

第三章：语法分析

SLR(1) 方法
LR(1) 语法分析

1. LR(0) 分析方法的不足

- ✪ LR(0)方法对文法的要求严格。
- ✪ LR(0)方法容易出现冲突状态。

1. LR(0) 分析方法的不足

1、如果某个状态有如下项目集：

$\{ A \rightarrow \alpha \bullet, D \rightarrow \mu \bullet d \gamma \}$ ，则存在移入-归约冲突。

可以如下解决：

- ◆ 若当前输入符在A的Follow集中，则应用 $A \rightarrow \alpha \bullet$ 归约；
- ◆ 若当前输入符为d则应移入。
- ◆ 而对当前输入符为d，d又在A的Follow集中，则无法解决。

1 LR(0) 分析方法的不足

如果某个状态有如下项目集：

$\{ A \rightarrow \alpha \bullet, B \rightarrow \beta \bullet \}$ ，则存在着归约-归约冲突。

可以如下解决：

- ◆ 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 集中
- ◆ 当前输入符应在 A 的 Follow 集中又在 B 的 Follow 集中，则无法解决。

2.1 SLR(1) 分析条件

- ◆ LRSM₀中存在着状态:

$$\begin{aligned} & \{ A_1 \rightarrow \alpha_1 \bullet, \\ & \quad \dots, \\ & \quad A_n \rightarrow \alpha_n \bullet, \\ & \quad B_1 \rightarrow \beta_1 \bullet a_1 r_1, \\ & \quad \dots, \\ & \quad B_m \rightarrow \beta_m \bullet a_m r_m \} \end{aligned}$$

则集合:

Follow(A_1)、 \dots 、Follow(A_n)、 $\{a_1, \dots, a_m\}$

两两相交为空。注: a_1, \dots, a_m 中可以有相同者。

2.2 SLR(1) 分析表的构造

假设 IS_k 为 LR(0) 项目集, 则

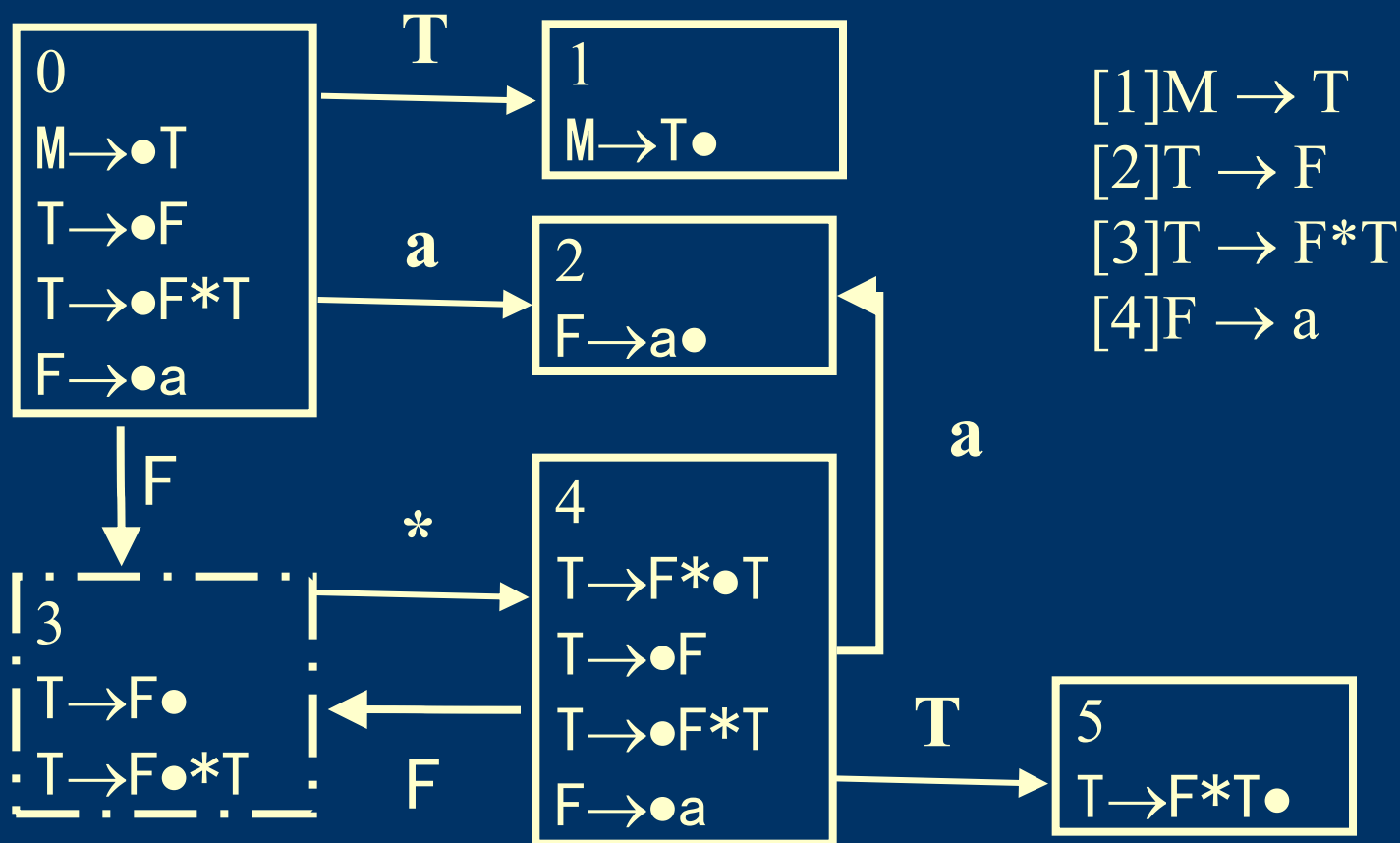
action 矩阵:

