

一、(30) 问答

1、解释 LL(1)文法

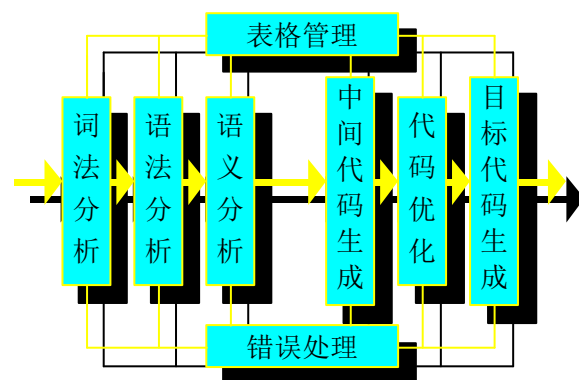
答：一个文法满足下面条件：

- (1) 文法不含左递归
- (2) 对于文法中每一个非终结符 A 的各个产生式的候选首符集量不相交。
- (3) 对于文法中的每个非终结符 A，若它存在某个候选首符集包含 ϵ ，那么

$$\text{First}(A) \cap \text{Follow}(A) = \phi$$

则称该文法为 LL(1)文法。

2、出编译程序总框图。



3、写出活跃变量数据流方程及在编译中的应用。

答：数据流方程：

$$\text{Liveout}(B) = \bigcup_{i \in S(b)} \text{Livein}(i)$$

$$\text{Livein}(B) = \text{Liveuse}(B) \cup (\text{Liveout}(B) - \text{Def}(B))$$

在编译中的应用：(1) 优化 (2) 目标代码生成 (3) 检查变量定值之前被引用

4、写出语言 $L = \{a^n b^m c^m d^n | n \geq 1 \text{ 且 } m \geq 1\}$ 的上下文无关文法。

答： $S \rightarrow aSd | aAd$

$A \rightarrow bAc | bc$

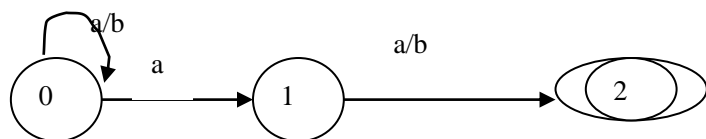
5、解释语法制导翻译法。

答：从概念上讲，基于属性文法的处理过程如下：对单词符号串进行语法分析，构造语法分析树，然后根据需要遍历语法树并在语法树的各结点处按语义规则进行计算。这种有源程序的语法结构驱动的处理办法就是语法制导翻译法

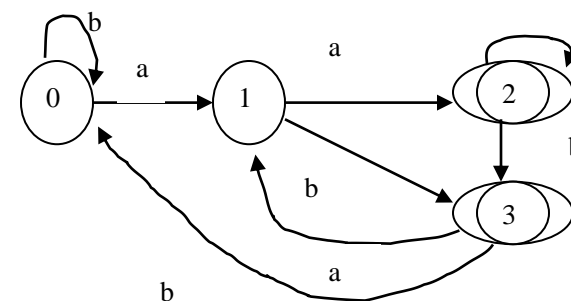
二、(15) 构造下列正规式相应的最简 DFA

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

解：1、由正规式到 NFA



	a	b
{0}	{0,1}	{0}
{0,1}	{0,1,2}	{0,2}
{0,1,2}	{0,1,2}	{0,2}
{0,2}	{0,1}	{0}



3、DFA 最小化

初始分划：{0,1} {2,3}

因为 $\{0,1\}_a = \{1,2\}$ 不属于上面任意一个子集，所以分开：{0}，{1}

因为 $\{2,3\}_a = \{1,2\}$ 不属于上面任意一个子集，所以分开：{2}，{3} 已经是最简

三、(20) 文法 G:

$$S \rightarrow L = R$$

$$S \rightarrow R$$

$$L \rightarrow *R$$

$$L \rightarrow i$$

$$R \rightarrow L$$

文法 G 是 LALR(1) 文法吗？如果是构造出分析表。

解答：1、 $S \rightarrow L = R$

$$2、S \rightarrow R$$

$$3、L \rightarrow *R$$

$$4、L \rightarrow i$$

$$5、R \rightarrow L$$

首先拓广文法

$$S' \rightarrow S$$

$$S \rightarrow L = R$$

$$S \rightarrow R$$

$$L \rightarrow *R$$

$$L \rightarrow i$$

$$R \rightarrow L$$

姓名

学号

级

专业

学院

密

封

线

构造 LR(1)项目集规范组

I0: $S' \rightarrow \bullet S \#$ $S \rightarrow \bullet L = R \#$ $S \rightarrow \bullet R \#$ $L \rightarrow \bullet * R = / \#$ $L \rightarrow \bullet i = / \#$ $R \rightarrow \bullet L \#$ I1: $S' \rightarrow S \bullet \#$ I2: $S \rightarrow L \bullet = R \#$ $R \rightarrow L \bullet \#$ I3: $S \rightarrow R \bullet \#$ I4: $L \rightarrow * \bullet R = / \#$ $R \rightarrow \bullet L = / \#$ $L \rightarrow \bullet * R = / \#$ $L \rightarrow \bullet i = / \#$ I5: $L \rightarrow i \bullet = / \#$ I6: $S \rightarrow L = \bullet R \#$ $L \rightarrow \bullet i \#$ $R \rightarrow \bullet L \#$ $L \rightarrow \bullet * R \#$ I7: $L \rightarrow * R \bullet = / \#$ I8: $R \rightarrow L \bullet = / \#$ I9: $S \rightarrow L = R \bullet \#$ I10: $L \rightarrow i \bullet \#$ I11: $L \rightarrow * \bullet R \#$ $R \rightarrow \bullet L \#$ $L \rightarrow \bullet * R \#$ $L \rightarrow \bullet i \#$ I12: $R \rightarrow L \bullet \#$ I13: $L \rightarrow * R \bullet \#$

因为每一个项目集都不存在冲突，所以该文法为 LR(1) 文法。合并同心集没有冲突，所以是 LALR(1) 文法。

I0: $S' \rightarrow \bullet S \#$ $S \rightarrow \bullet L = R \#$ $S \rightarrow \bullet R \#$ $L \rightarrow \bullet * R = / \#$ $L \rightarrow \bullet i = / \#$ $R \rightarrow \bullet L \#$ I1: $S' \rightarrow S \bullet \#$ I2: $S \rightarrow L \bullet = R \#$ $R \rightarrow L \bullet \#$ I3: $S \rightarrow R \bullet \#$ I4: $L \rightarrow * \bullet R = / \#$ $R \rightarrow \bullet L = / \#$ $L \rightarrow \bullet * R = / \#$ $L \rightarrow \bullet i = / \#$ I5: $L \rightarrow i \bullet = / \#$ I6: $S \rightarrow L = \bullet R \#$ $L \rightarrow \bullet i \#$ $R \rightarrow \bullet L \#$ $L \rightarrow \bullet * R \#$ I7: $L \rightarrow * R \bullet = / \#$ I8: $R \rightarrow L \bullet = / \#$ I9: $S \rightarrow L = R \bullet \#$

分析表为

状态	Action				Goto		
	*	=	I	#	S	L	R
0	s4		s5		1	2	3
1				acc			
2		s6		r5			
3				r2			
4	s4		s5			8	7
5		r4		r4			
6	s4		s5			8	9
7	r3			r3			
8	r5			r5			
9				r1			

四、(15) 用语法制导翻译的思想，把下面的语句翻译成四元式序列。

```

If a=1 or c<5 then
    While a<c and b<d do
        c=c+1
    else c=c+2

