

附录 部分习题参考答案

第 1 章参考答案：

1, 2, 3, 4, 5, 6, 7 解答：略！

第 2 章参考答案：

1, 2, 3: 解答：略！

4. 解答：

A: ① B: ③ C: ① D: ②

5. 解答：

用 E 表示<表达式>, T 表示<项>, F 表示<因子>, 上述文法可以写为：

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

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

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

最左推导：

$$E \Rightarrow E+T \Rightarrow E+T+T \Rightarrow T+T+T \Rightarrow F+T+T \Rightarrow i+T+T \Rightarrow i+F+T \Rightarrow i+i+T \\ \Rightarrow i+i+F \Rightarrow i+i+i$$

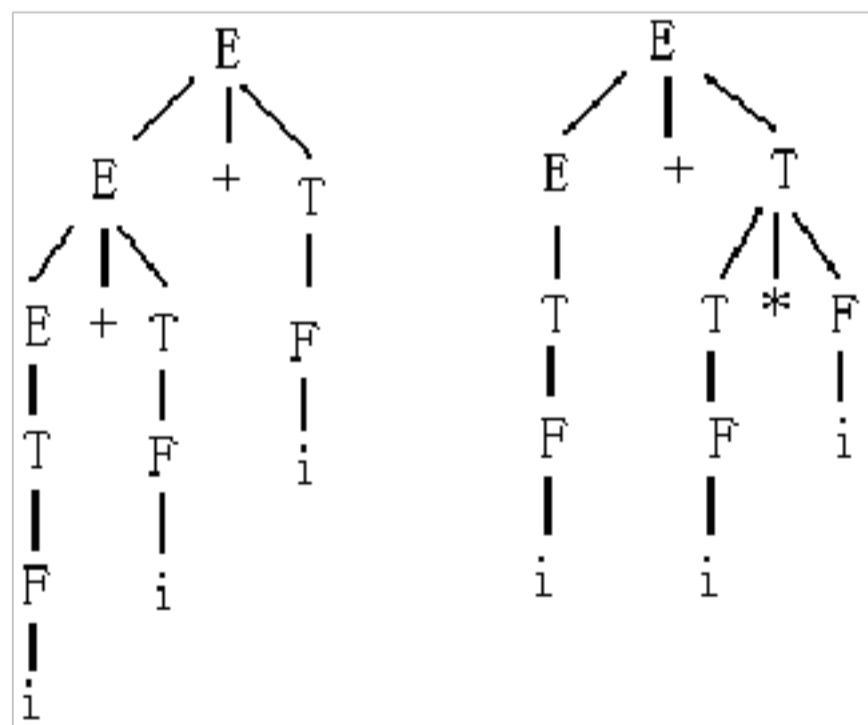
$$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow i+T \Rightarrow i+T * F \Rightarrow i+F * F \Rightarrow i+i * F \Rightarrow i+i * i$$

最右推导：

$$E \Rightarrow E+T \Rightarrow E+F \Rightarrow E+i \Rightarrow E+T+i \Rightarrow E+F+i \Rightarrow E+i+i \Rightarrow T+i+i \\ \Rightarrow F+i+i \Rightarrow i+i+i$$

$$E \Rightarrow E+T \Rightarrow E+T * F \Rightarrow E+T * i \Rightarrow E+F * i \Rightarrow E+i * i \Rightarrow T+i * i \Rightarrow F+i * i \Rightarrow i+i * i$$

$i+i+i$ 和 $i+i * i$ 的语法树如下图所示。



$i+i+i$ 、 $i+i * i$ 的语法树

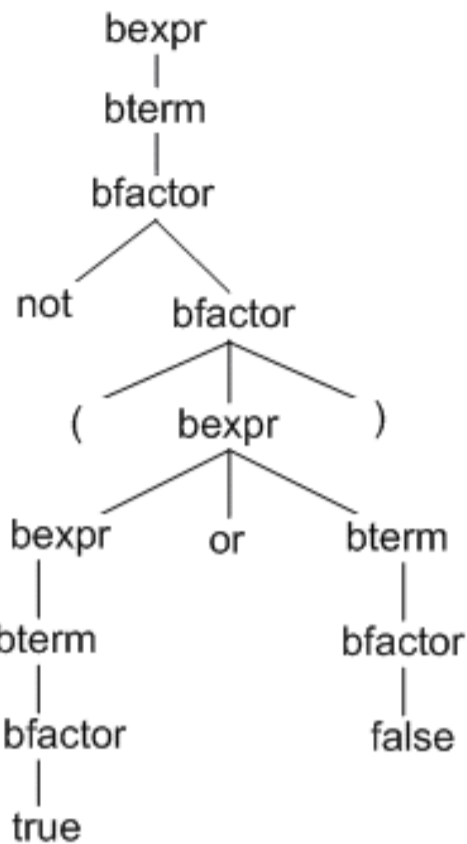
6. 解答：

(1) 终结符号为：{or, and, not, (,), true, false }

非终结符号为: {bexpr, bterm, bfactor}

开始符号为: bexpr

(2) 句子 not(true or false) 的语法树为:



7. 解答:

(1) 把 $an^ib^nc^i$ 分成 an^ib^n 和 c^i 两部分, 分别由两个非终结符号生成, 因此, 生成此文法的产生式为:

$$S \rightarrow AB$$

$$A \rightarrow aAb|ab$$

$$B \rightarrow cB|\square$$

(2) 令 S 为开始符号, 产生的 w 中 a 的个数恰好比 b 多一个, 令 E 为一个非终结符号, 产生含相同个数的 a 和 b 的所有串, 则产生式如下:

$$S \rightarrow aE|Ea|bSS|SbS|SSb$$

$$E \rightarrow aEbE|bEaE|\square$$

(3) 设文法开始符号为 S , 产生的 w 中满足 $|a| \leq |b| \leq 2|a|$ 。因此, 可想到 S 有如下的产生式 (其中 B 产生 1 到 2 个 b):

$$S \rightarrow aSBS|BSaS$$

$$B \rightarrow b|bb$$

(4) 解法一:

$$S \rightarrow \langle \text{奇数头} \rangle \langle \text{整数} \rangle \langle \text{奇数尾} \rangle$$

$$|\langle \text{奇数头} \rangle \langle \text{奇数尾} \rangle$$

$$|\langle \text{奇数尾} \rangle$$

$$\langle \text{奇数尾} \rangle \rightarrow 1|3|5|7|9$$

$$\langle \text{奇数头} \rangle \rightarrow 2|4|6|8|\langle \text{奇数尾} \rangle$$

$$\langle \text{整数} \rangle \rightarrow \langle \text{整数} \rangle \langle \text{数字} \rangle | \langle \text{数字} \rangle$$

$$\langle \text{数字} \rangle \rightarrow 0|\langle \text{奇数头} \rangle$$

解法二: 文法 $G = (\{S, A, B, C, D\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, P, S)$

$$S \rightarrow AB | B$$

$$A \rightarrow AC | D$$

$$B \rightarrow 1|3|5|7|9$$

$$D \rightarrow 2|4|6|8|B$$

$$C \rightarrow 0|D$$

(5) 文法 $G = (\{N, S, M, D\}, \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}, S, P)$

$$S \rightarrow N0 | N5$$

$$N \rightarrow MD | \square$$

$$M \rightarrow 1|2|3|4|5|6|7|8|9$$

$$D \rightarrow D0 | DM | \square$$

(6) $G[S]: S \rightarrow aSa | bSb | cSc | a | b | c | \square$

8. 解答:

(1) 句子 $abab$ 有如下两个不同的最左推导:

$$S \Rightarrow aSbS \Rightarrow abS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab$$

$$S \Rightarrow aSbS \Rightarrow abSaSbS \Rightarrow abaSbS \Rightarrow ababS \Rightarrow abab$$

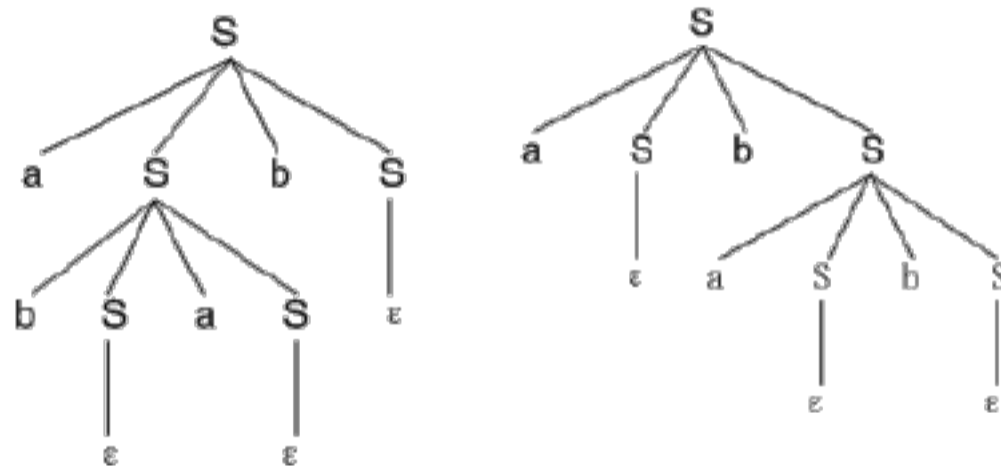
所以此文法是二义性的。

(2) 句子 $abab$ 的两个相应的最右推导:

$$S \Rightarrow aSbS \Rightarrow aSbaSbS \Rightarrow aSbaSb \Rightarrow aSbab \Rightarrow abab$$

$$S \Rightarrow aSbS \Rightarrow aSb \Rightarrow abSaSb \Rightarrow abSab \Rightarrow abab$$

(3) 句子 $abab$ 的两棵分析树:



(a)

(b)

(4) 此文法产生的语言是: 在 $\{a, b\}$ 上由相同个数的 a 和 b 组成的字符串。

9, 10: 解答: 略!

第 3 章习题解答:

1. 解答:

(1) √ (2) √ (3) × (4) × (5) √ (6) √

2. [分析]

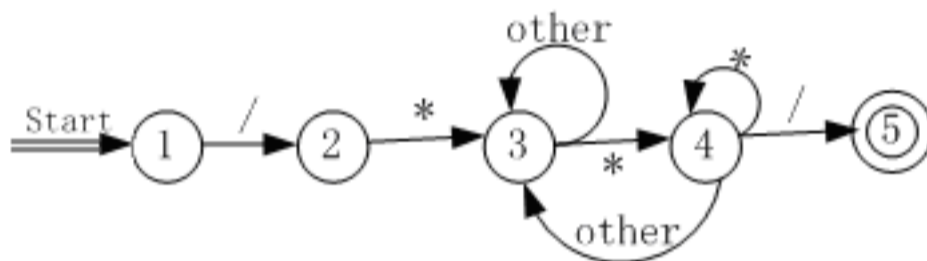
有限自动机分为确定有限自动机和非确定有限自动机。确定有限自动机的确定性表现在映射 $\delta: Q \times V_T \rightarrow Q$ 是单值函数, 也就是说, 对任何状态 $q \in Q$ 和输入字符串 $a \in V_T$, $\delta(q, a)$ 唯一确定下一个状态。显然, 本题给出的是一个确定的有限自动机, 它的状态转换图是 C 中的②。

它所接受的语言可以用正则表达式表示为 $00(0|1)^*$, 表示的含义为由两个 0 开始的后跟任意个 (包含 0 个) 0 或 1 组成的符号串的集合。

2. 解答: A: ④ B: ③ C: ② D: ② E: ④

3, 4. 解答: 略!

5. 解答:



6. 解答:

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

(2) $((1|2|\dots|9)(0|1|2|\dots|9))^*(0|5)$

(3) $(0|10011)(0|1)^*$

(4) $1^*0(0|10)^*(1)^*$

(5) $a^*b^*c^*\dots z^*$

(6) $(0|10)^*1$

(7) $(00|10)(01|10)^*(00|10)(00|10)^*$

(8) [分析]

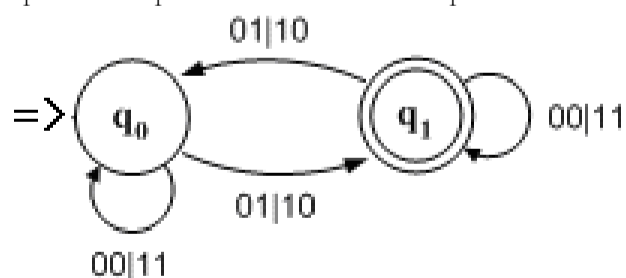
设 S 是符合要求的串, $|S|=2k+1$ ($k \geq 0$)。

则 $S \rightarrow S_1 0 S_2 1$, $|S_1|=2k$ ($k > 0$), $|S_2|=2k$ ($k \geq 0$)。

且 S_1 是 $\{0, 1\}$ 上的串, 含有奇数个 0 和奇数个 1。

S_2 是 $\{0, 1\}$ 上的串, 含有偶数个 0 和偶数个 1。

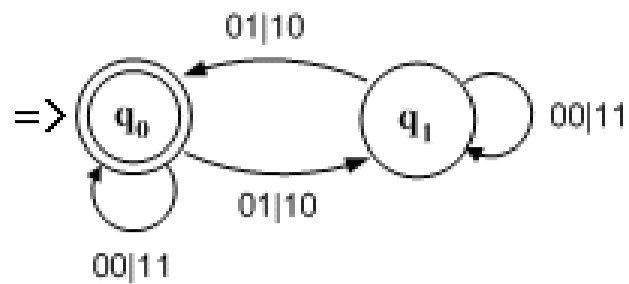
考虑有一个自动机 M_1 接受 S_1 , 那么自动机 M_1 如下:



和 $L(M_1)$ 等价的正规式, 即 S_1 为:

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

类似的考虑有一个自动机 M_2 接受 S_2 , 那么自动机 M_2 如下:



和 $L(M)$ 等价的正规式，即 S_2 为：

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

因此， S 为：

$$((00|11) | (01|10)(00|11)^*(01|10))^*(01|10)(00|11)^*0|$$

$$((00|11) | (01|10)(00|11)^*(01|10))^*1$$

7. 解答：

(1) 以 0 开头并且以 0 结尾的，由 0 和 1 组成的符号串。

(2) $\{\square \in \{0,1\}^*\}$

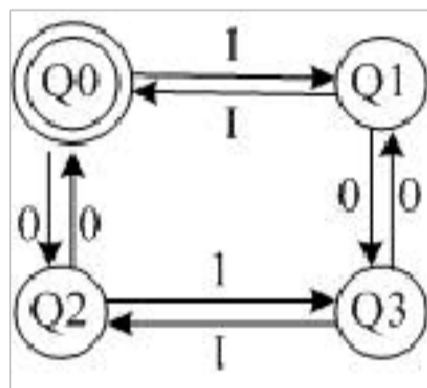
(3) 由 0 和 1 组成的符号串，且从右边开始数第 3 位为 0。

(4) 含 3 个 1 的由 0 和 1 组成的符号串。 $\{\square \in \{0,1\}^+, \text{且 } \square \text{ 中含有 3 个 1}\}$

(5) 包含偶数个 0 和 1 的二进制串，即 $\{\square \in \{0,1\}^*, \text{且 } \square \text{ 中有偶数个 0 和 1}\}$

8. 解答：

	0	1
Q0*	Q2	Q1
Q1	Q3	Q0
Q2	Q0	Q3
Q3	Q1	Q2



9. 解答：

(1) DFA $M = (\{0, 1\}, \{q_0, q_1, q_2\}, q_0, \{q_2\}, \delta)$

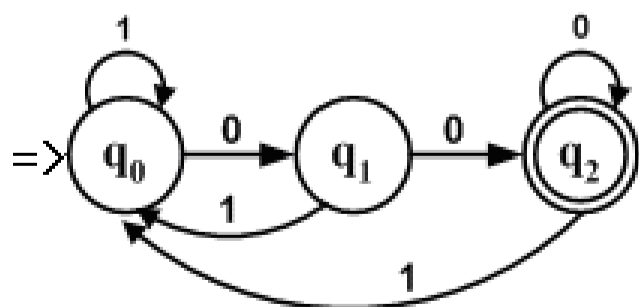
其中 δ 定义如下：

$$\delta(q_0, 0) = q_1 \quad \delta(q_0, 1) = q_0$$

$$\delta(q_1, 0) = q_2 \quad \delta(q_1, 1) = q_0$$

$$\delta(q_2, 0) = q_2 \quad \delta(q_2, 1) = q_0$$

状态转换图为：

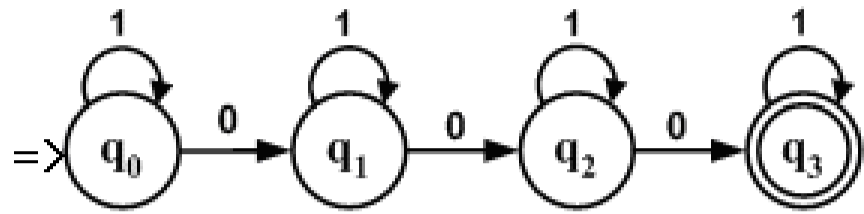


(2) 正规式： $101^*01^*01^*$

DFA $M = (\{0, 1\}, \{q_0, q_1, q_2, q_3\}, q_0, \{q_3\}, \delta)$ ，其中 δ 定义如下：

$$\begin{aligned} \delta(q_0, 0) &= q_1 & \delta(q_0, 1) &= q_0 \\ \delta(q_1, 0) &= q_2 & \delta(q_1, 1) &= q_1 \\ \delta(q_2, 0) &= q_3 & \delta(q_2, 1) &= q_2 \\ \delta(q_3, 1) &= q_3 \end{aligned}$$

状态转换图为：

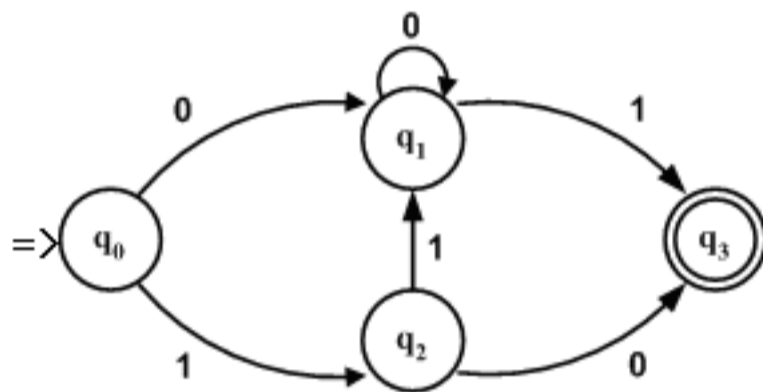


10. 解答：

(1) DFA $M = (\{0, 1\}, \{q_0, q_1, q_2, q_3\}, q_0, \{q_3\}, \delta)$, 其中 δ 定义如下：

$$\begin{aligned} \delta(q_0, 0) &= q_1 & \delta(q_0, 1) &= q_2 \\ \delta(q_1, 0) &= q_1 & \delta(q_1, 1) &= q_3 \\ \delta(q_2, 0) &= q_3 & \delta(q_2, 1) &= q_1 \end{aligned}$$

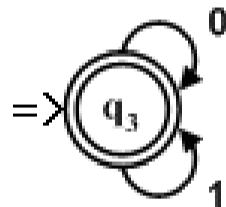
状态转换图为：



(2) DFA $M = (\{0, 1\}, \{q_0\}, q_0, \{q_0\}, \delta)$, 其中 δ 定义如下：

$$\delta(q_0, 0) = q_0 \quad \delta(q_0, 1) = q_0$$

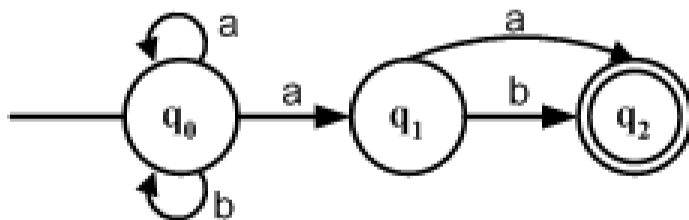
状态转换图为：



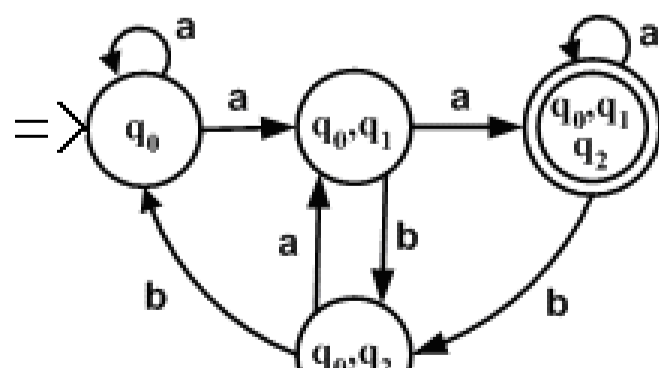
11 解答：

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

① 求出 NFA M:



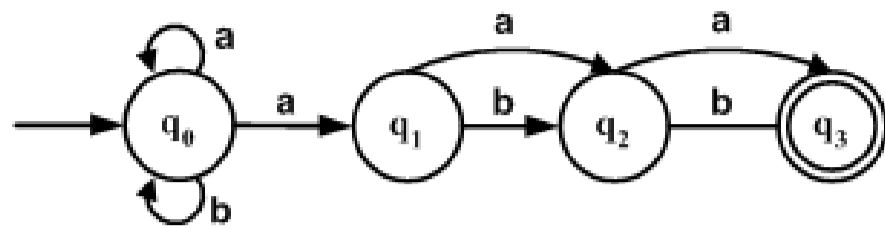
② 确定化，得到 DFA M:



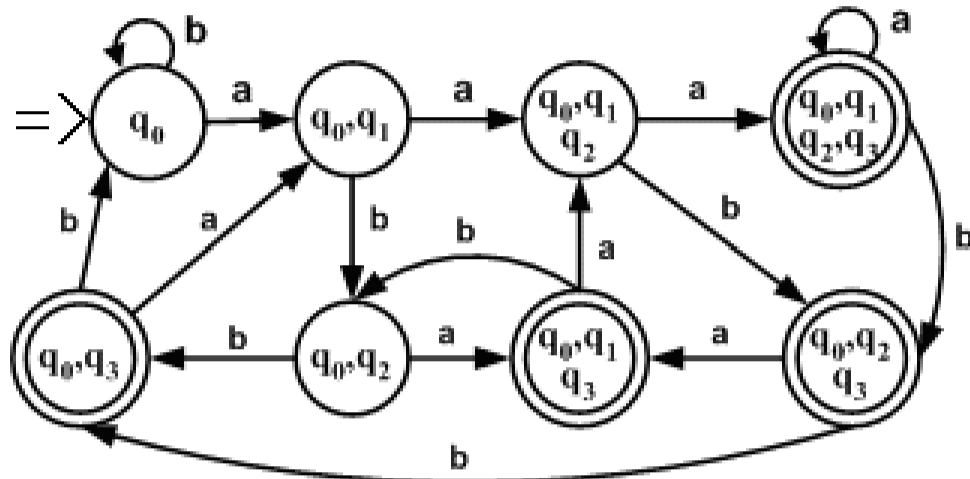
③ 化简：在第②步中求出的 DFA M 中没有等价状态，因此它就是最小化的 DFA M。

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

① 求 NFA M:



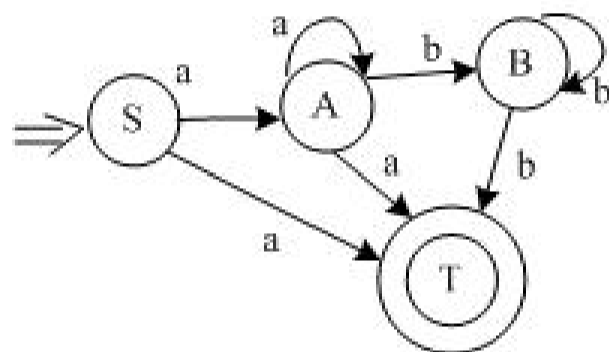
② 确定化, 得到 DFA M:



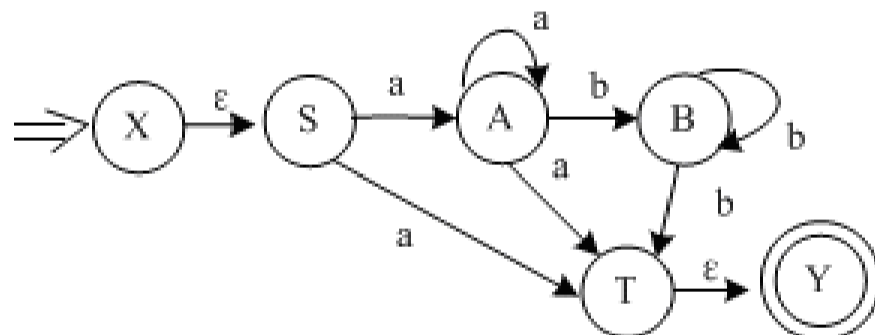
③ 化简, 在第②步中求出的 DFA M 中没有等价状态, 因此它已经是最小化的 DFA M 了。

12. 解答:

对应的 NFA 为:

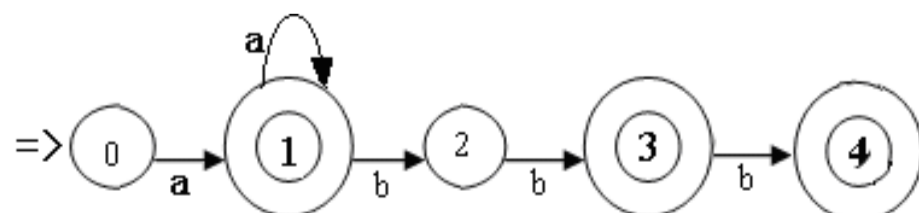


增加状态 X, Y, 再确定化:



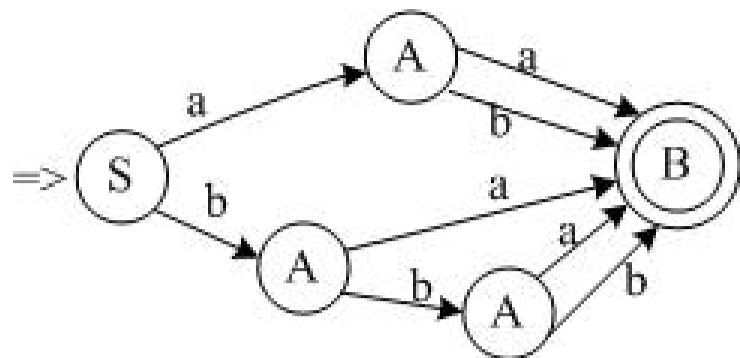
I	I_a	I_b
{x,5}	{A,T,Y}	{ }
{A,T,Y}	{A,T,Y}	{B}
{B}	{ }	{B,T,Y}
{B,T,Y}	{ }	{T,Y}
{T,Y}	{ }	{ }

得到的 DFA 为:



最小化: 该自动机已经是最小化的 DFA 了。

13. 解答:

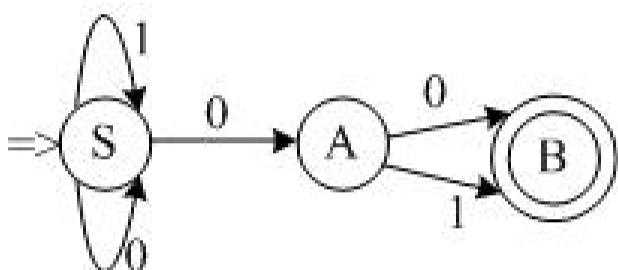


其中 a 代表 1 元硬币, b 代表 5 角硬币

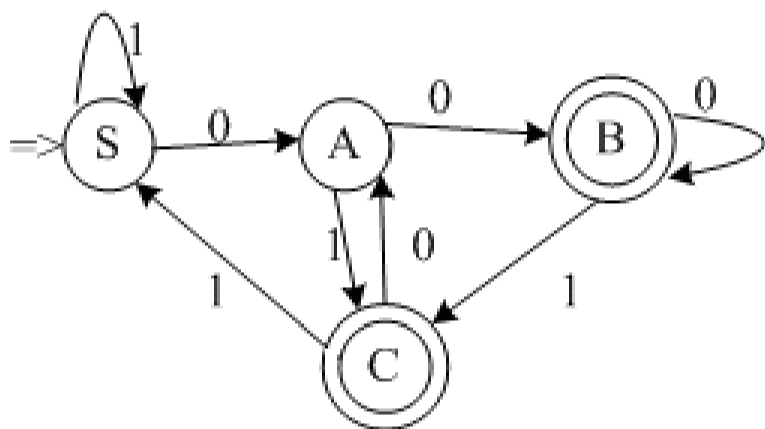
14. 解答:

正规式为: $(0|1)^*(00|01)$ 化简: $(0|1)^*(0|1)$

不确定的有穷自动机为:



确定化, 并最小化得到:



正规文法为:

$S \rightarrow 1S | 0A$

$A \rightarrow 0B | 0 | 1C | 1$

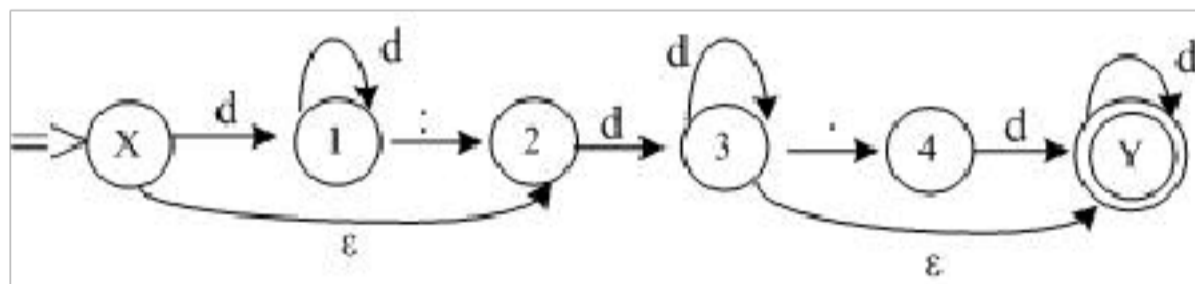
$B \rightarrow 0B | 0 | 1C | 1$

$C \rightarrow 1S | 0A$

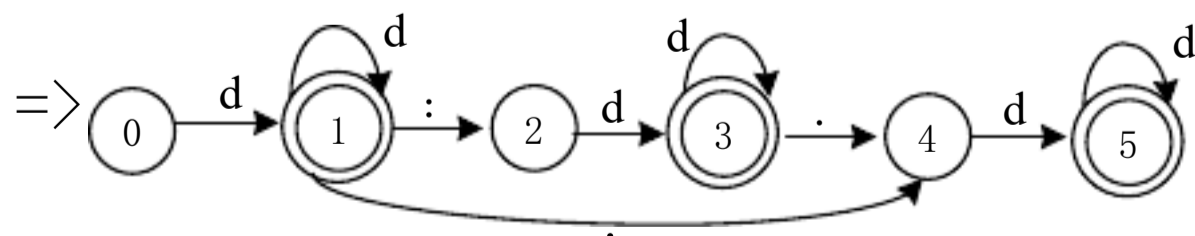
15. 解答:

① 正规式: $(dd^* | \epsilon)dd^*(dd^* | \epsilon)$, d 代表 a~z 的字母

② NFA 为:



③ DFA 为:



16. 解答:

词法分析器对源程序采取非常局部的观点, 因此象 C 语言的语句


```
fi (a == f(x)) ...
```

中，词法分析器把 `fi` 当作一个普通的标识符交给编译的后续阶段，而不会把它看成是关键字 `if` 的拼写错。

PASCAL 语言要求作为实型常量的小数点后面必须有数字，如果程序中出现小数点后面没有数字情况，它由词法分析器报错。

17. 解答：

此时编译器认为

```
/* then part
return q
else
/* else part */
```

是程序的注释，因此它不可能再发现 `else` 前面的语法错误。

分析 这是注释用配对括号表示时的一个问题。注释是在词法分析时忽略的，而词法分析器对程序采取非常局部的观点。当进入第一个注释后，词法分析器忽略输入符号，一直到出现注释的右括号为止，由于第一个注释缺少右括号，所以词法分析器在读到第二个注释的右括号时，才认为第一个注释处理结束。

为克服这个问题，后来的语言一般都不用配对括号来表示注释。例如 **Ada** 语言的注释始于双连字符 (`--`)，随行的结束而终止。如果用 **Ada** 语言的注释格式，那么上面函数应写成

```
long gcd(p,q)
long p,q;
{
    if (p%q == 0)
    -- then part
    return q
    else
    --else part
    return gcd(q, p%q);
}
```

18. 解答：略！

章习题解答:

2, 3, 4 解答 略!

解答:

(1) × (2) √ (3) × (4) √ (5) √ (6) √ (7) × (8) ×

6. 解答:

(1)A: ④ B : ③ C : ③ D : ④ E : ②

(2)A: ④ B : ④ C : ③ D : ③ E : ②

7. 解答:

消除给定文法中的左递归, 并提取公因子:

$bterm \rightarrow bfactor \{and\ bfactor\}$

$bfactor \rightarrow not\ bfactor \mid (bexpr) \mid true \mid false$

(2) 用类 C 语言写出其递归分析程序:

```
void bexpr();
{
    bterm();
    WHILE(lookahead == 'or') {
        match ('or');
        bterm();
    }
}

void bterm();
{
    bfactor();
    WHILE(lookahead == 'and'){
        match ('and');
        bfactor();
    }
}

void bfactor();
{
    if (llokahead == 'not') then {
        match ('not');
        bfactor();
    }
    else if(lookahead == '(') then {
        match ( '(' );
        bexpr();
        match (')');
    }
    else if(lookahead == 'true')
        then match ('true')
    else if (lookahead == 'false')
        then match ('false');
    else error;
}
```

