

编译原理练习题参考答案

第 4 章 语法分析

1. 请消除下列文法的左递归, 分别将非终结符排序为: S, Q, R 和 R, Q, S

$$S \rightarrow Qc|c \quad Q \rightarrow Rb|b \quad R \rightarrow Sa|a$$

若排序为 S, Q, R 最终文法为:

$$\begin{aligned} S &\rightarrow Qc|c \\ Q &\rightarrow Rb|b \\ R &\rightarrow bcaR' | caR' | aR' \\ R' &\rightarrow bcaR' | \varepsilon \end{aligned}$$

若排序为 R, Q, S , 最终文法为:

$$\begin{aligned} S &\rightarrow abcS' | bcS' | cS' \\ S' &\rightarrow abcS' | \varepsilon \\ Q &\rightarrow Sab | ab \\ R &\rightarrow Sa | a \end{aligned}$$

2. 判断文法是不是 LL(1)文法

$$S \rightarrow aA | BC \quad A \rightarrow bA | c \quad B \rightarrow Ac | \varepsilon \quad C \rightarrow d | \varepsilon$$

参考答案:

$$FIRST(S) = \{a, b, c, d, \varepsilon\}; \quad \text{因 } \varepsilon \in FIRST(B) \text{ 且 } \varepsilon \in FIRST(C)$$

$$FIRST(A) = \{b, c\}; \quad FIRST(B) = \{b, c, \varepsilon\}; \quad FIRST(C) = \{d, \varepsilon\};$$

$$FOLLOW(S) = \{\$ \}; \quad FOLLOW(A) = \{c, \$ \}; \quad FOLLOW(B) = \{d, \$ \};$$

$$FOLLOW(C) = \{\$ \};$$

$$SELECT(S \rightarrow aA) = \{a\}; \quad SELECT(S \rightarrow BC) = \{b, c, d, \$ \};$$

$$SELECT(A \rightarrow bA) = \{b\}; \quad SELECT(A \rightarrow c) = \{c\};$$

$$SELECT(B \rightarrow Ac) = \{b, c\}; \quad SELECT(B \rightarrow \varepsilon) = \{d, \$ \};$$

$$SELECT(C \rightarrow d) = \{d\}; \quad SELECT(C \rightarrow \varepsilon) = \{\$ \};$$

$$\text{因为 } SELECT(S \rightarrow aA) \cap SELECT(S \rightarrow BC) = \emptyset$$

$$SELECT(A \rightarrow bA) \cap SELECT(A \rightarrow c) = \emptyset$$

$$SELECT(B \rightarrow Ac) \cap SELECT(B \rightarrow \varepsilon) = \emptyset$$

$$SELECT(C \rightarrow d) \cap SELECT(C \rightarrow \varepsilon) = \emptyset$$

所以, 此文法是 LL(1)文法。

3. 练习: 请先改造文法, 消除左递归和左公因子。然后再求所有非终结符的 FIRST 集, FOLLOW 集, 求所有产生式的 SELECT 集并画出预测分析表, 此文法是否是 LL(1)文法。

$$S \rightarrow SAB|ab \quad A \rightarrow Ba|\varepsilon \quad B \rightarrow Db|D \quad D \rightarrow d|\varepsilon$$

参考答案：

(1) 消除左递归和左公因子后，文法为：

$S \rightarrow abS' \quad S' \rightarrow ABS' | \epsilon \quad A \rightarrow Ba | \epsilon \quad B \rightarrow DB' \quad B' \rightarrow b | \epsilon \quad D \rightarrow d | \epsilon$

(2) 求非终结符的 FIRST 集和 FOLLOW 集

$FIRST(S) = \{a\}; FIRST(S') = FIRST(A) = \{a, b, d, \epsilon\};$

$FIRST(B) = \{b, d, \epsilon\}$

$FIRST(B') = \{b, \epsilon\}; FIRST(D) = \{d, \epsilon\};$

$FOLLOW(A) = FOLLOW(B) = FOLLOW(B') = FOLLOW(D) = \{a, b, d, \$\};$

$FOLLOW(S) = FOLLOW(S') = \{\$\};$

(3) 求产生式的 SELECT 集

$SELECT(S \rightarrow abS') = \{a\};$

$SELECT(S' \rightarrow \epsilon) = FOLLOW(S') = \{\$\}$

$SELECT(S' \rightarrow ABS') = FIRST(ABS') - \{\epsilon\} \cup FOLLOW(S') = \{a, b, d, \$\};$

$SELECT(A \rightarrow Ba) = \{a, b, d\};$

$SELECT(A \rightarrow \epsilon) = FOLLOW(A) = \{a, b, d, \$\}$

$SELECT(B \rightarrow DB') = FIRST(DB') - \{\epsilon\} \cup FOLLOW(B) = \{a, b, d, \$\}$

