



编译原理

朱雪峰 博士

计算机科学与技术系

Tel: 89733787(O)

Email: xuefeng.zhu@cup.edu.cn



第六题（P36 第11题）

解：分析 L_2 ，要求b和c的个数一样多，因此可以使用一个非终结符去生成 $b^n c^n$ 串，而用另外一个非终结符去生成 a^i ，因此，可以模拟 L_1 使用一个非终结符去生成 $b^n c^n$ ，而用另外一个非终结符去生成 a^i 。

L_2 的文法：

$$S \rightarrow AB$$

$$A \rightarrow aA \mid \varepsilon$$

$$B \rightarrow bBc \mid bc$$



第六题（P36 第11题）

解：分析 L_3 ，可以将 $a^n b^n a^m b^m$ 分成两段考虑，即 $a^n b^n$ 和 $a^m b^m$ ，然后使用两个非终结符分别去产生它们。

L_3 的文法：

$$S \rightarrow AB$$

$$A \rightarrow aAb \mid \varepsilon$$

$$B \rightarrow aBb \mid \varepsilon$$



第六题（ P36 第11题）

解： L_4 不能采用分段处理的方式，它要求中间的0和1的个数相同，而且一前一后的1和0的个数相同。对于这种文法我们可以采用从里向外扩展的方式进行，即先用一个非终结符生成处于中间的 m 个0和 m 个1，然后，使用另外一个非终结符在该串的基础上扩充前后的 n 个1和 n 个0。

L_4 :

$$A \rightarrow 0A1 \mid \varepsilon$$

$$S \rightarrow 1S0 \mid A$$



第七题 (P64 第7题)

7. 构造下列正规式相应的DFA

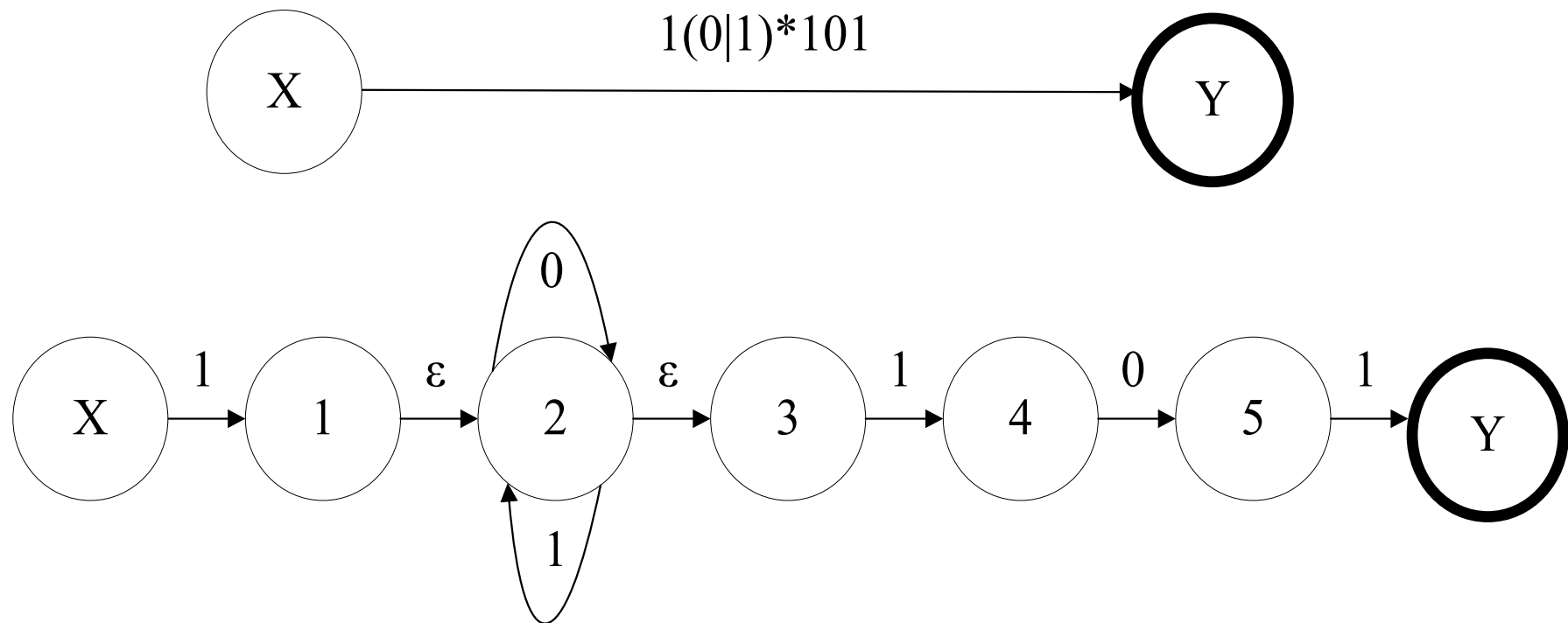
① $1(0|1)^*101$

② $1(1010^*|1(010)^*1)^*0$

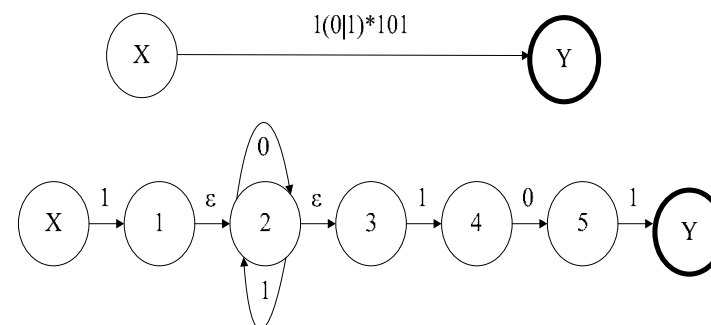
③ $0^*10^*10^*10^*$

④ $(00|11)^*((01|10)(00|11)^*(01|10)(00|11)^*)^*$

第七题 (P64 第7题①)



第七题 (P64 第7题①)

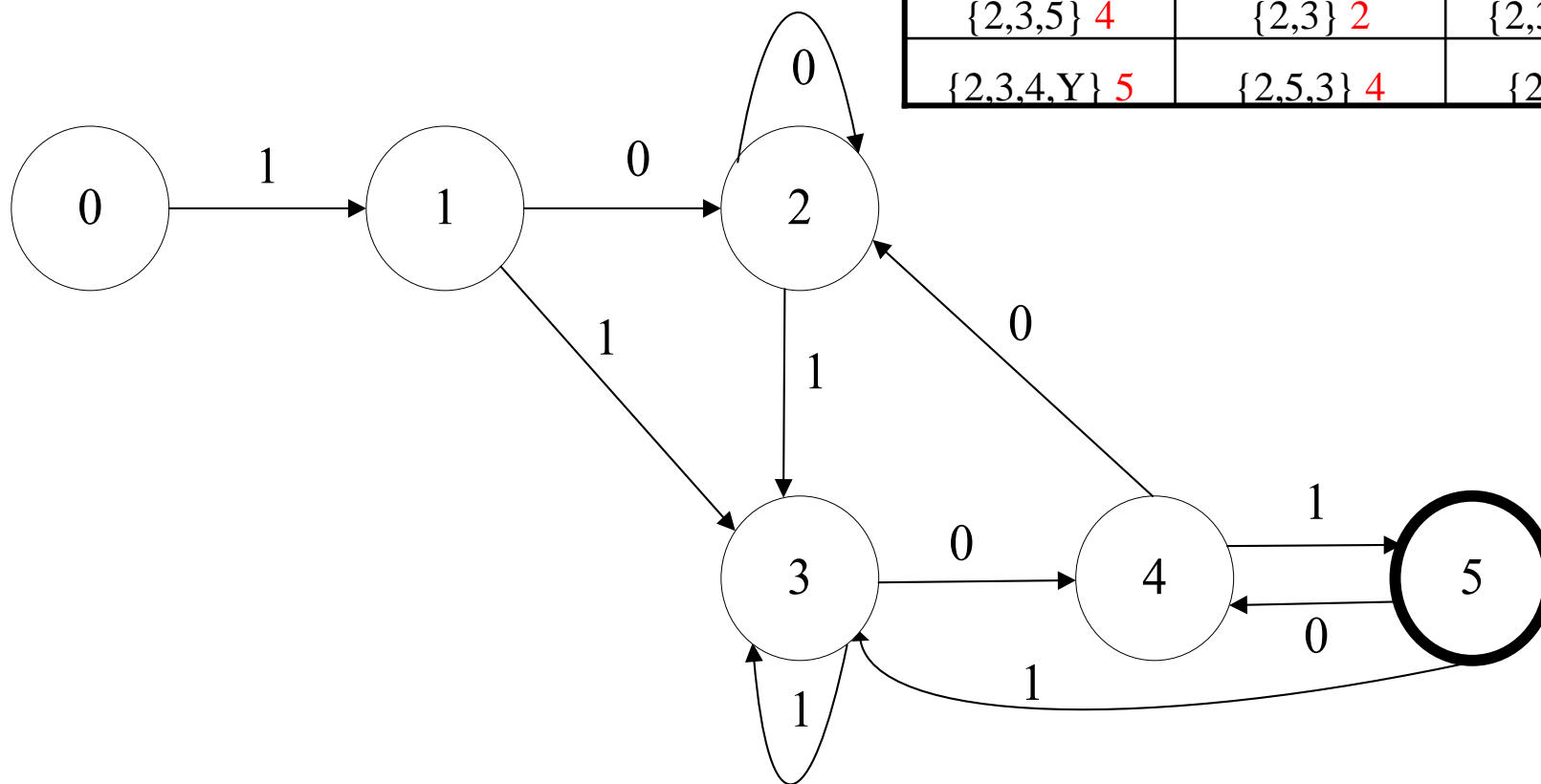


状态	0	1
{X} 0	\emptyset	{1,2,3} 1
{1,2,3} 1	{2,3} 2	{2,3,4} 3
{2,3} 2	{2,3} 2	{2,3,4} 3
{2,3,4} 3	{2,5,3} 4	{2,3,4} 3
{2,3,5} 4	{2,3} 2	{2,3,4,Y} 5
{2,3,4,Y} 5	{2,5,3} 4	{2,4,3} 3



第七题（ P64 第7题① ）

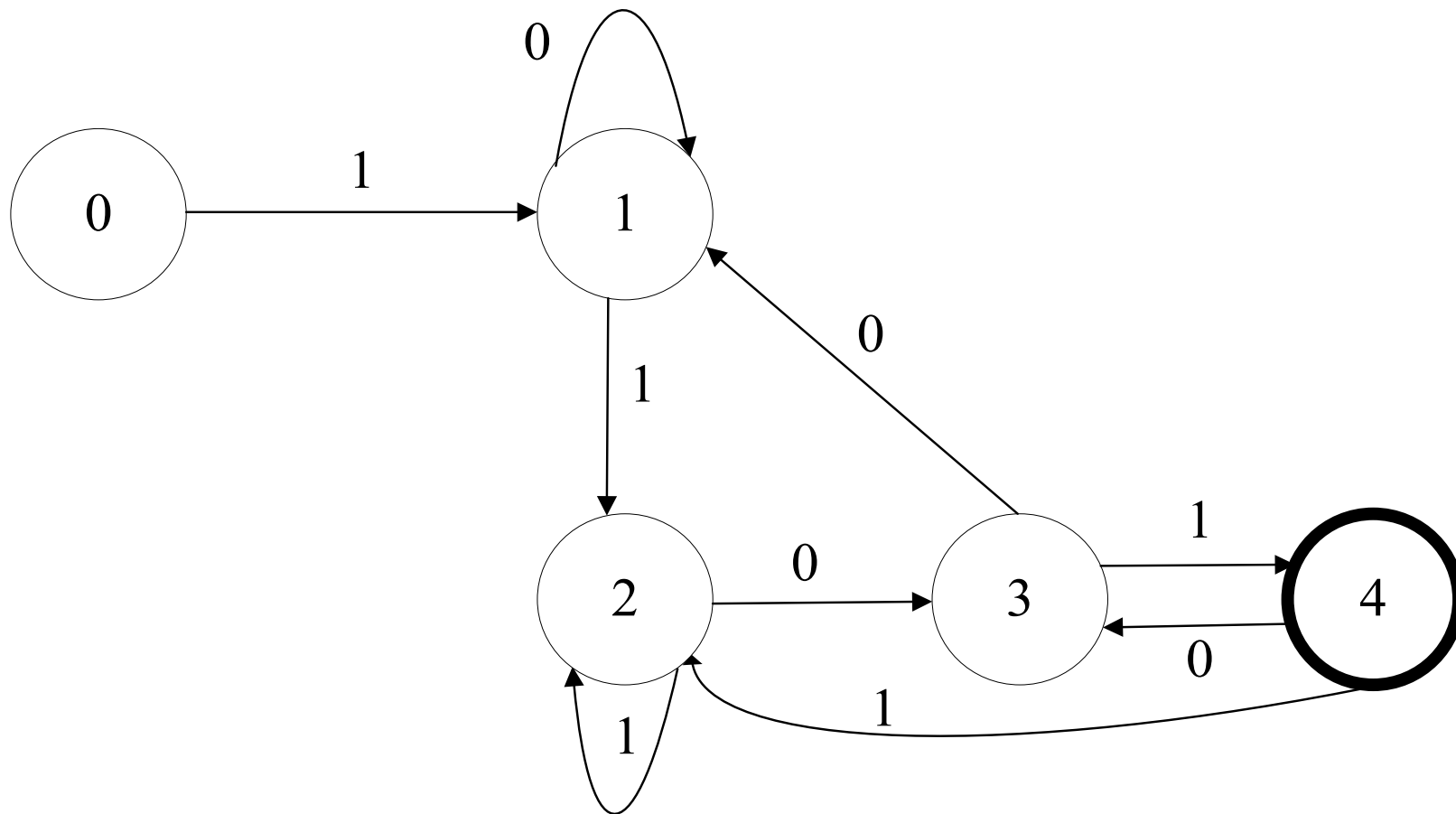
状态	0	1
{X} 0	\emptyset	{1,2,3} 1
{1,2,3} 1	{2,3} 2	{2,3,4} 3
{2,3} 2	{2,3} 2	{2,3,4} 3
{2,3,4} 3	{2,5,3} 4	{2,3,4} 3
{2,3,5} 4	{2,3} 2	{2,3,4,Y} 5
{2,3,4,Y} 5	{2,5,3} 4	{2,4,3} 3