8. 解答:

消除所给文法的左递归, 得 G':

$S \rightarrow (L) \mid a$

$L \rightarrow SL'$

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

实现预测分析器的不含递归调用的一种有效方法是使用一张分析表和一个栈进行联合控制, 下面构造预测分析表:

根据文法 G' 有:

$First(S) = \{ (, a \}$

$Follow(S) = \{), , , \# \}$

$First(L) = \{ (, a \}$

$Follow(L) = \{) \}$

$First(L') = \{ , \}$

$Follow(L') = \{) \}$

按以上结果, 构造预测分析表 M 如下:

非终结符号	输入符号				
	()	,	a	#
S	$S \rightarrow (L)$			$S \rightarrow a$	
L	$L \rightarrow SL'$			$L \rightarrow SL'$	
L'		$L' \rightarrow \epsilon$	$L' \rightarrow ,SL'$		

G是LL(1)的，因为它的LL(1)分析表不含多重定义入口。
 预测分析器对输入符号串(a,(a,a)) 做出的分析动作如下：

步骤	栈	剩余输入串	输出
1	#S	(a,(a,a))#	#
2)L(a,(a,a))#	
3)L	a,(a,a))#	
4)L'S	a,(a,a))#	$L \rightarrow SL'$
5)L'a	a,(a,a))#	$S \rightarrow a$
6)L'	,a,a))#	
7)L',S,	,a,a))#	$L' \rightarrow ,SL'$
8)L'S	(a,a))#	
9)L')L((a,a))#	$S \rightarrow (L)$
10)L')L	a,a))#	
11)L')L'S	a,a))#	$L \rightarrow SL'$
12)L')L'a	a,a))#	$S \rightarrow a$
13)L')L'	,a))#	
14)L')L',S,	,a))#	$L' \rightarrow ,SL'$
15)L')L'S	a))#	
16)L')L'a	a))#	$S \rightarrow a$
17)L')L')#	
18)L'))#	$L' \rightarrow$
19)L')#	
20))#	$L' \rightarrow \square$
21	#	#	

解答：

各非终结符的 First 集：

$$\square \cup \square \cup \{\square \cup \{b\}\} = \{a, b, \square\}$$

$$\text{First}(A) = \{b\} \cup \{\square\} = \{b, \square\}$$

$$\text{First}(B) = \{\square\} \cup \{a\} = \{a, \square\}$$

$$\square \cup \text{First}(D) \cup \text{First}(b) = \{a, b, c\}$$

$$\text{First}(D) = \{a\} \cup \{c\} = \{a, c\}$$

各个候选式的 First 集为：

$$\text{First}(AB) = \{a, b, \square\} \quad \text{First}(bC) = \{b\}$$

$$\text{First}(\square) = \{\square\} \quad \text{First}(b) = \{b\}$$

$$\text{First}(aD) = \{a\} \quad \text{First}(AD) = \{a, b, c\}$$

$$\text{First}(b) = \{b\} \quad \text{First}(aS) = \{a\}$$

$$\text{First}(c) = \{c\}$$

各非终结符的 Follow 集的计算：

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

$$\}) \cup \text{Follow}(S) \cup \text{First}(D) = \{a, \#, c\}$$

$$\text{Follow}(B) = \text{Follow}(S) = \{\#\}$$

$$\text{Follow}(C) = \text{Follow}(S) = \{\#\}$$

$$\text{Follow}(D) = \text{Follow}(B) \cup \text{Follow}(C) = \{\#\}$$

. 解答:

(1) 求 First 和 Follow 集

$$\begin{aligned} \text{First}(E) = \text{First}(T) &= \{(, a, b, \wedge\} && \textcircled{7} \\ \text{First}(E') &= \{+, \square\} && \textcircled{6} \\ \text{First}(T) = \text{First}(F) &= \{(, a, b, \wedge\} && \textcircled{4} \\ \text{First}(T') &= \{(, a, b, \wedge, \square\} && \textcircled{5} \\ \text{First}(F) = \text{First}(P) &= \{(, a, b, \wedge\} && \textcircled{3} \\ \text{First}(F') &= \{*, \square\} && \textcircled{2} \\ \text{First}(P) &= \{(, a, b, \wedge\} && \textcircled{1} \text{ (计算顺序)} \end{aligned}$$

$$\begin{aligned} \text{Follow}(E) &= \{\#,)\} && (1, 7) \\ \text{Follow}(E') &= \text{Follow}(E) = \{\#,)\} && (1) \text{ (使用的产生式)} \\ \text{Follow}(T) &= \text{First}(E') \setminus \{\square\} \cup \text{Follow}(T') && (1, 4) \\ &= \{+\} \cup \{), \#\} = \{+,), \#\} \\ \text{Follow}(T') &= \text{Follow}(T) = \{+,), \#\} && (3) \\ \text{Follow}(F) &= \text{First}(T') \setminus \{\square\} \cup \text{Follow}(T) && (3, 4) \\ &= \{(, a, b, \wedge, +,), \#\} \\ \text{Follow}(F') &= \text{Follow}(F) && (5) \\ &= \{(, a, b, \wedge, +,), \#\} \\ \text{Follow}(P) &= \text{First}(F') \setminus \{\square\} \cup \text{Follow}(F) && (5, 6) \\ &= \{*, (, a, b, \wedge, +,), \#\} \end{aligned}$$

(2) 证明

- ∵ a. 文法不含左递归;
- b. 每个非终结符的各个候选式的 First 集不相交;
- c. $\text{First}(E') \cap \text{Follow}(E') = \{+, \square\} \cap \{\#,)\} = \emptyset$
 $\text{First}(T') \cap \text{Follow}(T') = \{(, a, b, \wedge, \square\} \cap \{+,)\} = \emptyset$
 $\text{First}(F') \cap \text{Follow}(F') = \{*, \square\} \cap \{(, a, \wedge, +,), \#\} = \emptyset$

∴ 改造后的文法满足 LL(1)文法的三个条件, 是 LL(1)文法。

(3) 预测分析表如下所示。

	a	b	*	+	∧	()	#
E		E → TE'			E → TE'	E → TE'		
E'				E' → +E			E' → □	E' → □
T	T → FT'	T → FT'			T → FT'	T → FT'		
T'	T' → T	T' → T		T' → □	T' → T	T' → T	T' → □	T' → □
F	F → PF'	F → PF'			F → PF'	F → PF'		
F'	F' → □	F' → □	F' → *F'	F' → □	F' → □	F' → □	F' → □	F' → □
P	P → a	P → b			P → ∧	P → (E)		

11. 解答:

(1)

$S \rightarrow Abc$
$A \rightarrow a \mid \square$
$B \rightarrow b \mid \square$

- b. S,A,B 各候选式的 First 集不相交;
- c. $First(A) \cap Follow(A) = \{a, \} \cap \{b\} = \square$
 $First(B) \cap Follow(B) = \{b, \} \cap \square = \square$