- ◆ 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



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	S2			1	3
1			Ac		
2		R4	R4		
3		S4	R2		
4	S2			5	3
5			R3		

自动机

示例

2.4 SLR(1) 文法限定条件

限定条件

- ◆ SLR(1) 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。

2.5 SLR(1) 的分析过程

a*a*a

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

分析表

2.6 SLR(1)问题所在

- ◆ $Z \rightarrow B^1 a B^2 b B^3 c$
- ◆ $B \rightarrow d$

SLR(1)归约时向前看一个符号，但是不区分语法符号的不同出现。上述文法中，B出现了三次，很显然 B^1 的后继符只能是a， B^2 的后继符只能是b， B^3 的后继符只能是c，而 $\text{Follow}(B) = \{a, b, c\}$ ，用SLR(1)就失去了精度。

2.7 几种LR方法的简单对比

- ◆ LR(0)方法不依赖输入流，直接判定归约，容易出现冲突。
- ◆ SLR(1)方法简单的把非终极符的Follow集做为可归约的依据，并不精确。
- ◆ 一个非终极符在不同的位置上出现，它所允许的后继符是不同的。LR(1)针对不同产生式上的非终极符，分别定义其后继符集，减少了移入/归约、归约/归约冲突。

3.1 LR(1) 基本思想

- ◆ 构造各种LR分析器的任务就是构造其 action 表和 goto 表，其他部分基本相同。LR(1) 的基本思想是对非终极符的每个不同出现求其后继符，而不是给每个非终极符求其统一的后继符，我们称其为展望符集。

3.2 LR(1)的基本概念

- ◆ LR(1)项目: $[A \rightarrow \alpha \bullet \beta, a]$, 即LR(0)项目及一个 $V_T \cup \{\#\}$ 的展望符组成的二元组。

用IS表示LR(1)项目的集合, 简称LR(1)项目集。其中, 项 $Z \rightarrow \bullet \alpha$ 的展望符为 $\#$

- ◆ $IS_{(X)}$: LR(1)项目集IS对于X的投影
 $IS_{(X)} = \{ [A \rightarrow \alpha X \bullet \beta, a] \mid [A \rightarrow \alpha \bullet X \beta, a] \in IS \}$

3.2 LR(1)的基本概念

- ◆ $\text{CLOSURE}(IS) = IS \cup \{ [A \rightarrow \bullet \beta, a] \mid [B \rightarrow \alpha_1 \bullet A \alpha_2, b] \in \text{CLOSURE}(IS), A \rightarrow \beta \text{ 是产生式}, a \in \text{First}(\alpha_2 b) \}$
- ◆ GO: 若 IS 是一个 LR(1) 项目集, X 是一个文法符号, 则 $\text{GO}(IS, X) = \text{CLOSURE}(IS_{(X)})$ 。