第七题 (P64 第7题①)

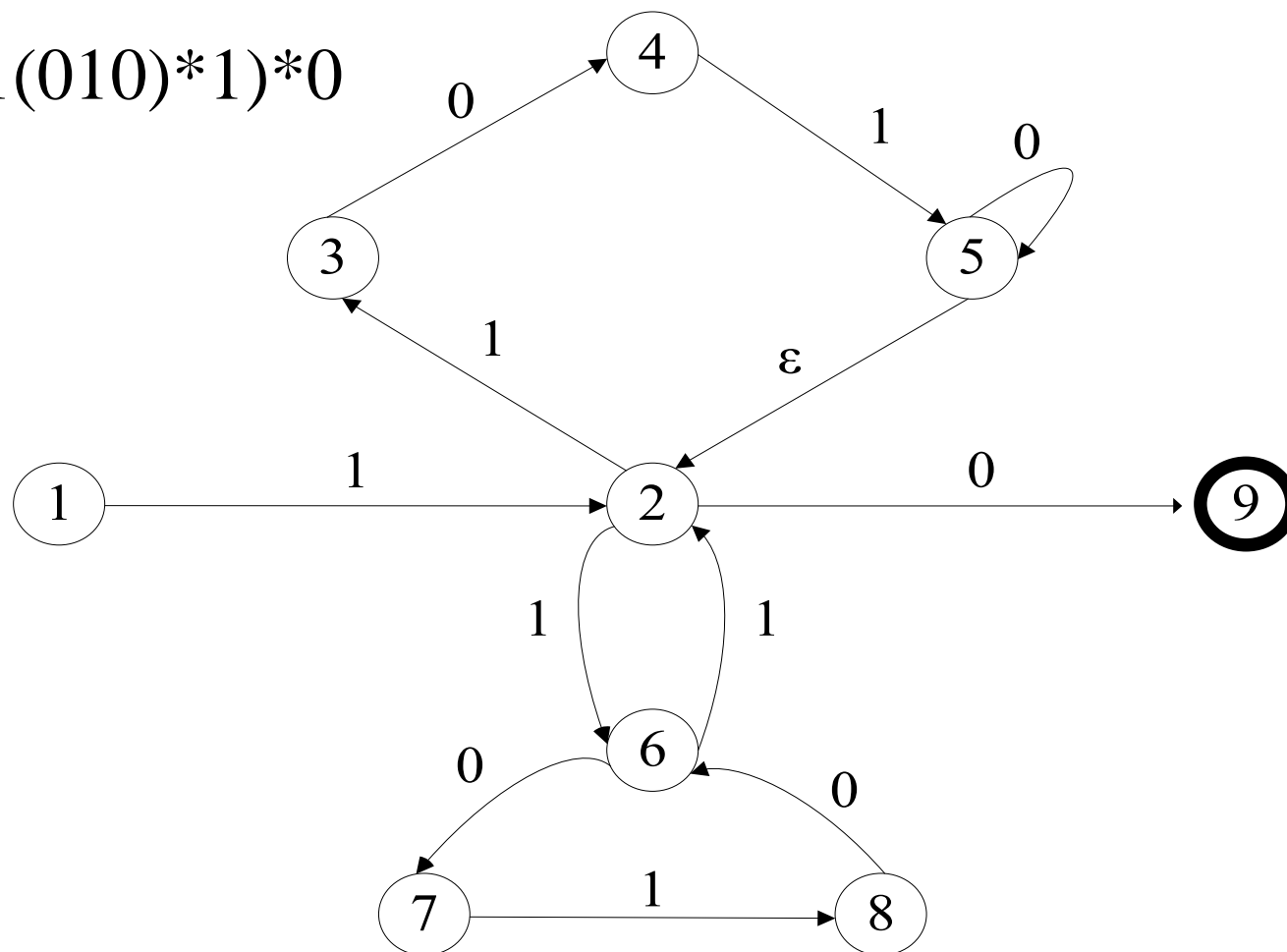
- 田 初始划分: $\{\{0,1,2,3,4\},\{5\}\}$, $\{0,1,2,3,4\}_0=\{2,4,_ \}$, $\{0,1,2,3,4\}_1=\{1,3,5\}$ 。由于0不能接受字符0, 需要把状态0划分出来; 另外, 只有状态4读入字符1后进入状态5, 因此将状态4划分出来, 划分的结果为4个集合: $\{\{0\},\{1,2,3\},\{4\},\{5\}\}$ 。
- 田 对于集合 $\{1,2,3\}$, 由于 $\{1,2,3\}_0=\{2,4\}$, 因此需要对 $\{1,2,3\}$ 进一步划分, 划分结果为5个集合: $\{\{0\},\{1,2\},\{3\},\{4\},\{5\}\}$ 。检查集合 $\{1,2\}$, 由于 $\{1,2\}_0=\{2\},\{1,2\}_1=\{3\}$, 不需要进一步的划分。所以最终划分结果为5个集合: $\{\{0\},\{1,2\},\{3\},\{4\},\{5\}\}$ 。

第七题（ P64 第7题①）



第七题 (P64 第7题②)

$1(1010^*|1(010)^*1)^*0$

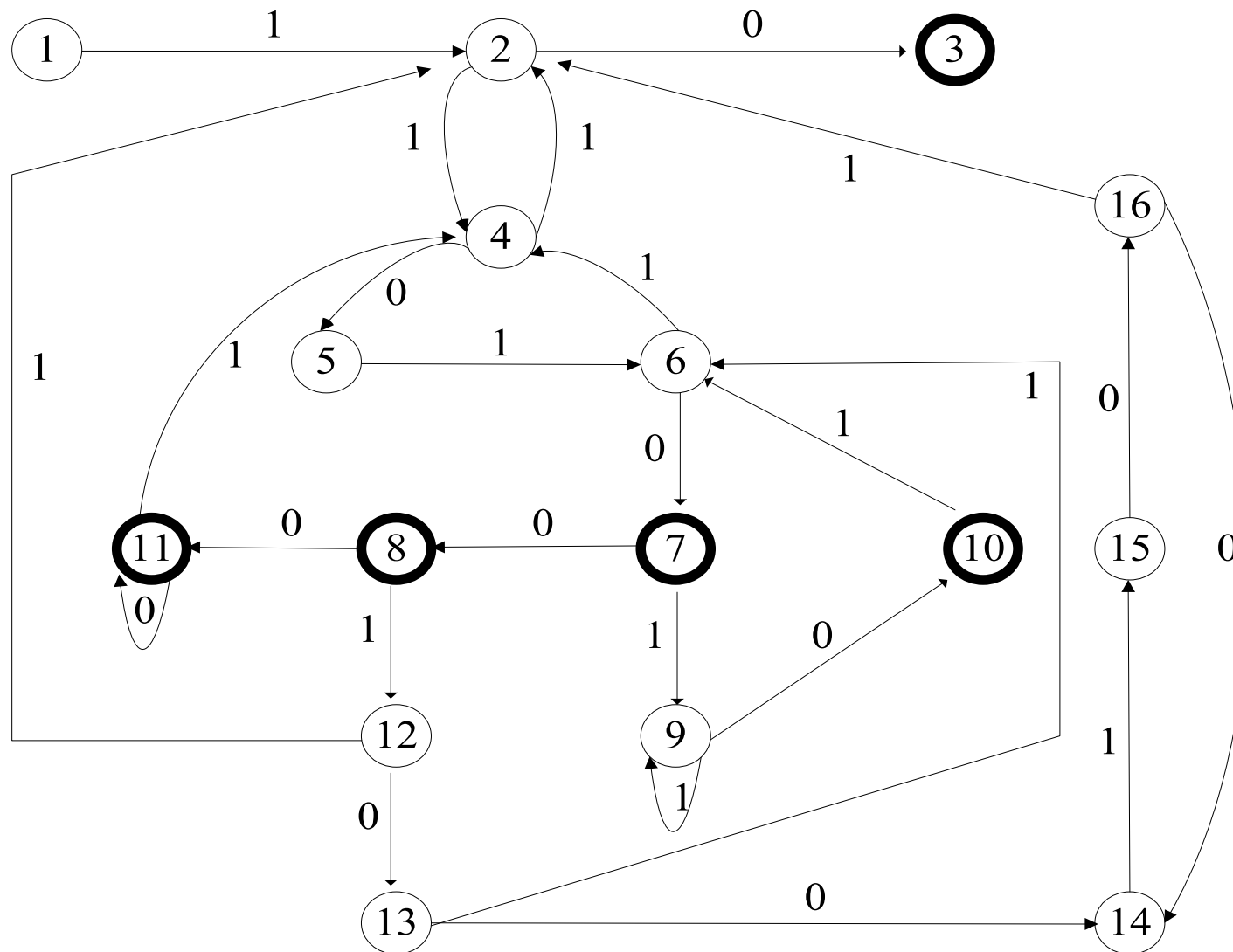


第七题（P64 第7题②）

状态	0	1
{1}	\emptyset	{2}
{2}	{9}	{3,6}
{9}	\emptyset	\emptyset
{3,6}	{4,7}	{2}
{4,7}	\emptyset	{5,8,2}
{5,8,2}	{5,2,6,9}	{3,6}
{5,2,6,9}	{5,2,7,9}	{3,6,2}
{5,2,7,9}	{5,2,9}	{3,6,8}
{3,6,2}	{4,7,9}	{3,6,2}

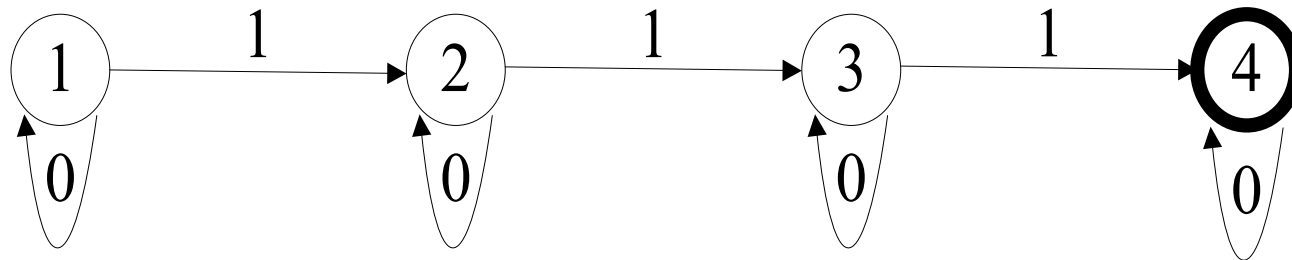
状态	0	1
{4,7,9}	\emptyset	{5,8,2}
{5,2,9}	{5,2,9}	{3,6}
{3,6,8}	{4,7,6}	{2}
{4,7,6}	{7}	{5,8,2}
{7}	\emptyset	{8}
{8}	{6}	\emptyset
{6}	{7}	{2}

第七题 (P64 第7题②)



