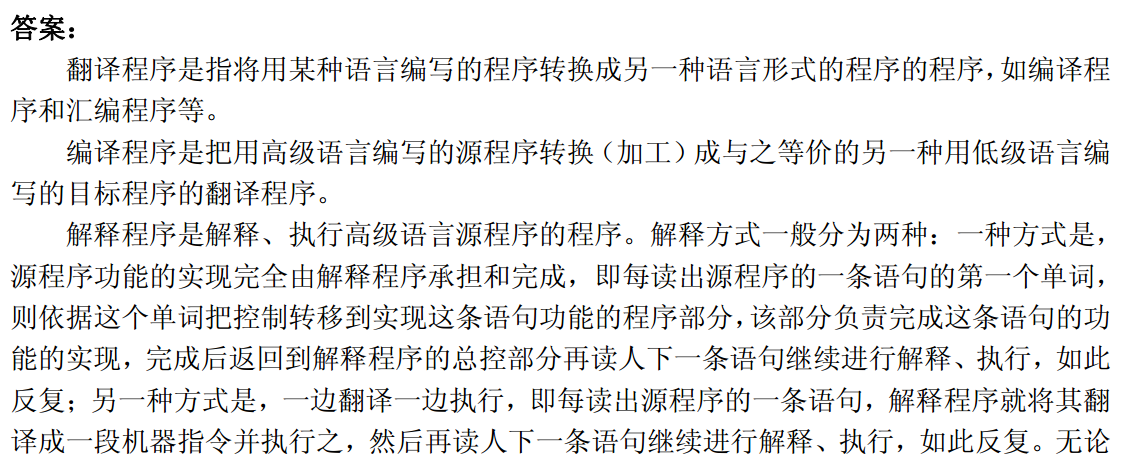
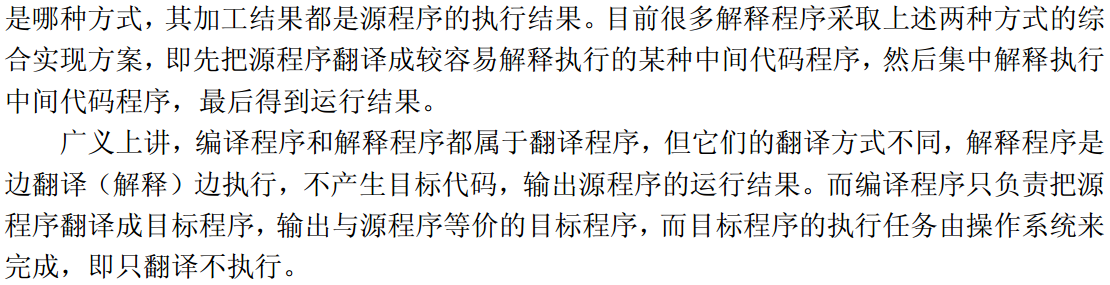
# Charpter 1