3.3 可归前缀图的构造

- ◆ Step1. 构造初始状态 IS_0 : $IS_0 = \text{CLOSURE}(\{ [Z \rightarrow \bullet S, \#] \})$, 并给 IS_0 标上NO。
- ◆ Step2. 从已构造的LRSM₁部分图选择被标为NO的任一状态 IS , 删除NO, 对每个符号 $X \in V_T \cup V_N$, 做下面动作:
 - 1) 令 $IS_j = \text{CLOSURE}(IS_{(X)})$ 。
 - 2) 若 IS_j 非空:
 - ①如果在LRSM₁部分图中已有 IS_j 项目集, 则在 IS 和 IS_j 之间画有向 X 边: $IS \xrightarrow{X} IS_j$ 。
 - ②如果在LRSM部分图中没有 IS_j 项目集, 则将 IS_j 作为LRSM的一个新的状态节点, 并给 IS_j 标上NO, 同在 IS 和 IS_j 之间画有向 X 边:
$$IS \xrightarrow{X} IS_j$$
- ◆ 重复Step2, 直至没有被标记为NO的状态结点为止。

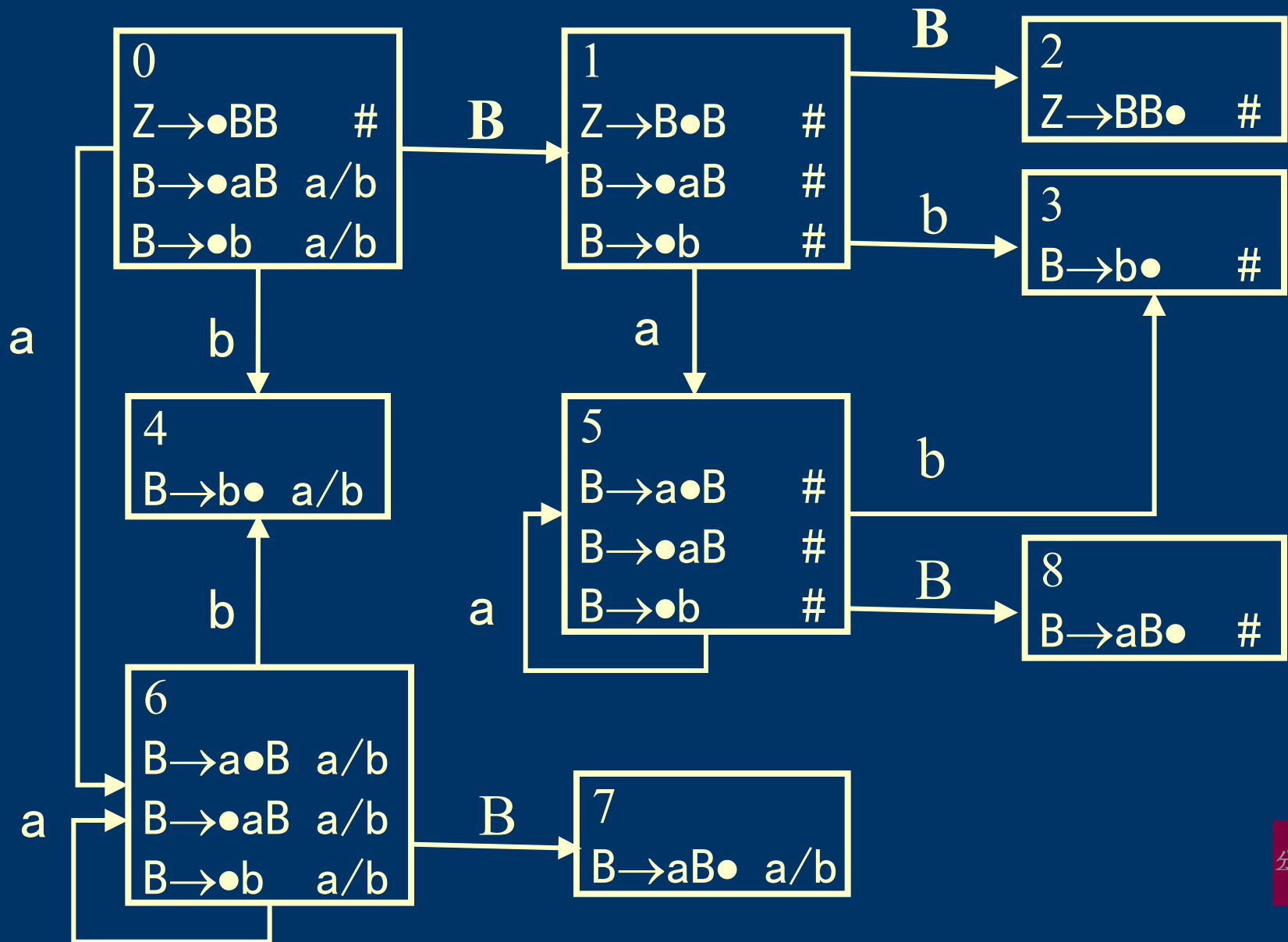
构造下面文法的LR(1)自动机

◆ 有文法：

$Z \rightarrow BB$

$B \rightarrow aB$

$B \rightarrow b$



3.4 LR(1) 分析表的构造

假设 IS_k 为 LR(1) 项目集则：

- ◆ 若 $[Z \rightarrow \alpha \bullet, \#] \in IS_k$ ，且 Z 为拓广产生式的左部非终极符，则 $action(IS_k, \#) = Accept$ 。
- ◆ 若 $[A \rightarrow \alpha \bullet, a] \in IS_k$ ，且产生式 $A \rightarrow \alpha$ 的编号为 j ，则 $action(IS_k, a) = R_j$ ，表示用编号为 j 的产生式进行归约。