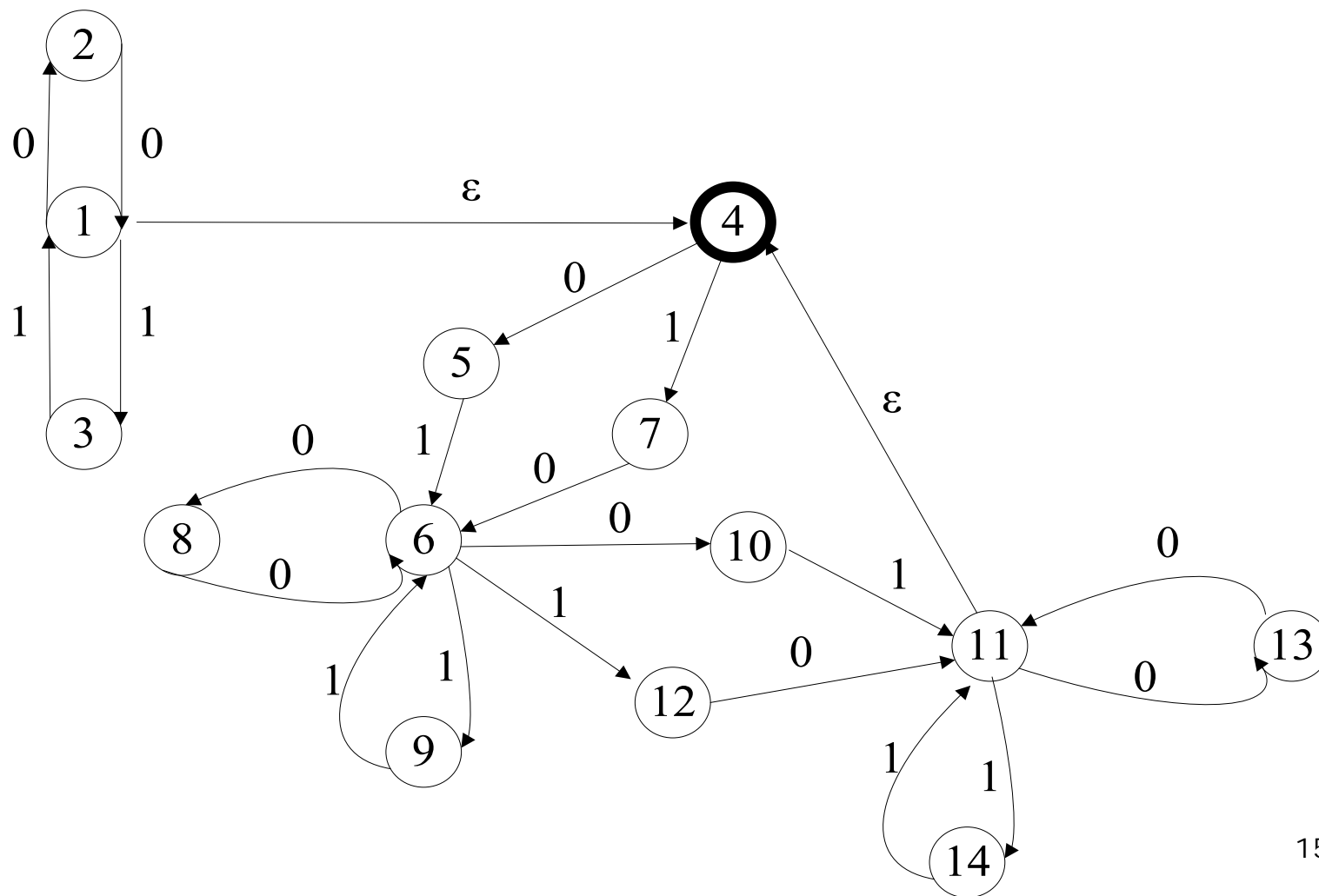
第七题 (P64 第7题③)

$0^*10^*10^*10^*$



第七题 (P64 第7题④)

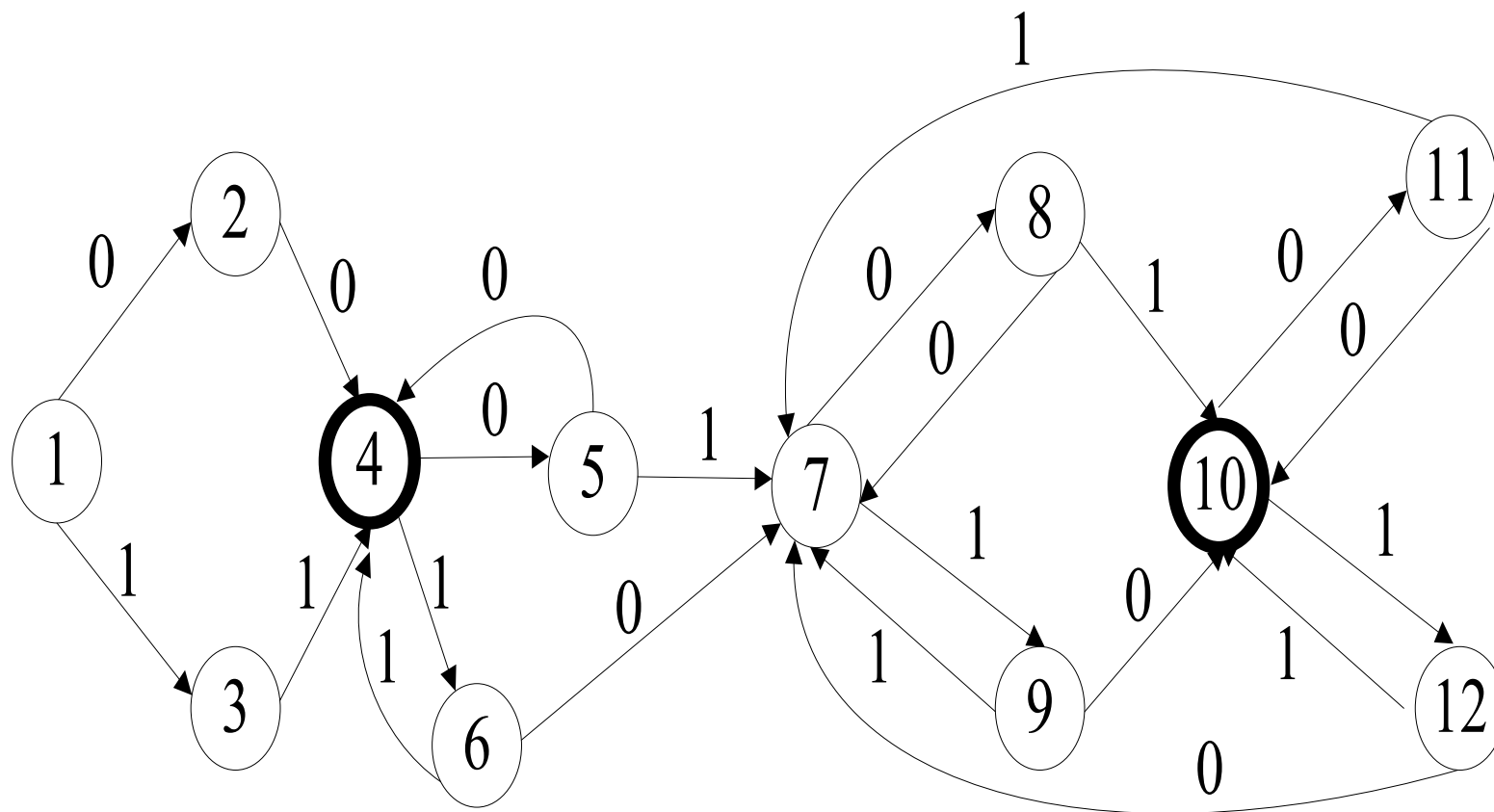
$(00|11)^*((01|10)(00|11)^*(01|10)(00|11)^*)^*$



第七题（P64 第7题④）

状态	0	1
{1}	{2}	{3}
{2}	{1,4}	\emptyset
{3}	\emptyset	{1,4}
{1,4}	{2,5}	{3,7}
{2,5}	{1,4}	{6}
{3,7}	{6}	{1,4}
{6}	{8,10}	{9,12}
{8,10}	{6}	{11,4}
{9,12}	{11,4}	{6}
{11,4}	{13,5}	{14,7}
{5,13}	{11,4}	{6}
{14,7}	{6}	{11,4}

第七题（ P64 第7题④）





第八题（P64 第8题）

8. 给出下面正规表达式：

- ① 以01结尾的二进制数串；
- ② 能被5整除的十进制整数；
- ③ 包含奇数个1或奇数个0的二进制数串；
- ④ 英文字母组成的所有符号串，要求符号串中的字母依照字典序排列；
- ⑤ 没有重复出现的数字的数字符号串的全体；
- ⑥ 最多有一个重复出现的数字的数字符号串的全体；
- ⑦ 不包含子串abb的由a和b组成的符号串的全体。



第八题（P64 第8题）

解：

- (1) 本题要求的是二进制串，即由0和1构成的串，并且必须以01结尾，所以本题可以分两部分去完成，一部分实现由0和1构成的任意串，一部分即01，然后将它们连接到一起就可以了，因此答案为 $(1|0)^*01$ 。



第八题（P64 第8题）

(2) 本题要求的是十进制整数，也就是由0~9这10个数字组成的字符串，并且不能以0开头（整数0除外），要求能够被5整除，则该串必须以0或5结尾。可以分成两种情况：一种情况是该整数只有1位，则该整数有0和5两种可能；另一种情况是该整数有多位，则该整数可以分为3部分来考虑。一是第一位必须不为0，二是最后一位必须为0或5，三是中间部分可有可无，并且可以由0~9之间任意数字构成，所以本题的正规表达式为：

$$(1|2|3|4|5|6|7|8|9)(0|1|2|3|4|5|6|7|8|9)^*(0|5) \mid (0|5)$$

第八题（P64 第8题）

- (3) 由于0和1在二进制串中的奇数出现具有对偶性，因此我们只考虑一种情况，另一种情况可以类似求得，考虑包含奇数个0的字符串：由于只关心0的个数的奇偶性，我们可以把二进制串分成多段来考虑，第一段为二进制串的开始到第一个0为止，这一段包含1个0，并且0的前面有0个或多个1；对于剩下的二进制串按照每段包含两个0的方式去划分，即以0开始，以0结尾，中间可以有0个或多个1.如果一个二进制串被这样划分完后，剩下的部分如果全部是全1串（这些全1串在前面划分的串之间或最后），则该二进制串就具有奇数个0，所以该二进制串可以这样描述：以第一段（ 1^*0 ）开始，后面由全1串（ 1^* ）以及包含两个0的串（ 01^*0 ）组成，所以包含奇数个0的正规式为： $1^*0(1|01^*0)^*$ 。本题答案为
- $$1^*0(1|01^*0)^* | 0^*1(0|10^*1)^*$$



第八题（ P64 第8题）

(4)

$$(a|A)^*(b|B)^*\cdots(z|Z)^*$$



第八题 (P64 第8题)

(5)

令 $r_i = i \mid \varepsilon \quad (i=0,1,2,\dots,9)$

$R_0|R_1|R_2|\dots|R_9$ 记为 $\sum_{i \in (0,1,2,\dots,9)} R_i$

令 $P(0,1,2,\dots,9)$ 表示 $i=0,1,2,\dots,9$ 的全排列

$\sum r_{i_0} r_{i_1} \cdots r_{i_9} \quad \text{其中 } i_0, i_1, \dots, i_9 \in P(0,1,\dots,9)$



第八题 (P64 第8题)

(6)

$$\sum_{i \in (0,1,2,\dots,9)} i \sum r_{i_0} r_{i_1} \cdots r_{i_9} \quad \text{其中 } i_0, i_1, \dots, i_9 \in P(0,1,\dots,9)$$



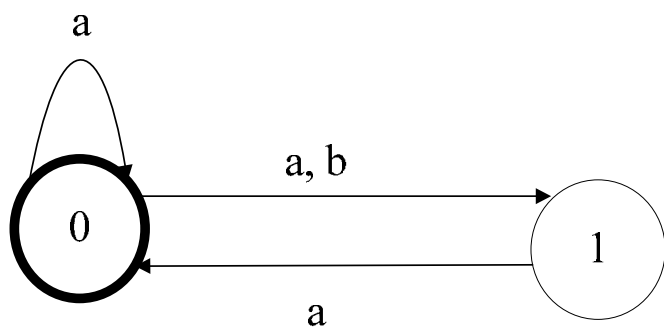
第八题（ P64 第8题）

(7) 直接写出满足条件的正规表达式。

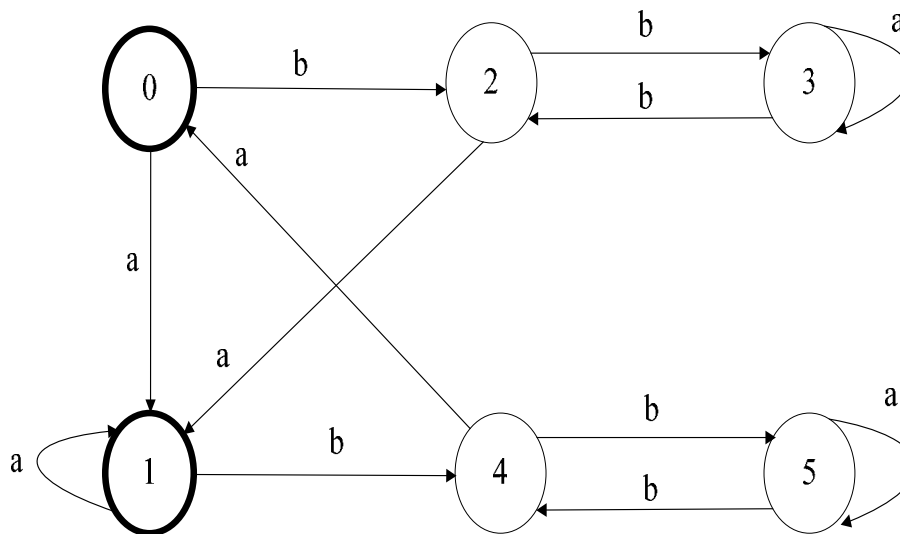
考虑满足条件的字符串：在串的开始部分可以有0个或多个b，但是一旦a出现以后，要么后面还是a，要么后面只能跟一个b然后还得跟a，由此我们得到满足条件的正规式为： $b^*(a|ab)^*$