$SELECT(B' \rightarrow b) = \{b\}$

$SELECT(D \rightarrow d) = \{d\}$

$SELECT(B' \rightarrow \epsilon) = FOLLOW(B') = \{a, b, d, \$\}$

$SELECT(D \rightarrow \epsilon) = FOLLOW(D) = \{a, b, d, \$\}$

(4) 画预测分析表

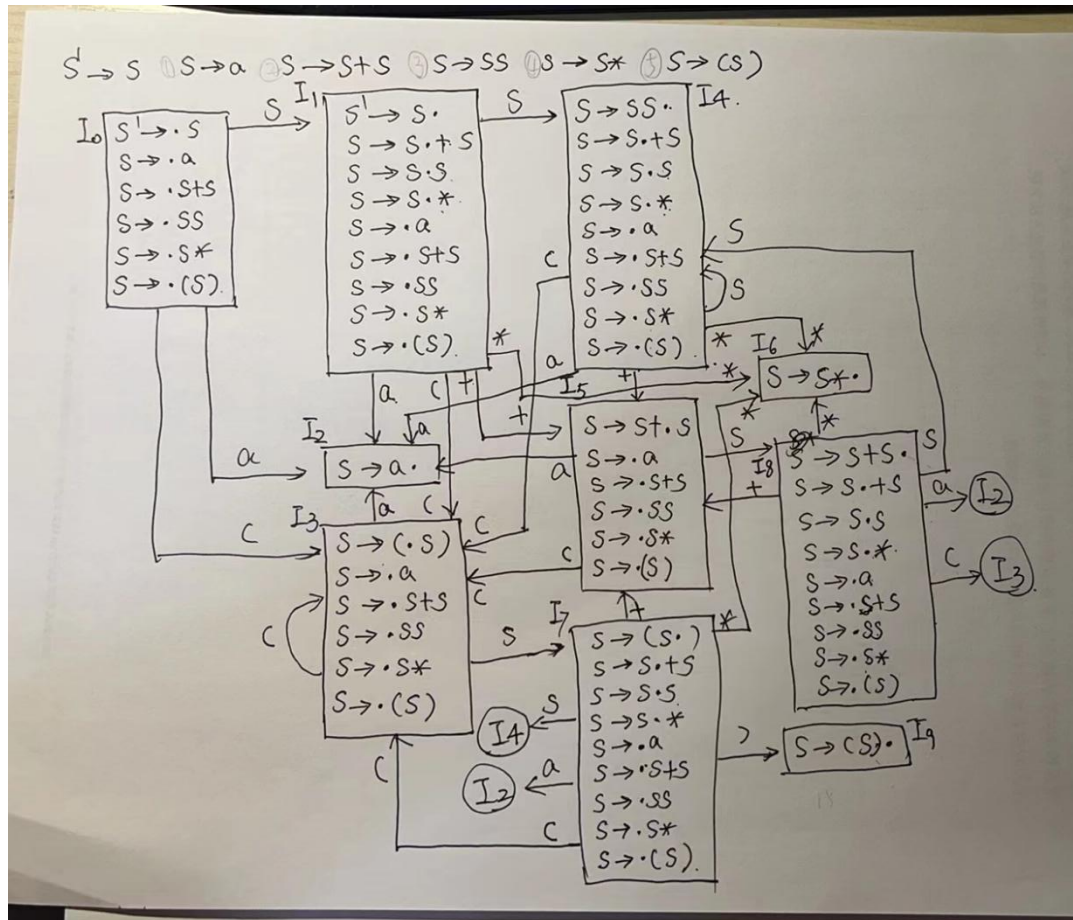
非终结符	输入符号			
	a	b	d	\$
S	$S \rightarrow abS'$			
S'	$S' \rightarrow ABS'$	$S' \rightarrow ABS'$	$S' \rightarrow ABS'$	$S' \rightarrow \epsilon$ $S' \rightarrow ABS'$
A	$A \rightarrow Ba$ $A \rightarrow \epsilon$	$A \rightarrow Ba$ $A \rightarrow \epsilon$	$A \rightarrow Ba$ $A \rightarrow \epsilon$	$A \rightarrow \epsilon$
B	$B \rightarrow DB'$	$B \rightarrow DB'$	$B \rightarrow DB'$	$B \rightarrow DB'$
B'	$B' \rightarrow \epsilon$	$B' \rightarrow b$ $B' \rightarrow \epsilon$	$B' \rightarrow \epsilon$	$B' \rightarrow \epsilon$
D	$D \rightarrow \epsilon$	$D \rightarrow \epsilon$	$D \rightarrow d$ $D \rightarrow \epsilon$	$D \rightarrow \epsilon$

因为预测分析表中有冲突的表项，所以，此文法为非 LL(1) 文法。

4. 画出文法的 LR(0)自动机，并画出 LR(0)分析表，说明是不是 LR(0)文法，为什么？

$S \rightarrow a \mid S + S \mid S S \mid S * \mid (S)$

答：



	ACTION						GOTO
	a	+	*	()	\$	
0	S2			()	\$	S
1	S2	S5	S6	S3			1
2	r1	r1	r1	r1	r1	acc	4
3	S2			S3			7
4	S2/r3	S5/r3	S6/r3	S3/r3	r3	r3	4
5	S2			S3			8
6	r4	r4	r4	r4	r4	r4	
7	S2	S5	S6	S3	S9		4
8	S2/r2	S5/r2	S6/r2	S3/r2	r2	r2	4
9	r5	r5	r5	r5	r5	r5	