3.4 LR(1) 分析表的构造

- ◆ 若 $[A \rightarrow \alpha \bullet a \beta, b] \in IS_k$, 且 $GO(IS_k, a) = IS_i$, $a \in V_T$, 则 $action(IS_k, a) = S_i$, 表示把状态 IS_i 和展望符 a 入栈。
- ◆ 若 $GO(IS_k, A) = IS_i$, $A \in V_N$, 则 $goto(IS_k, A) = i$ 。
- ◆ 其它情形, 则 $Error(n)$, 表示出错标志, 也可不填。

3.5 LR(1) 文法的定义

- ◆ 对于一个文法，若按照上述算法构造的分析表中没有冲突动作，则称该文法为 LR(1) 文法。

LR(1) 分析表

- ◆ 有文法:
- [1] $Z \rightarrow BB$
- [2] $B \rightarrow aB$
- [3] $B \rightarrow b$

	action表			goto表
	a	b	#	B
0	S6	S4		1
1	S5	S3		2
2			AC	
3			R3	
4	R3	R3		
5	S5	S3		8
6	S6	S4		7
7	R2	R2		
8			R2	

状态栈	符号栈	输入串	Action	GoTo
0	#	abaab#	S6	
0,6	#a	baab#	S4	
0,6,4	#ab	aab#	R3	7
0,6,7	#aB	aab#	R2	1
0,1	#B	aab#	S5	
0,1,5	#Ba	ab#	S5	
0,1,5,5	#Baa	b#	S3	
0,1,5,5,3	#Baab	#	R3	8
0,1,5,5,8	#BaaB	#	R2	8
0,1,5,8	#BaB	#	R2	2
0,1,2	#BB	#	AC	

