1X_2\dots X_n, a_1a_{i+1}\dots a_n\#)$ ，且 $\text{Action}(IS_n, a) = R_j$ ， $a \in V_T \cup \{\#\}$ ，则按照第 j 个产生式进行归约，符号栈和状态栈相应元素退栈，归约后的文法符号入栈。假设第 j 个产生式为 $A \rightarrow a$ ， $k = |a|$ ($a = X_{n-k+1}\dots X_n$)，则归约后的格局变为：

$$(\#S_1S_2\dots S_{n-k}S, \#X_1X_2\dots X_{n-k}A, a_1a_{i+1}\dots a_n\#)$$

其中 $S = \text{Goto}(S_{n-k}, A)$ 。

1.4 LR驱动程序

成功: 若状态栈的栈顶状态为 S_i , 输入流当前值为 $\#$, 且 $\text{action}(S_i, \#)=\text{Accept}$, 则分析成功。

失败: 若状态栈的栈顶状态为 S_i , 输入流当前值为 a , 且 $\text{action}(S_i, a)=\text{Error}$ 或空, 则转向出错处理程序。

1.5 LR(0)文法的限定条件

- 定义：在可归前缀图中，如果一个状态含有两个或两个以上的以上的归约项目，则称有归约/归约冲突。如果同时含有移入项目和归约项目，称为移入/归约冲突，这类状态称为二义性状态。