第九题（P64 第12题）

12. 将图3.18的(a)和(b)分别确定化和最小化。

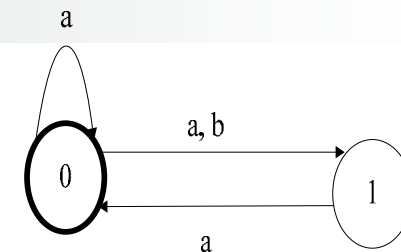


(a) 需确定化的有限自动机



(b) 需最小化的有限自动机

第九题（P64 第12题）



(a) 需确定化的有限自动机

解：图(a)中为一个NFA，所以需要先将其确定化。得到DFA，然后再将其最小化。

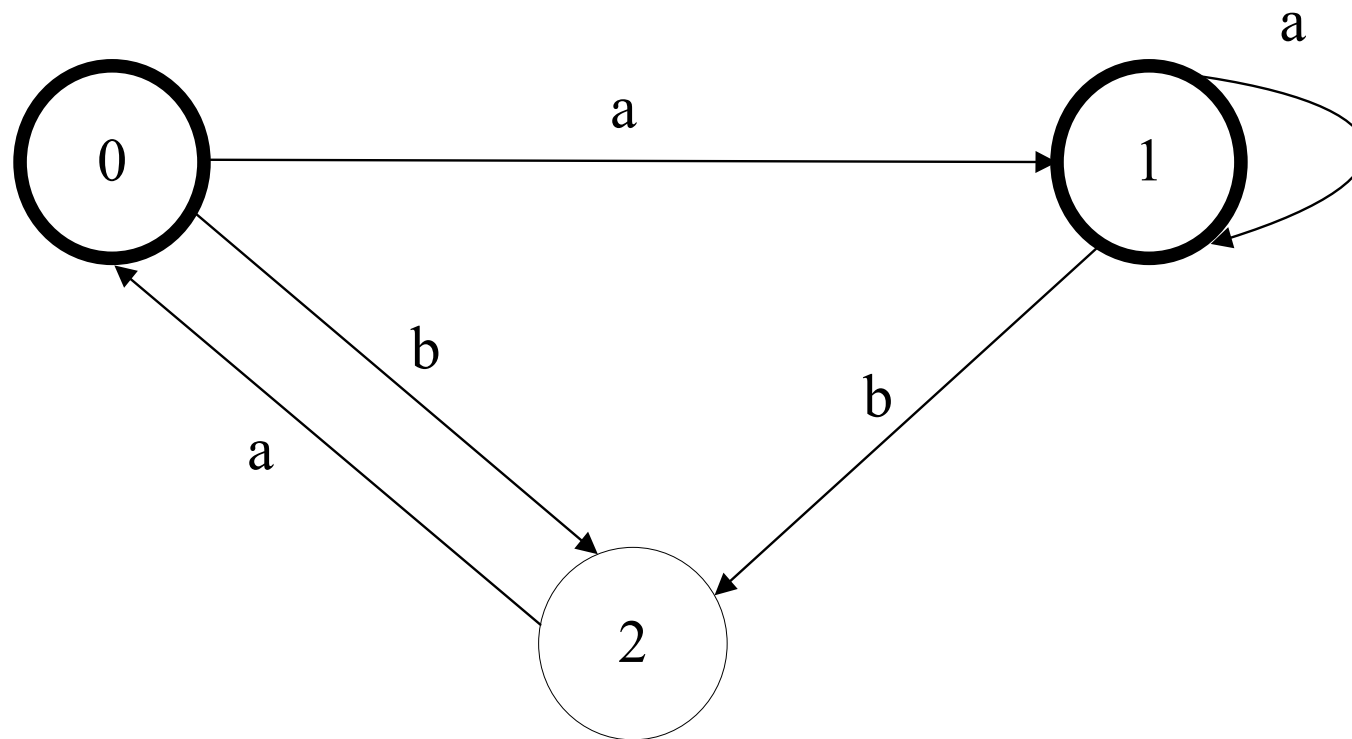
第一步：确定化，得到DFA。确定化的结果见左表所列。给状态编号，得到右表所列的状态转换矩阵。

状态	a	b
{0}	{0,1}	{1}
{0,1}	{0,1}	{1}
{1}	{0}	\emptyset

状态	a	b
0	1	2
1	1	2
2	0	

第九题（P64 第12题）

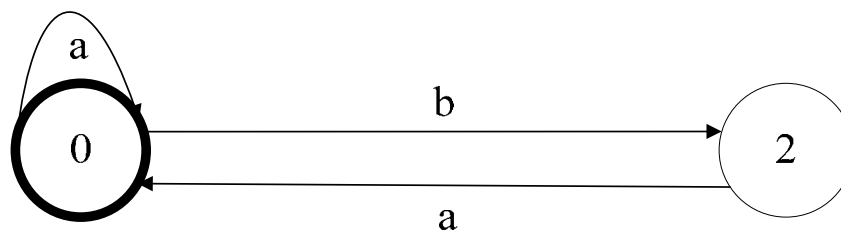
解：根据状态转换矩阵得到如图所示的DFA



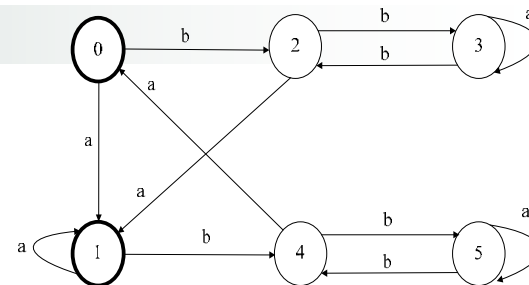
第九题（P64 第12题）

解：第二步：最小化。首先将状态划分成两个集合 $\{0,1\}$ 和 $\{2\}$ ，由于 $\{0,1\}_a=\{1\}$ 、 $\{0,1\}_b=\{2\}$ ，所以 $\{0,1\}$ 和 $\{2\}$ 就是最后的分划，因此得到如下图所示的最小化的DFA

。



第九题（P64 第12题）



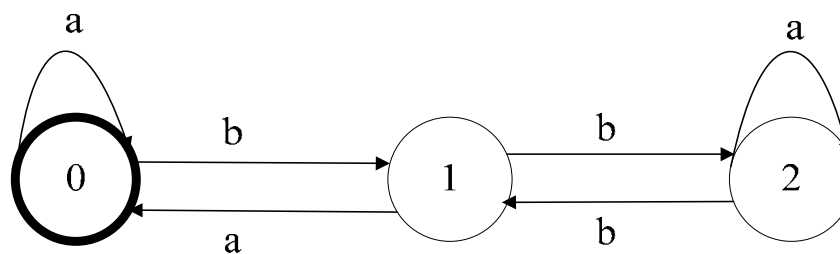
(b) 需最小化的有限自动机

解：图(b)中所示为一个DFA，不需要确定化，因此只需最小化即可。

首先将状态划分为两个集合 $\{\{0,1\},\{2,3,4,5\}\}$ ，由于 $\{0,1\}_a=\{1\}$ 、 $\{0,1\}_b=\{2,4\}$ 和 $\{2,3,4,5\}_a=\{1,3,0,5\}$ 、 $\{2,3,4,5\}_b=\{2,3,4,5\}$ ，所以可以进一步划分为 $\{\{0,1\},\{2,4\},\{3,5\}\}$ 。然后有 $\{0,1\}_a=\{1\}$ 、 $\{0,1\}_b=\{2,4\}$ 、 $\{2,4\}_a=\{1,0\}$ 、 $\{2,4\}_b=\{3,5\}$ 、 $\{3,5\}_a=\{3,5\}$ 、 $\{3,5\}_b=\{2,4\}$ ，因此，最后划分结果就是 $\{\{0,1\},\{2,4\},\{3,5\}\}$

第九题（ P64 第12题）

解：最小化后的DFA如下图所示：





第十题（ P65 第14题）

14. 构造一个DFA，它接受 $\Sigma=\{0,1\}$ 上所有满足如下条件的字符串：每个1都有0直接跟在右边。



第十题（P65 第14题）

第一步：写出正规表达式。根据题意，该DFA接受的字符串由0和1组成，并且每个1的后面都有0直接跟在右边，因此，可以将该字符串理解为由0和10构成的串，则相应的正规表达式就是 $(0|10)^*$

第二步：构造NFA。首先得出图1。然后按照P50的替换规则对图1进行分解后得到如图2所示的NFA。

第十题（ P65 第14题）

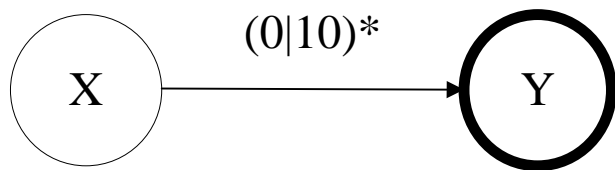


图1 初始NFA

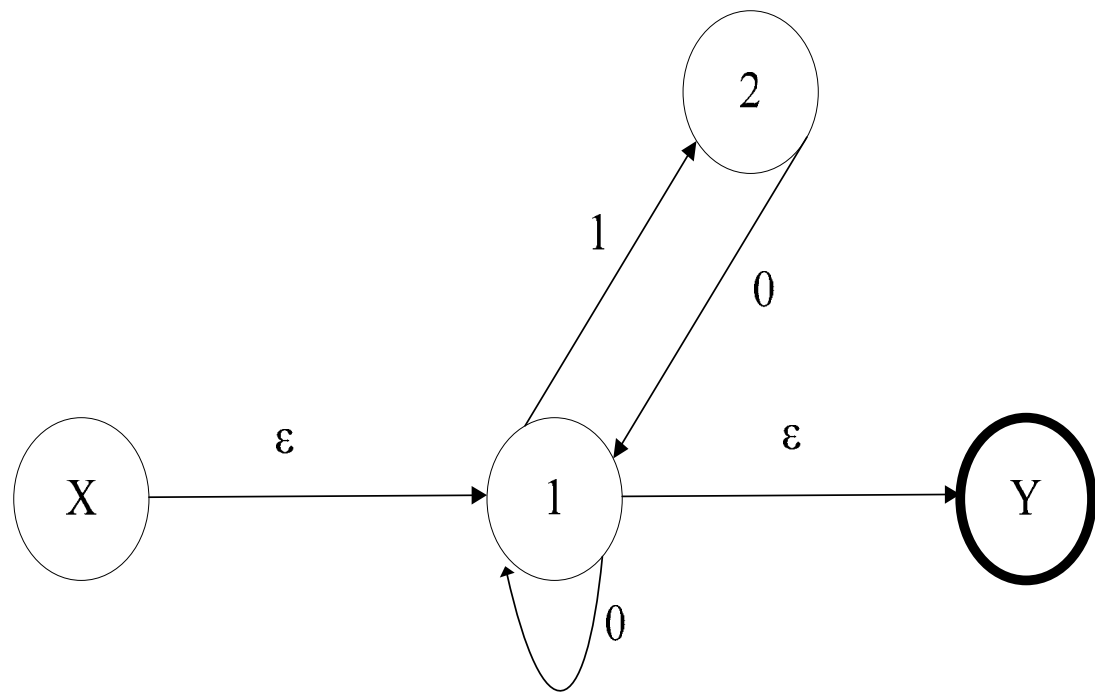


图2 转换后的NFA

第十题（P65 第14题）

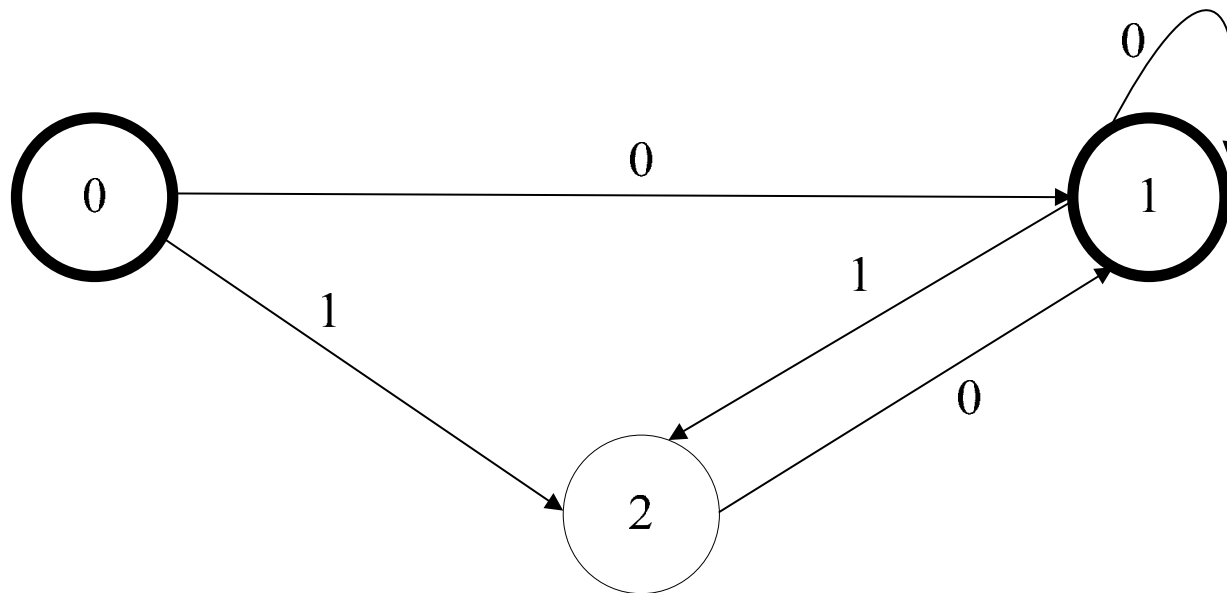
第三步：确定化，得到DFA。确定化的结果见左表所列，给状态编号，得到如右表所列的状态转换矩阵。