# Charpter 2

1. 文法G=（{A,B,S}, {a,b,c}, P,S）

其中P为： S → Ac | aB

A→ab

A→bc

写出L(G[S])的全部元素。

答案：L(G[S])={abc}

2. 文法G[N]为

N →D | ND

D →0 |1|2|3|4|5|6|7|8|9

G[N]的语言是什么？

答案：

因为：N=>ND=>NDD=>......=>NDDD....D=>D.......D 所以

G[N]的语言是允许0开头的非负十进制整数。

或其语言是V+， 这里V={0，1，2，3，4，5，6，7，8，9}

4. 证明文法G=（{E，O}, {(,),+,\*,v,d}, P,E}是二义性的，其中P为

E → EOE | (E) | v | d

O → + | \*

答案：

该题简单的证明方法就是找出一个句型。画出两种不同形态的语法推导树。

例如句型： v+v\*d, 其语法推导树为：

 或 

所以文法是二义性的。

11. 一个上下文无关文法申城句子abbaa的唯一语法树如下：



（1） 给出该句子相应的最左推导和最右推导

（2） 该文法的产生式集合P可能有哪些元素？

（3） 找出改句子的所有短语、简短短语、句柄

答案：

（1）最左推导：

S=>ABS=>aBS=>aSBBS=>aBBS=>abBS=>abbS=>abbAa=>abbaa

最右推导：

S=>ABS=>ABAa=> ABaa => ASBBaa = ASBbaa = ASbbaa => Abbaa =>abbaa

（2） 产生式集合可能包含： S→ ABS S→Aa s S→ε

A→a B→SBB

