**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 |  | | | | |

1. （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。

1. （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)= φ ,M(z,1)={y}，构造相应的 DFA 并最小化。

1. （10分）假定语言X的字母表∑={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：

尝试构造词法分析程序。

1. （15分）已知文法G（S）：S→Sa|Nb|c N→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。

1. （10分）已知文法：S→a|^|(T) T→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。

1. （10分）对下面的文法G：E->TE' E'->+E| ε T->FT' T' ->T| ε F-> PF' F'-> \*F'| ε P->(E)|a|b|^

计算这个文法的每个非终结符的 FIRST 集和 FOLLOW 集。

1. （10分）已知文法为：A->aAd|aAb|ε

（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

1. （15分）已知文法G：S-> (L) | a L-> 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

1. （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+l

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→AB|B

A→AC|D

B→1|3|5|7|9

D→B|2|4|6|8

C→0|D

2.文法G(S):

S→ABC|C

C→1|3|5|7|9

B→0|1|2|3|4|5|6|7|8|9|ε|BB

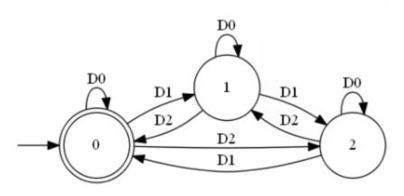
A→1|2|3|4|5|6|7|8|9

（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

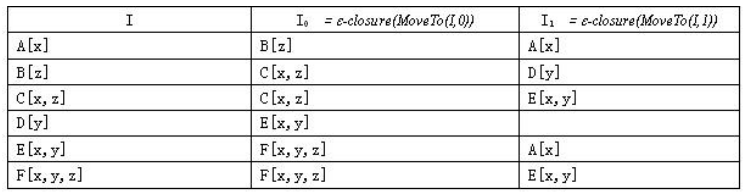
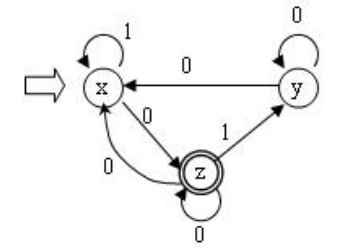
D1 = 1|4|7

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）。



其中，含z的是终态，即BCF是终态。

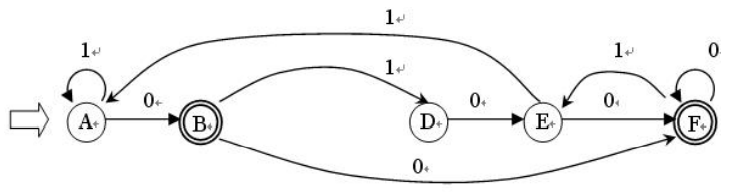
使用划分法进行DFA确定化。

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

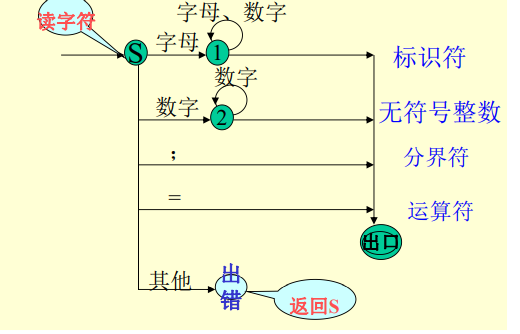
OUTPUT (2, 整数)；

BREAK;

CASE 3 OUTPUT ( 3 ,“;” )；BREAK;

CASE 4 : OUTPUT ( 4 ,“=”)；BREAK;

DEFAULT: Error();



**四、消除左递归，递归分析伪代码。**

S→fN′bS′|cS′

S′→aS′|dN′bS′|ε

N′→eN′|ε

**procedure S;**

if sym='f'

advance;

N’;

if sym='b'

advance;

S’;

advance;

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’;

advance;

else error;

**procedure N’;**

if sym=‘e’

advance;

N’;

**五、短语、直接短语、句柄、最左推导。（注意箭头双横线）**

短语：a1 S a2 S,a2 (S,a2) a1, (S,a2) (a1, (S,a2))

直接短语：a1 S a2

句柄：a1

对(((a,a),^,(a)),a) 的最左推导为：

S=>(T) =>(T,S) =>(S,S) =>((T),S)

=>((T,S),S) =>((T,S,S),S) =>((S,S,S),S)

=>(((T),S,S),S) =>(((T,S),S,S),S) =>(((S,S),S,S),S)

=>(((a,S),S,S),S) =>(((a,a),S,S),S) =>(((a,a),^,S),S)

=>(((a,a),^,(T)),S) =>(((a,a),^,(S)),S)

=>(((a,a),^,(a)),S) =>(((a,a),^,(a)),a)

**六、求First和Follow**

FIRST 集合有：

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

FIRST(E')={+,ε}

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

FIRST(T')=FIRST(T)∪{ε}={(,a,b,^,ε};

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

FIRST(F')=FIRST(P)={\*,ε};

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

FOLLOW 集合有：

FOLLOW(E)={),#};

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

FOLLOW(T)=FIRST(E')∪FOLLOW(E)={+,),#};//不包含ε

FOLLOW(T')=FOLLOW(T)=FIRST(E')∪FOLLOW(E)={+,),#};

