

中国矿业大学 2020~2021 学年第二学期

《编译技术》复习试卷

考试时间：114.514 分钟 考试方式：闭卷

命题人：不愿透露名字，因为题量有点大

学院	班级	姓名	学号		
题号	一	二	三	四	总分
得分					
阅卷人					

一、（10 分）解决以下问题：

（1）设计一个文法，使其语言是奇数集，且每个奇数不以 0 开头。

（2）设计 DFA 以识别所有能被 3 整除的无符号十进制数。

二、（10 分）已知 $NFA = (\{x, y, z\}, \{0, 1\}, M, \{x\}, \{z\})$ ，其中：

$M(x, 0) = \{z\}$, $M(y, 0) = \{x, y\}$, $M(z, 0) = \{x, z\}$, $M(x, 1) = \{x\}$, $M(y, 1) = \varnothing$, $M(z, 1) = \{y\}$ ，构造相应的 DFA 并最小化。

三、（10 分）假定语言 X 的字母表 $\Sigma = \{A-Z, a-z, 0-9, ;, =\}$ ，单词符号定义如下：

1、标识符：字母打头的字母数字串；2、无符号整数：无符号数字串；3、分界符：；4、运算符=
写出构造词法分析程序伪代码。

四、（15 分）已知文法 $G(S)$ ： $S \rightarrow Sa|Nb|c$ $N \rightarrow Sd|Ne|f$

（1）试构造等价的无左递归的文法。

（2）写出递归预测分析程序的伪代码。

五、（10 分）已知文法： $S \rightarrow a|^(T)$ $T \rightarrow T,S|S$

（1）写出 $(a, (S, a))$ 的短语、直接短语、句柄。

（2）求对 $((((a, a), ^, (a)), a))$ 的最左推导。

六、（10 分）对下面的文法 G ： $E \rightarrow TE'$ $E' \rightarrow +E|\epsilon$ $T \rightarrow FT'$ $T' \rightarrow >T|\epsilon$ $F \rightarrow PF'$ $F' \rightarrow *F|\epsilon$ $P \rightarrow (E)|a|b|^\wedge$
计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

七、（10 分）已知文法为： $A \rightarrow aAd|aAb|\epsilon$

（1）判断该文法是否是 LR(0)文法，是否是 SLR(1) 文法

（2）若是 SLR(1) 文法，构造相应分析表

（3）对输入串 $ab\#$ 给出分析过程

八、（15 分）已知文法 G ： $S \rightarrow (L)|a$ $L \rightarrow L,S|S$

（1）写一个翻译方案，它输出配对括号的个数。例如：对于 $(a, (a, a))$ ，输出的结果是 2

（2）写一个翻译方案，它输出每个 a 的嵌套深度。例如：对于 $(a, (a, a))$ ，输出的结果是 1 2 2

（3）写一个翻译方案，它打印每个 a 在句子中是第几个字符。例如：对于 $(a, (a, a))$ ，输出的结果是 2 5 7

九、（10 分）将下面的语句翻译成四元式序列：

```
while A<C and B<D do
  if A=1 then C:=C+1
  else while A≤D do
    A:=A+2;
```

中国矿业大学 2020~2021 学年第二学期

《编译技术》复习试卷参考答案

一、设计文法、设计 DFA。

(1) 答案不唯一。

1. 文法 $G(N)$:

$N \rightarrow AB|B$
 $A \rightarrow AC|D$
 $B \rightarrow 1|3|5|7|9$
 $D \rightarrow B|2|4|6|8$
 $C \rightarrow 0|D$

2. 文法 $G(S)$:

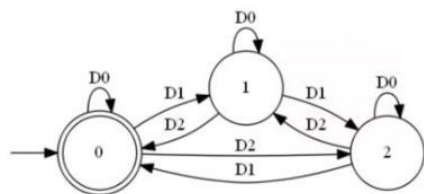
$S \rightarrow ABC|C$
 $C \rightarrow 1|3|5|7|9$
 $B \rightarrow 0|1|2|3|4|5|6|7|8|9|\epsilon|BB$
 $A \rightarrow 1|2|3|4|5|6|7|8|9$