```

解：

分析栈

输入栈

动作

```

If a=1 or m1 c<5 then m2
While m3 a<c and m4 b<d do
m5 c=c+1 N

```

else m6 c=c+2

if a=1

if E11

```

E11.t=100
E11.f=101
100: (j=<,a,1,0)→104
101: (j,-,-,0)→102

```

if E11 and m1

if E11 and m1 c<5

if E11 and m1 E12

m1.q=102

```

E12.t=102
E12.f=103
102: (j<,c,5,0)→104
103: (j,-,-,0)→112
Backpatch(E11.f,m1.q)
E1.t={100,102}
E1.f=103
m2.q=104
m3.q=104

E21.t=104
E21.f=105
104: (j<,a,c,0)→106
105: (j,-,-,0)
m4.q=106

```

If E1 then m2

If E1 then m2 while m3

If E1 then m2 while m3 a<c

If E1 then m2 while m3 E21

If E1 then m2 while m3 E21 and m4

If E1 then m2

while m3 E21 and m4 b<d

If E1 then m2

while m3 E21 and m4 E22

If E1 then m2

while m3 E2 do

```

E22.t=106
E22.f=107
106: (j<,b,d,0)→108
107: (j,-,-,0)

Backpatch(E21.t,m4.q)
E2.t={106}
E2.f={105,107}

```

If E1 then m2
while m3 E2 do m5
If E1 then m2
while m3 E2 do m5 c=c+1
If E1 then m2
while m3 E2 do m5 S11

m5.q=108

If E1 then m2 S1

108:(+, c,1,t1)
109:(=,t1,-,c)
S11.n=null
Backpatch(E2.t,m5.q)
S1.n=E2.f={ 105,107}
110:(j,-,-104)
111:(j,-,-0)
N,n=111

If E1 then m2 S1 N

If E1 then m2 S1 N

else m6

If E1 then m2 S1 N

else m6 c=c+2

If E1 then m2 S1 N

else m6 S2

m6.q=112

S2.n=null;

112:(+, c,2,t2)

113:(=,t2,-,c)

Backpatch(E1.t,m2.q)

Backpatch(E1.f,m6.q)

S.n={S1.n,N.n,S2.n}={105,107,111}.

S

五、(20) 对以下基本块，

1、B1 应用 DAG 优化，假设只有 L 在基本块后面还要被引用写出优化后的中间代码序列。

2、B2 应用 DAG 节点排序算法重新排序，假设可用寄存器 R，L 是基本块出口处的活跃变量，生成较优的目标代码。

B1:

F:=B*C

D:=B/C

G:=B*C

H:=G*E

F:=H*G

L:=F

M:=L

B2:

T1:=C+D

T2:=T1-A

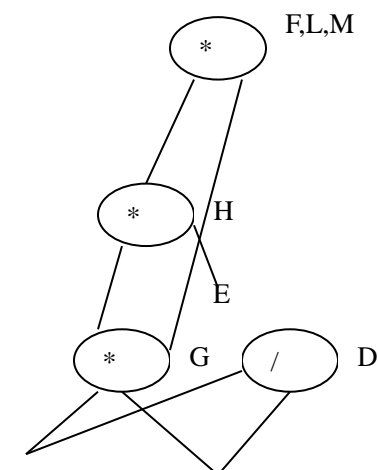
T3:=T2*B

T4:=T1+T3

T5:=T3-F

L:=T4*T5

解 1、B1 的 DAG



只有 L 在基本块后面还要被引用写出优化后的中间代码序列:

S1=B*C

S2=S1*E

L=S2*S1

2、B2 的 DAG

DAG 排序逆序

L,T4,T5,T3

T2,T1

正序为:

T1:=C+D

T2:=T1-A

T3:=T2*B

T5:=T3-F

T4:=T1+T3

L:=T4*T5

生成的优化

目标代码:

LD R,C

ADD R,D

ST R,T1

SUB R,A

MUL R,B

ST R,T3

SUB R,F

ST R,T5

LD R,T1

ADD R,T3

MUL R,T5

ST R,L

