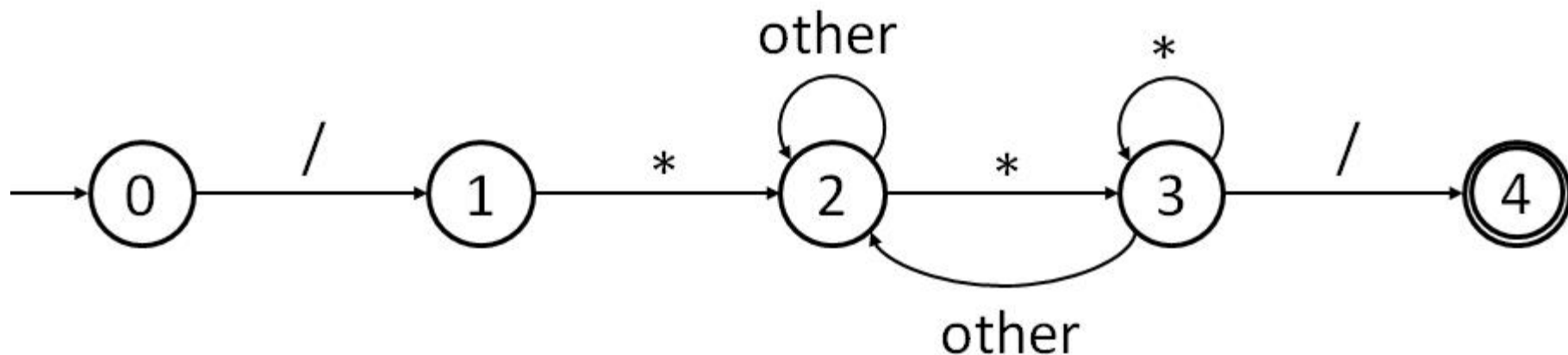


# 识别注释的DFA

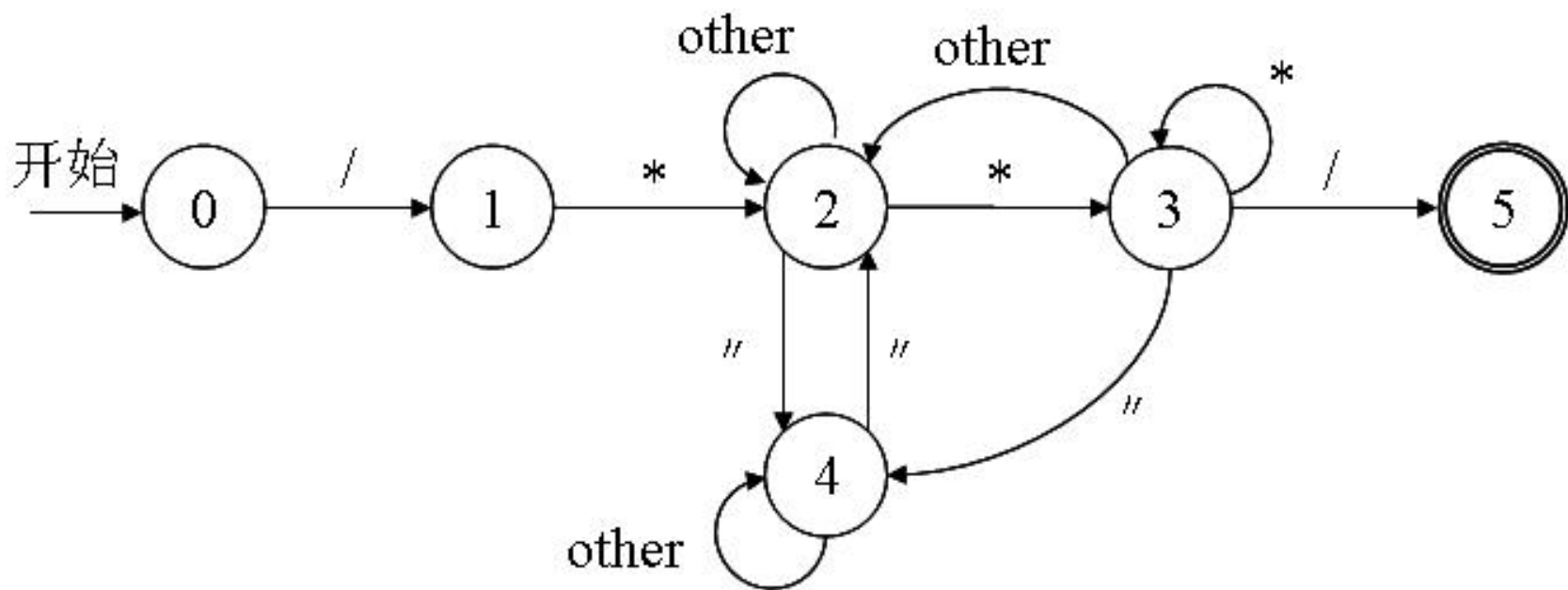


## 习题3.1:

设某程序设计语言规定，其程序中的注释是由“/\*”和“\*/”括起来的字符串，注释中不能出现“\*/”，除非它们出现在双引号中（假设双引号必须配对使用），请给出识别该语言注释结构的DFA D。

# 解答:

- 识别形如`/*....."...".....*/`的注释的DFA



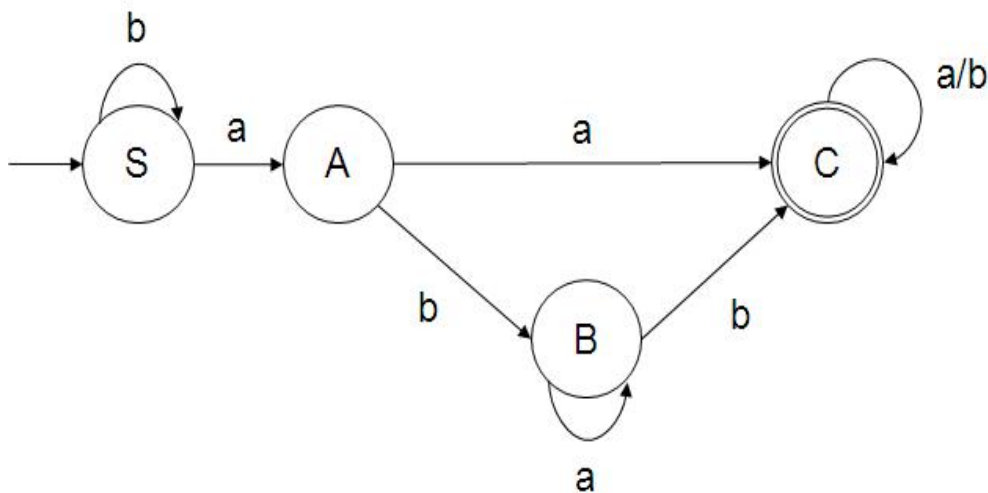
# 课堂练习1

- 自动机 M 的状态转换矩阵如下所示，其中初态是 S，终态是 C。

- (1) 画出相应的状态转换图；
- (2) 写出与之等价的右线性文法。

$$\begin{array}{c} S \\ A \\ B \\ C \end{array} \begin{array}{cc} a & b \\ \left[ \begin{array}{cc} A & S \\ C & B \\ B & C \\ C & C \end{array} \right] \end{array}$$