(2) 所以我们可以通过记录十进制数每位数与三相除的余数，如果最后的余数之和能被三整除那该数就可以被三整除；否则不能被三整除。（除三余一的数和除三余二的数组组合一定能被三整除，除三余一的数和除三余一的数除三组合一定余二）

$D0 = 0|3|6|9$

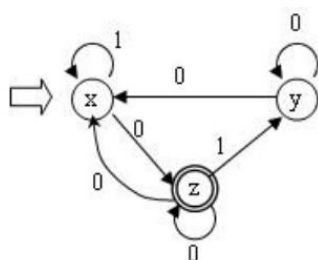
$D1 = 1|4|7$

$D2 = 2|5|8$



二、NFA 的确定化，DFA 的最小化。

绘制 NFA。根据闭包进行 NFA 的确定化（这个求闭包不像 ϵ -closure，每次只多进行一步，而不是连续走）。



I	$I_0 = \epsilon\text{-closure}(\text{MoveTo}(I, 0))$	$I_1 = \epsilon\text{-closure}(\text{MoveTo}(I, 1))$
A[x]	B[z]	A[x]
B[z]	C[x, z]	D[y]
C[x, z]	C[x, z]	E[x, y]
D[y]	E[x, y]	
E[x, y]	F[x, y, z]	A[x]
F[x, y, z]	F[x, y, z]	E[x, y]

其中，含 z 的是终态，即 BCF 是终态。

使用划分法进行 DFA 确定化。

	0	1	
A	B	A	1
D	E	-	
E	F	A	
B	C	D	2
C	C	E	
F	F	E	

	0	1	
A	B	A	1
D	E	-	3
E	F	A	
B	C	D	2
C	C	E	
F	F	E	

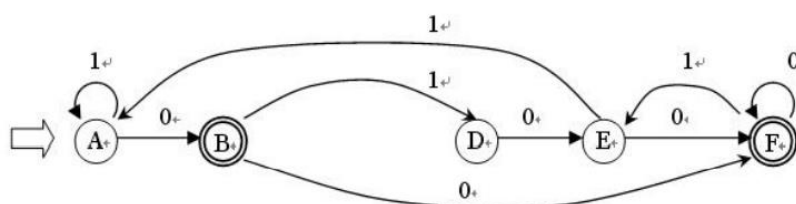
	0	1	
A	B	A	1
D	E	-	3
E	F	A	4
B	C	D	2
C	C	E	
F	F	E	

	0	1	
A	B	A	1
D	E	-	3
E	F	A	4
B	C	D	2
C	C	E	5
F	F	E	

观察，直接 C=F 就行

修改上面的表格得到最小 DFA：

	0	1
A	B	A
D	E	-
E	F	A
B	F	D
F	F	E



三、词法分析程序

GETNEXTCHAR() ; 例 1 程序语言的词法分析程序

SWITCH(CHCODE);

CASE 1:

WHILE (ISLETTER OR ISDIGIT) DO

SAVE(); // 当前字符放入一临时字符数组;

GETNEXTCHAR(); //从缓冲区取下一字符

UNGETCH; //回退一字符

OUTPUT(1, 标识符名字);

BREAK;

CASE 2:

WHILE ISDIGIT DO

SAVE(); // 当前字符放入一临时字符数组;