状态	0	1
{X,1,Y}	{1,Y}	{2}
{1,Y}	{1,Y}	{2}
{2}	{1,Y}	\emptyset

状态	0	1
0	1	2
1	1	2
2	1	

第十题（P65 第14题）

根据状态转换矩阵得到的DFA如下图所示：



确定化的DFA

第十题（P65 第14题）

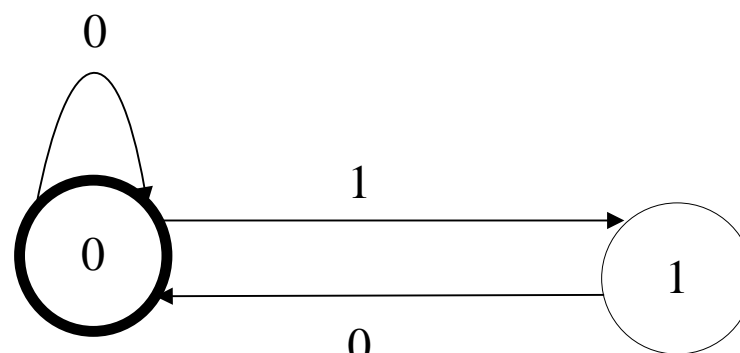
第四步，使用P57的方法对该
DFA进行最小化。最小化分
析过程如下：

初始分划： $\{0,1\}$ 、 $\{2\}$

由于 $\{0,1\}_0 = \{1\}$,

$\{0,1\}_1 = \{2\}$

最小化后的DFA如下图所示，
该DFA即为所求。



最小化的DFA



第十一题（ P81 第1题）

1. 考虑下面文法 G_1 :

$$S \rightarrow a \mid \wedge \mid (T)$$

$$T \rightarrow T, S \mid S$$

- ① 消去 G_1 的左递归。然后，对每个非终结符，写出不带回溯的递归子程序。
- ② 经改写后的文法是否是LL(1)的？给出它的预测分析表。



第十一题（P81 第1题）

解：（1）按照T、S的顺序消除左递归，得到文法：

$G'(S):$

$S \rightarrow a \mid \wedge \mid (T)$

$T \rightarrow ST'$

$T' \rightarrow , S T' \mid \varepsilon$

对每个非终结符，写出不带回溯的递归子程序（略，不讲）

第十一题 (P81 第1题)

(2) 首先计算每个非终结符的FIRST集合和FOLLOW集合
(P78和P79页计算FIRST和FOLLOW的算法)

$\text{FIRST}(S) = \{a, \Lambda, (\}$

$\text{FIRST}(T) = \{a, \Lambda, (\}$

$\text{FIRST}(T') = \{, , \varepsilon \}$

$\text{FOLLOW}(S) = \{ \# , , ,) \}$

$\text{FOLLOW}(T) = \{) \}$

$\text{FOLLOW}(T') = \{) \}$

$\# \in \text{FOLLOW}(S)$

$) \in \text{FOLLOW}(T)$

$\text{FIRST}(T') / \{ \varepsilon \} \subseteq \text{FOLLOW}(S)$

$\text{FOLLOW}(T) \subseteq \text{FOLLOW}(T')$

$\text{FOLLOW}(T) \subseteq \text{FOLLOW}(S)$

第十一题（P81 第1题）

(2) 构造预测分析表如下：P79页构造分析表 M 的算法

	a	\wedge	()	,	#
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'				$T \rightarrow \varepsilon$	$T \rightarrow ,ST'$	

从表中可以看出没有多重入口，所以改造后的文法是LL(1)文法

第十二题（P81 第2题）

2. 对下面的文法G:

$$E \rightarrow TE'$$

$$E' \rightarrow +E \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow T \mid \varepsilon$$

$$F \rightarrow PF'$$

$$F' \rightarrow *F' \mid \varepsilon$$

$$P \rightarrow (E) \mid a \mid b \mid \wedge$$

- ① 计算这个文法的每个非终结符的FIRST和FOLLOW。
- ② 证明这个文法是LL(1)的。
- ③ 构造它的预测分析表。
- ④ 构造它的递归下降分析程序。

第十二题（P81 第2题）

(1) 计算每个非终结符的FIRST集合和FOLLOW集合：

$$\text{FIRST}(E) = \{ (, a, b, \Lambda \}$$

$$\text{FIRST}(E') = \{ +, \varepsilon \}$$

$$\text{FIRST}(T) = \{ (, a, b, \Lambda \}$$

$$\text{FIRST}(T') = \{ (, a, b, \Lambda, \varepsilon \}$$

$$\text{FIRST}(F) = \{ (, a, b, \Lambda \}$$

$$\text{FIRST}(F') = \{ *, \varepsilon \}$$

$$\text{FIRST}(P) = \{ (, a, b, \Lambda \}$$

第十二题 (P81 第2题)

田 计算每个非终结符的FIRST集合和FOLLOW集合:

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

$\text{FOLLOW}(E') = \{) , \# \}$

$\text{FOLLOW}(T) = \{ (, + ,) , \# \}$

$\text{FOLLOW}(T') = \{ + ,) , \# \}$

$\text{FOLLOW}(F) = \{ (, a , b , \wedge , + ,) , \# \}$

$\text{FOLLOW}(F') = \{ (, a , b , \wedge , + ,) , \# \}$

$\text{FOLLOW}(P) = \{ * , (, a , b , \wedge , + ,) , \# \}$

$\# \in \text{FOLLOW}(E)$

$) \in \text{FOLLOW}(E)$

$\text{FIRST}(E') / \{ \epsilon \} \subseteq \text{FOLLOW}(T)$

$\text{FIRST}(T') / \{ \epsilon \} \subseteq \text{FOLLOW}(F)$

$\text{FIRST}(F') / \{ \epsilon \} \subseteq \text{FOLLOW}(P)$

$\text{FOLLOW}(E) \subseteq \text{FOLLOW}(E')$

$\text{FOLLOW}(E) \subseteq \text{FOLLOW}(T)$

$\text{FOLLOW}(T) \subseteq \text{FOLLOW}(T')$

$\text{FOLLOW}(T) \subseteq \text{FOLLOW}(F)$

$\text{FOLLOW}(F) \subseteq \text{FOLLOW}(F')$

$\text{FOLLOW}(F) \subseteq \text{FOLLOW}(P)$

$\text{FOLLOW}(E') \subseteq \text{FOLLOW}(E)$

$\text{FOLLOW}(T') \subseteq \text{FOLLOW}(T)$



第十二题（ P81 第2题）

（2）根据P73，可以证明这个文法是LL(1)的。或根据第(3)步构造的预测分析表无多重入口，也可证明。

第十二题（P81 第2题）

(3) 构造预测分析表如下：

	+	*	()	a	b	\wedge	#
E			$E \rightarrow TE'$		$E \rightarrow TE'$	$E \rightarrow TE'$	$E \rightarrow TE'$	
E'	$E' \rightarrow +E$			$E' \rightarrow \varepsilon$				$E' \rightarrow \varepsilon$
T			$T \rightarrow FT'$		$T \rightarrow FT'$	$T \rightarrow FT'$	$T \rightarrow FT'$	
T'	$T' \rightarrow \varepsilon$		$T' \rightarrow T$	$T' \rightarrow \varepsilon$	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow \varepsilon$
F			$F \rightarrow PF'$		$F \rightarrow PF'$	$F \rightarrow PF'$	$F \rightarrow PF'$	
F'	$F' \rightarrow \varepsilon$	$F' \rightarrow *F'$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$	$F' \rightarrow \varepsilon$
P			$P \rightarrow (E)$		$P \rightarrow a$	$P \rightarrow b$	$P \rightarrow \wedge$	



第十二题（ P81 第2题）

（4）构造它的递归下降分析程序。（略，不讲）



第十三题（P82 第3题）

（选作，不讲）



第十四题（ P133 第1题）

1. 令文法 G_1 为：

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

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

$$F \rightarrow (E) \mid i$$

证明 $E+T * F$ 是它的一个句型，指出这个句型的所有短语、直接短语和句柄。

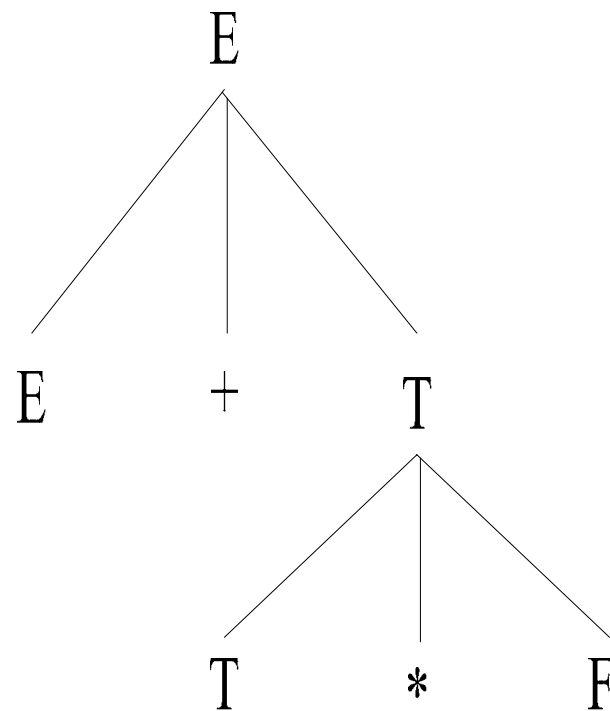
第十四题（P133 第1题）

解：因为 $E \Rightarrow E+T \Rightarrow E+T^*F$ ，
所以 $E+T^*F$ 是该文法的一个句型。

短语： $E+T^*F$ 、 T^*F

直接短语： T^*F

句柄： T^*F





第十五题（P133 第2题）

2. 考虑下面的表格结构文法 G_2 :

$$S \rightarrow a \mid \wedge \mid (T)$$

$$T \rightarrow T, S \mid S$$

- ① 给出 $(a, (a, a))$ 和 $((a, a), \wedge, (a)), a)$ 的最左和最右推导。
- ② 指出 $((a, a), \wedge, (a)), a)$ 的规范归约及每一步的句柄。根据这个规范归约，给出“移进-归约”的过程，并给出它的语法树自下而上的构造过程。



第十五题（ P133 第2题）

解：

- ① 最左推导和最右推导过程（略）最左推导指每一步都是对最左的非终结符进行替换，最右推导是指每一步都是对最右的非终结符进行替换。
- ② 规范归约过程及每一步的句柄，“移进-归约”过程，以及语法树的自下而上构造过程等均可按照语法树进行。具体过程非常繁杂简单，略，不讲。



第十六题（ P133 第3题）

3. 考虑下面的表格结构文法 G_2 :

$$S \rightarrow a \mid \wedge \mid (T)$$

$$T \rightarrow T, S \mid S$$

- ① 计算文法 G_2 的FIRSTVT和LASTVT。
- ② 计算 G_2 的优先关系。 G_2 是一个算符优先文法吗？
- ③ 计算 G_2 的优先函数。
- ④ 给出输入串 $(a, (a, a))$ 的算符优先分析过程。

第十六题 (P133 第3题)

① 计算FIRSTVT和LASTVT集合

根据P91规则(1)，我们有 $a, \wedge, (\in \text{FIRSTVT}(S)$ ；

$, \in \text{FIRSTVT}(T)$ 。根据规则(2)，我们有：

$\text{FIRSTVT}(S) \subseteq \text{FIRSTVT}(T)$ 。因此，我们有：

$\text{FIRSTVT}(S) = \{a, \wedge, (\}$ ， $\text{FIRSTVT}(T) = \{, , a, \wedge, (\}$

同理，我们有：

$\text{LASTVT}(S) = \{a, \wedge,) \}$ ， $\text{LASTVT}(T) = \{, , a, \wedge,) \}$

第十六题（P133 第3题）

② 计算该文法的优先关系

根据文法我们有：

$(\prec \text{FIRSTVT}(T) \prec$

$\text{LASTVT}(T) \succ)$ 、

$, \prec \text{FIRSTVT}(S) \prec$

$\text{LASTVT}(T) \succ)$ 、

$\# \prec \text{FIRSTVT}(S) \prec$

$\text{LASTVT}(S) \succ \#$ 、

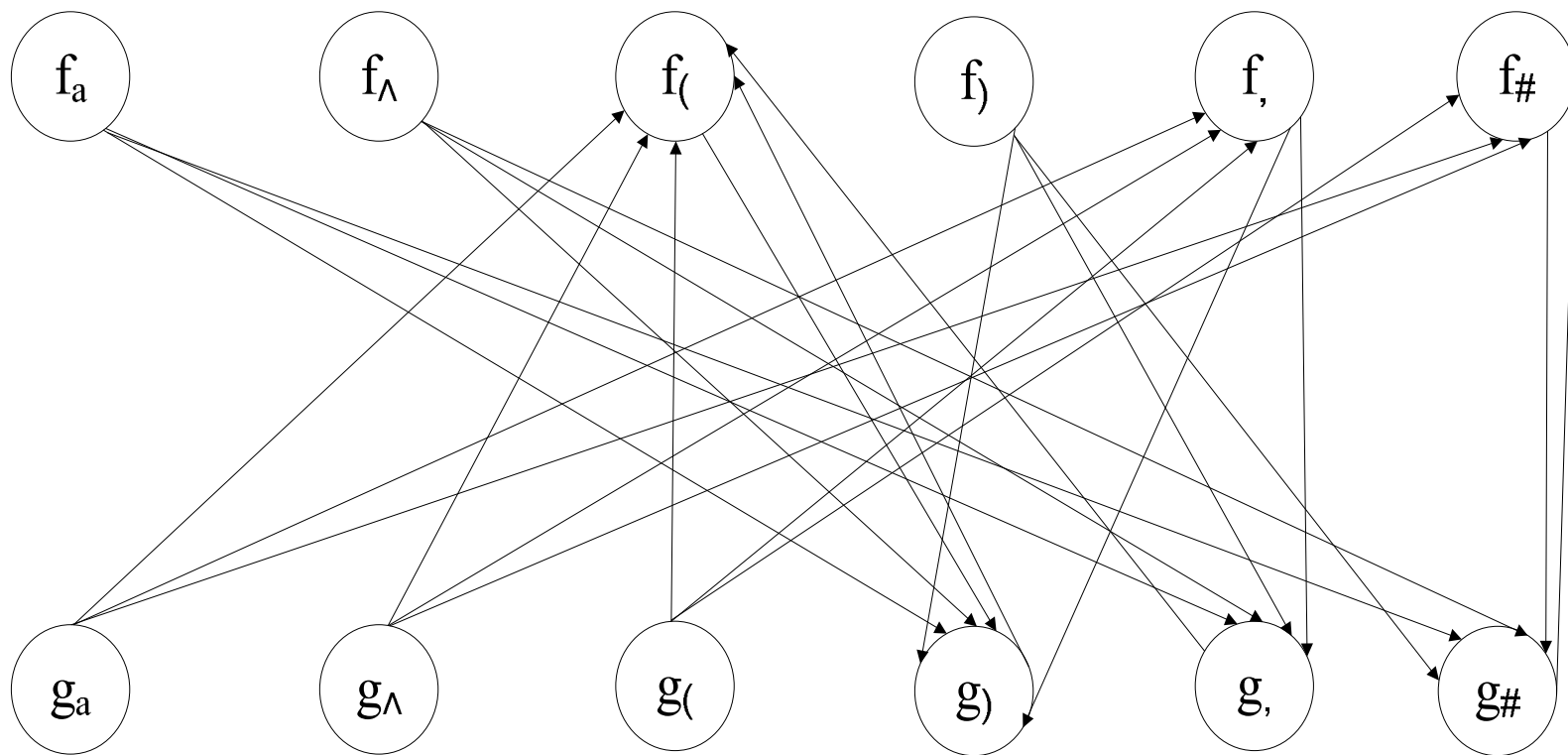
$\# \preceq \#$ 、 (\preceq) 。

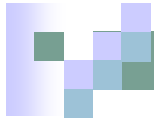
	a	\wedge	()	,	#
a				\succ	\succ	\succ
\wedge				\succ	\succ	\succ
(\prec	\prec	\prec	\preceq	\prec	
)				\succ	\succ	\succ
,	\prec	\prec	\prec	\succ	\succ	
#	\prec	\prec	\prec			\preceq