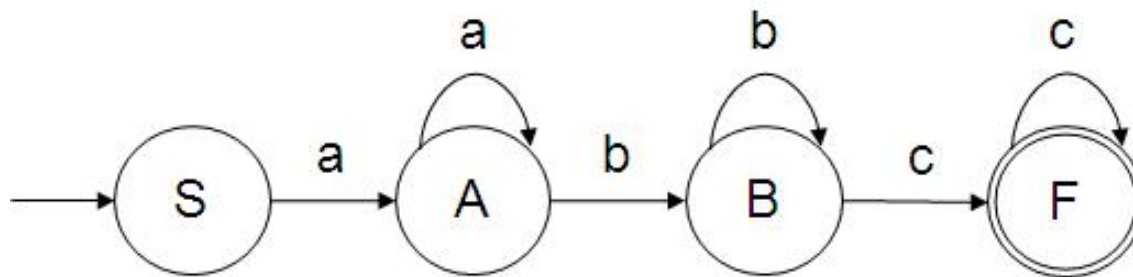
- 解答：



$S \rightarrow aA \mid bS$   
 $A \rightarrow aC \mid bB$   
 $B \rightarrow aB \mid bC$   
 $C \rightarrow aC \mid bC \mid \varepsilon$

# 课堂练习2

- 自动机  $M$  的状态转换图如下所示。
  - (1) 该自动机识别的语言是什么？
  - (2) 给出与之等价的右线性文法。



解答：

- (1) 根据自动机知其产生的语言是：  $L = \{a^m b^n c^i \mid m, n, i \geq 1\}$
- (2) 与之等价的右线性文法是：

$S \rightarrow aA$

$A \rightarrow aA \mid bB$

$B \rightarrow bB \mid cF$

$F \rightarrow cF \mid \varepsilon$

或者：  $S \rightarrow aA$

$A \rightarrow aA \mid bB$

$B \rightarrow bB \mid cF \mid c$

$F \rightarrow cF \mid c$

# 课堂练习3

- 已知正则表达式： $(a^*|b)^*(c|d)$ ，判断下面哪几个正则表达式与其等价，请简述理由。

(1)  $a^*(c|d)|b(c|d)$

(2)  $a^*(c|d)^*|b(c|d)^*$

(3)  $a^*(c|d)|b^*(c|d)$

(4)  $(a|b)^*c|(a|b)^*d$

(5)  $(a^*|b)^*c|(a^*|b)^*d$

- 解答：  
(1)、(2)、(3)与所给正则表达式不等价；  
(4)和(5)与所给正则表达式等价。

# 课堂练习4

## ■ 有限自动机M:

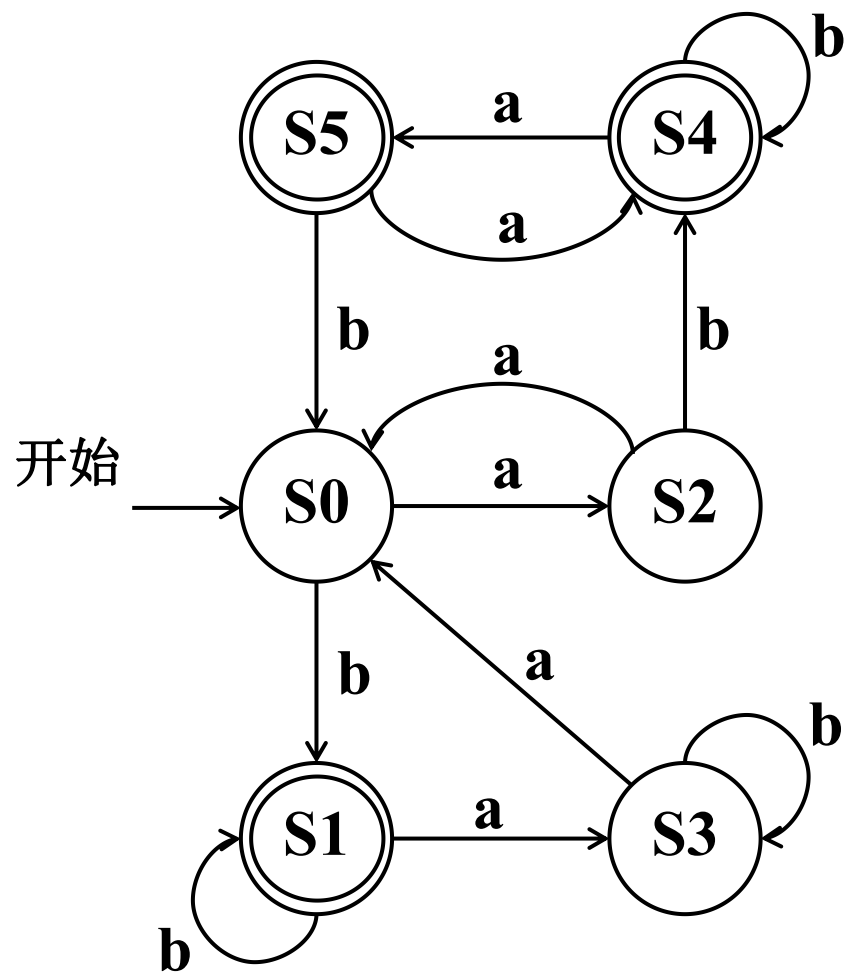
$$M = (\{a, b\}, \{S_0, S_1, S_2, S_3, S_4, S_5\}, S_0, \{S_1, S_4, S_5\}, \delta)$$