因 LR(0)分析表的第 4 和第 8 行有冲突，所以，不是 LR(0)文法。

5. 画出文法的 LR(0)自动机, 并画出 SLR 分析表, 说明是不是 SLR 文法, 为什么?

$$S \rightarrow a \mid S + S \mid S S \mid S * \mid (S)$$

LR(0)自动机问题 4。

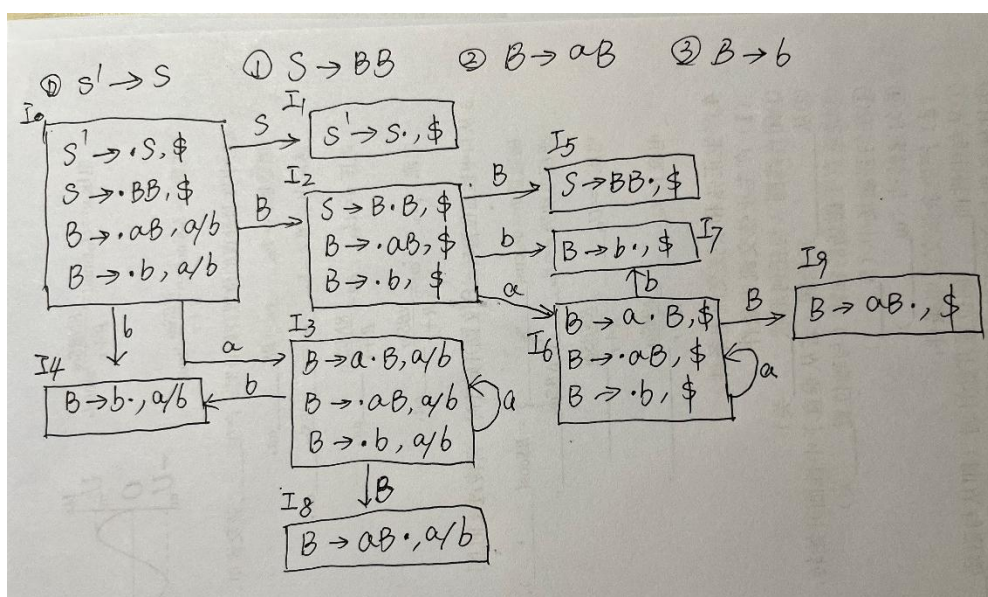
$FOLLOW(S) = \{ a, +, *, (,), \$ \}$, 因此 SLR 分析表问题 4。因 SLR 表项有冲突,

所以不是 SLR 文法。

6. 画出文法的 LR(1)自动机, 并构建 LR(1)分析表

$$0) S' \rightarrow S \quad 1) S \rightarrow BB \quad 2) B \rightarrow aB \quad 3) B \rightarrow b$$

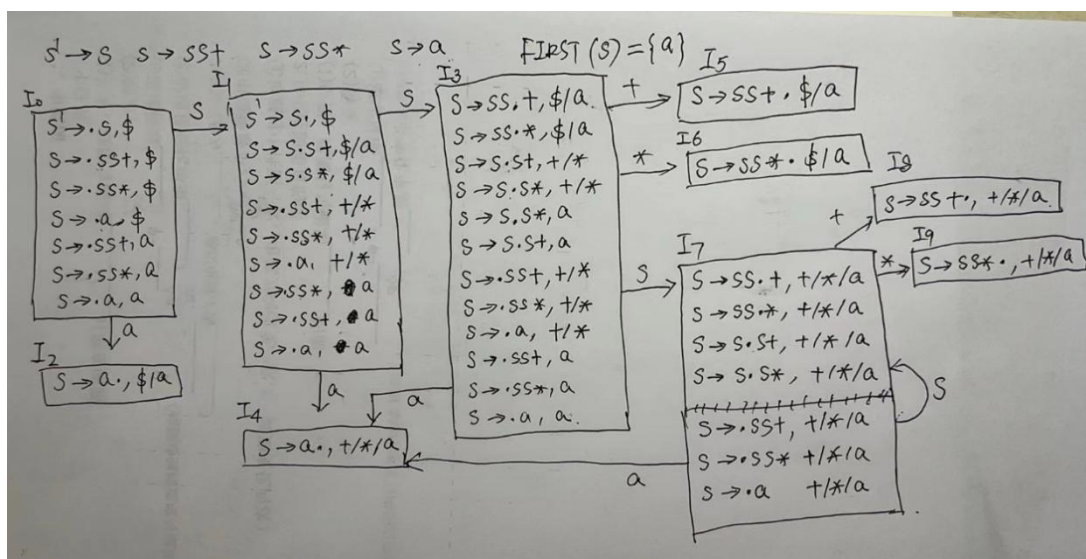
$$FIRST(B) = \{ a, b \}$$



状态	ACTION			GOTO	
	a	b	\$	S	B
0	S3	S4		1	2
1			acc		
2	S6	S7			5
3	S3	S4			8
4	r3	r3			
5			r1		
6	S6	S7			9
7			r3		
8	r2	r2			
9			r2		

