

## 第 8 章 语法制导翻译和中间代码生成

### 第 1 题

给出下面表达式的逆波兰表示(后缀式):

(1)  $a * (-b + c)$

(2)  $\text{if}(x+y)*z=0 \text{ then } s := (a+b)*c \text{ else } s := a*b*c$

答案:

给出下面表达式的逆波兰表示(后缀式):

(1)  $ab-c+*$

(2)  $xy+z*0=sab+c*:=sab*c*:=Y$  (注:  $Y$  表示 if-then-else 运算)

如果写成这样:  $xy+z*0=sab+c*:=sabc**:=Y$ , 则是错误的, 因为写表达式和赋值语句的中间代码序列, 或是写它们的代码生成过程, 必须注意按照算符优先序进行, 这实际上是按照 LR 分析过程进行的。例如: 写出赋值语句  $a:=a+b*c*(d+e)$  的四元式中间代码, 当前四元式序号为 100。不能写成:

```
100  (+,d,e,t1)
101  (*,b,c,t2)
102  (*,t2,t1,t3)
103  (+,a,t3,t4)
104  (:=,t4,-,a)
```

应该写成:

```
100  (*,b,c,t1)
101  (+,d,e,t2)
102  (*,t1,t2,t3)
103  (+,a,t3,t4)
104  (:=,t4,-,a)
```

### 第 2 题

请将表达式  $-(a+b)*(c+d)-(a+b+c)$  分别表示成三元式、间接三元式、四元式序列、树形、逆波兰, 当前序号为 100。

答案:

三元式:

```
100  (+, a, b)
101  (+, c, d)
102  (*, (1), (2))
```

103 (-, (3), /)

104 (+, a, b)

105 (+, (5), c)

106 (- (4), (6))

**间接三元式:**

间接三元式序列	间接码表
100 (+, a, b)	(100)
101 (+, c, d)	(101)
102 (*, (1), (2))	(102)
103 (-, (3), /)	(103)
104 (+, (1), c)	(100) (104)
105 (-, (4), (1))	(105)

或者:

**间接三元式:**

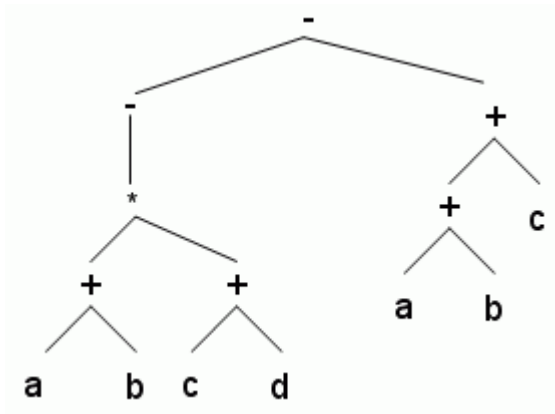
100 (+, a, b)  
101 (+, c, d)  
102 (\*, (1), (2))  
103 (-, (3), /)  
104 (+, (1), c)  
105 (-, (4), (1))

间接码表: 100 101 102 103 100 104 105

**四元式:**

100 (+, a, b, t1)  
101 (+, c, d, t2)  
102 (\*, t1, t2, t3)  
103 (-, t3, /, t4)  
104 (+, a, b, t5)  
105 (+, t5, c, t6)  
106 (-, t4, t6, t7)

**树形:**



逆波兰:  $ab+cd+*-ab+c+-$

[典型例题]:

写出 if A and B and C > D then

if A < B then F:=1

else F:=0

else G:=G+1; 的四元式序列, 翻译过程中, 采用 then 与 else 的最近匹配原则。

- |                 |                            |
|-----------------|----------------------------|
| (1) (jnz,A,_,3) | /* A and B and C>D 的四元式 */ |
| (2) (j,_,_,13)  |                            |
| (3) (jnz,B,_,5) |                            |
| (4) (j,_,_,13)  |                            |
| (5) (j>,C,D,7)  |                            |
| (6) (j,_,_,13)  |                            |
| (7) (j<,A,B,9)  | /* A < B 的四元式 */           |
| (8) (j,_,_,11)  |                            |
| (9) (:=,1,_,F)  | /* F:=1 */                 |
| (10) (j,_,_,14) |                            |
| (11) (:=,0,_,F) | /* F:=0 */                 |
| (12) (j,_,_,14) |                            |
| (13) (:=,G,1,G) |                            |
| (14)            |                            |