$\delta$ 由如右的状态转移矩阵给出。

- (1) 试画出该自动机的状态转换图;
- (2) 试找出一个长度最小的输入串,  
使得在识别此输入串的过程中,  
每一状态至少经历一次;
- (3) 试找出一个长度最小的输入串,  
使得每一状态转换至少经历一次。

	a	b
S <sub>0</sub>	S <sub>2</sub>	S <sub>1</sub>
S <sub>1</sub>	S <sub>3</sub>	S <sub>1</sub>
S <sub>2</sub>	S <sub>0</sub>	S <sub>4</sub>
S <sub>3</sub>	S <sub>0</sub>	S <sub>3</sub>
S <sub>4</sub>	S <sub>5</sub>	S <sub>4</sub>
S <sub>5</sub>	S <sub>4</sub>	S <sub>0</sub>

# 课堂练习4参考答案



**baaaba**

**aaabbaaabbbabab**

## 练习4.1

有如下文法：

$$bexpr \rightarrow bexpr \text{ or } bterm \mid bterm$$
$$bterm \rightarrow bterm \text{ and } bfactor \mid bfactor$$
$$bfactor \rightarrow \text{not } bfactor \mid (bexpr) \mid \text{true} \mid \text{false}$$

请构造一个可以用来分析该文法所产生的句子的递归调用分析程序。



# Step 1: 消除左递归

$bexpr \rightarrow bterm \ E'$

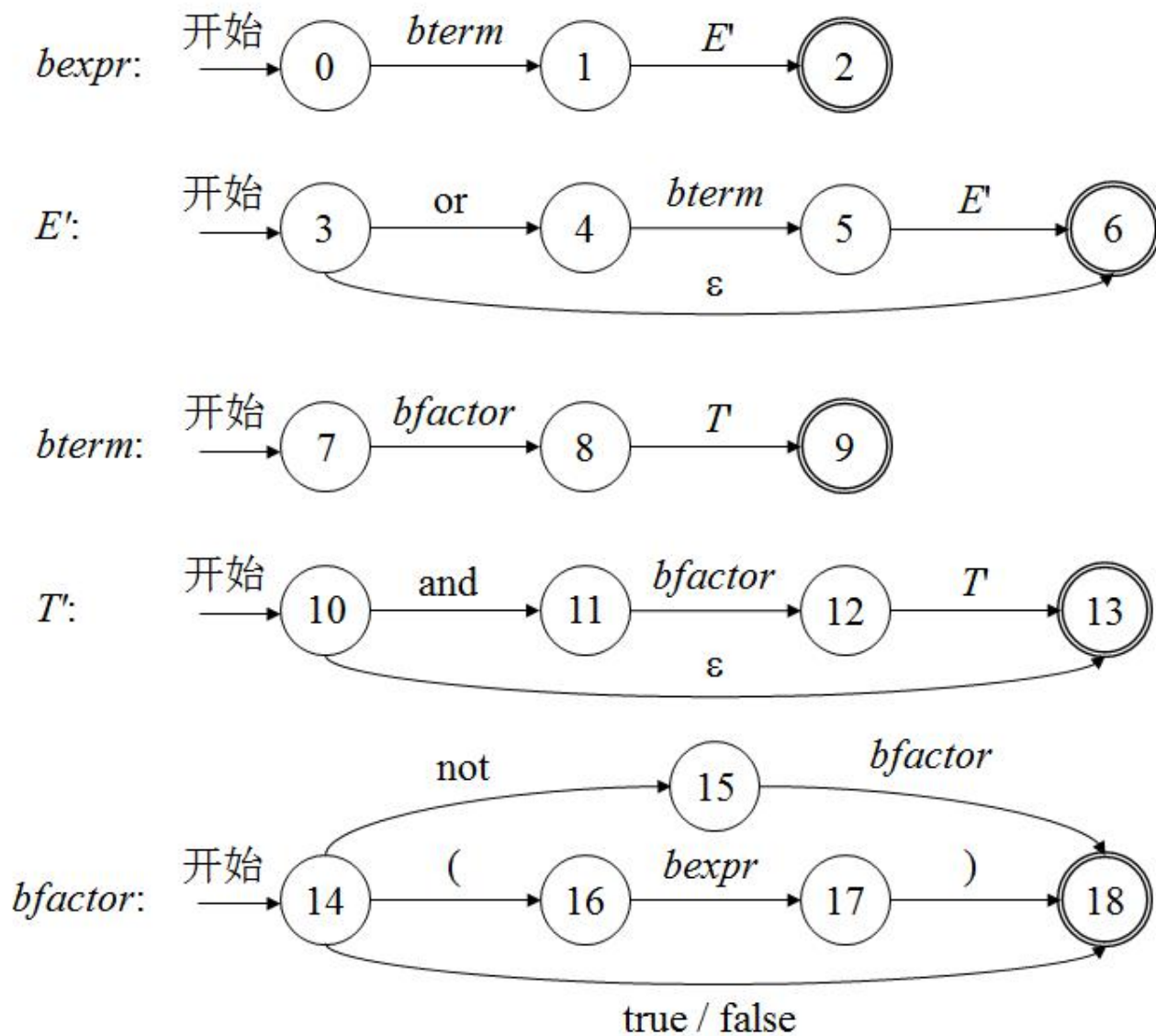
$E' \rightarrow \text{or } bterm \ E' \mid \varepsilon$