习题

设文法 $G[S]$ 为：

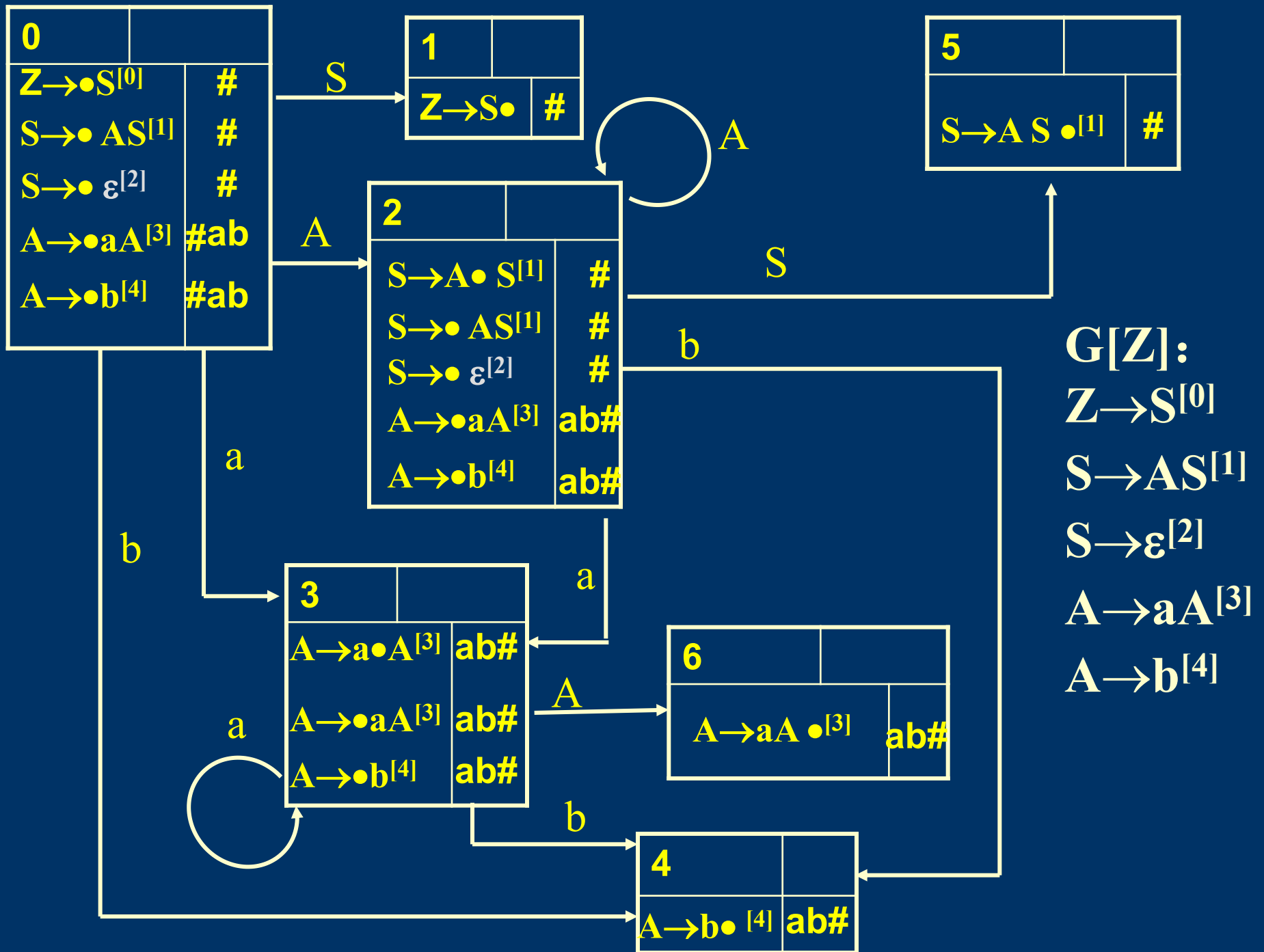
$S \rightarrow AS$

$S \rightarrow \varepsilon$

$A \rightarrow aA$

$A \rightarrow b$

证明 $G[S]$ 是LR(1)文法；构造它的LR(1)分析表；给出符号串 $abab\#$ 的分析过程



action

	a	b	#
0	S3	S4	R2
1			Acc
2	S3	S4	R2
3	S3	S4	
4	R4	R4	R4
5			R1
6	R3	R3	R3

goto

	A	S
0	2	1
1		
2	2	5
3	6	
4		
5		
6		

G[Z]:
Z→**S**^[0]
S→**A****S**^[1]
S→**ε**^[2]
A→**a****A**^[3]
A→**b**^[4]

状态栈	符号栈	输入流
0	#	abab#
03	#a	bab #
034	#ab	ab #
036	#aA	ab #
02	#A	ab #
023	#Aa	b #
0234	#Aab	#
0236	#AaA	#
022	#AA	#
0225	#AAS	#
025	#AS	#
01	#S	#