GETNEXTCHAR ; //从缓冲区取下一字符

UNGETCH; //回退一字符

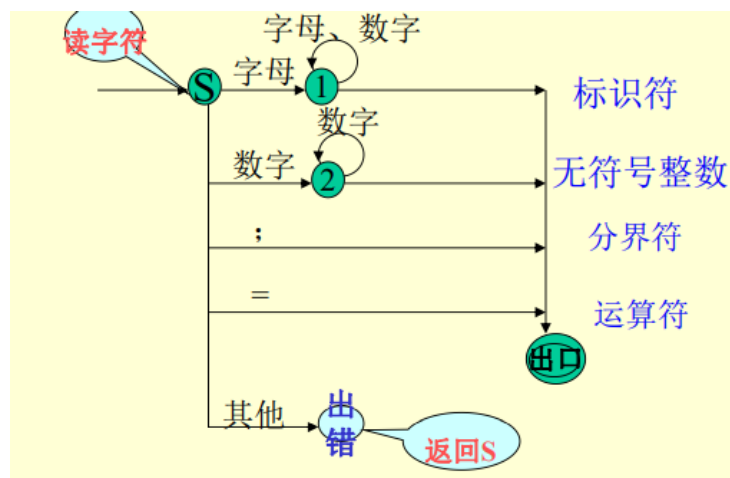
OUTPUT (2, 整数);

BREAK;

CASE 3 OUTPUT (3 ,“;”); BREAK;

CASE 4 : OUTPUT (4 ,“=”); BREAK;

DEFAULT: Error();



诚信关乎个人一生，公平竞争赢得尊重。

以下行为是严重作弊行为，学校将给予留校察看或开除学籍处分：1. 替他人考试或由他人替考；2. 通讯工具作弊；3. 团伙作弊。

四、消除左递归，递归分析伪代码。

$S \rightarrow fN'bS'|cS'$

$S' \rightarrow aS'|dN'bS'|\varepsilon$

$N' \rightarrow eN'|\varepsilon$

procedure S;

if sym='f'

 advance;

 N';

 if sym='b'

 advance;

 S';

 else error;

else if sym='c'

 advance;

 S';

else error;

procedure S';

if sym='a'

 advance;

 S';

else if sym='d'

 advance;

 N';

 if sym='b'

 advance;

 S';

 else error;

procedure N';

if sym='e'

 advance;

 N';

诚信关乎个人一生，公平竞争赢得尊重。

以下行为是严重作弊行为，学校将给予留校察看或开除学籍处分：1. 替他人考试或由他人替考；2. 通讯工具作弊；3. 团伙作弊。

五、短语、直接短语、句柄、最左推导。（注意箭头双横线）

短语： $a1 \quad S \quad a2 \quad S, a2 \quad (S, a2) \quad a1, (S, a2) \quad (a1, (S, a2))$

直接短语： $a1 \quad S \quad a2$

句柄： $a1$

对 $((a, a), ^, (a)), a$ 的最左推导为：

$$\begin{aligned} S &\Rightarrow (T) \Rightarrow (T, S) \Rightarrow (S, S) \Rightarrow ((T), S) \\ &\Rightarrow ((T, S), S) \Rightarrow ((T, S, S), S) \Rightarrow ((S, S, S), S) \\ &\Rightarrow (((T), S, S), S) \Rightarrow (((T, S), S, S), S) \Rightarrow (((S, S), S, S), S) \\ &\Rightarrow (((a, S), S, S), S) \Rightarrow (((a, a), S, S), S) \Rightarrow (((a, a), ^, S), S) \\ &\Rightarrow (((a, a), ^, (T)), S) \Rightarrow (((a, a), ^, (S)), S) \\ &\Rightarrow (((a, a), ^, (a)), S) \Rightarrow (((a, a), ^, (a)), a) \end{aligned}$$

六、求 First 和 Follow

FIRST 集合有：

$FIRST(E) = FIRST(T) = FIRST(F) = FIRST(P) = \{(a, b, ^)\};$

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

$FIRST(T) = FIRST(F) = FIRST(P) = \{(a, b, ^)\};$

$FIRST(T') = FIRST(T) \cup \{\varepsilon\} = \{(a, b, ^, \varepsilon)\};$

$FIRST(F) = FIRST(P) = \{(a, b, ^)\};$

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

$FIRST(P) = \{(a, b, ^)\};$

FOLLOW 集合有：

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

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

$FOLLOW(T) = FIRST(E') / \varepsilon \cup FOLLOW(E) = \{+, \varepsilon, \#\};$ // 不包含 ε