$bterm \rightarrow bfactor \ T'$

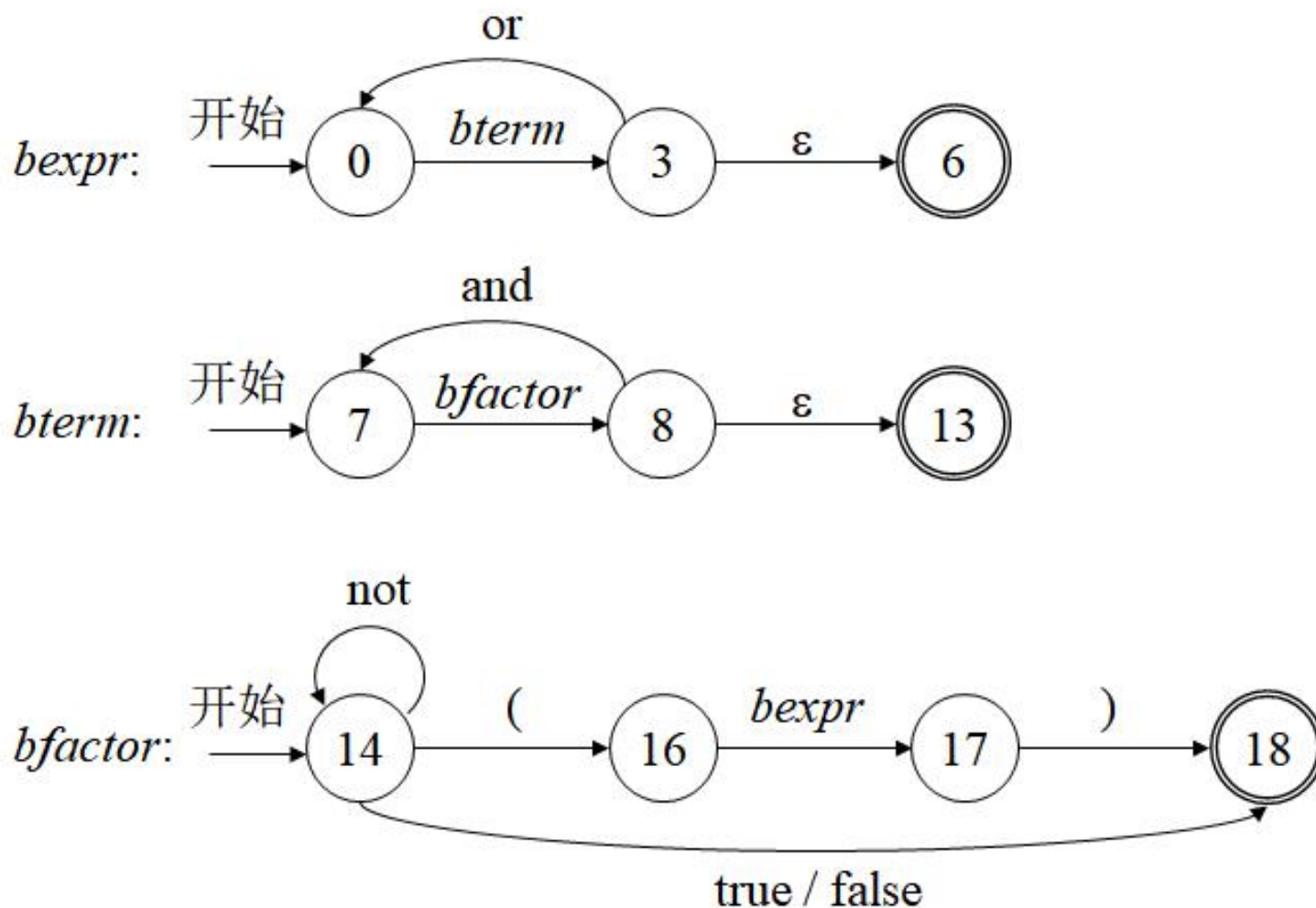
$T' \rightarrow \text{and } bfactor \ T' \mid \varepsilon$

$bfactor \rightarrow \text{not } bfactor \mid (bexpr) \mid \text{true} \mid \text{false}$

## Step 2: 文法 $G'$ 的预测分析程序状态转换图



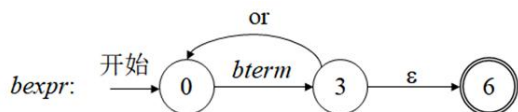
## Step 3: 化简后的预测分析程序状态转换图



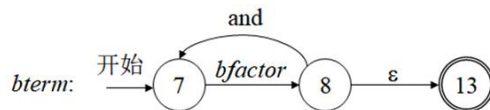
# Step4:根据状态转换图进行程序设计

- *bexpr*的函数
- *bterm*的过程
- *bfactor*的过程

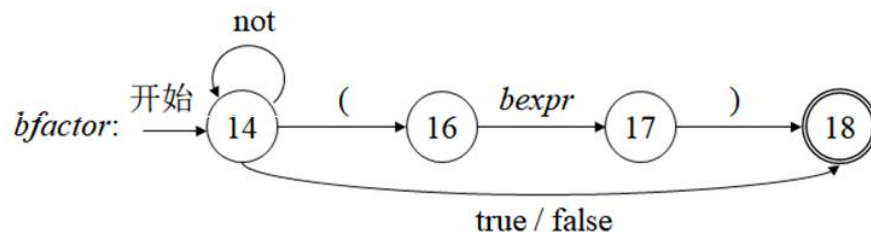
```
void proc_expr(void) {  
    proc_term();  
    if (char=='or') {  
        forward pointer;  
        proc_expr();  
    }  
}
```



```
void proc_term(void) {  
    proc_factor();  
    if (char=='and') {  
        forward pointer;  
        proc_term();  
    }  
}
```



```
void proc_factor(void) {  
    if (char=='not'){  
        forward pointer;  
        proc_factor();  
    }  
    else if (char=='(') {  
        forward pointer;  
        proc_expr();  
        if (char==')')  
            forward pointer;  
        else error();  
    }  
    else if (char=='true')||(char=='false')  
        forward pointer;  
    else error();  
}
```



**练习：** 判断下面的文法是否为LL(1)文法？  
 若不是，可否改写为LL(1)文法？  
 若可以，请构造其LL(1)分析表。

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L, S \mid S$$

**解答：**

- 文法含有左递归，故不是LL(1)文法
- 改写文法：消除左递归

$$S \rightarrow (L) \mid a$$

$$L \rightarrow SL'$$

$$L' \rightarrow ,SL' \mid \epsilon$$

$$\text{FIRST}((L)) \cap \text{FIRST}(a) = \phi$$

$$\text{FIRST}(,SL') \cap \text{FOLLOW}(L') = \phi$$

	FIRST	FOLLOW
S	( a	\$ , )
L	( a	)
L'	, ε	)

	a	(	)	,	\$
S	$S \rightarrow a$	$S \rightarrow (L)$			
L	$L \rightarrow SL'$	$L \rightarrow SL'$			
L'			$L' \rightarrow \epsilon$	$L' \rightarrow ,SL'$	

# 练习:

- 1)  $S \rightarrow aAcBe$
- 2)  $A \rightarrow b$
- 3)  $A \rightarrow Ab$
- 4)  $B \rightarrow d$

构造LL(1)分析表，分析  $abbcbde$ 。

# 解答:

$S \rightarrow aAcBe$   
 $A \rightarrow bA'$   
 $A' \rightarrow bA' \mid \epsilon$   
 $B \rightarrow d$

	First	follow
S	a	\$
A	b	c
A'	b, $\epsilon$	c
B	d	e

	a	b	c	d	e	\$
S	$S \rightarrow aAcBe$					
A		$A \rightarrow bA'$				
A'		$A' \rightarrow bA'$	$A' \rightarrow \epsilon$			
B				$B \rightarrow d$		