（3） a 是相对于A的短语

ε是相对于S的短语

b是相对于B的短语

bb是相对于B的短语

aa是相对于S的短语

abbaa是相对于S的短语

直接短语有a b ε

句柄是a

12

（2）、 S 🡪 aS | aSb | ε

（5）、 S 🡪 aSbb | aSbbb | ε

（6）、 S🡪 aSa | bSb | ε

18

（2）、 S 🡪 aA 或 S 🡪 aS | aB

A 🡪 aA | bB B 🡪 bB | b

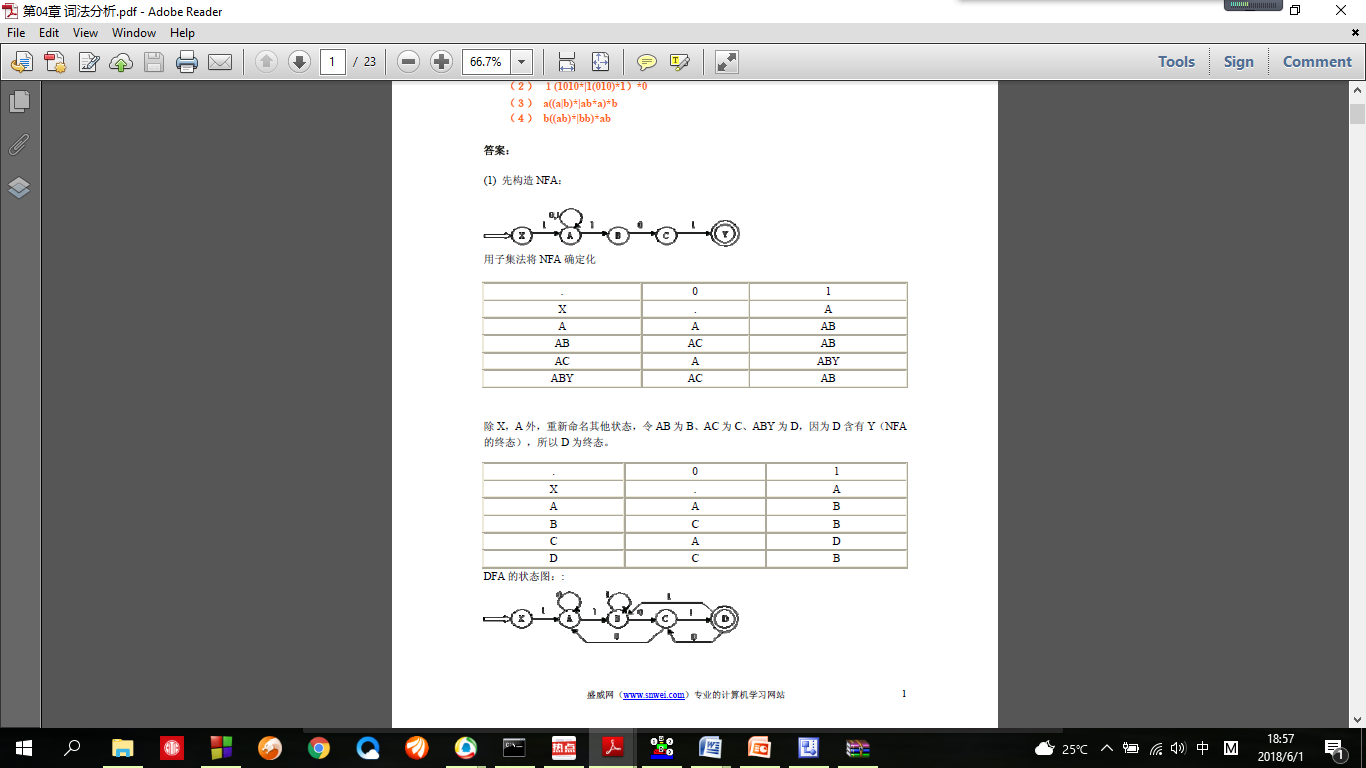
B 🡪 bB | ε

# Charpter 3

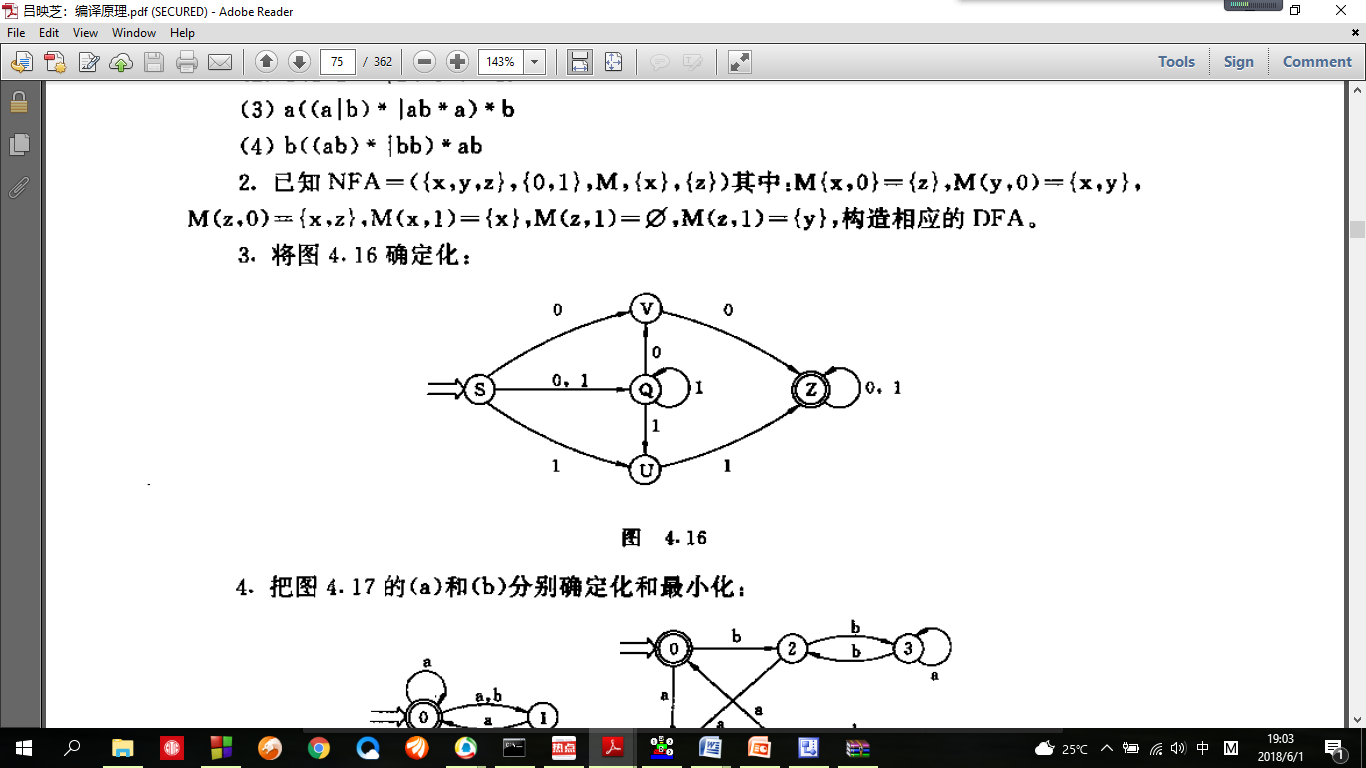
1. 构造下列正规式相应的DFA

（1）1（0|1）\*101

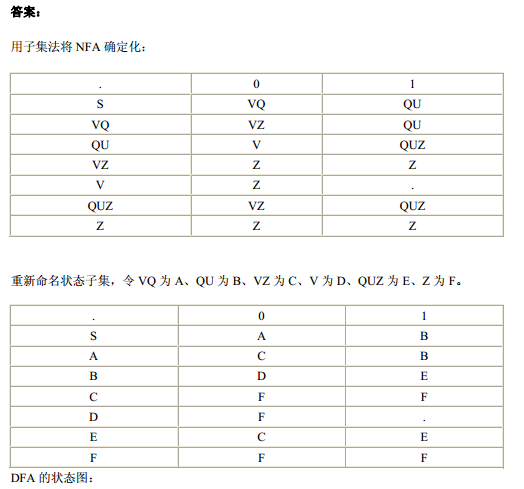
答案：

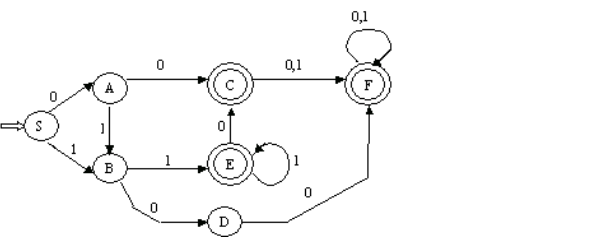


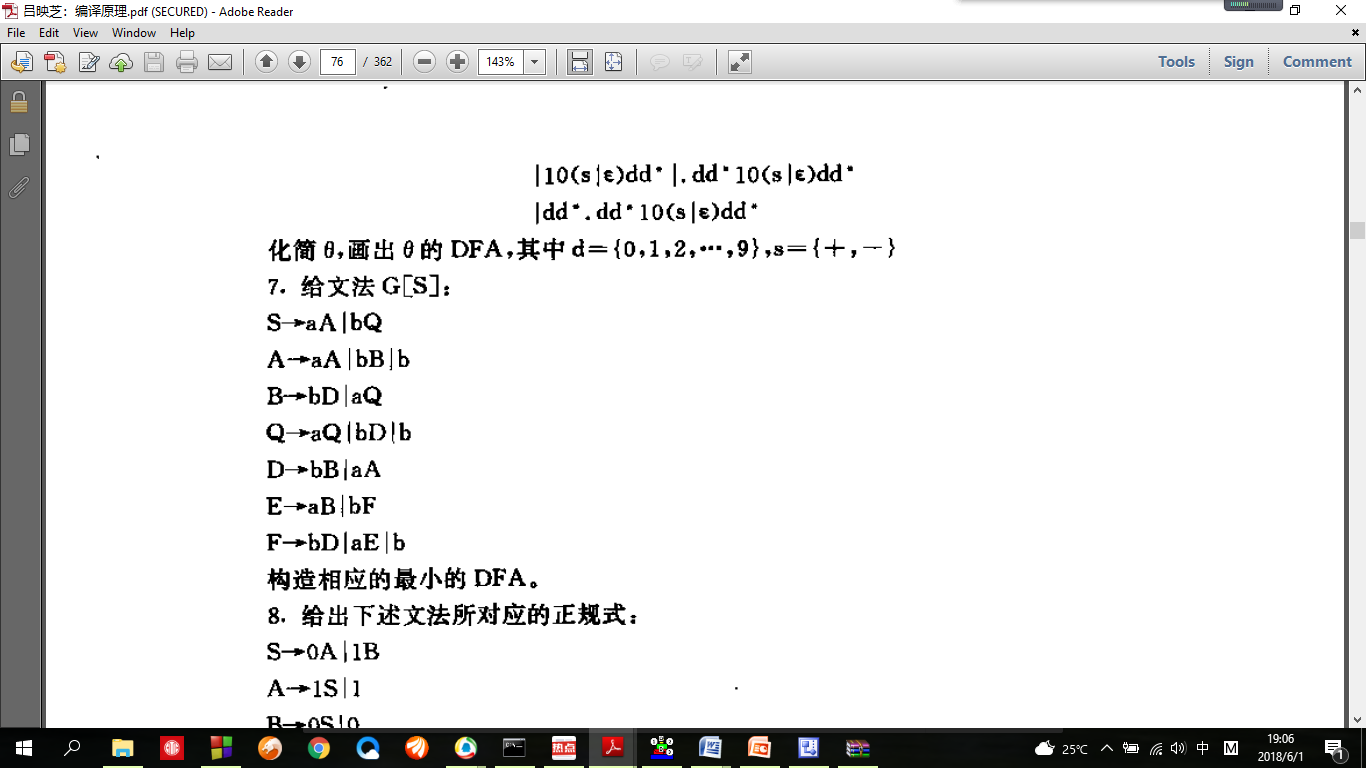
3 将下图NFA确定化

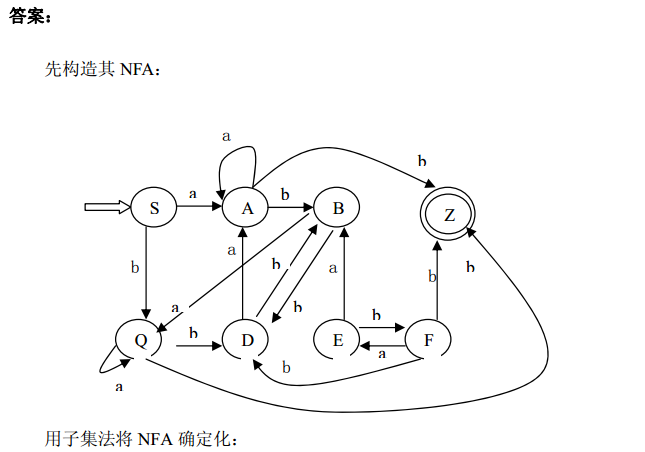


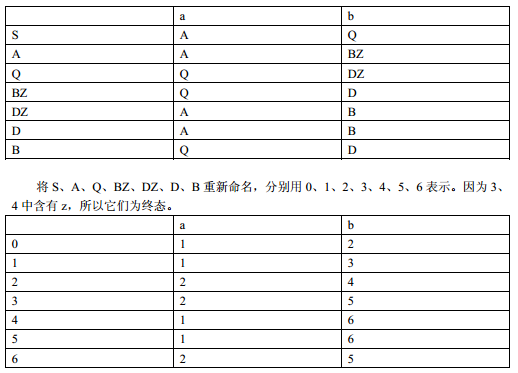
答案：

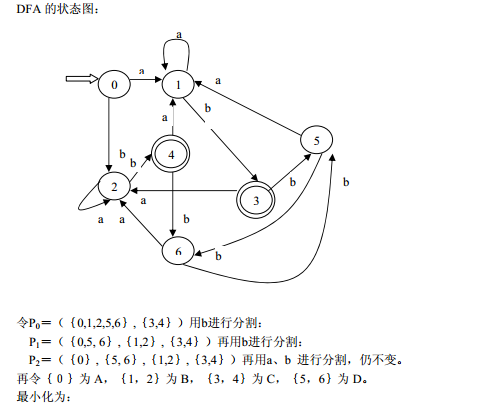


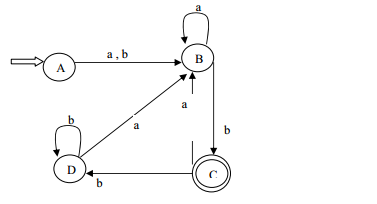


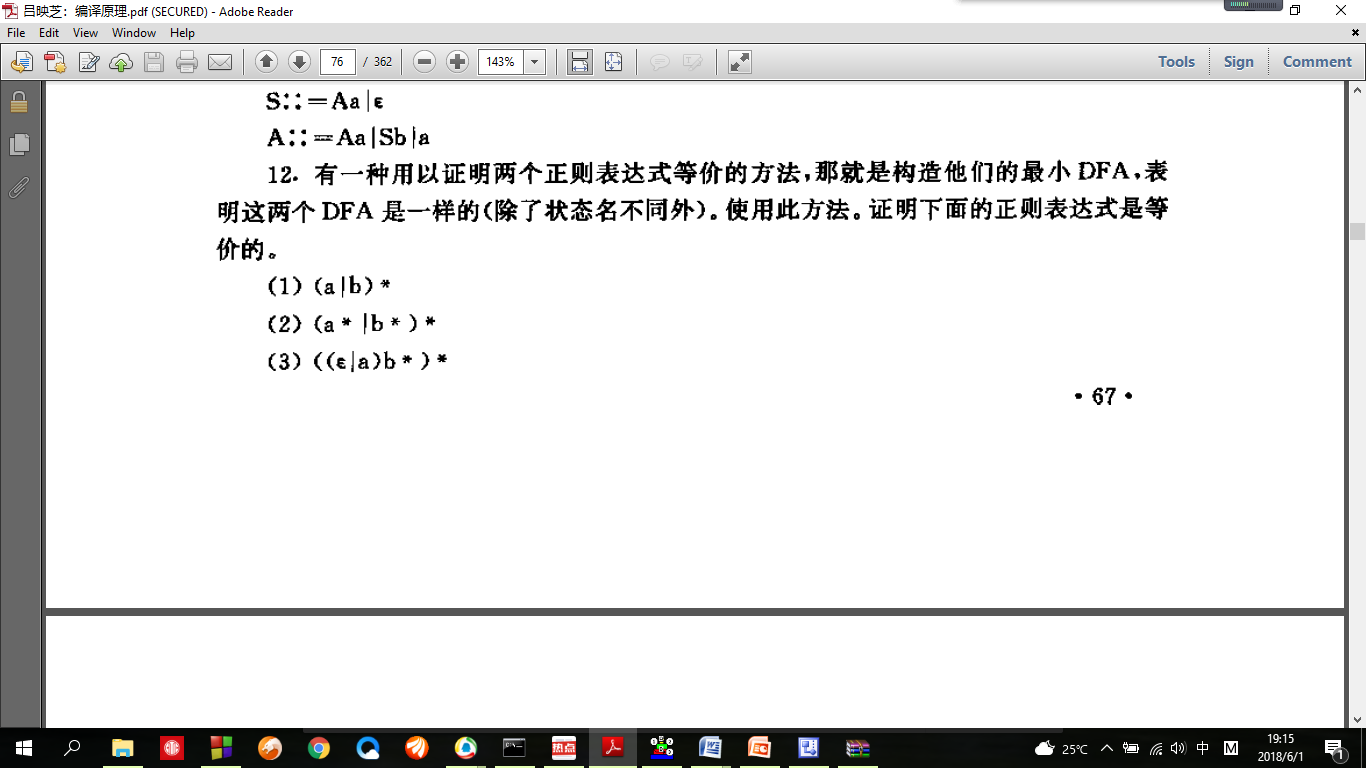










11.