$FOLLOW(T') = FOLLOW(T) = FIRST(E') / \varepsilon \cup FOLLOW(E) = \{+, \varepsilon, \#\};$

$FOLLOW(F) = FIRST(T') \cup FOLLOW(T) = \{(a, b, ^, +, \varepsilon), \#\};$ // 不包含 ε

$FOLLOW(F') = FOLLOW(F) = FIRST(T') / \varepsilon \cup FOLLOW(T) = \{(a, b, ^, +, \varepsilon), \#\}$

$FOLLOW(P) = FIRST(F') / \varepsilon \cup FOLLOW(F') = \{(a, b, ^, +, *, \varepsilon), \#\}$

七、LR 自底向上分析。

增加一个非终结符 S' 后，产生原文法的增广文法有：

$S' \rightarrow A$

$A \rightarrow aAd | aAb | \varepsilon$

下面构造它的 LR(0)项目集规范族为：（最好画 DFA，可能题目就直接让你画）

当前符号	a	b	d	#	A
$I_0:$ $S' \rightarrow \cdot A$ $A \rightarrow \cdot aAd$ $A \rightarrow \cdot aAb$ $A \rightarrow \cdot$	$I_2:$ $A \rightarrow a \cdot Ad$ $A \rightarrow a \cdot Ab$ $A \rightarrow \cdot aAd$ $A \rightarrow \cdot aAb$ $A \rightarrow \cdot$				$I_1:$ $S' \rightarrow A \cdot$
$I_1:$ $S' \rightarrow A \cdot$				acc	
$I_2:$ $A \rightarrow a \cdot Ad$ $A \rightarrow a \cdot Ab$ $A \rightarrow \cdot aAd$ $A \rightarrow \cdot aAb$ $A \rightarrow \cdot$	I_2				$I_3:$ $A \rightarrow aA \cdot d$ $A \rightarrow aA \cdot b$
$I_3:$ $A \rightarrow aA \cdot d$ $A \rightarrow aA \cdot b$		$I_4:$ $A \rightarrow aAb \cdot$	$I_5:$ $A \rightarrow aAd \cdot$		
$I_4:$ $A \rightarrow aAb \cdot$					
$I_5:$ $A \rightarrow aAd \cdot$					

从上表可看出,状态 I_0 和 I_2 存在移进-归约冲突，该文法不是 LR(0)文法。

对于 I_0 来说有： $\text{Follow}(A) \cap \{a\} = \{b, d, \#\} \cap \{a\} = \Phi$

所以在 I_0 状态下面临，输入符号为 a 时移进，为 $b, d, \#$ 时归约，为其他时报错。

对于 I_2 来说有也有与 I_0 完全相同的结论。（即 $\text{Follow}(A) \cap \{a\} = \{b, d, \#\} \cap \{a\} = \Phi$ ）

这就是说，以上的移进归约冲突是可以解决的，因此该文法是 SLR(1)文法。

其 SLR(1)分析表为：

状态	ACTION				GOTO
	a	b	d	#	A
0	S_2	r_1	r_2	r_3	1
1				acc	
2	S_2	r_1	r_2	r_3	3
3		S_1	S_5		
4		r_2	r_2	r_2	
5		r_1	r_1	r_1	

对输入串 $ab\#$ 给出分析过程为：

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	#	ab#	S_2	
2	02	#a	b#	r_3	3
3	023	#aA	b#	S_1	
4	0234	#aAb	#	r_2	1
5	01	#A	#	acc	

八、语法制导翻译。（没有说明自顶向下还是自底向上，不用管是否消除左递归，第三问两种答案）

$S \rightarrow (L) \mid a$ (a) 写一个翻译方案，它输出配对括号的个数。如对于句子 $(a, (a, a))$ ，输出结果为2。

$S' \rightarrow S$ {print(S.num)}
 $S \rightarrow (L)$ {S.num=L.num+1}
 $S \rightarrow a$ {S.num=0}
 $L \rightarrow L1, S$ {L.num=L1.num+S.num}
 $L \rightarrow S$ {L.num=S.num}