# 课堂练习1

有如下文法：

$$E \rightarrow E \vee T \mid T$$

$$T \rightarrow T \wedge F \mid F$$

$$F \rightarrow \neg F \mid (E) \mid t \mid f$$

- (1) 该文法是LL(1)文法吗？说明理由。  
若是，做(3)，若不是，做(2)
- (2) 请改写该文法为LL(1)文法，继续做(3)。
- (3) 构造每个非终结符号的FIRST和FOLLOW函数，  
继续做(4)。
- (4) 构造LL(1)分析表。

# 参考答案

$E \rightarrow E \vee T \mid T$   
 $T \rightarrow T \wedge F \mid F$   
 $F \rightarrow \neg F \mid (E) \mid t \mid f$

- (1) 由于该文法存在左递归，所以不是LL(1)文法。
- (2) 改写文法。消除其中的左递归，得到文法G'：

$E \rightarrow TE'$

$E' \rightarrow \vee TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow \wedge FT' \mid \varepsilon$

$F \rightarrow \neg F \mid (E) \mid t \mid f$

- (3) 每个非终结符号的FIRST和FOLLOW集合如右：

	FIRST	FOLLOW
E	$\neg, (, t, f$	$\$, )$
E'	$\vee, \varepsilon$	$\$, )$
T	$\neg, (, t, f$	$\vee, \$, )$
T'	$\wedge, \varepsilon$	$\vee, \$, )$
F	$\neg, (, t, f$	$\wedge, \vee, \$, )$

- (4) 文法的LL(1)分析表如下：

	$\neg$	$\wedge$	$\vee$	t	f	(	)	\$
E	$E \rightarrow TE'$			$E \rightarrow TE'$	$E \rightarrow TE'$	$E \rightarrow TE'$		
E'			$E' \rightarrow \vee TE'$				$E' \rightarrow \varepsilon$	$E' \rightarrow \varepsilon$
T	$T \rightarrow FT'$			$T \rightarrow FT'$	$T \rightarrow FT'$	$T \rightarrow FT'$		
T'		$T' \rightarrow \wedge FT'$	$T' \rightarrow \varepsilon$				$T' \rightarrow \varepsilon$	$T' \rightarrow \varepsilon$
F	$F \rightarrow \neg F$			$F \rightarrow t$	$F \rightarrow f$	$F \rightarrow (E)$		



# 课堂练习2

证明下面的文法是LL(1)的，但不是SLR(1)的。

$$S \rightarrow (X \mid E] \mid F)$$
$$X \rightarrow E) \mid F]$$
$$E \rightarrow A$$
$$F \rightarrow A$$
$$A \rightarrow \varepsilon$$

# 解答：证明该文法是LL(1)文法

$S \rightarrow (X \mid E \mid F)$   
 $X \rightarrow E) \mid F]$   
 $E \rightarrow A$   
 $F \rightarrow A$   
 $A \rightarrow \varepsilon$

- 该文法每个非终结符号的FIRST集和FOLLOW集合如下：

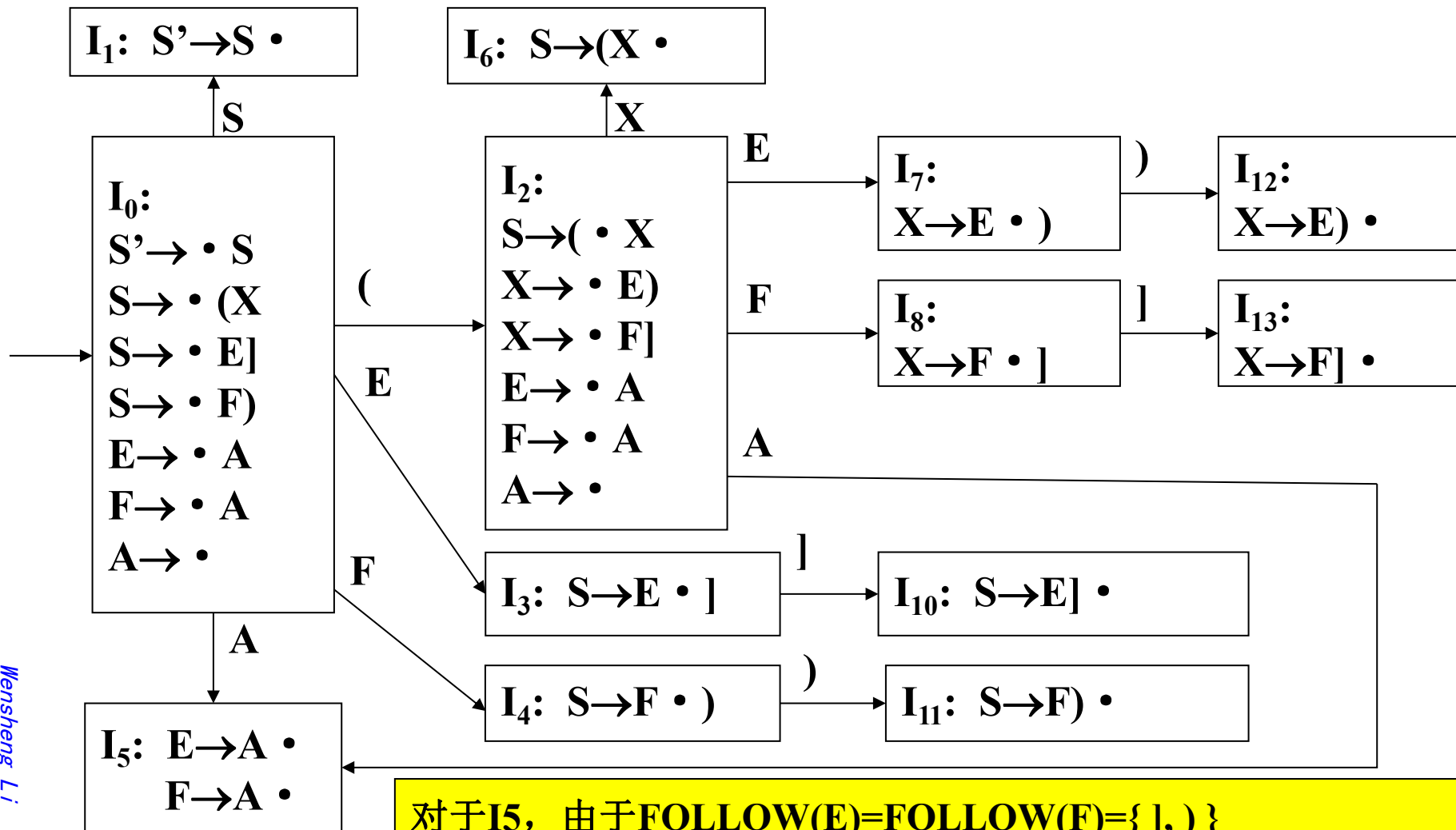
	FIRST	FOLLOW
S	(, [, )	\$
X	[, )	\$
E	$\varepsilon$	[, )
F	$\varepsilon$	[, )
A	$\varepsilon$	[, )

