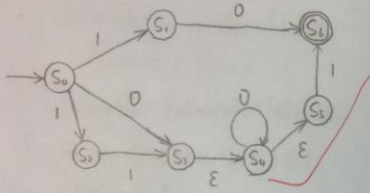


1、共 (15 分)

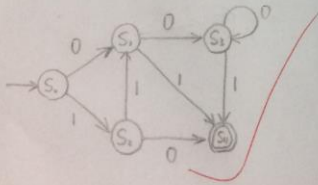
构造识别正规式  $10 | (0 | 11)0^*1$  的极小化 DFA  $M$ 。

首先构造识别该正规式的 NFA



用子集构造法构造对应 DFA

子集	DFA 状态
$\{S_0\}$	$S_0$
$\{S_1, S_4, S_5\}$	$S_1$
$\{S_2, S_3\}$	$S_2$
$\{S_4, S_5\}$	$S_3$
$\{S_6\}$	$S_4$

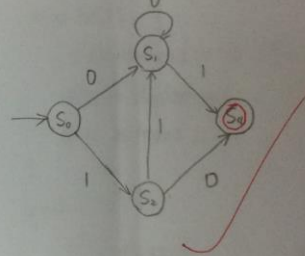


对该 DFA 应用极小化方法

有划分  $I_0 = \{0, 1, 2, 3\}$ ,  $I_4 = \{4\}$

$\rightarrow 0: I_0 = \{0, 1, 3\}$ ,  $I_2 = \{2\}$ ,  $I_4 = \{4\}$

$\rightarrow 1: I_0 = \{0\}$ ,  $I_1 = \{1, 3\}$ ,  $I_2 = \{2\}$ ,  $I_4 = \{4\}$

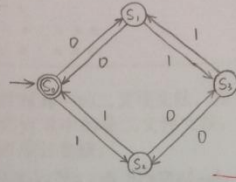


2、共 (10 分)

针对正规集  $L = \{0 \text{ 和 } 1 \text{ 的个数均为偶数的 } 0-1 \text{ 串}\}$ :

- (1) 先直接给出识别  $L$  的极小化 DFA  $M$ ;
- (2) 再给出描述  $L$  的正规式  $R$ 。

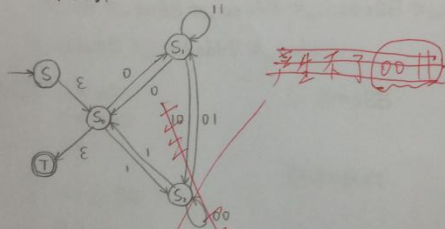
(1)



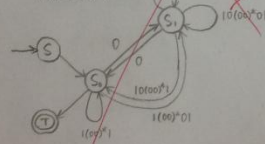
(2) 增加新的开始状态  $S$  和终止状态  $T$ .

它们分别经  $\epsilon$  指向  $S_0$  和经  $\epsilon$  离开  $S_3$ .

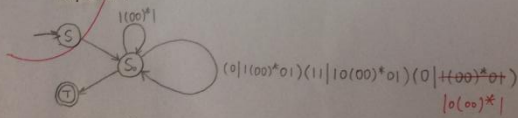
删去  $S_1$ :



删去  $S_2$ :



删去  $S_3$ :



得正规式:

$$\left( \left( \epsilon | 0(00)^*01 | 11 | 10(00)^*01 \right) \left( 0 | 1(00)^*01 \right) \right) \left( 0 | 1(00)^*01 \right)^* \left( \epsilon | 1(00)^*1 \right)^*$$

2. 共 (10 分)

针对正规集  $L = \{0 \text{ 和 } 1 \text{ 的个数均为偶数的 } 0-1 \text{ 串}\}$ ;  
 (1) 先直接给出识别  $L$  的极小化 DFA  $M$ ;  
 (2) 再给出描述  $L$  的正规式  $R$ .



已为 DFA.

极小化: 终态集合  $I_0 = \{1\}$ , 非终态集合  $I_1 = \{2, 3, 4\}$

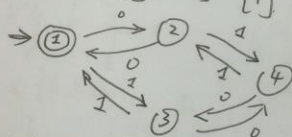
$2 \xrightarrow{0} 1 \in I_0, 3 \xrightarrow{0} 4 \in I_1, 4 \xrightarrow{0} 3 \in I_1$

$\therefore I_1$  分为  $I_2 = \{2\}, I_3 = \{3, 4\}$

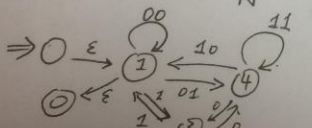
又:  $3 \xrightarrow{1} 2 \in I_0, 4 \xrightarrow{1} 3 \in I_3$