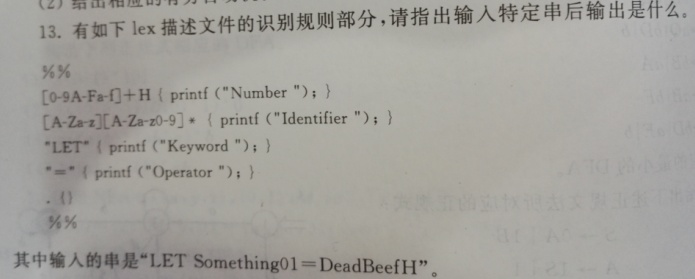
解答：先求出DFA:

由正规式（1），很容易看出其DFA如下，当然也是最小的



由正规式（2），先构造NFA，再确定化得DFA如下，最后最小化得到和上面一样的DFA，（过程省略），即完成证明。

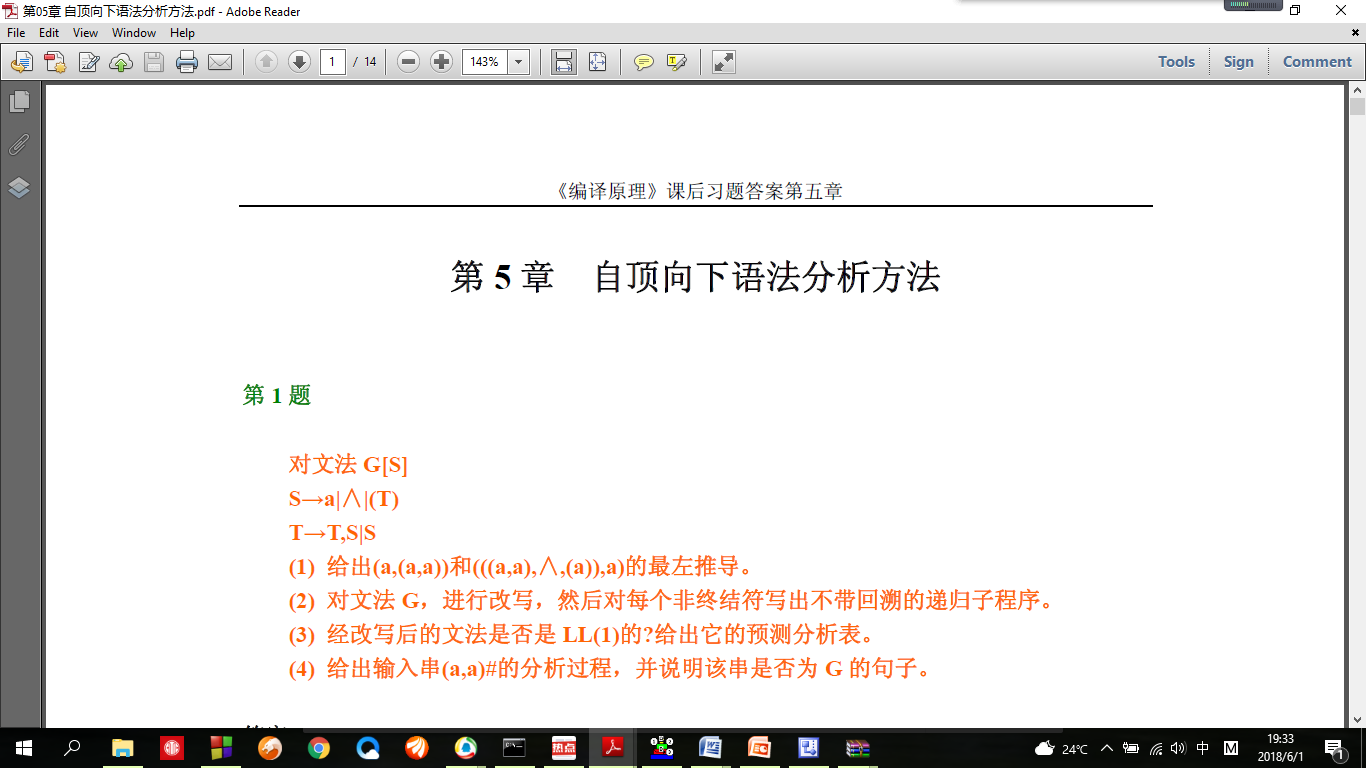
13.

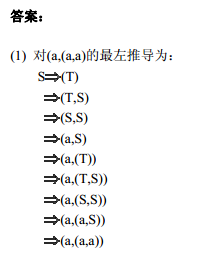


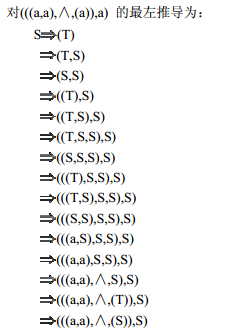
输出：Identifier Indentifier Operator Number