[典型例题]:

写出 WHILE A<C AND B<D DO

IF A=1 THEN C:=C+1 ELSE

WHILE A<=D DO A:=A+2;的四元式序列。

- |       |              |
|-------|--------------|
| (100) | (j<,A,C,102) |
| (101) | (j,=,1,114)  |
| (102) | (j<,B,D,104) |

- (103) (j,-,114)
- (104) (j=,A,1,106)
- (105) (j,-,109)
- (106) (+,C,1,T)
- (107) (:=,T,-,C)
- (108) (j,-,100)
- (109) (j<=,A,D,111)
- (110) (j,-,100)
- (111) (+,A,2,T)
- (112) (:=,T,-,A)
- (113) (j,-,109)
- (114)

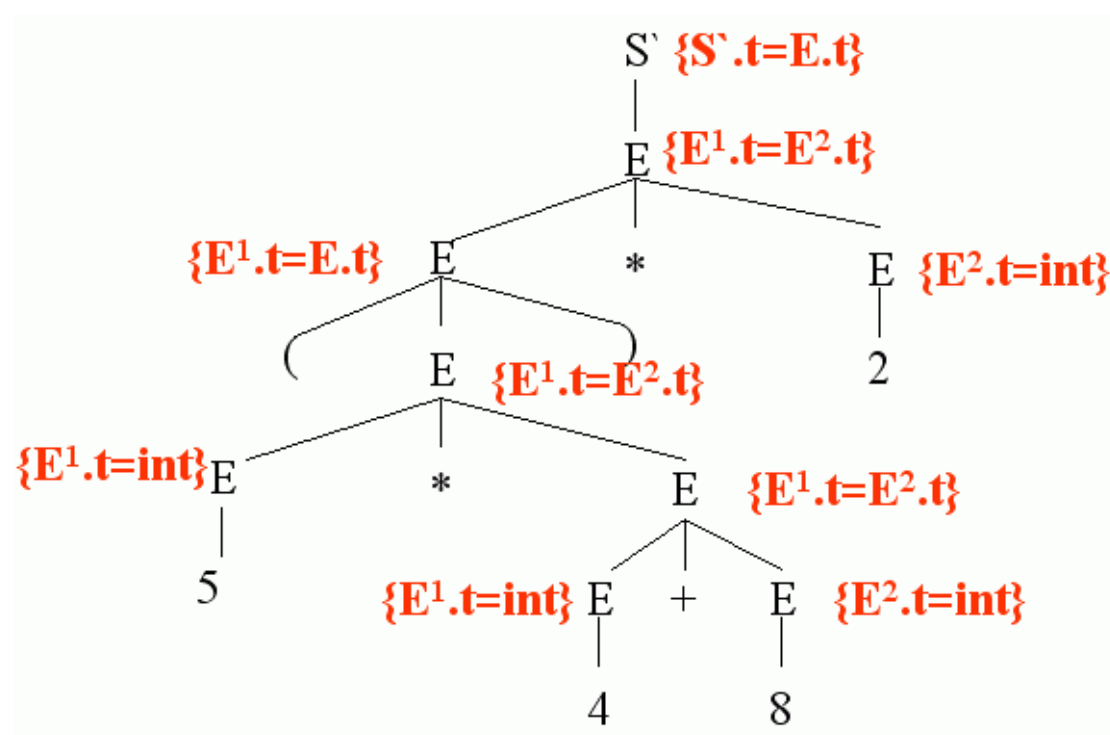
## 第3题

采用语法制导翻译思想，表达式  $E$  的“值”的描述如下：

产生式	语义动作
(0) $S' \rightarrow E$	{print E.VAL}
(1) $E \rightarrow E^1 + E^2$	{E.VAL := E <sup>1</sup> .VAL + E <sup>2</sup> .VAL}
(2) $E \rightarrow E^1 * E^2$	{E.VAL := E <sup>1</sup> .VAL * E <sup>2</sup> .VAL}
(3) $E \rightarrow (E^1)$	{E.VAL := E <sup>1</sup> .VAL}
(4) $E \rightarrow n$	{E.VAL := n.LEXVAL}

如采用 LR 分析方法，给出表达式  $(5*4+8)*2$  的语法树并在各结点注明语义值 VAL。

答案：



采用语法制导翻译思想，表达式 E 的“值”的描述如下：

产生式	语义动作
(0) $S' \rightarrow E$	{print E.VAL}
(1) $E \rightarrow E^1 + E^2$	{E.VAL := E <sup>1</sup> .VAL + E <sup>2</sup> .VAL}
(2) $E \rightarrow E^1 * E^2$	{E.VAL := E <sup>1</sup> .VAL * E <sup>2</sup> .VAL}
(3) $E \rightarrow (E^1)$	{E.VAL := E <sup>1</sup> .VAL}
(4) $E \rightarrow n$	{E.VAL := n.LEXVAL}

假如终结符 n 可以是整数或实数，算符+和\*的运算对象类型一致，语义处理增加“类型匹配检查”，请给出相应的语义描述。

答案：

```

(0)  $S' \rightarrow E$  { if error≠1 then print E.VAL}