	(	)	[	\$
S	$S \rightarrow (X$	$S \rightarrow F)$	$S \rightarrow E]$	
X		$X \rightarrow E)$	$X \rightarrow F]$	
E		$E \rightarrow A$	$E \rightarrow A$	
F		$F \rightarrow A$	$F \rightarrow A$	
A		$A \rightarrow \varepsilon$	$A \rightarrow \varepsilon$	

- 该文法的LL(1)分析表如右：
- 结论：LL(1)分析表中不含有多重定义的入口，所以该文法是LL(1)文法
- 也可以分析产生式，根据候选式的first集合互不相交来说明。

# 证明该文法不是SLR(1)文法

- 构造该文法的LR(0)项目集规范族及识别所有活前缀的DFA



对于 $I_5$ , 由于 $FOLLOW(E)=FOLLOW(F)=\{ ], ) \}$   
该集合中存在归约-归约冲突, 所以该文法不是SLR(1)文法。

## 课堂练习3

说明下面的文法是LR(1)文法，但不是SLR(1)文法。

$$X \rightarrow Ma \mid bMc \mid dc \mid bda$$
$$M \rightarrow d$$

# 解答:

首先, 拓广文法

(0)  $S \rightarrow X$

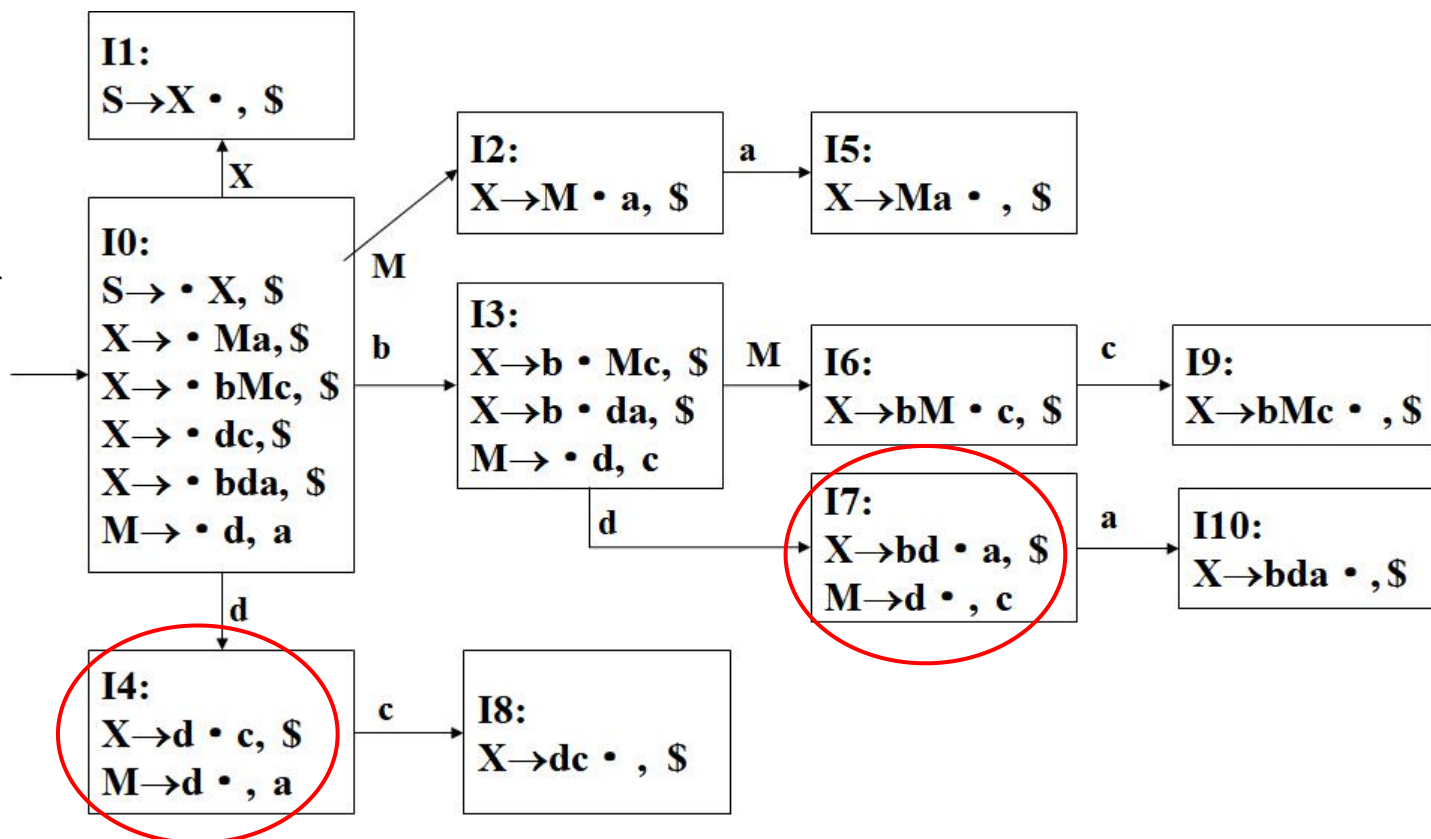
(1)  $X \rightarrow Ma$

(2)  $X \rightarrow bMc$

(3)  $X \rightarrow dc$

(4)  $X \rightarrow bda$

(5)  $M \rightarrow d$



其次, 构造文法的LR(1)项目集规范族及识别其所有活前缀的DFA。

判断该文法是LR(1)文法: 集合I0、I3中没有归约项目, 所以, 不存在冲突;

集合I1、I2、I5、I6、I8、I9、I10各只有一个归约项目, 所以这些集合中没有冲突;

集合I4和I7中既有移进项目又有归约项目, 但是归约符号和移进符号不同, 所以也没有冲突。

结论: 是LR(1)