因此，G2是算符优先文法

第十六题（P133 第3题）

③ 方向图如下图，优先函数





第十六题（ P133 第3题）

③ 优先函数

	a	\wedge	()	,	#
f	6	6	2	6	4	2
g	7	7	7	2	3	2

第十六题 (P133 第3题)

④ 输入串(a, (a, a))的算符优先分析过程

栈	输入字符串	动作
#	(a, (a, a))#	预备
#(a, (a, a))#	移进
#(a	, (a, a))#	移进
#(S	, (a, a))#	归约
#(T	, (a, a))#	归约?
#(T,	(a, a))#	移进
#(T,(a, a))#	移进
#(T,(a	, a))#	移进
#(T,(S	, a))#	归约
#(T,(T	, a))#	归约?

第十六题（ P133 第3题）

④ 输入串(a, (a, a))的算符优先分析过程

栈	输入字符串	动作
#(T,(T,	a))#	移进
#(T,(T, a))#	移进
#(T,(T, S))#	归约
#(T,(T))#	归约
#(T,(T))#	移进
#(T, S)#	归约
#(T)#	归约
#(T)	#	移进
#S	#	归约
acc		



第十七题（P134 第5题）

5. 考虑文法

$$S \rightarrow AS \mid b$$

$$A \rightarrow SA \mid a$$

- ① 列出这个文法的所有LR(0)项目。
- ② 构造这个文法的LR(0)项目集规范族及识别活前缀的DFA。
- ③ 这个文法是SLR的吗？若是，构造出它的SLR分析表。
- ④ 这个文法是LALR或LR(1)的吗？



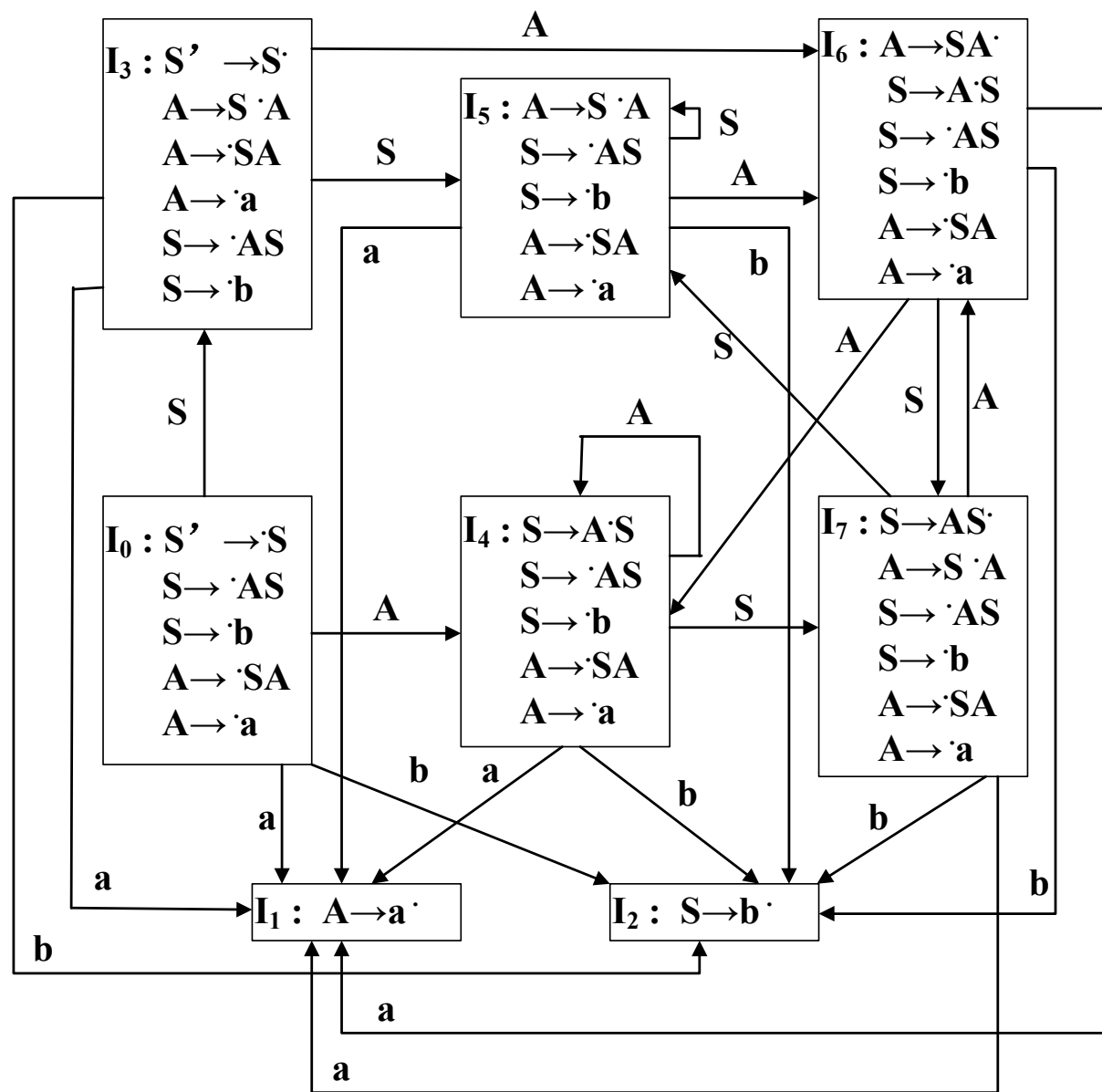
第十七题（P134 第5题）

① 文法的所有LR(0)项目如下：

- | | | |
|------------------------------|-----------------------------|------------------------------|
| 0. $S' \rightarrow \cdot S$ | 1. $S' \rightarrow S \cdot$ | 2. $S \rightarrow \cdot AS$ |
| 3. $S \rightarrow A \cdot S$ | 4. $S \rightarrow AS \cdot$ | 5. $S \rightarrow \cdot b$ |
| 6. $S' \rightarrow b \cdot$ | 7. $A \rightarrow \cdot SA$ | 8. $A \rightarrow S \cdot A$ |
| 9. $A \rightarrow SA \cdot$ | 10. $A \rightarrow \cdot a$ | 11. $A \rightarrow a \cdot$ |



② 通过
CLOSURE和
GO函数
构造的
LR(0)项
目集规
范族及
识别活
前缀的
DFA如
下图所
示：





第十七题（P134 第5题）

③ I_3 、 I_6 和 I_7 包含“移进-归约”冲突。

I_3 : FOLLOW(S')= $\{\#\}$, 不包含a、b。

I_6 : FOLLOW(S)= $\{\#,a,b\}$ 包含a、b , “移进-归约”冲突无法消解。

I_7 : FOLLOW(A)= $\{a,b\}$ 包含a、b , “移进-归约”冲突无法消解。

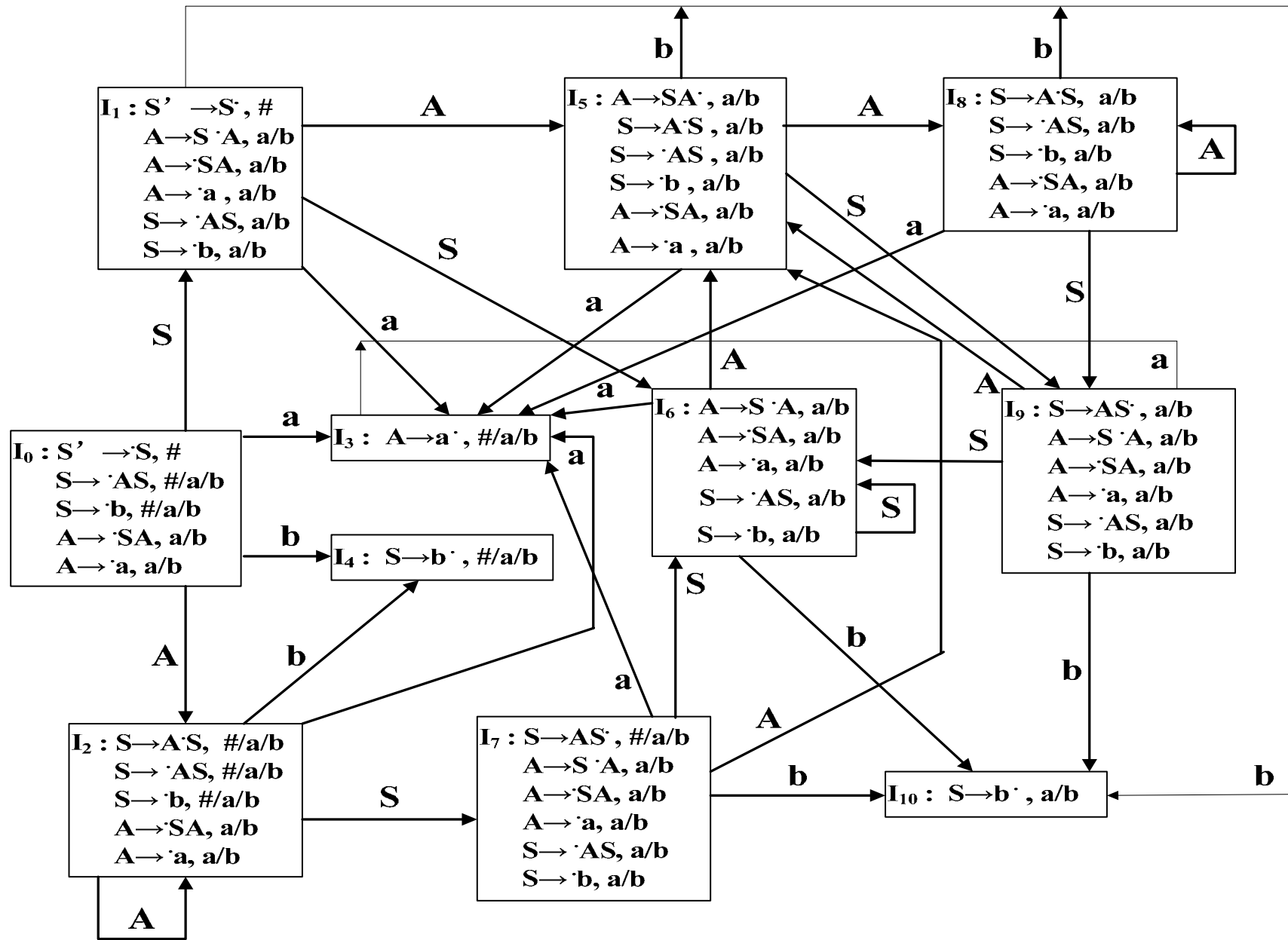
所以不是SLR文法。



第十七题（P134 第5题）

④ 构造LR(1)项目集规范族如图所示：

对于状态5，因为包含项目 $[A \rightarrow SA\cdot, a/b]$ ，所以遇到搜索符号a或b时，应该用 $A \rightarrow SA$ 归约。又因为状态5包含项目 $[A \rightarrow \cdot a, a/b]$ ，所以遇到搜索符号a时，应该移进。因此存在“移进-归约”矛盾，所以这个文法不是LR(1)文法。（当然也不是LALR文法）