$\therefore I_3$  划分为  $I_4 = \{3\}, I_5 = \{4\}$

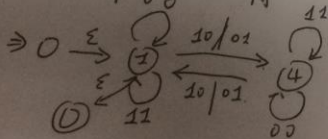
$\therefore M$  为



(2) 添加新的开始  $\epsilon$  连接状态后, 再去掉状态 2. 得



再去掉状态 3. 得



再去掉状态 4. 得正规式为

$$\left\{ (11|00)^* [(10|01)(11|00)^*(10|01)]^* \right\}^*$$

即

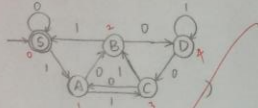
$$\left[ (11|00) \mid (10|01)(11|00)^*(10|01) \right]^*$$

3、共 (20 分)

- (1) 给出产生可被 5 整除的二进制串集(含空串)的上下文无关文法  $G_0$ ;  
 (2) 并针对  $G_0$ , 给出 First 与 Follow 集合, LL(1) 分析表;  
 (3) 为  $G_0$  设计相应的递归下降分析程序。

- (1)  $S \rightarrow 0S \mid 1A \mid \epsilon$   
 $A \rightarrow 0B \mid 1C$   
 $B \rightarrow 1S \mid 0D$   
 $C \rightarrow 0A \mid 1B$   
 $D \rightarrow 0C \mid 1D$

(对应 DFA:



- (2)  $First(S) = \{0, 1, \epsilon\}$   
 $First(A) = First(B) = First(C)$   
 $= First(D) = \{0, 1\}$   
 $Follow(S) = Follow(A) = Follow(B)$   
 $= Follow(C) = Follow(D) = \{\$, \}$

LL(1) 分析表:

	0	1	\$
S	$S \rightarrow 0S$	$S \rightarrow 1A$	$S \rightarrow \epsilon$
A	$A \rightarrow 0B$	$A \rightarrow 1C$	
B	$B \rightarrow 0D$	$B \rightarrow 1S$	
C	$C \rightarrow 0A$	$C \rightarrow 1B$	
D	$D \rightarrow 0C$	$D \rightarrow 1D$	

```

(3) void match (terminal t) {
    if (lookahead == t) lookahead = nextToken();
    else error();
}

void S() {
    if (lookahead == '0') { match('0'); S(); }
    else if (lookahead == '1') { match('1'); A(); }
    else if (lookahead == '$') { succeed(); }
    else error();
}

void A() {
    if (lookahead == '0') { match('0'); B(); }
    else if (lookahead == '1') { match('1'); C(); }
    else error();
}

void B() {
    if (lookahead == '0') { match('0'); D(); }
    else if (lookahead == '1') { match('1'); S(); }
    else error();
}

void C() {
    if (lookahead == '0') { match('0'); A(); }
    else if (lookahead == '1') { match('1'); B(); }
    else error();
}

void D() {
    if (lookahead == '0') { match('0'); C(); }
    else if (lookahead == '1') { match('1'); D(); }
    else error();
}
    
```

4、共 (20 分)

文法 G1 和 G2 中有一个是二义性文法，另一个是非二义性文法。

文法 G1, s 为开始符号。  
 $S \rightarrow aBS \mid bAS$   
 $S \rightarrow \varepsilon$   
 $A \rightarrow a \mid bAA$   
 $B \rightarrow b \mid aBB$

文法 G2, s 为开始符号。  
 $S \rightarrow aB \mid bA$   
 $S \rightarrow \varepsilon$   
 $A \rightarrow aS \mid bAA$   
 $B \rightarrow bS \mid aBB$

aababb

- (1) 针对其中的二义性文法，用串 abaabb 证明其二义性；  
 (2) 针对其中的非二义性文法，给出读过活前缀 aBaaBB 经过的所有 LR(0) 项目集簇。

(1) 文法 G2 是二义的，因为它对 aababb 有两个最左推导：

$S \Rightarrow aB \Rightarrow aBB \Rightarrow aabSB \Rightarrow aabB \Rightarrow aabaBB \Rightarrow aababSB$   
 $\Rightarrow aababB \Rightarrow aababBS \Rightarrow aababb$

$S \Rightarrow aB \Rightarrow aBB \Rightarrow aabSB \Rightarrow aabaBB \Rightarrow aababSB$   
 $\Rightarrow aababB \Rightarrow aababBS \Rightarrow aababb$

(2)  $I_0: S \rightarrow \cdot aBS$   
 $S \rightarrow \cdot bAS$   
 $S \rightarrow \cdot$

$I_5: B \rightarrow aBB \cdot$

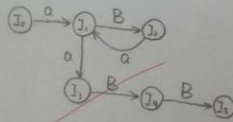
$I_1: S \rightarrow a \cdot BS$   
 $B \rightarrow \cdot b$   
 $B \rightarrow \cdot aBB$

$I_2: S \rightarrow aB \cdot S$   
 $S \rightarrow \cdot aBS$   
 $S \rightarrow \cdot bAS$   
 $S \rightarrow \cdot$

$I_3: B \rightarrow a \cdot BB$   
 $B \rightarrow \cdot b$   
 $B \rightarrow \cdot aBB$

$I_4: B \rightarrow aB \cdot B$   
 $B \rightarrow \cdot b$   
 $B \rightarrow \cdot aBB$