然后, 构造文法的LR(0)项目集规范族及识别其所有活前缀的DFA。

$FOLLOW(S) = \{ \$ \}$   
 $FOLLOW(X) = \{ \$ \}$   
 $FOLLOW(M) = \{ a, c \}$

I4、I7中存在移进-归约冲突,  $FOLLOW(M) = \{ a, c \}$   
这种冲突用SLR(1)方法无法解决,  
所以该文法不是SLR(1)文法。

# 课堂练习4

已知文法G[A]为：

$$A \rightarrow aABe|a$$

$$B \rightarrow Bb|d$$

$$A \rightarrow aA'$$

$$A' \rightarrow ABe \mid \varepsilon$$

$$B \rightarrow dB'$$

$$B' \rightarrow bB' \mid \varepsilon$$

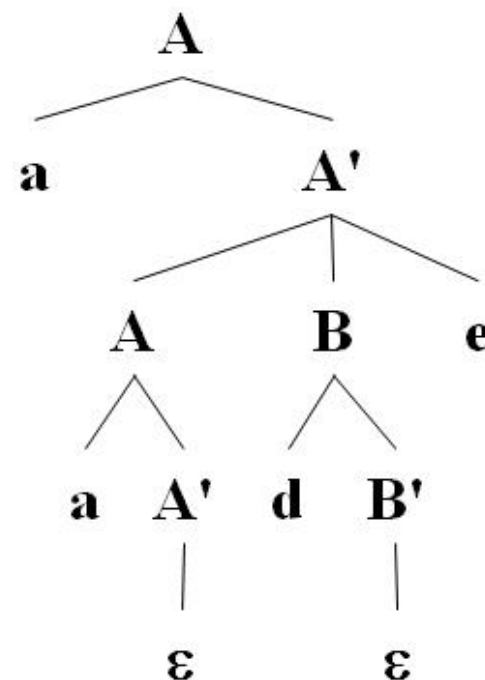
	First	Follow
A	a	\$, d
A'	a, $\varepsilon$	\$, d
B	d	e
B'	b, $\varepsilon$	e

(1) 试给出与G[A]等价的LL(1)文法G'[A]

(2) 构造G'[A]的预测分析表

(3) 给出输入串aade的分析过程。

	a	b	d	e	\$
A	$A \rightarrow aA'$				
A'	$A' \rightarrow ABe$		$A' \rightarrow \varepsilon$		$A' \rightarrow \varepsilon$
B			$B \rightarrow dB'$		
B'		$B' \rightarrow bB'$		$B' \rightarrow \varepsilon$	



# aade的分析过程

	a	b	d	e	\$
A	$A \rightarrow aA'$				
A'	$A' \rightarrow ABe$		$A' \rightarrow \epsilon$		$A' \rightarrow \epsilon$
B			$B \rightarrow dB'$		
B'		$B' \rightarrow bB'$		$B' \rightarrow \epsilon$	

步骤	栈	输入	分析动作
(1)	\$A	aade\$	$A \rightarrow aA'$
(2)	\$A'a	aade\$	
(3)	\$A'	ade\$	$A' \rightarrow ABe$
(4)	\$eBA	ade\$	$A \rightarrow aA'$
(5)	\$eBA'a	ade\$	
(6)	\$eBA'	de\$	$A' \rightarrow \epsilon$
(7)	\$eB	de\$	$B \rightarrow dB'$
(8)	\$eB'd	de\$	
(9)	\$eB'	e\$	$B' \rightarrow \epsilon$
(10)	\$e	e\$	
(11)	\$	\$	分析成功

# 课堂练习5

有如下文法G[A]:

$A \rightarrow BA \mid a$

$B \rightarrow aB \mid b$

(1) 判断该文法是以下哪些类型的文法，要求给出判断过程。

LL(1)、LR(0)、SLR(1)