规则存在先后顺序，所以LET 字符串匹配的是[A-Za-a][A-Za-z0-9]\*

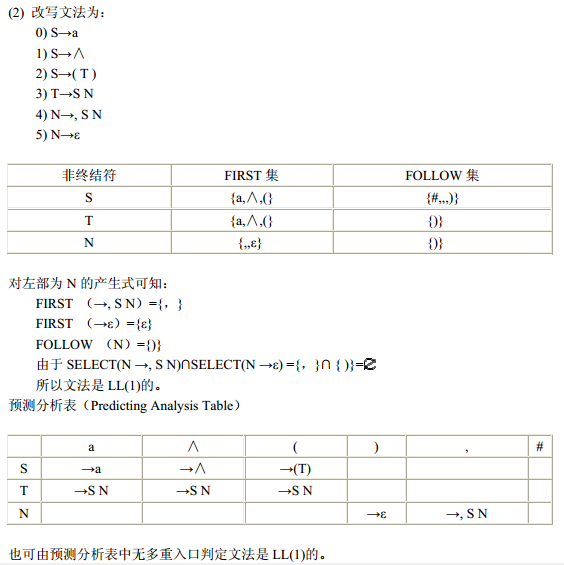
# Charpter 4

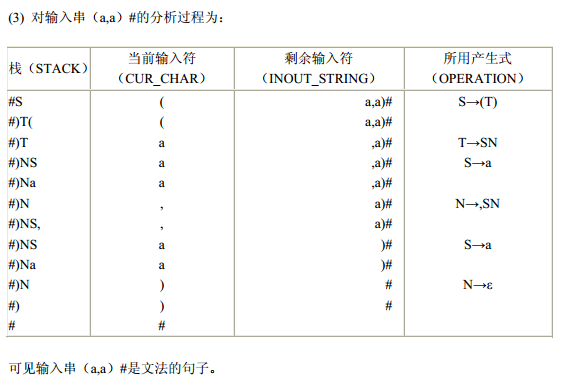


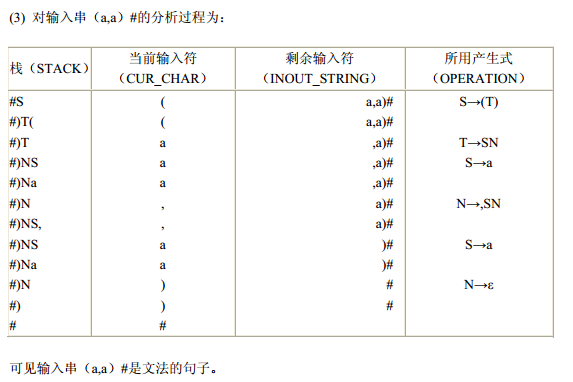












4. 证明下述文法不是LL（1）的

S→C$

C→bA | aB

A→a | aC | bAA

B→b | bC | aBB

是否能构造一等价文法，使其是LL（1）的？并给出判断过程。

**答案：**

由于A和B的规则中都包含公共左因子，所以：

a属于 select(A→a )并且 a select(A→aC)，A的这两条规则的select集合相交不为空，故文法不是LL（1）的

通过提取公共左因子来改写：

S🡪C$

C🡪 bA | aB

A🡪 aC'| bAA

B🡪 bC'| aBB

C'🡪C| ε

计算：first(S)={a,b} first(C)={a,b} first(A)={a,b }

first(B)={ a,b } first(C')={ a,b,ε}

follow(S)={# } follow (C)={$} follow (A)={a,b,$}

follow (B)={a,b,$} follow (C')={a,b,$}

这样按定义求select集合

select(S🡪C$)={a,b}

select(C🡪 bA)={b} select(C🡪aB)={a}

select(A🡪 aC')={a} select(A🡪 bAA)={b}

select(B🡪 bC')={b} select(B🡪aBB)={a}

select(C'🡪 C)={a,b} select(C'🡪ε)={a,b,$}

非终结符C'的各规则的select集合两两相交为空，不是LL（1）文法。

7. 对于一个文法若消除了左递归，提取了提取公共左因子后，是否一定为LL（1）文法？，试对下面文法进行改写，并对改写后的文法进行判断。

（4）S🡪AS | b

A🡪SA | a

答案：对于一个文法若消除了左递归，提取了提取公共左因子后，不能保证一定为LL（1）文法，需要验证。

这个文法是间接左递归，所以要消除直接和间接左递归。

首先给一个非终结符的排列： S A

对于第一个非终结符S没有直接左递归，不需要处理

对于第二个非终结符A，将前面的非终结符的规则带入，得到：

A🡪ASA | bA | a 出现了直接左递归，需要消除，最后得到的文法：

S🡪AS | b

A🡪 bAA' | aA'