$S \rightarrow (L) \mid a$ (d) 写一个翻译方案，它打印出每个a在句子中是第几个字符。例如：当句子是 $(a, (a, (a, a)), (a))$ 时，打印的结果是2 5 8 10 14。

为文法符号S和L分别定义一个继承属性in和一个综合属性out，分别表示在句子中，该文法符号推出的字符序列的前面有多少个字符，和该文法符号推出的字符序列的最后一个字符在句子中是第几个字符。所求的翻译方案如下：

$S' \rightarrow \{S.in=0\} S$
 $S \rightarrow \{L.in=S.in+1\} (L) \{S.out=L.out+1\}$
 $S \rightarrow a \{S.out=S.in+1; \text{print}(S.out)\}$
 $L \rightarrow \{L1.in=L.in\} L1, \{S.in=L1.out+1\} S \{L.out=S.out\}$
 $L \rightarrow \{S.in=L.in\} S \{L.out=S.out\}$

$S \rightarrow (L) \mid a$ (c) 写一个翻译方案，它输出每个a的嵌套深度。例如：对于句子 $(a, (a, a))$ ，输出的结果是1 2 2。

由于a的嵌套深度不是由a本身能决定的，所以一定要用继承属性。用继承属性depth表示嵌套深度，所求的翻译方案如下：

$S' \rightarrow \{S.depth=0\} S$
 $S \rightarrow \{L.depth=S.depth+1\} (L)$
 $S \rightarrow a \{print(S.depth)\}$
 $L \rightarrow \{L1.depth=L.depth\} L1, \{S.depth=L.depth\} S$
 $L \rightarrow \{S.depth=L.depth\} S$

$S \rightarrow (L) \mid a$ (d) 写一个翻译方案，它打印出每个a在句子中是第几个字符。例如：当句子是 $(a, (a, (a, a)), (a))$ 时，打印的结果是2 5 8 10 14。

为文法符号S和L分别定义一个继承属性in和一个综合属性total，分别表示在句子中，该文法符号推出的字符序列的前面有多少个字符，和该文法符号推出的字符总数（即：out=in+total）。所求的翻译方案如下：

$S' \rightarrow \{S.in=0\} S$
 $S \rightarrow \{L.in=S.in+1\} (L) \{S.total=L.total+2\}$
 $S \rightarrow a \{S.total=1; \text{print}(S.in+1)\}$
 $L \rightarrow \{L1.in=L.in\} L1, \{S.in=L.in+L1.total+1\} S \{L.total=L1.total+1+S.total\}$
 $L \rightarrow \{S.in=L.in\} S \{L.total=S.total\}$

补充一题

$S \rightarrow (L) \mid a$ (b) 写一个翻译方案，它输出配对括号的最大嵌套深度。如对于句子 $(a, (a, a))$ ，输出结果为2。

$S' \rightarrow S$ {print(S.num)}
 $S \rightarrow (L)$ {S.num=L.num+1}
 $S \rightarrow a$ {S.num=0}
 $L \rightarrow L1, S$ {L.num=max(L1.num, S.num)}
 $L \rightarrow S$ {L.num=S.num}

九、中间代码生成，四元式。

100 (j<,A,C,102)

101(j,_,113) /*E1 为 F*/

102 (j<,B,D,104) /*E1 为 T*/

103 (j,_,113) /*E1 为 F*/

104 (j=,A,1,106) /*Ez 为 T*/

105 (j,_,108) /*Ez 为 F*/

106 (+,C,1,C) /*C:=C+1*/

107 (j,_,112) /*跳过 else 后的语句*/

108 (j≤,A,D,110) /*E3 为 T*/

109 (j,_,112) /*E3 为 F*/

110 (+,A,2,A) /*A:=A+2*/

111 (j,_,108) /*转回内层 while 语句开始处*/

112(j,_,100) /*转回外层 while 语句开始处*/

113