(2) 构造该文法的LR(1)项目集规范族及识别其所有活前缀的DFA。

(3) 构造该文法的LR(1)分析表

(4) 给出对输入符号串abb的分析过程。

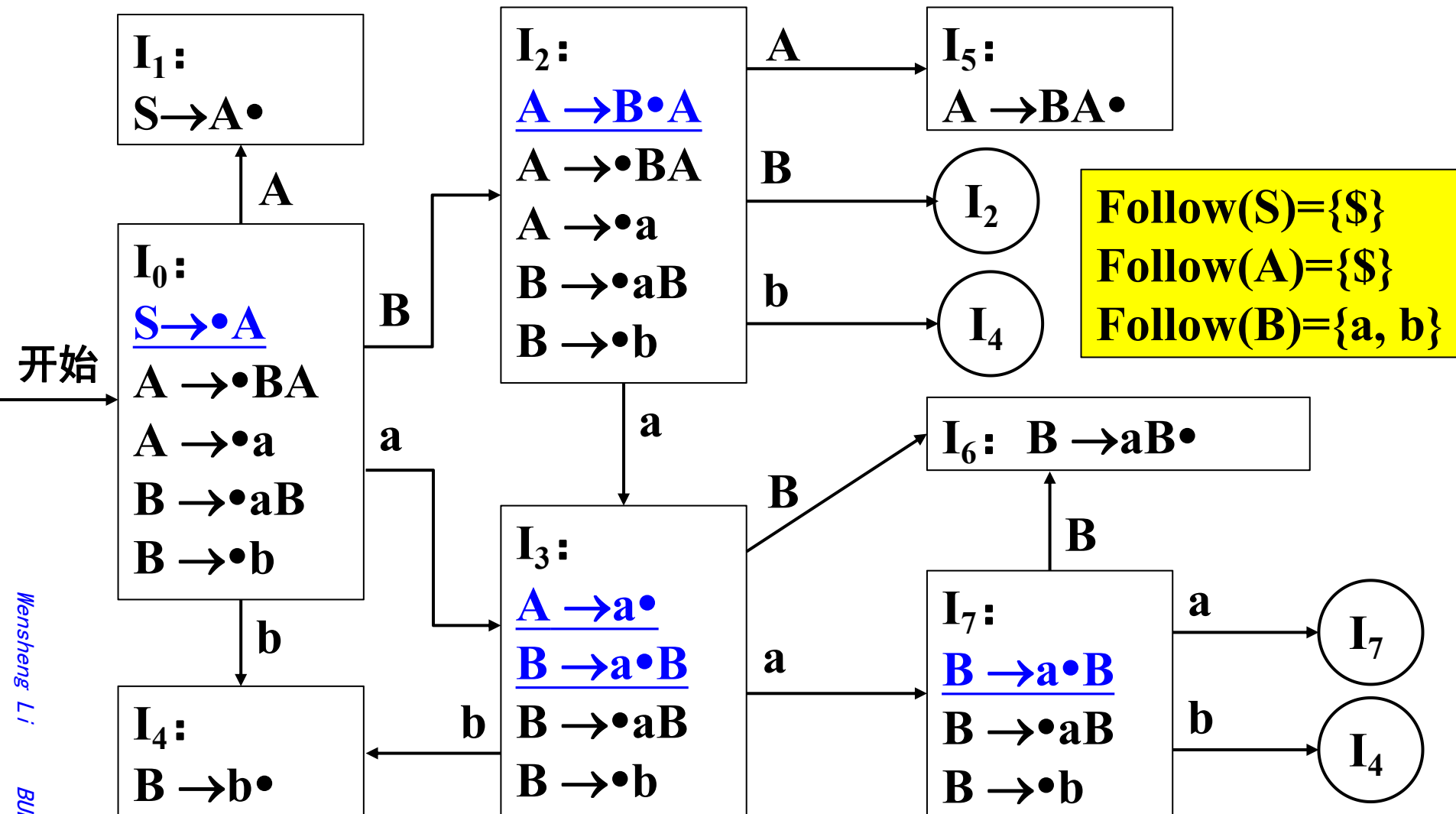


LR(0)文法，项目集中：

- (1) 要么所有元素都是移进-待约项目
- (2) 要么只含有唯一的归约项目

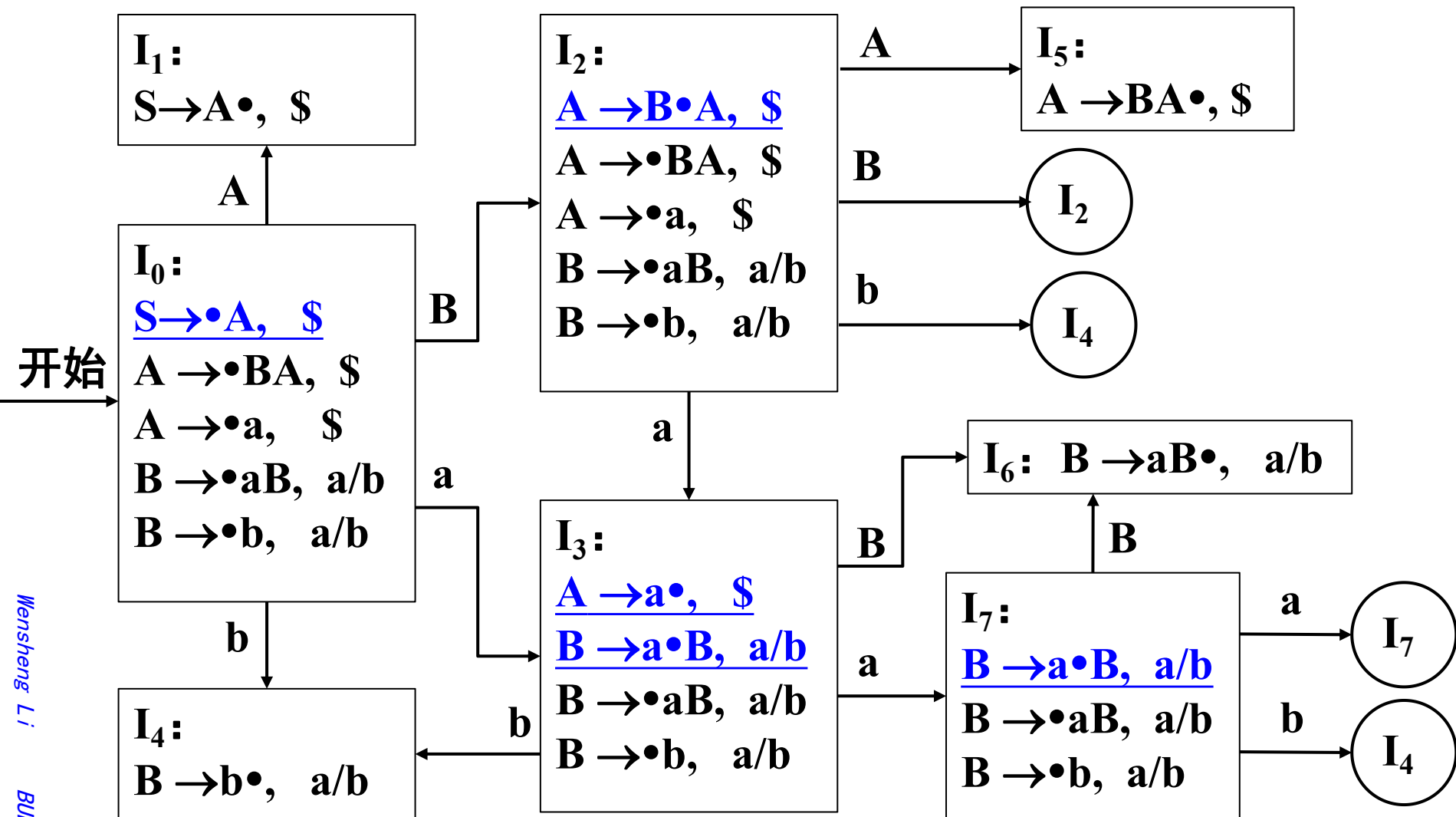
- (0)  $S \rightarrow A$
- (1)  $A \rightarrow BA$
- (2)  $A \rightarrow a$
- (3)  $B \rightarrow aB$
- (4)  $B \rightarrow b$

LR(0)项目集规范族及识别其所有活前缀的DFA:



# 参考答案（续）

LR(1)项目集规范族及识别其所有活前缀的DFA:



# 文法的LR(1)分析表

状态	Action			goto	
	a	b	\$	A	B
0	S3	S4		1	2
1			ACC		
2	S3	S4		5	2
3	S7	S4			6
4	R4	R4			
5			R1		
6	R3	R3			
7	S7	S4			6

# abb的分析过程

步骤	栈	输入	分析动作
(1)	0	abb\$	S3
(2)	0 3 - A	bb\$	S4
(3)	0 3 4 - a b	b\$	R4 B→b
(4)	0 3 6 - a B	b\$	R3 B→aB
(5)	0 2 - B	b\$	S4
(6)	0 2 4 - B b	\$	error 弹出栈顶状态4
(7)	0 2 - B	\$	goto(2, A)=5 将状态5压入栈顶
(8)	0 2 5 - B A	\$	R1 A→BA
(9)	0 1 - A	\$	accept