其中有转换关系：



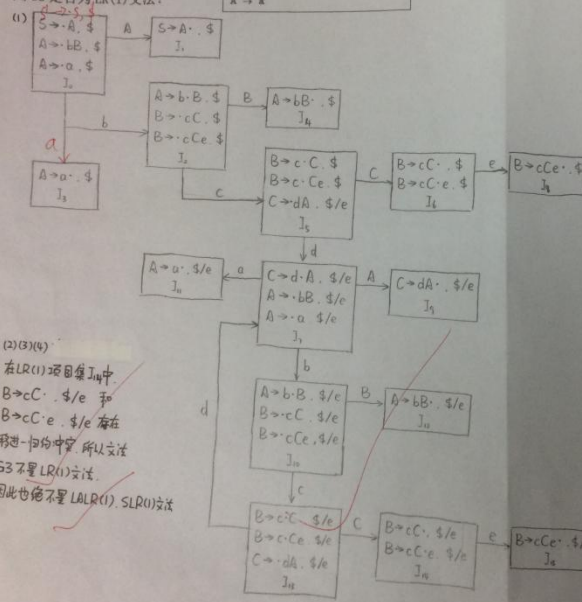


5. 共 (20 分)

针对右图中的文法 G3,

- (1) 给出 LR(1) 项目集族
- (2) G3 是否为 SLR(1) 文法?
- (3) G3 是否为 LALR(1) 文法?
- (4) G3 是否为 LR(1) 文法?

文法 G3, a 为开始符号。  
 $S \rightarrow A$   
 $A \rightarrow bB$   
 $B \rightarrow cC$   
 $C \rightarrow dA$   
 $A \rightarrow a$



(2)(3)(4)  
 在 LR(1) 项目集  $I_{10}$  中,  
 $B \rightarrow cC \cdot \$/e$  和  
 $B \rightarrow cCe \cdot \$/e$  存在  
 移进-归约冲突, 所以文法  
 G3 不是 LR(1) 文法,  
 因此它也不是 LALR(1), SLR(1) 文法

6. 共 (15 分)

针对以下 C/C++ 程序:

```
int j, a[10];
void main()
{
    int *p = a;
    for(j=0; j<10 ;++) a[j] = 0; //语句 1
    for(j=0; j<10 ;++) _____; //语句 2
    for(j=0; j<10 ;++) cout << a[j] << " "; //语句 3
}
```

在语句 2 的下划线处, 分别进行如下三种操作:

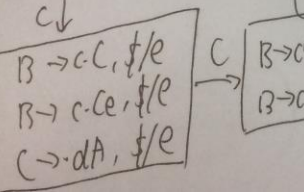
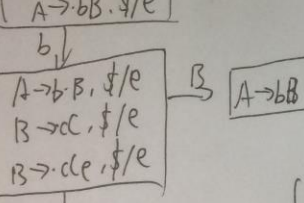
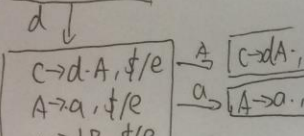
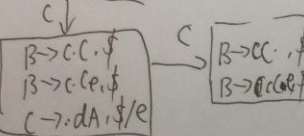
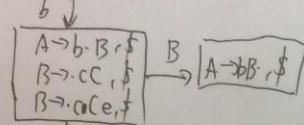
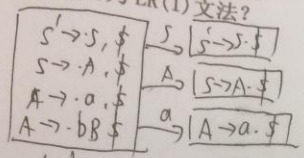
- (1)  $*p++ = j * 10$
- (2)  $++*p = j * 10$
- (3)  $++*p++ = j * 10$

试分别给出语句 3 的输出结果。

(1)	0	10	20	30	40	50	60	70	80	90
(2)	90	0	0	0	0	0	0	0	0	0
(3)	0	10	20	30	40	50	60	70	80	90

LR(1) 文法?

$C \rightarrow dA$   
 $A \rightarrow a$



d

这个文法不是 LR(1) 文法，  
不是 LR(1) 文法，  
不是 LR(1) 文法，  
对串 bcd bcd a e 的分析，有：

$S \Rightarrow A \Rightarrow bB \Rightarrow bcC \Rightarrow bcdA$   
 $\Rightarrow bcd bB \Rightarrow bcd b cCe \Rightarrow bcd b c d a e$   
 $\Rightarrow bcd b d a e$

$S \Rightarrow A \Rightarrow bB \Rightarrow bcCe \Rightarrow bcd a e$   
 $\Rightarrow bcd b c e \Rightarrow bcd b c e \Rightarrow bcd b c d a e$   
 $\Rightarrow bcd b c d a e$

该文法是二义性文法，故不是 LR(1), LALR(1), SLR(1) 文法。  
同样，对栈式，不论是 LR(1), LALR(1), SLR(1) 文法，  
都不能判断对输入串是归约还是移进，  
存在归约-移进冲突，故不是 LR(1), LALR(1), SLR(1) 文法。

(\*)