(1)  $E \rightarrow E^1 + E^2$  { if E1.TYPE=int AND E2.TYPE=int then
    begin
        E.VAL:=E1.VAL + E2.VAL;
        E.YTPE:=int;
    end
    else if E1.TYPE=real AND E2.TYPE=real then
    begin
        E.VAL:=E1.VAL + E2.VAL;
        E.YTPE:=real;
    end
    else error=1
}
(2)  $E \rightarrow E^1 * E^2$  { if E1.TYPE=int AND E2.TYPE=int then
    begin
        E.VAL:=E1.VAL * E2.VAL;;
        E.YTPE:=int;
    end
    else if E1.TYPE=real AND E2.TYPE=real then
    begin
        E.VAL:=E1.VAL * E2.VAL;;
        E.YTPE:=real;
    end
    else error=1
}
(3)  $E \rightarrow (E^1)$  { E.VAL:=E1.VAL;
    E.TYPE:=E1.TYPE }
(4)  $E \rightarrow n$  { E.VAL:=n.LEXVAL;
    E.TYPE:=n.LEXTYPE }

```

## 第5题

令  $S.val$  为下面的文法由  $S$  生成的二进制数的值(如, 对于输入 101.101,  $S.val=5.625$ );

$S \rightarrow L.L \mid L$

$L \rightarrow LB \mid B$

$B \rightarrow 0 \mid 1$

按照语法制导翻译的方法, 对每个产生式给出相应的语义规则。(中国科学院计算所 1995 年)

答案: 加入新的开始符号  $S'$  和规则  $S' \rightarrow S$ , 得到增广文法。语法制导定义如下:

产生式	语义规则
$S' \rightarrow S$	$\text{print}(S.val)$
$S \rightarrow L_1.L_2$	$S.val := L_1.val + L_2.val / 2^{L_2.length}$
$S \rightarrow L$	$S.val := L.val$
$L \rightarrow L_1B$	$L.val := L_1.val * 2 + B.val$ $L.length := L_1.length + 1$
$L \rightarrow B$	$L.val := B.val$ $L.length := 1$
$B \rightarrow 0$	$B.val := 0$
$B \rightarrow 1$	$B.val := 1$

如果题目是  $S::=L.L \mid L$   $L::=LB \mid B$   $B::=0 \mid 1$  则写成:

$S'::=S \quad \{ \text{print}(S.val); \}$

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

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

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

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

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

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

## 第 6 题

下面文法产生的表达式是对整型和实型常数应用算符+形成的。当两个整数相加时,结果为整数,否则为实数。

$$E \rightarrow E+T \mid T$$

$$T \rightarrow \text{num.num} \mid \text{num}$$

- (1) 给出语法制导定义确定每个子表达式的类型。
- (2) 把表达式翻译成前缀形式, 并且决定类型。试用一元运算符 `inttoreal` 把整型值转换为相等的实型值, 以使得前缀表达式中两个运算对象是同类型的。

答案:

- (1) 设 `type` 是综合属性, 代表各非终结符的“类型”属性

语法制导定义

产生式	语义规则
$E \rightarrow E_1+T$	IF ( $E_1.type=integer$ ) and ( $T.type=integer$ ) THEN $E.type:=integer$ ELSE $E.type:=real$
$E \rightarrow T$	$E.type:=T.type$
$T \rightarrow \text{num.num}$	$T.type:=real$
$T \rightarrow \text{num}$	$T.type:=integer$

- (2) 设 `code` 为综合属性, 代表各非终结符的代码属性  
`type` 为综合属性, 代表各非终结符的类型属性  
`inttoreal` 把整型值转换为相等的实型值  
`vtochar` 将数值转换为字符串



## 语法制导定义

产生式	语义规则
$S \rightarrow E$	print E.code
$E \rightarrow E_1 + T$	<pre> IF (E1.type=integer) and (T.type=integer) THEN   begin     E.type:=integer     E.code:='+'  E1.code  T.code;   end ELSE begin   E.type:=real   IF E1.type=integer THEN     begin       E1.type:=real       E1.val:=inttoreal(E1.val)       E1.code=vtochar(E1.val)     end   IF T.type:=integer THEN     begin       T.type:=real       T.val:=inttoreal(T.val)       T.code=vtochar(T.val)     end   E.code:='+'  E1.code  T.code; End </pre>
$E \rightarrow T$	<pre> E.type:=T.type E.val:=T.val E.code=vtochar(E.val) </pre>
$T \rightarrow \text{num.num}$	<pre> T.type:=real T.val:=num.num.lexval T.code=vtochar(T.val) </pre>
$T \rightarrow \text{num}$	<pre> T.type:=integer T.val:=num.lexva T.code=vtochar(T.val) </pre>

## 第 7 题

假设变量的说明是由下列文法生成的：

$D \rightarrow i L$

$L \rightarrow, i L \mid : T$

$T \rightarrow \text{integer} \mid \text{real}$

建立一个语法制导定义，把每一个标志符的类型加在符号表中。

答案：

type 为综合属性，代表类型属性，

函数 addtype 实现向符号表中 i 对应项填类型信息。

语法制导定义

产生式	语义动作
$D \rightarrow i L$	$D.Type := L.Type$ $\text{addtype}(i.\text{entry}, D.type)$
$L \rightarrow, i L1$	$L.Type := L1.Type$ $\text{addtype}(i.\text{entry}, L.type)$
$L \rightarrow : T$	$L.type := T.type$
$T \rightarrow \text{integer}$	$T.type := \text{integer}$
$T \rightarrow \text{real}$	$T.type := \text{real}$

## 附加题

### 问题 1:

请将下列语句

**while (A<B do if (C>D) then X:=Y+Z**

**翻译成四元式**

### 答案:

假定翻译的四元式序列从 (100) 开始:

```
(100)  if A<B goto (102)
(101)  goto (107)
(102)  if C<D got (104)
(103)  goto (100)
(104)  T := Y+Z
(105)  X := T
(106)  goto (100)
(107)
```