∴该文法为 LL(1)文法。

(2)

$S \rightarrow Ab$
$A \rightarrow a \mid B \mid \square$
$B \rightarrow b \mid \square$

- a. 文法不含左递归;
 - b. S,A,B 各候选式的 First 集不相交;
 - c. $First(A) \cap Follow(A) = \{a,b, \} \cap \square = \square$
- ∴ 该文法不是 LL(1)文法。

12. 解答:

① 最右推导:

$E \Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow (E + T) \Rightarrow (E + F) \Rightarrow (E + i) \Rightarrow (T + i) \Rightarrow (T * F + i)$

语法树:

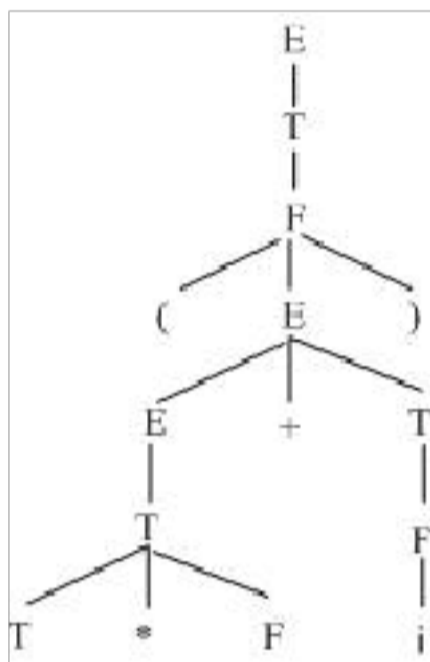


图4.1 句型(T*F+i)的语法树

② 短语: (T*F+i), T*F+i, T*F, i

素短语: T*F, i

最左素短语: T*F

③ 由于 $E \Rightarrow E + T \Rightarrow E + T * F$, 故 $E + T * F$ 为该文法的句型

短语: T*F、E+T*F

直接短语: T*F

句柄: T*F

13. 解答:

最左推导:

$S \Rightarrow (T) \Rightarrow (T,S) \Rightarrow (S,S) \Rightarrow (a,S) \Rightarrow (a,(T)) \Rightarrow (a,(T,S))$
 $\Rightarrow (a,(S,S)) \Rightarrow (a,(a,S)) \Rightarrow (a,(a,a))$

最右推导:

$S \Rightarrow (T) \Rightarrow (T,S) \Rightarrow (T,(T)) \Rightarrow (T,(T,S)) \Rightarrow (T,(T,a))$
 $\Rightarrow (T,(T,a)) \Rightarrow (T,(a,a)) \Rightarrow (S,(a,a)) \Rightarrow (a,(a,a))$

文法中 S 和 T 的 FirstVT 和 LastVT 集为:

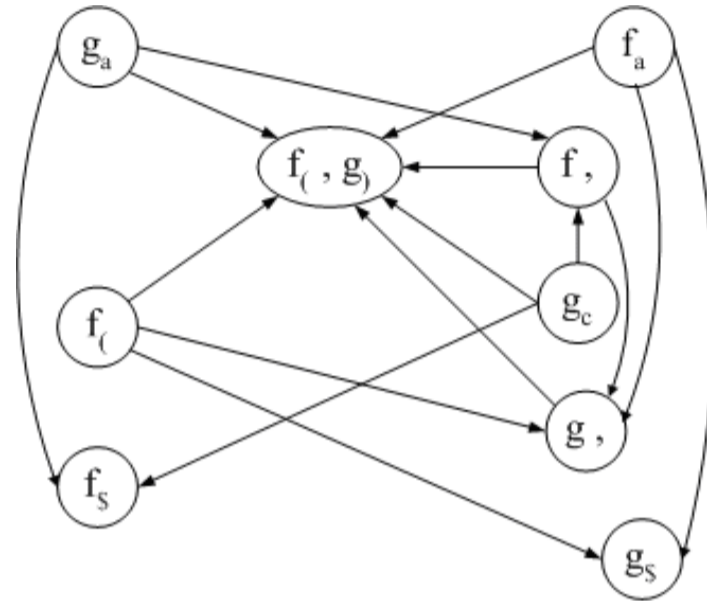
$FirstVT(S) = \{a, (\}$ $FirstVT(T) = \{., a, (\}$

$lastVT(S) = \{a,) \}$ $lastVT(T) = \{., a,) \}$

文法 G[S] 的算符优先关系表:

	a	()	,	#
a			·>	·>	·>
(<·	<·	≡	<·	
)			·>	·>	·>
,	<·	<·	·>	·>	
#	<·	<·			≡

对每个终结符或 建立符号 f 与 g, 把 f 和 g 分成一组。根据 G[S] 的算符优先关系表, 画出如下的有向图。



优先函数如下:

	a	()	,	#
f	2	0	2	2	0
g	3	3	0	1	0

用算符优先分析法分析句子(a,(a,a))。

栈	输入	动作
#	(a, (a, a))#	初始
#(a, (a, a))#	移进
#(a	, (a, a))#	移进
#(N	, (a, a))#	归约
#(N,	(a, a))#	移进
#(N,(a, a))#	移进
#(N,(a	, a))#	移进
#(N,(N	, a))#	归约
#(N,(N,	a))#	移进
#(N,(N,a)#	移进
#(N,(N,N)#	归约
#(N,(N)#	归约
#(N,(N))#	移进
#(N,N)#	归约
#(N)#	归约
#(N)	#	移进
#N	#	归约, 接受

解答:

(1) . 拓广文法

(0) $S' \rightarrow S$

(1) $S \rightarrow aS$

(2) $S \rightarrow bS$

(3) $S \rightarrow c$

(2) . 写出所有的项目

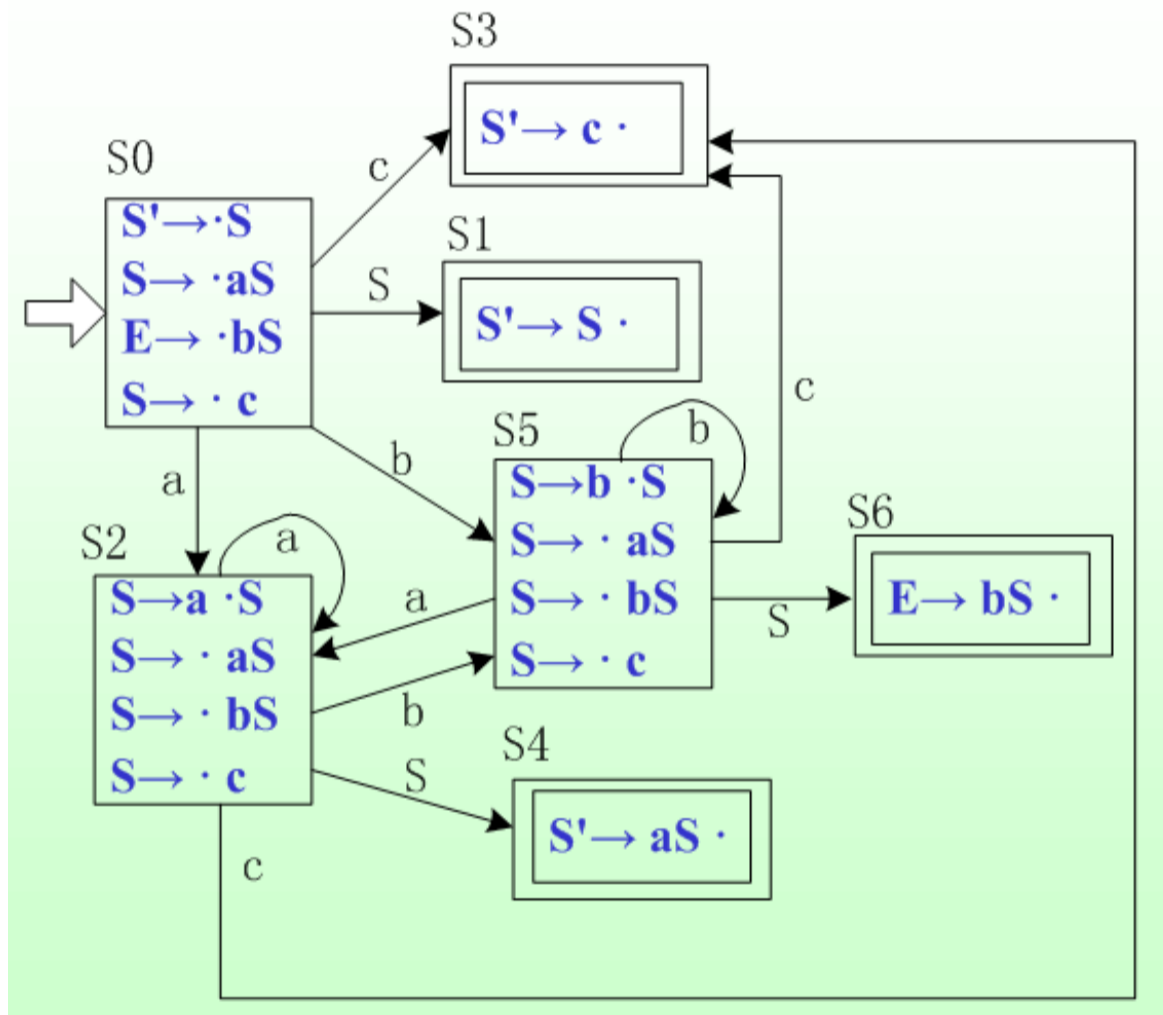
1. $S' \rightarrow \cdot S$ 2. $S' \rightarrow S \cdot$

3. $S \rightarrow \cdot aS$ 4. $S \rightarrow a \cdot S$ 5. $S \rightarrow aS \cdot$

6. $S \rightarrow \cdot bS$ 7. $S \rightarrow b \cdot S$ 8. $S \rightarrow bS \cdot$

9. $S \rightarrow \cdot c$ 10. $S \rightarrow c \cdot$

(3) . 求项目集规范族



每个项目集中的各个项目不冲突,则是 LR(0)文法。

(4) . 构造 LR(0)分析表

状态	ACTION				GOTO
	a	b	c	#	
S_0	S_2	S_5	S_3		1
S_1				acc	
S_2	S_2	S_5	S_3		4
S_3	r_3	r_3	r_3	r_3	
S_4	r_1	r_1	r_1	r_1	

S_5	S_2	S_5	S_3		6
S_6	r_2	r_2	r_2	r_2	

15. 解答:

(1) 该文法的拓广文法 G 为

0. $S' \rightarrow S$
1. $S \rightarrow Sab$
2. $S \rightarrow bR$
3. $R \rightarrow S$
4. $R \rightarrow a$

其 LR(0) 项目集规范族和识别活前缀的 DFA 如下:

$$I_0 = \{S' \rightarrow \cdot S, S \rightarrow \cdot Sab, S \rightarrow \cdot bR\}$$

$$I_1 = \{S' \rightarrow S \cdot, S \rightarrow S \cdot ab\}$$

$$I_2 = \{S \rightarrow b \cdot R, R \rightarrow \cdot S, R \rightarrow \cdot a, S \rightarrow \cdot Sab, S \rightarrow \cdot bR\}$$

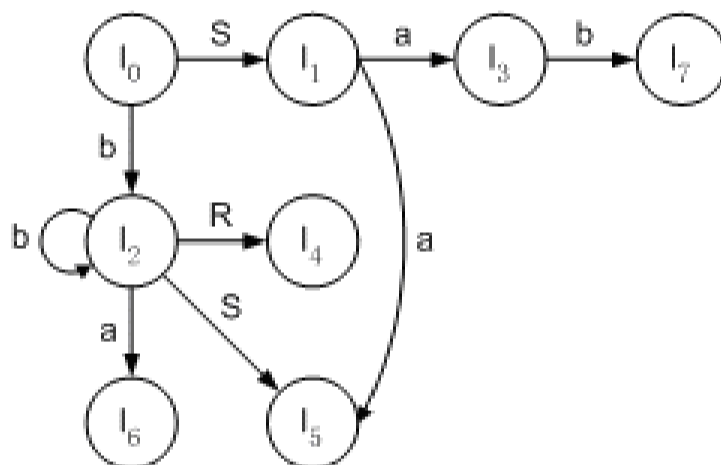
$$I_3 = \{S \rightarrow Sa \cdot b\}$$

$$I_4 = \{S \rightarrow bR \cdot\}$$

$$I_5 = \{R \rightarrow S \cdot, S \rightarrow S \cdot ab\}$$

$$I_6 = \{R \rightarrow a \cdot\}$$

$$I_7 = \{S \rightarrow Sab \cdot\}$$



显然, I_1 和 I_5 存在移进-归约冲突。求 S' 和 R 的 Follow 集:

$$\text{Follow}(S') = \{\#\}$$

$$\text{Follow}(R) = \text{Follow}(S) = \{a, \#\}$$

在 I_5 中, 出现的移进-归约冲突, 且 $\text{Follow}(R) \cap \{a\} = \{a\}$, 不能用 SLR(1) 方法解决。因此, 此文法不是 SLR(1) 文法。

(2) 该文法的拓广文法 G' 为

0. $S' \rightarrow S$
1. $S \rightarrow aSAB$
2. $S \rightarrow BA$
3. $A \rightarrow aA$
4. $A \rightarrow B$
5. $B \rightarrow b$

其 LR(0) 项目集规范族和识别活前缀的 DFA 如下:

$$I_0 = \{S' \rightarrow \cdot S, S \rightarrow \cdot aSAB, S \rightarrow \cdot BA, B \rightarrow \cdot b\}$$

$$I_1 = \{S' \rightarrow S \cdot\}$$

$$I_2 = \{B \rightarrow b \cdot\}$$

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/278124101042006024>