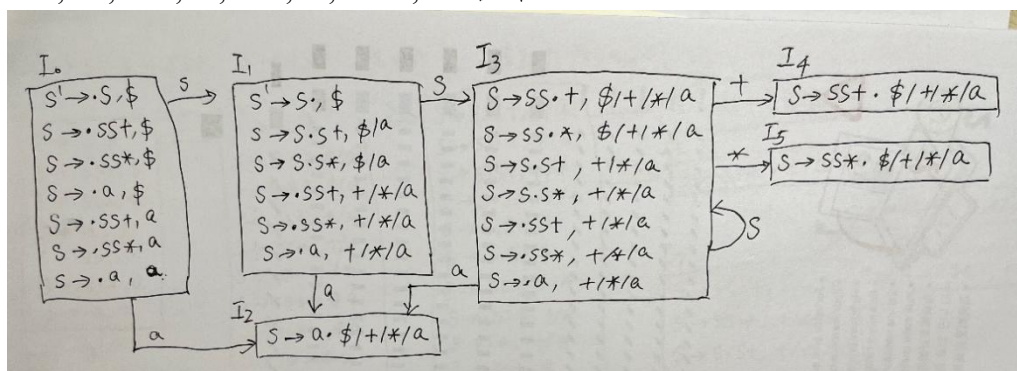
7. 文法 $S \rightarrow SS+ \mid SS* \mid a$ 构造

① 规范 LR(1)项集族 (DFA)



② LALR 项集族 (DFA)

I_2, I_4 ; I_3, I_7 ; I_5, I_8 ; I_6, I_9 合并。



第 5 章语法制导翻译

1. 下面的文法生成了含“小数点”的二进制数：

$S \rightarrow L_1.L_2|L$

$L \rightarrow L_1B|B$

$B \rightarrow 0|1$

(1) 设计一个 S 属性的 SDT 来计算 S.val，即输入串的十进制数值。比如，串 101.101 应该被翻译成十进制数 5.625。

(2) 参考图 5-20，将(1)中的 SDT 改写为在自底向上语法分析栈中实现的栈操作。

(1) 答： S-属性 SDT

$S \rightarrow L_1.L_2 \quad \{ \quad S.val = L_1.val + L_2.val / 2^{L_2.length}; \quad \}$

$S \rightarrow L \quad \{ \quad S.val = L.val; \quad \}$

$L \rightarrow L_1B \quad \{ \quad L.val = L_1.val * 2 + B.val; \quad L.length = L_1.length + 1; \quad \}$

$L \rightarrow B \quad \{ \quad L.val = B.val; \quad L.length = 1; \quad \}$

$B \rightarrow 0 \quad \{ \quad B.val = 0; \quad \}$

$B \rightarrow 1 \quad \{ \quad B.val = 1; \quad \}$

(2) 答：

$S \rightarrow L_1.L_2 \quad \{ \quad \text{stack[top-1].val} = \text{stack[top-1].val} + \text{stack[top].val} /$
 $\text{pow}(2, \text{stack[top].length});$
 $\text{top} = \text{top} - 2; \quad \}$

$S \rightarrow L$

$L \rightarrow L_1B \quad \{ \quad \text{stack[top-1].val} = \text{stack[top-1].val} * 2 + \text{stack[top].val};$
 $\text{stack[top-1].length} = \text{stack[top-1].length} + 1;$
 $\text{top} = \text{top} - 1; \quad \}$

$L \rightarrow B \quad \{ \quad \text{stack[top].length} = 1; \quad \}$

$B \rightarrow 0$

$B \rightarrow 1$

2. 下面的文法生成了含“小数点”的二进制数：

$S \rightarrow L_1.L_2|L \quad L \rightarrow L_1B|B \quad B \rightarrow 0|1$

消除左递归和左公因子后，为 LL(1) 文法：

$S \rightarrow LS' \quad S' \rightarrow .L | \epsilon \quad L \rightarrow BL' \quad L' \rightarrow BL' | \epsilon \quad B \rightarrow 0|1$

为其设计 L 属性 SDD 如下：

$S \rightarrow LS' \quad L.side = left$
 $S.val = L.val + S'.val$

$S' \rightarrow .L \quad L.side = right$
 $S'.val = L.val$

$S' \rightarrow \epsilon \quad S'.val = 0$

$L \rightarrow BL' \quad L'.side = L.side$
 if L.side=left then L.val= B.val*2^{L'.length}+ L'.val
 if L.side=right then L.val= (B.val + L'.val)/2
 L.length= L'.length +1