2.1 SLR(1)方法

有些情况LR(0)方法解决不了，例如有文法：

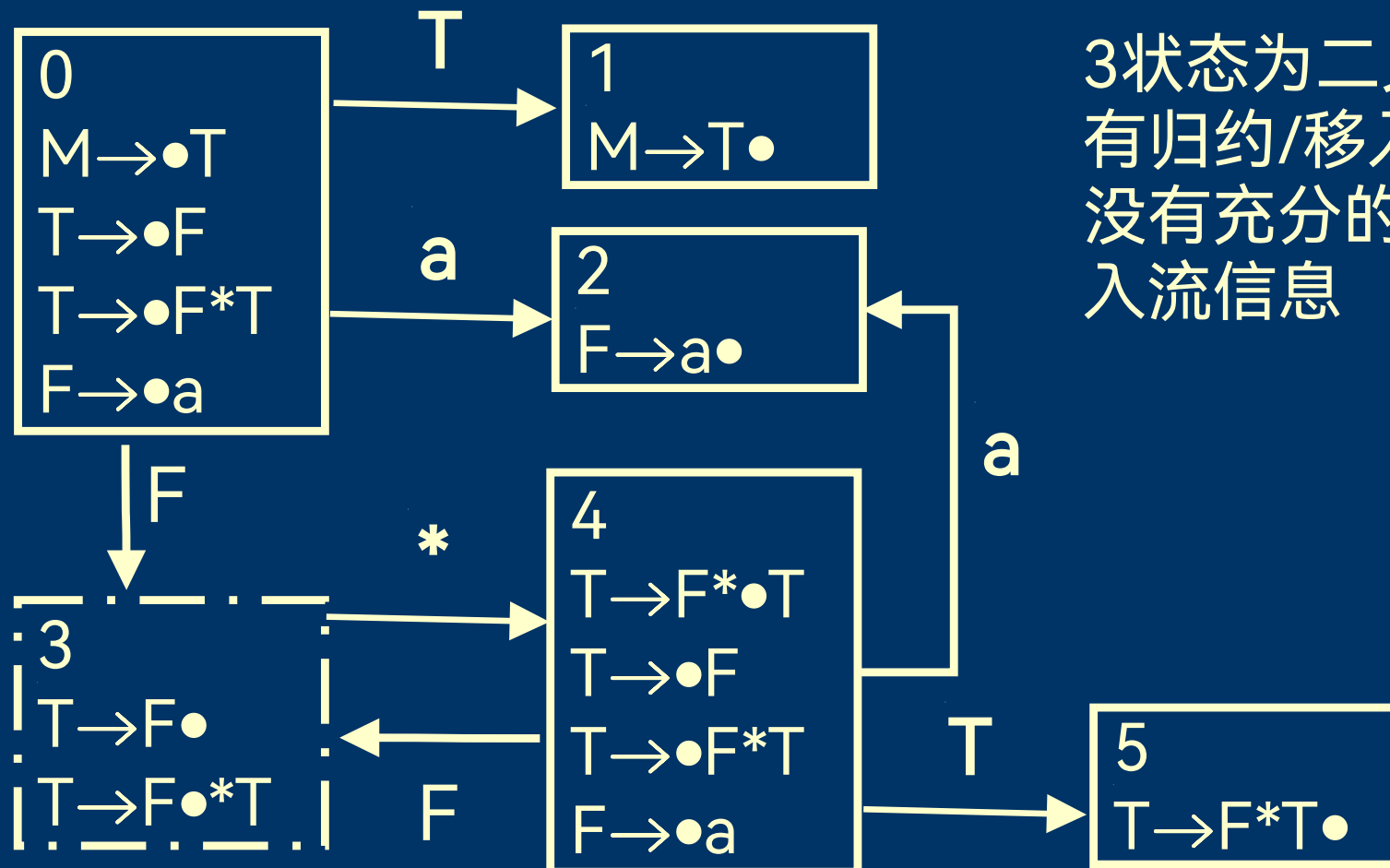
[1] $M \rightarrow T$

[2] $T \rightarrow F$

[3] $T \rightarrow F * T$

[4] $F \rightarrow a$

2.1 SLR(1)方法



3状态为二义性节点,
有归约/移入冲突
没有充分的利用输入流信息

2.1 SLR(1)方法

如果某个状态有如下项目集：

$\{ A \rightarrow \alpha \bullet, B \rightarrow \beta \bullet, D \rightarrow \mu \bullet d \gamma \}$ ，则存在着归约-归约，移入-归约冲突

- 若用 $A \rightarrow \alpha \bullet$ 归约，则当前输入符应在 A 的 Follow 集中
- 若用 $B \rightarrow \beta \bullet$ 归约，则当前输入符应在 B 的 Follow 集
- 若移入，则当前输入符应为 d 。

2.1 SLR(1)方法

SLR(1)分析条件

□ LRSM₀中存在着状态

$$\{ A_1 \rightarrow a_1 \bullet, \dots, A_n \rightarrow a_n \bullet, \\ B_1 \rightarrow \beta_1 \bullet a_1 r_1, \dots, B_m \rightarrow \beta_m \bullet a_m r_m \}$$

则集合:

Follow(A₁)、...、Follow(A_n)、a₁,...,a_m}两
两之间互不相交

2.2 SLR(1)分析表的构造

假设 IS_k 为LR(0)项目集，则

action矩阵:

- 若 $A \rightarrow \alpha \bullet a \beta \in IS_k$ ，且 $GO(IS_k, a) = IS_i$ ， $a \in V_T$ ，则 $action(IS_k, a) = S_i$ ，表示移入动作。
- 若 $A \rightarrow \alpha \bullet \in IS_k$ ，则对任意 $a \in V_T$ ， $a \in Follow(A)$ ，令 $action(IS_k, a) = R_j$ ，其中产生式 $A \rightarrow \alpha$ 的编号为j，表示用编号为j的产生式进行归约。
- 若 $Z \rightarrow \alpha \bullet \in IS_k$ ，且Z为拓广产生式的左部非终极符(文法的开始符)，则 $action(IS_k, \#) = Accept$ 。
- 其它情形，则 $Error(n)$ ，表示出错标志，也可不填。

goto矩阵:

- 若 $GO(IS_k, A) = IS_i$ ， $A \in V_N$ ，则 $goto(IS_k, A) = i$ 。

2.3 SLR(1)语法分析表

[1] $M \rightarrow T$

[2] $T \rightarrow F$

[3] $T \rightarrow F * T$

[4] $F \rightarrow a$

$\text{Follow}(M) = \{\#\}$

$\text{Follow}(T) = \{\#\}$

$\text{Follow}(F) = \{*, \#\}$

	action表			goto表	
	a	*	#	T	F
0	S_2			1	3
1			Ac		
2		R_4	R_4		
3		S_4	R_2		
4	S_2			5	3
5			R_3		

2.4 SLR(1)文法限定条件

限定条件

- 语法分析表单值

LR(0)和SLR(1)分析能力对比

- LR(0)只看分析栈的内容，不考虑当前输入符；SLR(1)考虑输入符，用follow集来解决冲突，因此SLR(1)要比LR(0)分析能力强。

SLR(1)的分析过程

状态栈	符号栈	输入串	分析动作	转向状态
0		a*a*a#	S2	
02	a	*a*a#	R4	3
03	F	*a*a#	S4	
034	F*	a*a#	S2	
0342	F*a	*a#	R4	3
0343	F*F	*a#	S4	
03434	F*F*	a#	S2	
034342	F*F*a	#	R4	3
034343	F*F*F	#	R2	5
034345	F*F*T	#	R3	5
0345	F*T	#	R3	5
01	T	#	AC	