A'🡪SAA' |ε

计算得到：

first(S)={a,b} first(A)={a,b} first(A')={a,b, ε}

follow(S)={a,b，#} follow (A)={a,b} follow (A')={a,b, ε}

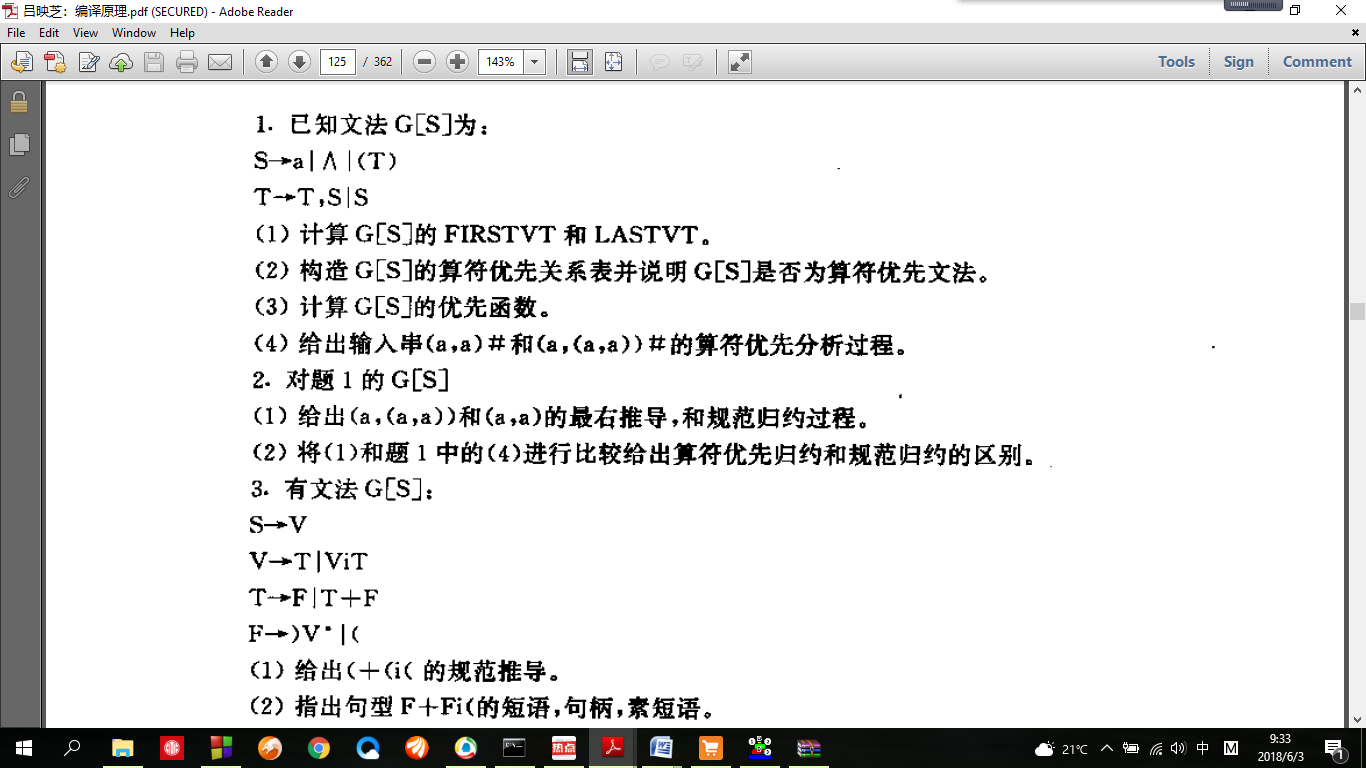
再依据first和follow计算select集合，首先对S的规则，

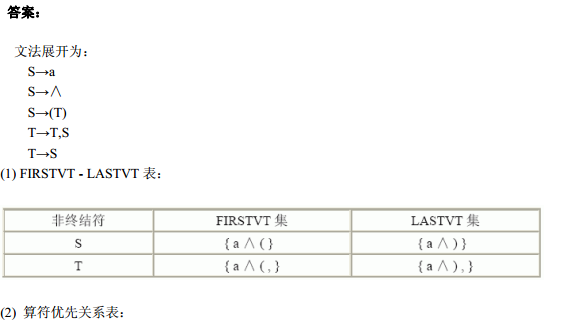
select（S🡪AS）={a,b}

select（b）={b} 其交集不为空，所以不是LL(1)文法。

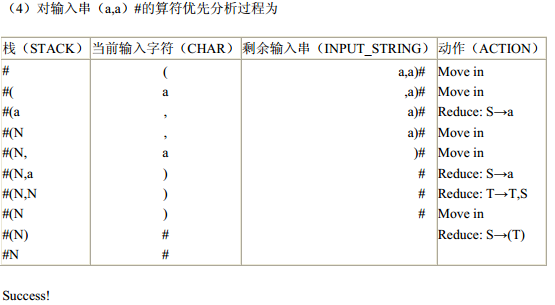
对其它规则的select集合就没必要再计算了。

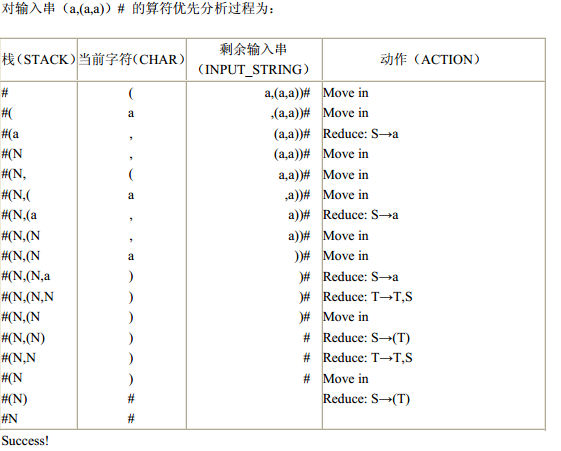
# Charpter 5

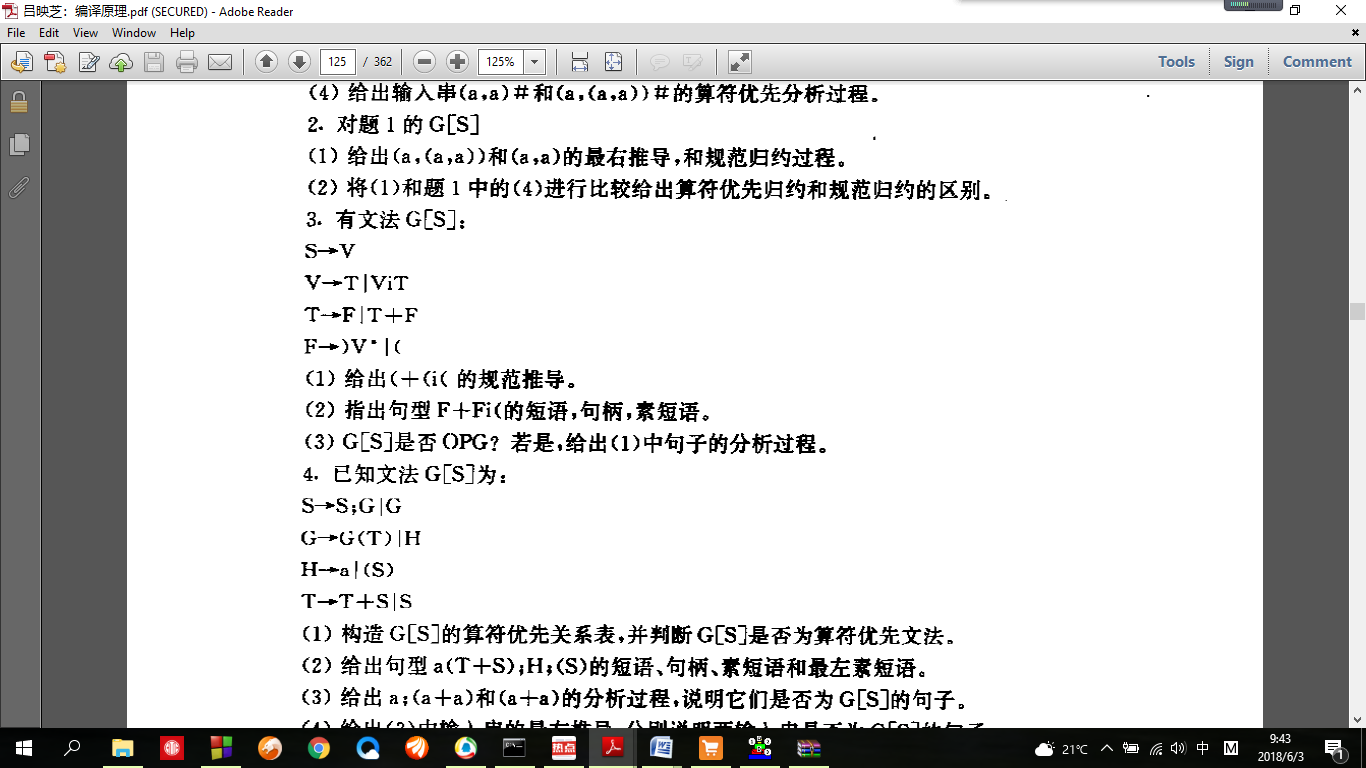