$L' \rightarrow BL_1'$ $L_1'.side = L'.side$
 if $L'.side = \text{left}$ then $L'.val = B.val * 2^{L_1'.length} + L_1'.val$
 if $L'.side = \text{right}$ then $L'.val = (B.val + L_1'.val)/2$
 $L'.length = L_1'.length + 1$
 $L' \rightarrow \epsilon$ $L'.val = 0; L'.length = 0$
 $B \rightarrow 0$ $B.val = 0$
 $B \rightarrow 1$ $B.val = 1$

(1) 找出哪些是继承属性，哪些是综合属性，并将 L-SDD 转换为 L 属性的 SDT。

$S \rightarrow$ { a1: $L.side = \text{left};$ }
 LS' { a2: $S.val = L.val + S'.val;$ }
 $S' \rightarrow \cdot$ { a3: $L.side = \text{right};$ }
 L { a4: $S'.val = L.val;$ }
 $S' \rightarrow \epsilon$ { a5: $S'.val = 0;$ }
 $L \rightarrow B$ { a6: $L'.side = L.side;$ }
 L' { a7: if $L.side = \text{left}$ then $L.val = B.val * 2^{L'.length} + L'.val$;
 if $L.side = \text{right}$ then $L.val = (B.val + L'.val)/2$;
 $L.length = L'.length + 1;$ }
 $L' \rightarrow B$ { a8: $L_1'.side = L'.side;$ }
 L_1' { a9: if $L'.side = \text{left}$ then $L'.val = B.val * 2^{L_1'.length} + L_1'.val$;
 if $L'.side = \text{right}$ then $L'.val = (B.val + L_1'.val)/2$;
 $L'.length = L_1'.length + 1;$ }
 $L' \rightarrow \epsilon$ { a10: $L'.val = 0; L'.length = 0;$ }
 $B \rightarrow 0$ { a11: $B.val = 0;$ }
 $B \rightarrow 1$ { a12: $B.val = 1;$ }

(2) 为了让(1)中 SDT 能在自顶向下分析中实现，请写出产生式 $L \rightarrow BL'$ 入栈时，与右部各记录相关联的语义动作的栈操作。(参考第 5 章 PPT73-76 页)。

答：

$S \rightarrow$ { a1: $L.side = \text{left};$ }
 LS' { a2: $S.val = L.val + S'.val;$ }

符号	属性	执行代码	动作
a1		stack[top-1].side = left; top=top-1;	
L	side	若选: $L \rightarrow BL'$ 推导 stack[top-3].Lside = stack[top].side; stack[top].Lside = stack[top].side; top=top+5;	复制到 {a6} 和 {a7}
Lsyn	val, length	stack[top-3].Lval = stack[top].val; top=top-1;	复制到 {a2}
S'			
S'syn	val	stack[top-1].S'val = stack[top].val; top=top-1;	复制到 {a2}
a2	Lval, S'val	stack[top-1].val; = stack[top].Lval+stack[top].s'val; top=top-1;	

$S' \rightarrow \cdot$ { a3: $L.side = \text{right};$ }

L { a4: S'.val=L.val; }

符号	属性	执行代码	动作
.			
a3		stack[top-1].side = right; top = top-1;	
L	side	若选: L->BL' 推导 stack[top-3].Lside = stack[top].side; stack[top].Lside = stack[top].side; top=top+5;	复制到{a6} 和{a7}
Lsyn	val, length	stack[top-1].Lval = stack[top].val; top = top-1;	复制到{a4}
a4	Lval	stack[top-1] = stack[top].Lval; top = top-1;	

S' → ε { a5: S'.val= 0; }

符号	属性	执行代码	动作
a5		stack[top-1].val = 0; top = top-1;	

L → B { a6: L'.side=L.side; }

L' { a7: if L.side=left then L.val= B.val*2^{L'.length}+ L'.val ;
if L.side=right then L.val= (B.val + L'.val)/2;
L.length= L'.length +1 }

符号	属性	执行代码	动作
B	-		
Bsyn	val	stack[top-4].Bval = stack[top].val; top = top-1;	复制到{a7}
a6	Lside	stack[top-1].side = stack[top].Lside; top = top-1;	
L'	L'side	根据当前输入符号选择产生式进行推导 若选: L'->BL' 推导: stack[top+3].L'side = stack[top].side; stack[top].L'side = stack[top].side; top=top+5; 若选: L'->ε 推导: 无需操作	复制到{a8} 和{a9}
L'syn	val , length	stack[top-1].L'val = stack[top].val; stack[top-1].L'length = stack[top].length; top = top-1;	复制到{a7}
a7	Lside, Bval, L'val, L'length	if (stack[top].Lside ==left) stack[top-1].val = stack[top].Bval* pow(2,stack[top].L'length)+stack[top].L'val; if(stack[top].Lside==right) stack[top-1].val = (stack[top].Bval+stack[top].L'val)/2; stack[top-1].length = stack[top].L'length; top=top-1;	

L' → B { a8: L₁'.side=L'.side;}

L₁' { a9: if L'.side=left then L'.val= B.val*2^{L₁'.length}+ L₁'.val ;
if L'.side=right then L'.val= (B.val + L₁'.val)/2;
L'.length= L₁'.length +1; }