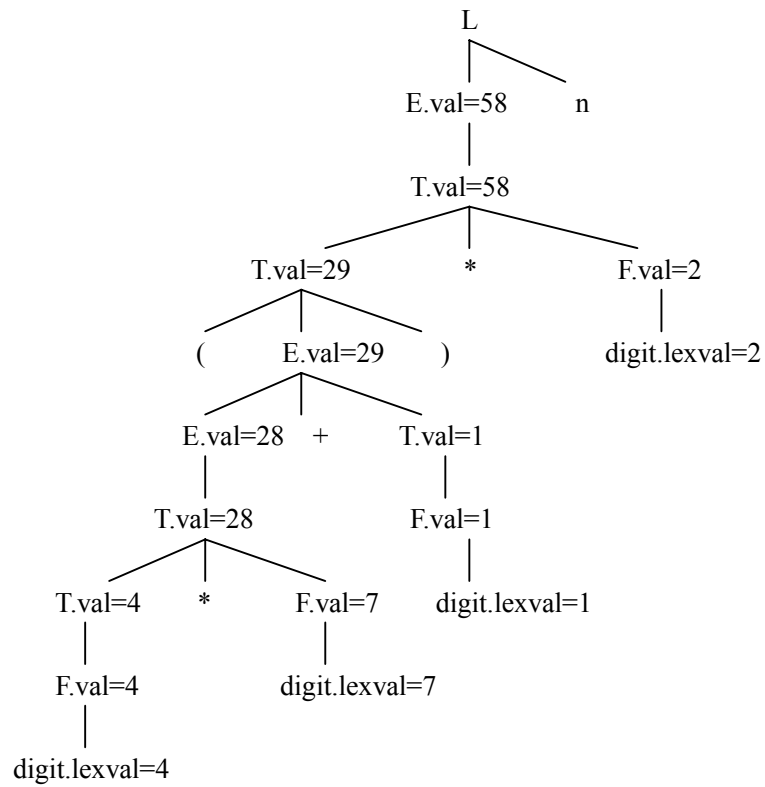
### 问题 2:

对于输入的表达式 $(4*7+1)*2$ , 根据下表的语法制导定义建立一棵带注释的分析树。  
val:表示非终结符的整数值, 综合属性, lexval 是单词 digit 的属性

语法制导定义

产生式	语义规则
$L \rightarrow E$	print (E.val)
$E \rightarrow E^1 + T$	$E.val := E^1.val + T.val$
$E \rightarrow T$	$E.val := T.val$
$T \rightarrow T^1 * F$	$T.val := T^1.val * F.val$
$T \rightarrow F$	$T.val := F.val$
$F \rightarrow (E)$	$F.val := E.val$
$F \rightarrow \text{digit}$	$F.val := \text{digit.lexval}$

答案:



## 问题 3:

请按语法制导的定义，将后缀表达式翻译成中缀表达式。注意，不允许出现冗余括号，后缀表达式的文法如下：

$$E \rightarrow EE+$$

$$E \rightarrow EE*$$

$$E \rightarrow id$$

答案:

语法制导定义

产生式	语义规则
$S \rightarrow E$	print E.code
$E \rightarrow E_1 E_2 +$	E.code=E <sub>1</sub> .code  '+'  E <sub>2</sub> .code; E.op='+'
$E \rightarrow E_1 E_2 *$	IF E <sub>1</sub> .op='+' AND E <sub>2</sub> .op='+' THEN E.code="("  E <sub>1</sub> .code  ")"*'('  E <sub>2</sub> .code  ')'; ELSE IF E <sub>1</sub> .op='+' THEN E.code="("  E <sub>1</sub> .code  ")"*'  E <sub>2</sub> .code'; ELSE IF E <sub>2</sub> .op='+' THEN E.code=E <sub>1</sub> .code  '*'('  E <sub>2</sub> .code  ')'; ELSE E.code=E <sub>1</sub> .code  '*'  E <sub>2</sub> .code  ';
$E \rightarrow id$	E.code:=id.lexeme;