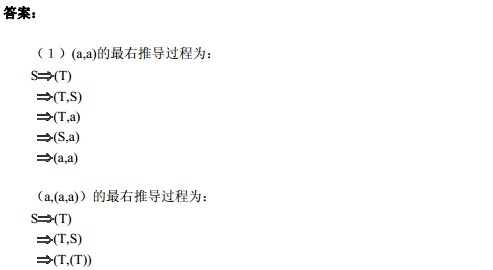




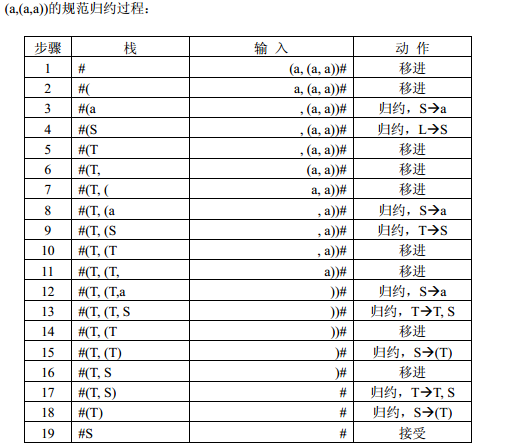


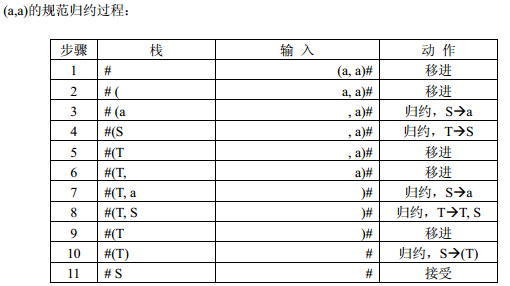


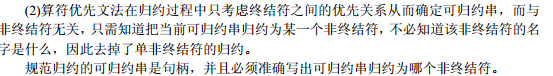










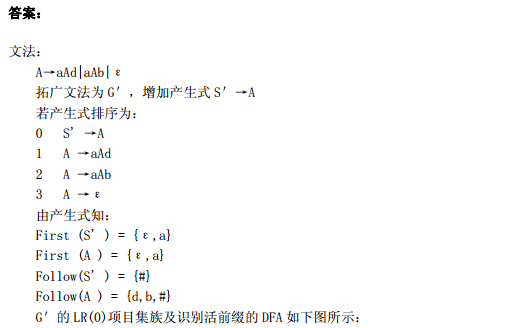


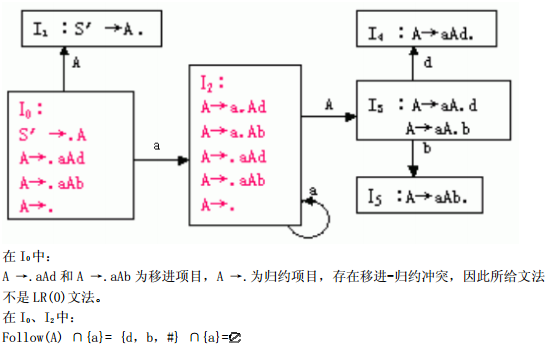
# Charpter 6

1. 已知文法

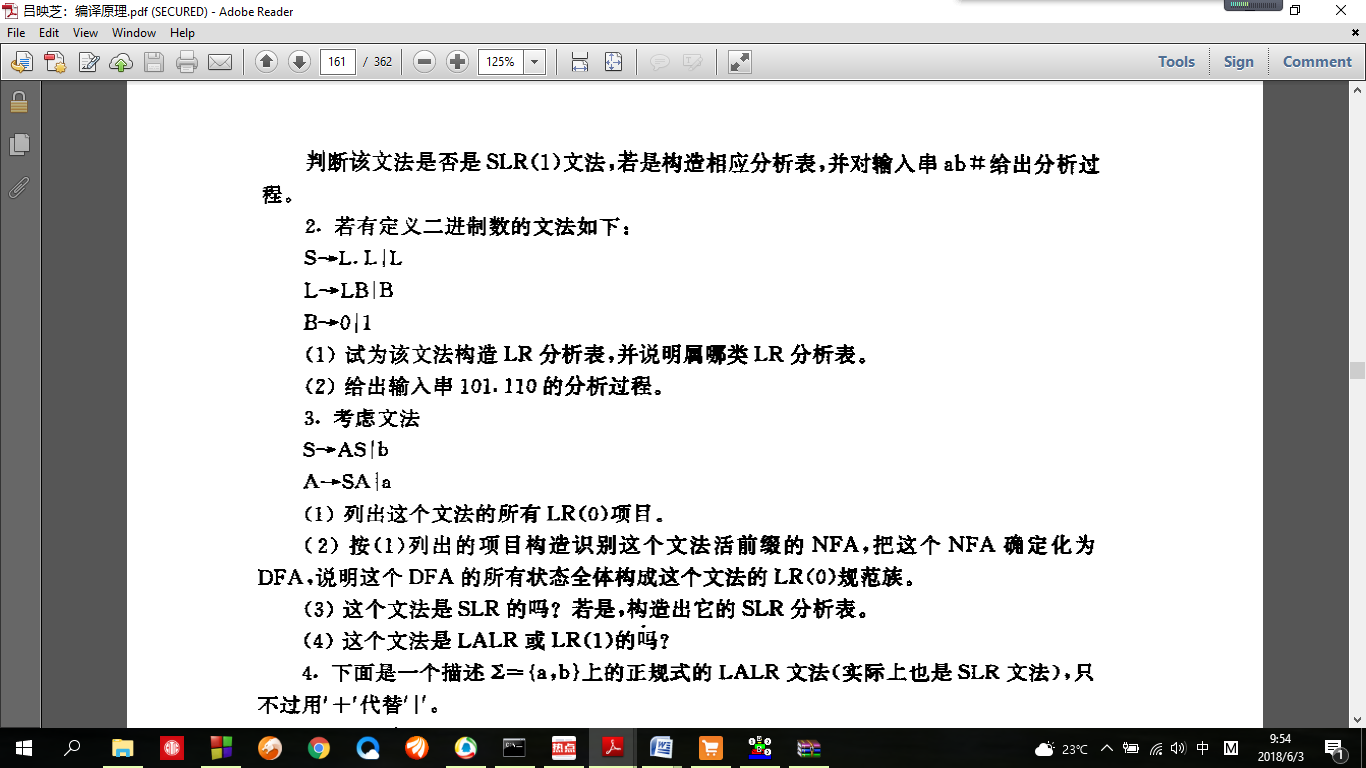
A🡪aAd |aAb|ε

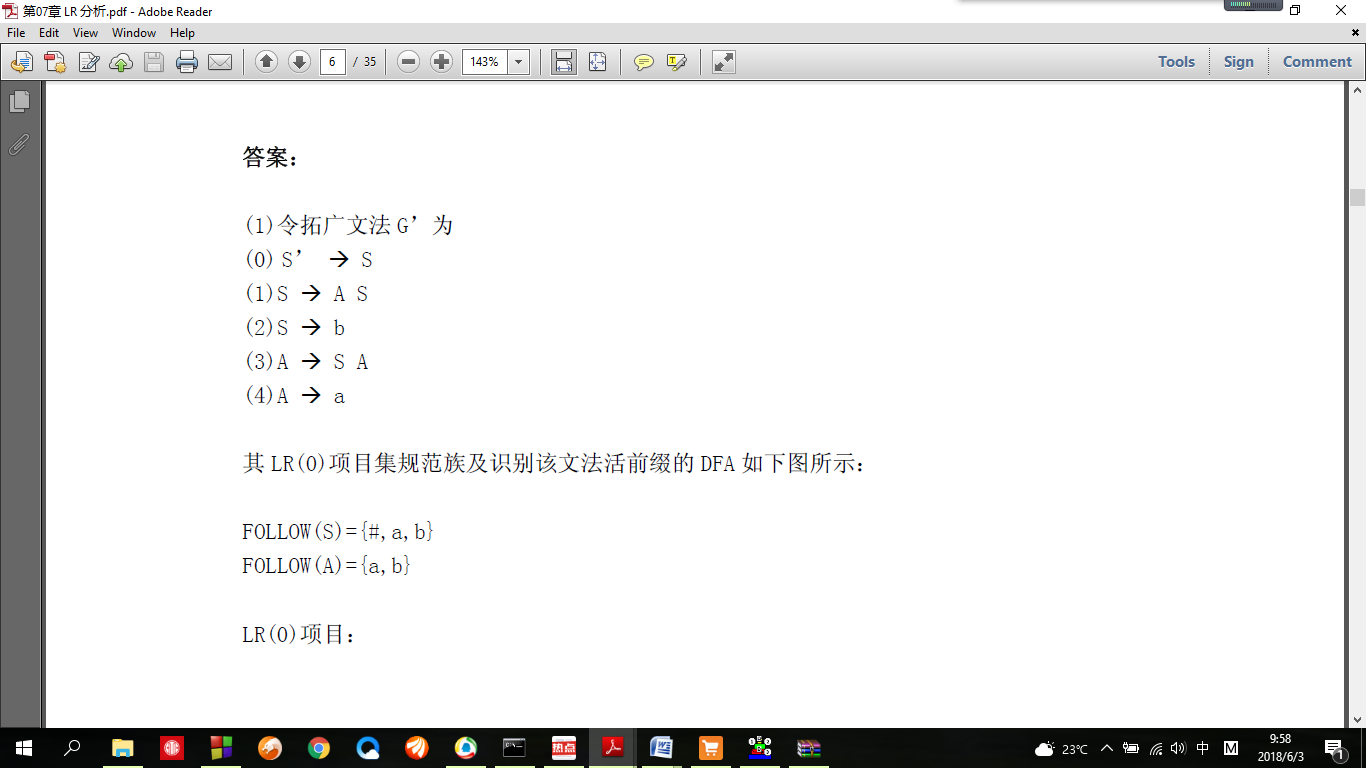
判断该文法是否是SLR(1)文法，如是，构造相应分析表，并对输入串ab#给出分析过程。