符号	属性	执行代码	动作
B	-		
Bsyn	val	stack[top-4].Bval = stack[top].val; top = top-1;	复制到{a9}
a8	Lside	stack[top-1].side = stack[top].Lside; top = top-1;	

L'	L'side	根据当前输入符号选择产生式进行推导 若选: $L' \rightarrow BL'$ 推导: stack[top+3].L'side = stack[top].side; stack[top].L'side = stack[top].side; top=top+5; 若选: $L' \rightarrow \epsilon$ 推导: 无需操作	复制 到 {a8} 和 {a9}
L'syn	val , length	stack[top-1].L'val = stack[top].val; stack[top-1].L'length = stack[top].length; top = top-1;	复制 到 {a9}
a9	L'side, Bval, L'val, L'length	if (stack[top].L'side == left) stack[top-1].val = stack[top].Bval* pow(2,stack[top].L'length)+stack[top].L'val; if(stack[top].L'side==right) stack[top-1].val = (stack[top].Bval+stack[top].L'val)/2; stack[top-1].length = stack[top].L'length; top=top-1;	

$L' \rightarrow \epsilon$ { a10: L'.val=0; L'.length=0; }

符号	属性	执行代码	动作
a10		stack[top-1].val = 0; stack[top-1].length = 0; top = top-1;	

$B \rightarrow 0$ { a11: B.val=0; }

符号	属性	执行代码	动作
0			
a11		stack[top-1].val = 0; top = top-1;	

$B \rightarrow 1$ { a12: B.val=1; }

符号	属性	执行代码	动作
1			
a12		stack[top-1].val = 1; top = top-1;	

3. 将上面练习中的 L 属性的 SDT 改造成可以在自底向上语法分析中实现的 SDT。

答:

$S \rightarrow M L S'$ { a2: S.val= L.val+ S'.val; }

$M \rightarrow \epsilon$ { M.syn = left }

$S' \rightarrow \cdot N L$ { a4: S'.val= L.val; }

$N \rightarrow \epsilon$ { M.syn = right }

$S' \rightarrow \epsilon$ { a5: S'.val= 0; }

$L \rightarrow B P L'$ { a7: if L.side=left then L.val = B.val*2^{L'.length}+ L'.val ;

if L.side=right then L.val = (B.val + L'.val)/2;

L. length = L'.length +1; }

$P \rightarrow \epsilon$ { P.inh = L.side; P.syn = P.inh; }

$L' \rightarrow B \quad Q \ L_1' \{ \quad a_9: \quad \text{if } L'.\text{side}=\text{left} \text{ then } L'.\text{val} = B.\text{val} * 2^{L_1'.\text{length}} + L_1'.\text{val};$
 $\quad \text{if } L'.\text{side}=\text{right} \text{ then } L'.\text{val} = (B.\text{val} + L_1'.\text{val})/2;$
 $\quad \quad L'.\text{length} = L_1'.\text{length} + 1; \quad \}$
 $Q \rightarrow \epsilon \{ Q.\text{inh} = L'.\text{side}; Q.\text{syn} = Q.\text{inh}; \}$
 $L' \rightarrow \epsilon \quad \{ a_{10}: L'.\text{val} = 0; L'.\text{length} = 0; \}$
 $B \rightarrow 0 \quad \{ a_{11}: B.\text{val} = 0; \}$
 $B \rightarrow 1 \quad \{ a_{12}: B.\text{val} = 1; \}$

第 6 章 中间代码生成

练习 1：使用讲义中的翻译方案翻译下面赋值语句生成三地址码序列。假设 a 和 b 的类型表达式都是 `array(3, array(5, real))`，real 类型占 8 个存储单元。

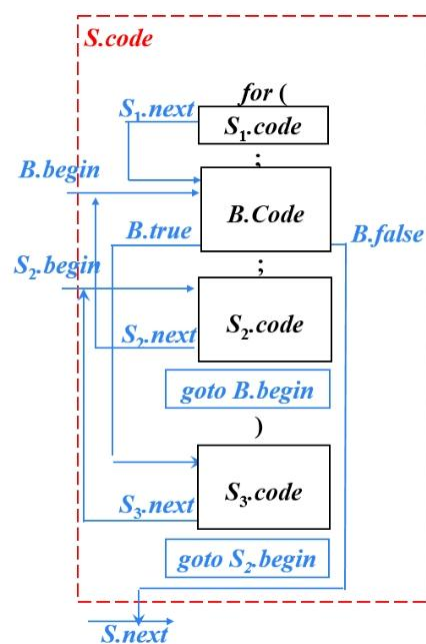
$x = a[i][j] + b[i][j]$

参考答案：

$t_1 = i * 40$
 $t_2 = j * 8$
 $t_3 = t_1 + t_2$
 $t_4 = a[t_3]$
 $t_5 = i * 40$
 $t_6 = j * 8$
 $t_7 = t_5 + t_6$
 $t_8 = b[t_7]$
 $t_9 = t_4 + t_8$
 $x = t_9$