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

FOLLOW(F')=FOLLOW(F)=FIRST(T')∪FOLLOW(T)={(,a,b,^,+,),#}

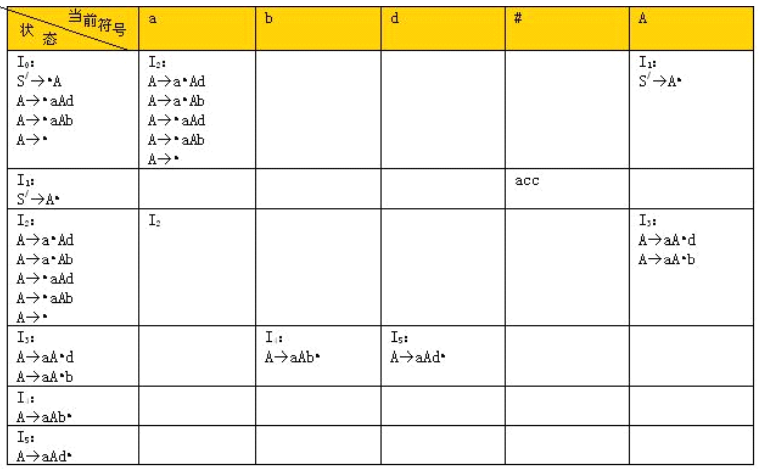
**七、LR自底向上分析。**

增加一个非终结符S’后，产生原文法的增广文法有：

S'->A

A->aAd|aAb|ε

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）



从上表可看出,状态I0和I2存在移进-归约冲突，该文法不是LR(0)文法。

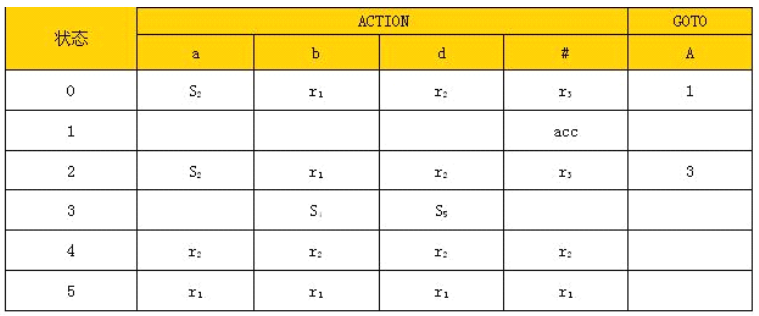
对于I0来说有：Follow(A)∩{a}={b,d,#}∩{a}=Φ

所以在 I0 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。

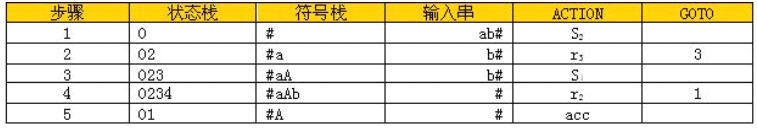
对于 I2 来说有也有与 I0 完全相同的结论。（即Follow(A)∩{a}={b,d,#}∩{a}=Φ）

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。

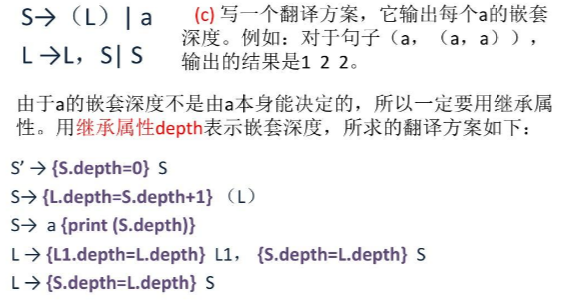
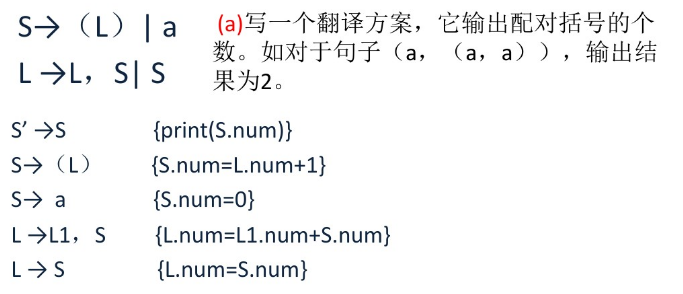
其 SLR(1)分析表为：

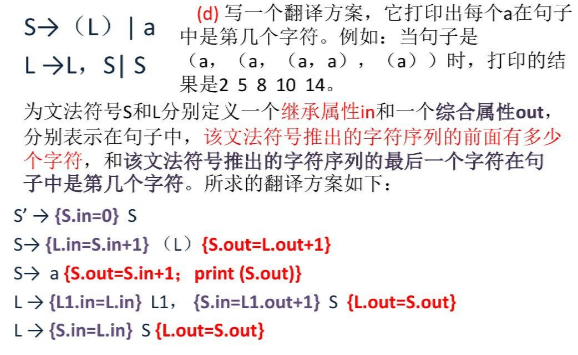
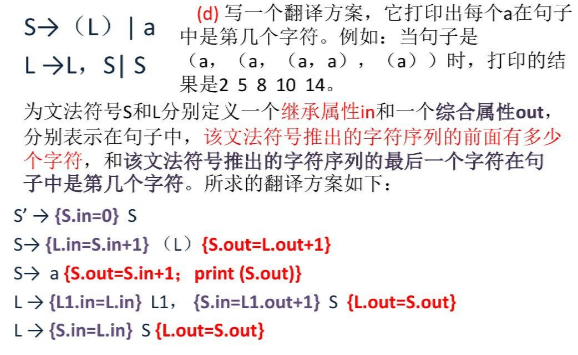


对输入串 ab#给出分析过程为：



**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）**





**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) /\*El 为 T\*/

103 (j,\_,\_,113) /\*El 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