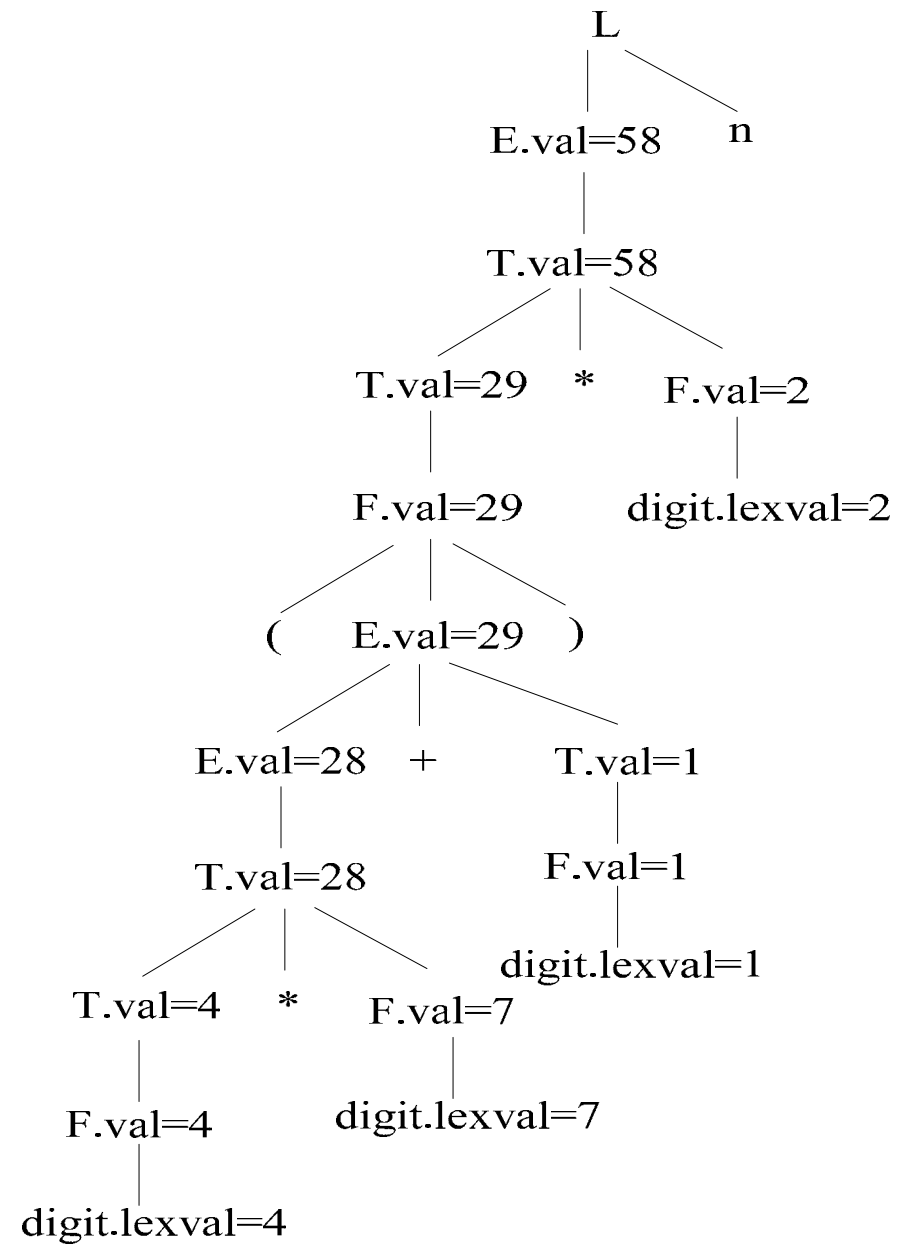
第十八题（P135 第8题）

（选作，不讲）



第二十题（P164 第1题）

1. 按照表6.1所示的属性文法，构造表达式 $(4*7+1)*2$ 的附注语法树。





第二十一题（ P164 第2题）

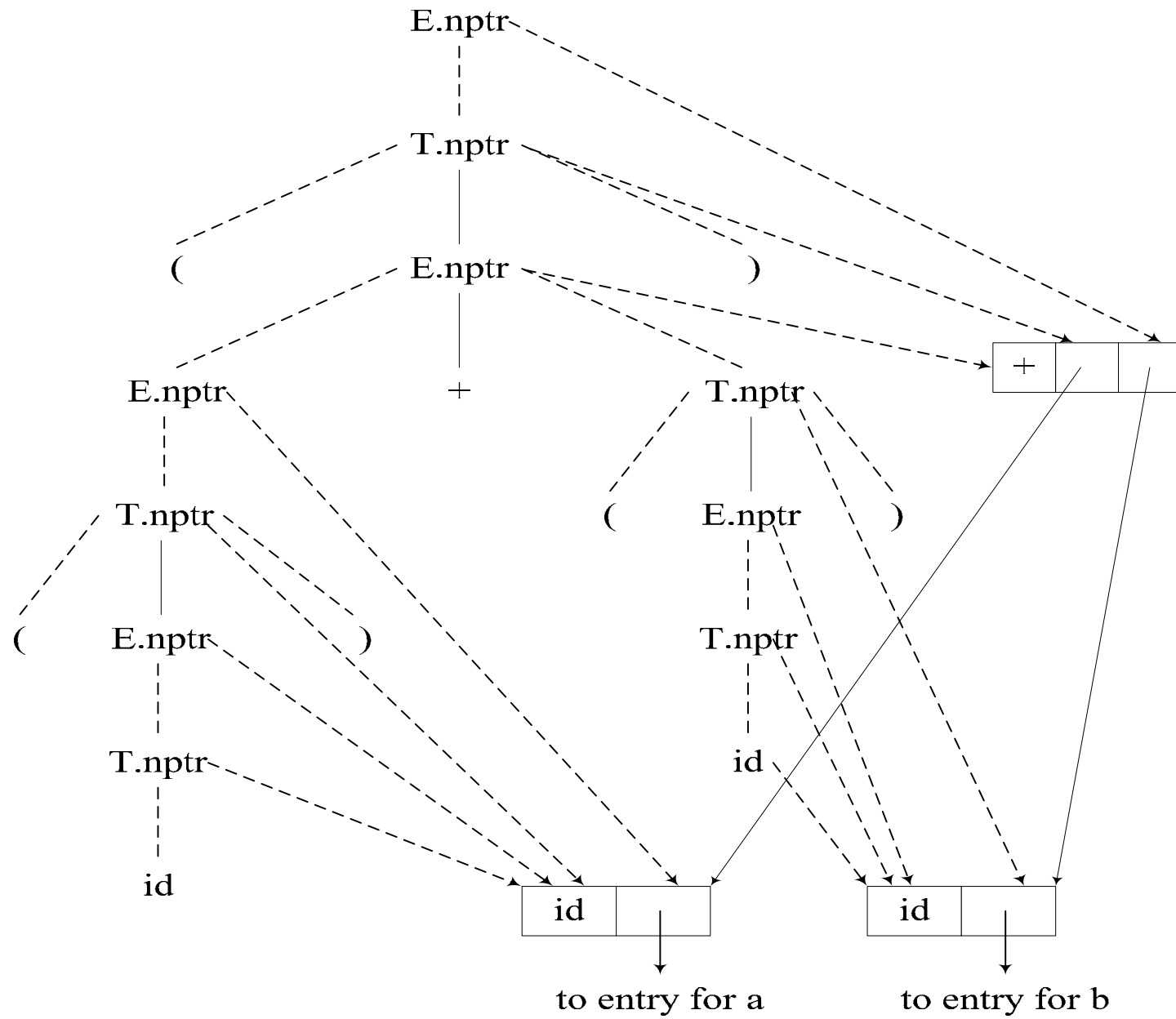
2. 对表达式 $((a) + (b))$:

- ① 按照表6.4所示的属性文法构造该表达式的抽象语法树;
- ② 按照图6.17所示的翻译模式, 构造该表达式的抽象语法树。



第二十一题（ P164 第2题）

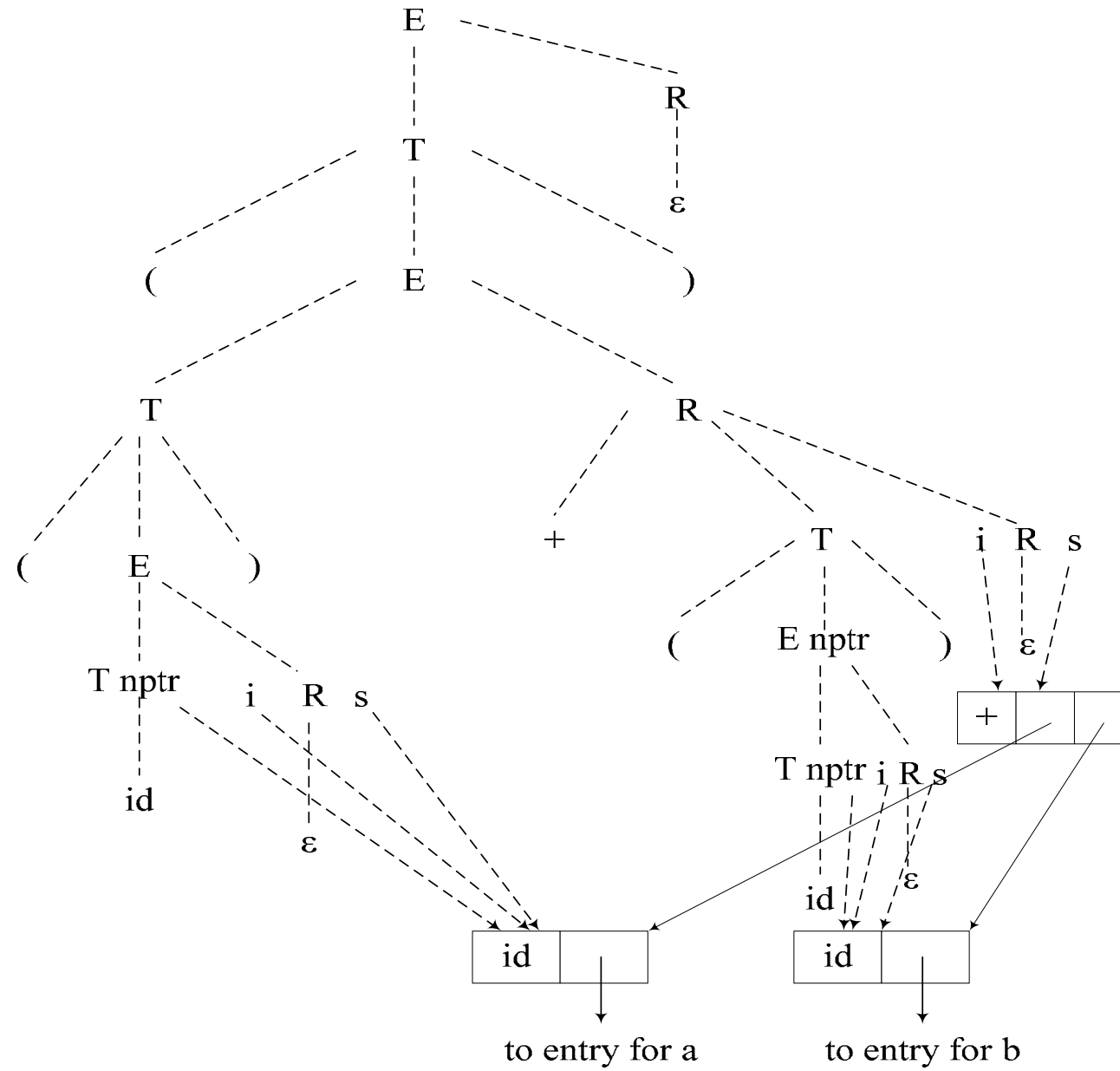
- ① 按表6.4所列的属性文法，表达式 $((a) + (b))$ 带注释的语法分析树如图所示。其中，构造出的抽象语法树用实线表示，语法分析树用虚线表示，语法分析树之中的E和T标识的结点中用综合属性nptr来保存指向抽象语法树中该非终结符号对应的结点的指针（图中用带箭头的虚线表示）。





第二十一题（ P164 第2题）

- ② 按图6.17所示的翻译模式，表达式 $((a) + (b))$ 带注释的语法分析树如图所示。





第二十二题（ P164 第5题）

5. 下列文法对整形常数和实型常数施用加法运算符+生成表达式；当两个整型数相加时，结果仍为整型数，否则，结果为实型数：

$$E \rightarrow E+T | T$$

$$T \rightarrow \text{num.num} | \text{num}$$

- ① 试给出确定每个子表达式结果类型的属性文法；
- ② 扩充(1)的属性文法，使之把表达式翻译成后缀形式，同时也能确定结果的类型。应该注意施用一元运算符`inttoreal`把整型数转换成实型数，以便使后缀形如加法运算符的两个操作数具有相同的类型。



第二十二题（ P164 第5题）

解：确定每个子表达式结果类型的属性文法是比较容易定义的。关键是如何扩充此属性文法，使之把表达式翻译成后缀形式。我们将不在name或num.num向T归约的时候输出该运算对象，而是把运算对象的输出放在T或E+T向E归约的时候。



第二十二题（ P164 第5题）

解：这是因为考虑输出类型转换算符`inttoreal`的动作可能在 $E+T$ 向 E 归约的时候进行，如果这时两个运算对象都在前面`name`或`num.num`向 T 归约的时候已经输出，需要为第1个运算对象输出类型转换符时就已经为时太晚。另外需要注意的是，在 $E+T$ 向 E 归约时，该加法运算的第1个运算对象已经输出。所以 $E \rightarrow E+T$ 的语义规则不需要有输出 E 运算对象的动作。



第二十二题（P164 第5题）

- (1) 为文法符号E和T配以综合属性type，用来表示它们的类型。类型值分别用int和real来表示。确定每个子表达式结果类型的属性文法如下：

产生式	语义规则
$E \rightarrow E+T$	$E.type := \text{if } E_1.type = \text{int and } T.type = \text{int then int else real}$
$E \rightarrow T$	$E.type := T.type$
$T \rightarrow \text{num.num}$	$T.type := \text{real}$
$T \rightarrow \text{num}$	$T.type := \text{int}$