练习 2：根据给定的代码结构图，参考 6.4 控制流语法的翻译，构造非回填的 SDT，实现 for 控制流的翻译。

for 循环语句： $S \rightarrow \text{for } (S_1; B; S_2) S_3$



答:

```
S → for ( { S1.next = newlabel(); }  
S1 { label( S1.next); }  
; { B.begin = newlabel(); label(B.begin); B.true = newlabel(); B.false = S.next; }  
B ; { S2.begin = newlabel(); label(S2.begin); S2.next = B.begin; }  
S2 { gen( 'goto' B.begin); label(B.true); S3.next = S2.begin; }  
S3 { gen( 'goto' S2.begin); }
```

3. 试将下面的语句翻译成三地址码和对应的四元式序列（指令标号从 1 开始，跳转指令目标标号是数字代表的标号），不必考虑避免生成冗余的 goto 语句。

(1) while a<c and b<d do

```
    if a=1 then c=c+1  
    else while a<=d do  
        a=a+2
```

答:

三地址码

```
(1) if a<c goto (3)  
(2) goto()  
(3) if b<d goto (5)  
(4) goto ()  
(5) if a=1 goto (7)  
(6) goto(10)  
(7) t1=c+1  
(8) c=t1  
(9)goto(1)  
(10) if a<=d goto(12)  
(11) goto(1)  
(12)t2=a+2  
(13) a=t2  
(14)goto(10)
```

四元式

```
(1) (j<, a, c, 3)  
(2) (j, -, -, -)  
(3) (j<, b, d, 5)
```

- (4) (j, -, -, -)
- (5) (j=, a, 1, 7)
- (6) (j, -, -, 10)
- (7) (+, c, 1, t1)
- (8) (=, t1, -, c)
- (9) (j, -, -, 1)
- (10) (j<=, a, d, 12)
- (11) (j, -, -, 1)
- (12) (+, a, 2, t2)
- (13) (=, t2, -, a)
- (14) (j, -, -, 10)

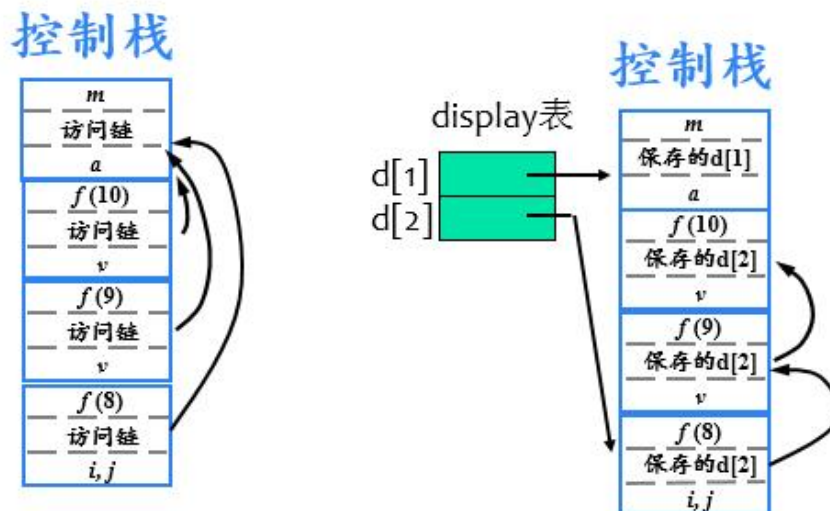
第 7 章 运行存储分配

练习：设有如下的 Pascal 程序：

```
program main( input, output);  
  
  var k: integer;  
  function f(n: integer): integer;  
    begin if n <=0 then f:=1 else f:=n* f(n-1) end;  
  begin  
    k:= f(10);  
  
    writeln(k);  
  end
```

- (1)若采用访问链存取非局部名字，当第 3 次(递归地)进入后，试绘制运行栈中的活动记录示意图(至少标示活动名称，画出访问链和控制链及其指向的活动记录)。
- (2) 若采用 display 表方式实现非局部数据的访问，参考教材图 7-14 重做(1)。