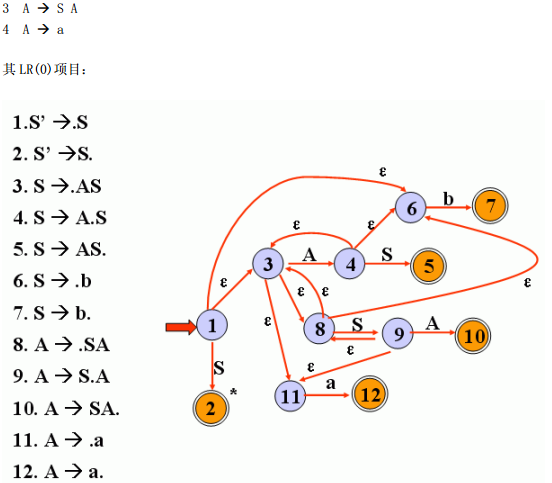




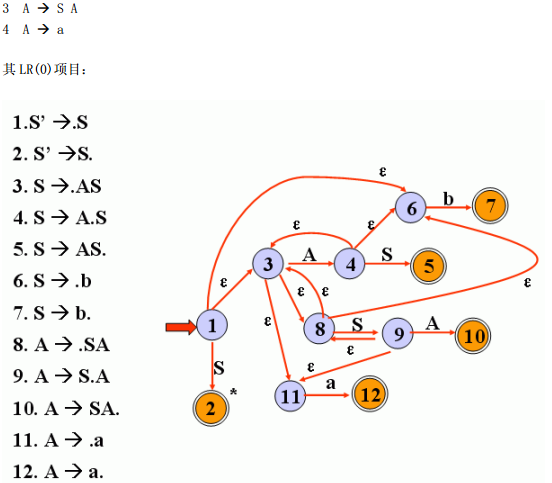




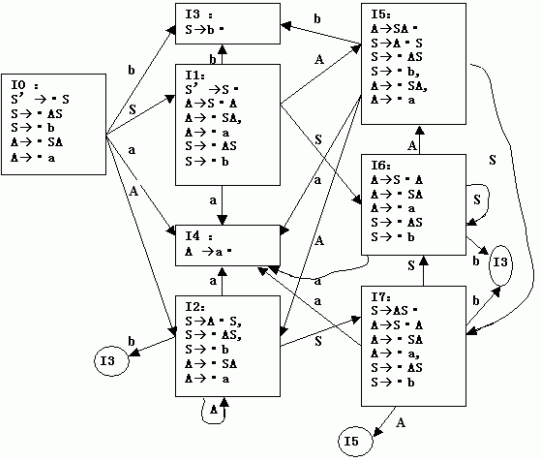


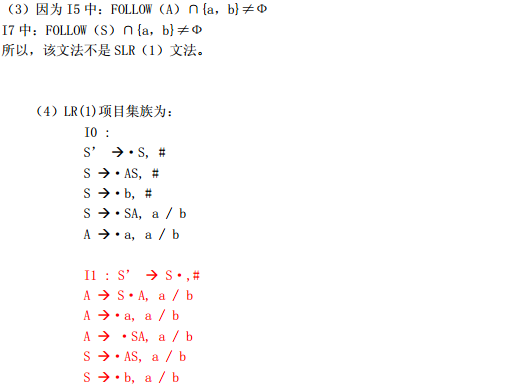


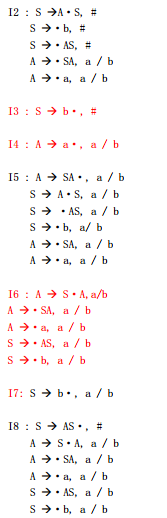
（2）NFA为：

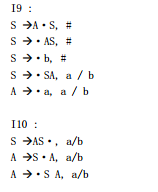


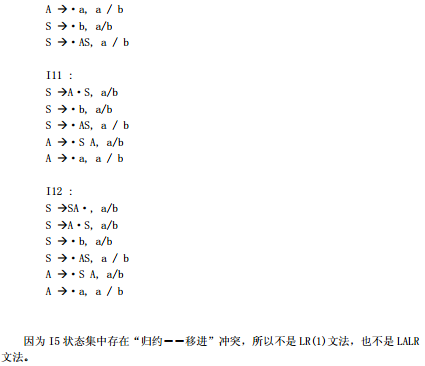
确定化的DFA为：











# Charpter 7

4. 以下是简单表达式（只含加、减运算）计算的一个属性文法G（E）:

E→TR {R.in:=T.val; E.val:=R.val}

R→+TR1 {R1.in:=R.in+T.val; R.val:=R1.val}

R→-TR1 {R1.in:=R.in-T.val; R.val:=R1.val}

R→ε {R.val:=R.in; }

T→num {T.val:=lexval(num);}

其中，lexval(num)表示从词法分析程序中得到的常数值。

试给出表达式3+4-5的语法分析树和相应的带标注语法分析树。

答案：

语法分析树：

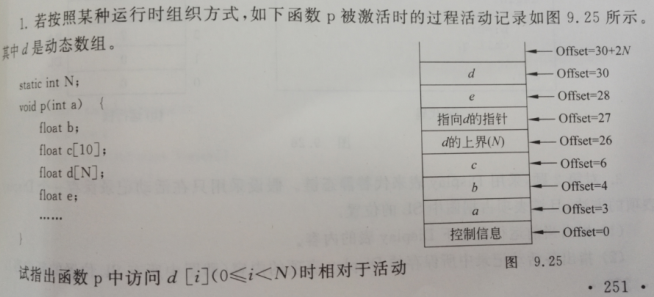


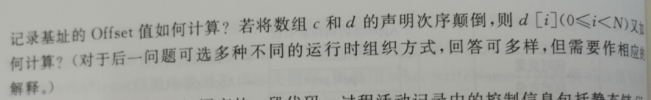
带标注语法分析树如下，由于是一个L属性文法，所以注意各属性的值是通过语法分析树的遍历时完成的属性计算。



# Charpter 8

# Charpter 9

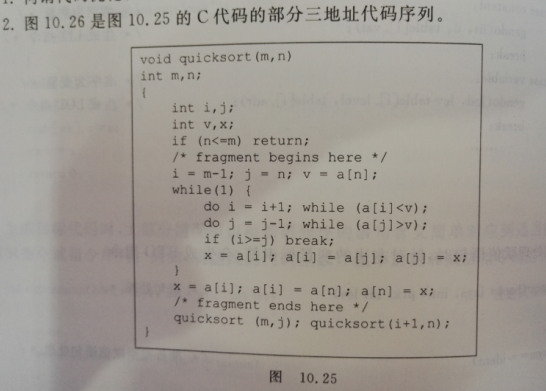


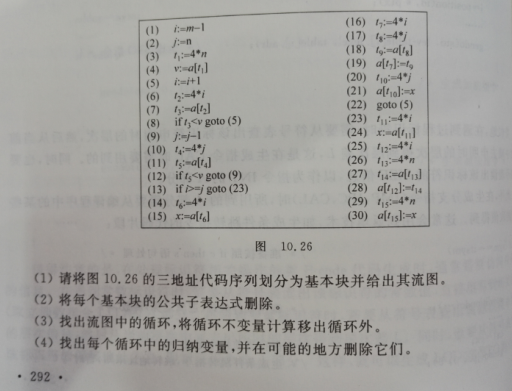


函数访问d[i] 的Offset的值为： offset = 30 + 2i

若将数组C和d 的声明次序颠倒，由于d是一个可变数组，所以d的数组元素存放的位置不会因为组C和d 的声明次序颠倒而发生改变，d[i]的起始偏移量仍为30，offset不变。

# Charpter 10





（1） 答案：如下代码，红色标记入口语句

（1） i:=m-1

（2） j:=n

（3） t1:=4\*n

（4） v:=a[t1]

（5） i:=i+1

（6） t2:=4\*i

（7） t3:=a[t2]

（8） if t3<v goto （5）

（9） j:=j-1

（10）t4:=4\*j

（11）t5:=a[t4]

（12）if t5<v goto （9）

（13）if i>=j goto （23）

（14）t6:=4\*i

（15）x:=a[t6]

（16）t7:=4\*i

（17）t8:=4\*j

（18）t9:=a[t8]

（19）a[t7] :=t9

（20）t10:=4\*j

（21）a[t10] :=x

（22）goto （5）

（23）t11:=4\*i

（24）x:=a[t11]

（25）t12:=4\*i

（26）t13:=4\*n

（27）t14:=a[t13]

（28）a[t12] :=t14

（29）t15:=4\*n

（30）a[t15] :=x

这样就得到6个基本快

B1：（1）（2）（3）（4）

B2：（5）（6）（7）（8）

B3：（9）（0）（11）（12）

B4：（13）

B5：（14）......（22）

B6：（23）......（30）



（2）基本块B1--B4都没有公共子表达式

B5： 红色标记处为删除公共子表达式的地方

（14）t6:=4\*i

（15）x:=a[t6]

（16）t7:= t6

（17）t8:=4\*j

（18）t9:=a[t8]

（19）a[t7] :=t9

（20）t10:= t8

（21）a[t10] :=x

（22）goto （5）

B6:

（23）t11:=4\*i

（24）x:=a[t11]

（25）t12:= t11

（26）t13:=4\*n

（27）t14:=a[t13]

（28）a[t12] :=t14

（29）t15:= t13

（30）a[t15] :=x