第二十二题（P164 第5题）

(2) 下面属性文法将表达式的后缀表示打印输出，其中 lexeme 属性表示单词的拼写。

产生式	语义规则
$E \rightarrow E_1 + T$	<pre>if E₁.type=real and T.type=int then begin E.type:=real; print (T.lexeme); print ('inttoreal') end if E₁.type=int and T.type=real then begin E.type:=real ; print ('inttoreal'); print (T.lexeme) end else begin E.type:=E₁.type; print(T.lexeme) end; print('+');</pre>



第二十二题（P164 第5题）

(2) 下面属性文法将表达式的后缀表示打印输出，其中 lexeme 属性表示单词的拼写。

产生式	语义规则
$E \rightarrow T$	$E.type := T.type; \text{print}(T.lexeme)$
$T \rightarrow num_1.num_2$	$T.type := \text{real}; T.lexeme := num_1.lexeme "." num_2.lexeme$
$T \rightarrow num$	$T.type := \text{int}; T.lexeme := num.lexeme$



第二十三题（ P164 第7题）

7. 下列文法由开始符号S产生一个二进制数，令综合属性val给出该数的值：

$$S \rightarrow L.L | L$$

$$L \rightarrow LB | B$$

$$B \rightarrow 0 | 1$$

试设计求S.val的属性文法，其中，已知B的综合属性c，给出由B产生的二进位的结果值。例如，输入101.101时，S.val=5.625，其中第一个二进位的值是4，最后一个二进位的值是0.125。

第二十三题（P164 第7题）

解：假设一个二进制数为： $a_m \dots a_1 a_0 . b_n \dots b_1 b_0$ ，其中， a_i ($i=0$ 、 1 、 \dots 、 m)和 b_j ($j=0$ 、 1 、 \dots 、 n)为二进制数字，则该二进制的十进制值为：

$$a_m * 2^m + \dots + a_1 * 2^1 + a_0 * 2^0 + b_n * 2^{-1} + \dots + b_1 * 2^{-n} + b_0 * 2^{-n-1} =$$
$$a_m * 2^m + \dots + a_1 * 2^1 + a_0 * 2^0 + (b_n * 2^n + \dots + b_1 * 2^1 + b_0 * 2^0) / 2^{n+1}$$

根据此公式，我们在语法制导翻译时，对二进制数字串L，不仅要计算其值L.val，还要计算其长度L.length（二进制数字个数）。如果L为小数部分，则其值L.val要除以 $2^{L.length}$ 。



第二十三题（ P164 第7题）

解：因此，我们所设计的属性文法如下：

产生式	语义动作
$S' \rightarrow S$	$\{\text{print}(S.\text{val})\}$
$S \rightarrow L_1.L_2$	$\{S.\text{val} := L_1.\text{val} + L_2.\text{val} / 2^{L_2.\text{length}}\}$
$S \rightarrow L$	$\{S.\text{val} := L.\text{val}\}$
$L \rightarrow L_1B$	$\{L.\text{val} := L_1.\text{val} * 2 + B.c\}$ $\{L.\text{length} := L_1.\text{length} + 1\}$
$L \rightarrow B$	$\{L.\text{val} := B.c; L.\text{length} := 1\}$
$B \rightarrow 1$	$\{B.c := 1\}$
$B \rightarrow 0$	$\{B.c := 0\}$



第二十四题（ P165 第11题）

选做题（略）



第二十五题（ P165 第12题）

选做题（略）



第二十六题（ P217 第1题）

1. 给出下面表达式的逆波兰表示（后缀式）：

$a*(-b+c)$ $\text{not } A \text{ or not } (C \text{ or not } D)$

$a+b*(c+d/e)$ $(A \text{ and } B) \text{ or } (\text{not } C \text{ or } D)$

$-a+b*(-c+d)$ $(A \text{ or } B) \text{ and } (C \text{ or not } D \text{ and } E)$

$\text{if } (x+y)*z=0 \text{ then } (a+b) \uparrow c \text{ else } a \uparrow b \uparrow c$



第二十六题（ P217 第1题）

解： $a\ b\ @\ c\ +\ *$

$A\ \text{not}\ C\ D\ \text{not}\ \text{or}\ \text{not}\ \text{or}$

$a\ b\ c\ d\ e\ /\ +\ *\ +$

$A\ B\ \text{and}\ C\ \text{not}\ D\ \text{or}\ \text{or}$

$a\ @\ b\ c\ @\ d\ +\ *\ +$

$A\ B\ \text{or}\ C\ D\ \text{not}\ E\ \text{and}\ \text{or}\ \text{and}$

$x\ y\ +\ z\ *\ 0 = a\ b\ +\ c\ \uparrow\ a\ b\ c\ \uparrow\ \uparrow\ \text{if_then_else}$



第二十七题（ P217 第3题）

3. 请将表达式 $-(a+b)*(c+d)-(a+b+c)$ 分别表示为三元式、间接三元式和四元式序列。



第二十七题 (P217 第3题)

三元式序列:

- (1) (+, a, b)
- (2) (@, (1), $_$)
- (3) (+, c, d)
- (4) (*, (2), (3))
- (5) (+, a, b)
- (6) (+, (5), c)
- (7) (-, (4), (6))

四元式序列:

- (1) (+, a, b, T_1)
- (2) (@, T_1 , $_$, T_2)
- (3) (+, c, d, T_3)
- (4) (*, T_2 , T_3 , T_4)
- (5) (+, a, b, T_5)
- (6) (+, T_5 , c, T_6)
- (7) (-, T_4 , T_6 , T_7)



第二十七题（ P217 第3题）

间接三元式序列：

(1) (+, a, b)

(2) (@, (1), _)

(3) (+, c, d)

(4) (*, (2), (3))

(5) (+, (1), c)

(6) (-, (4), (5))

间接码表

(1)

(2)

(3)

(4)

(1)

(5)

(6)



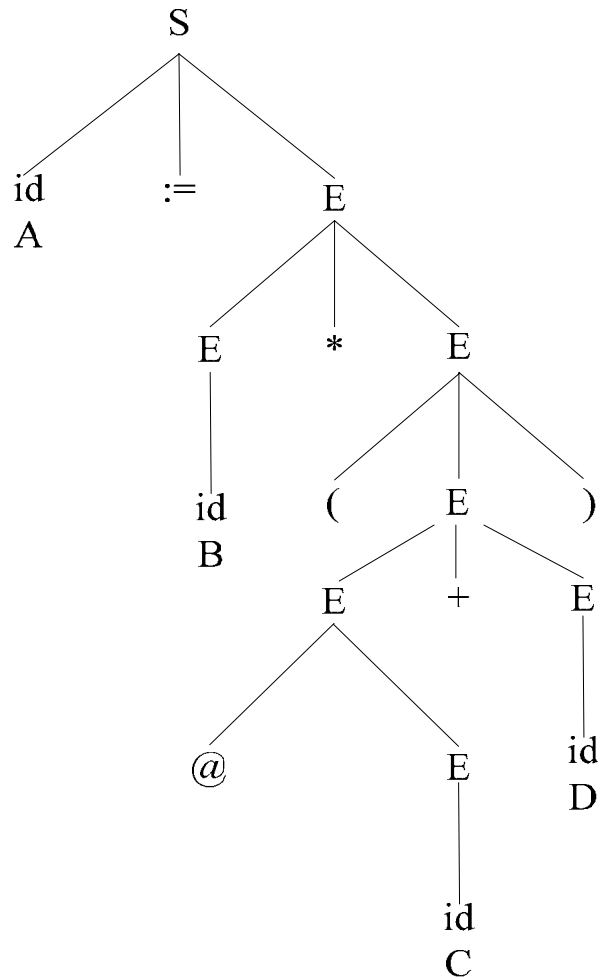
第二十八题（ P218 第4题）

4. 按7.3节所说的办法，写出下面赋值语句

$$A:=B*(-C+D)$$

的自下而上语法制导翻译过程。给出所产生的三地址代码。

第二十八题（P218 第4题）



```

S→id:=E      { p:=lookup(id.name);
                    if p≠nil then emit(p ':= ' E.place)
                    else error }

E→E1+E2      { E.place:=newtemp;
                    emit(E.place ':= ' E1.place '+' E2.place)}

E→E1*E2      { E.place:=newtemp;
                    emit(E.place ':= ' E1.place '*' E2.place)}

E→-E1         { E.place:=newtemp;
                    emit(E.place' := ' 'uminus'E1.place)}

E→(E1)        { E.place:=E1.place}

E→id          { p:=lookup(id.name);
                    if p≠nil then E.place:=p
                    else error }
  
```



第二十八题（ P218 第4题）

根据语法分析树，最终生成的四元式序列如下：

$$T_1 := @C$$

$$T_2 := T_1 + D$$

$$T_3 := B * T_2$$

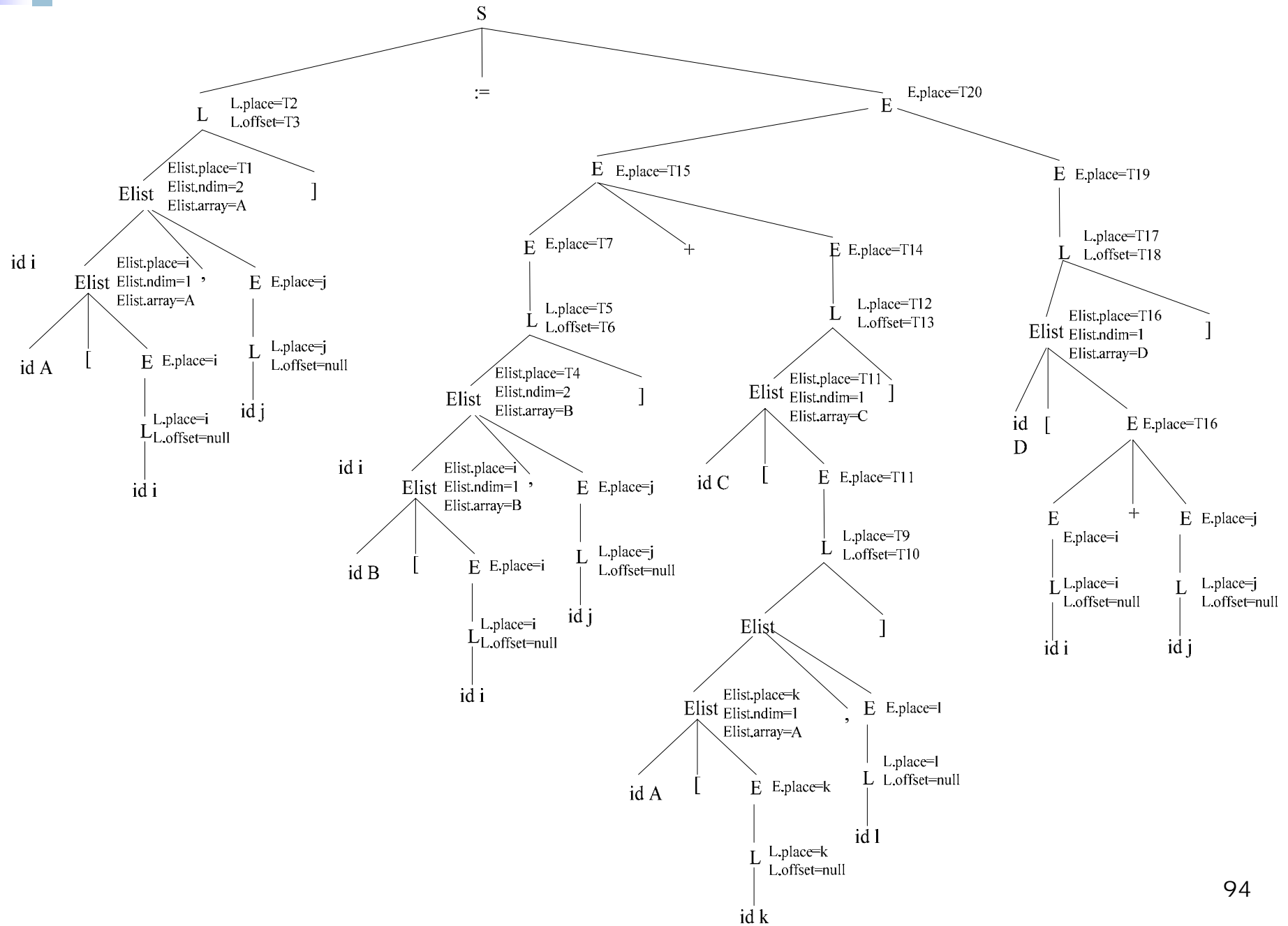
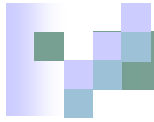
$$A := T_3$$



第二十九题（ P218 第5题）

5. 按7.3.2所给的翻译模式，把下列赋值语句翻译为三地址代码：

$$A[i, j] := B[i, j] + C[A[k, l]] + D[i + j]$$



第二十九题（P218 第5题）

所得结果如下：

T1:=i*20

T8:=k*20

T15:=T7+T14

T1:=T1+j

T8:=T8+1

T16:=i+j

T2:=A-84

T9:=A-84

T17:=D-4

T3:=4*T1

T10:=4*T8

T18:=4*T16

T4:=i*20

T11:=T9[T10]

T19:=T17[T18]

T4:=T4+j

T12:=C-4

T20:=T15+T19

T5:=B-84

T13:=4*T11

T2[T3]:=T20

T6:=4*T4

T14:=T12[T13]

T7:=T5[T6]