设栈底在上，栈顶在下，控制链直接指向当前活动记录栈底方向紧临的调用者的活动记录。访问链结构如下图。

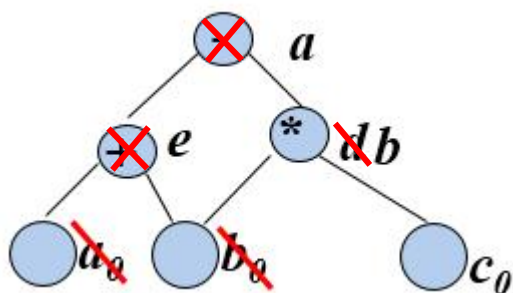


第 8 章 代码优化

练习 1: 为下面的基本块构造 DAG, 假设只有 b 在出口处活跃, 重组基本块的代码。

$d = b * c$
 $e = a + b$
 $b = b * c$
 $a = e - d$

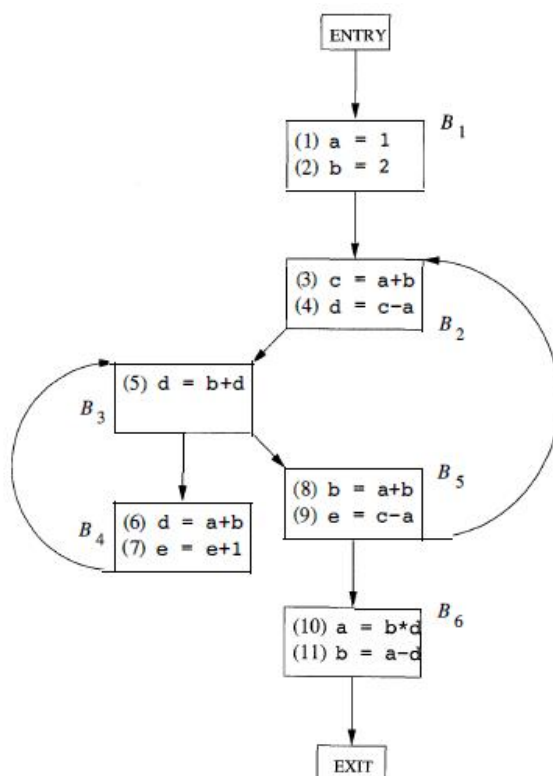
参考答案:



重组后的代码:

$b = b * c$

练习 2. 给出如图所示的流图, 计算活跃变量分析中各个基本块的 def 、 use 、 IN 和 OUT 集合。



参考答案

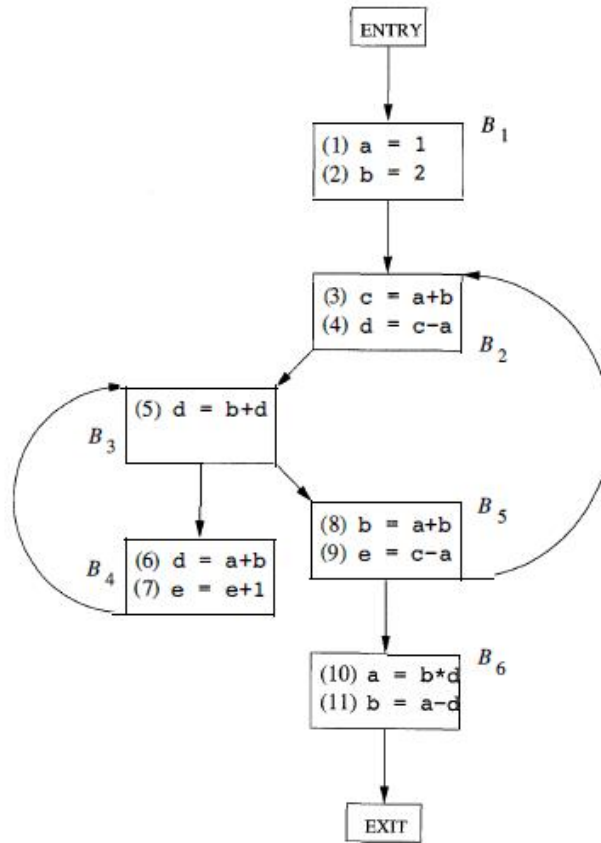
➤ $USCB1 = \Phi$

- $\text{def}_{B1} = \{ a, b \}$
- $\text{use}_{B2} = \{ a, b \}$
- $\text{def}_{B2} = \{ c, d \}$
- $\text{use}_{B3} = \{ b, d \}$
- $\text{def}_{B3} = \Phi$
- $\text{use}_{B4} = \{ a, b, e \}$
- $\text{def}_{B4} = \{ d \}$
- $\text{use}_{B5} = \{ a, b, c \}$
- $\text{def}_{B5} = \{ e \}$
- $\text{use}_{B6} = \{ b, d \}$
- $\text{def}_{B6} = \{ a \}$

	OUT[B]	IN[B]
B6	Φ	bd
B5	abde	abcd
B4	abcde	abce
B3	abcde	abcde
B2	abcde	abe
B1	abe	e

练习 3. 对于下图中的流图：

- (1) 找出流图中的自然循环。
- (2) B_1 中的语句 (1) 和 (2) 都是复制语句且 a 和 b 都被赋予了常量值。我们可以对 a 和 b 的哪些引用进行复制传播，并把对它们的引用替换为对一个常量的引用？在所有可能的地方进行这种替换。
- (3) 找出所有的全局公共子表达式。



参考答案：

- (1)
 - 循环{B₃、B₄}
 - 循环{B₂、B₃、B₄、B₅}
- (2) 可以对 a 在语句 (3) (4) (6) (8) (9) 处的使用进行复制传播，b 都不能复制传播。
- (3) 语句 (6) 中的 a+b
 语句 (8) 中的 a+b
 语句 (9) 中的 c-a