## 问题 4:

有文法:

$$S \rightarrow (L)a$$

$$L \rightarrow L,S|S$$

给此文法配上语义动作子程序(或者说为此文法写一个语法制导定义)，它输出配对括号的个数。如对于句子(a,(a,a))，输出是 2。(中国科学院计算所 1994)

答案:

加入新开始符号 S'和产生式  $S' \rightarrow S$ ，设 num 为综合属性，代表值属性，则语法制导定义如下:

产生式	语义规则
$S' \rightarrow S$	print(S.num)
$S \rightarrow (L)$	S.num:=L.num+1
$S \rightarrow a$	S.num:=0
$L \rightarrow L_1,S$	L.num:=L <sub>1</sub> .num+S.num
$L \rightarrow S$	L.num:=S.num

## 问题 5:

文法 G 的产生式如下:

$S \rightarrow (L)a$

$L \rightarrow L, S | S$

①试写出一个语法制导定义, 它输出配对括号个数;

②写一个翻译方案, 打印每个 a 的嵌套深度。如((a),a),打印 2,1。(中国科学院软件所 1999)

答案:

①为 S,L 引入综合属性 num, 代表配对括号个数;

语法制导定义

产生式	语义动作
$S' \rightarrow S$	print(S.num)
$S \rightarrow (L)$	$S.num := L.num + 1$
$S \rightarrow a$	$S.num := 0$
$L \rightarrow L_1, S$	$L.num := L_1.num + S.num$
$L \rightarrow S$	$L.num := S.num$

②引入继承属性 f, 代表嵌套深度

$S' \rightarrow \{S.f := 0\} \quad S$

$S \rightarrow '(' \quad \{L.f := S.f + 1;\}$

$L$

$)'$

$S \rightarrow a \quad \{\text{print}(S.f);\}$

$L \rightarrow \{L_1.f := L.f;\}$

$L_1, \{S.f := L.f\}$

$S$

$L \rightarrow \{S.f := L.f;\}$

$S$

## 问题 6:

对下面的文法，只利用综合属性获得类型信息。

$D \rightarrow L, id \mid L$

$L \rightarrow T \ id$

$T \rightarrow int \mid real$

## 答案:

语法制导定义

产生式	语义规则
$D \rightarrow L, id$	$D.type := L.type$ $addtype(id.entry, L.type)$
$D \rightarrow L$	$D.type := L.type$
$L \rightarrow T \ id$	$L.type := T.type$ $addtype(id.entry, T.type)$
$T \rightarrow int$	$T.type := integer$
$T \rightarrow real$	$T.type := real$

## 问题 7:

下面文法产生的表达式是对整型和实型常数应用算符+形成的。当两个整数相加时,结果为整数,否则为实数。

$$E \rightarrow TR$$

$$R \rightarrow + TR \mid \varepsilon$$

$$T \rightarrow \text{num.num} \mid \text{num}$$

a) 给出语法制导定义确定每个子表达式的类型。

b) 把表达式翻译成前缀形式, 并且决定类型。试用一元运算符 `inttoreal` 把整型值转换为相等的实型值, 以使得前缀表达式中两个运算对象是同类型的。

## 答案:

a) 设 `type` 是综合属性, 代表各非终结符的“类型”属性  
 设 `in` 是继承属性,

翻译方案

产生式	语义规则
$E \rightarrow T$ $R$	$\{R.i := T.type\}$ $\{E.Type := R.s\}$
$R \rightarrow +$ $T$ $R1$	$\{ \text{IF } (R.i = \text{integer}) \text{ and } (T.type = \text{integer}) \text{ THEN}$ $\quad R1.i := \text{integer}$ $\text{ELSE}$ $\quad R1.i := \text{real} \}$ $\{R.s := R1.s\}$
$R \rightarrow \varepsilon$	$\{R.s := R.i\}$
$T \rightarrow \text{num.num}$	$T.type := \text{real}$
$T \rightarrow \text{num}$	$T.type := \text{integer}$

b) 设属性 `s` 和 `i` 用于传递属性 `type`, 属性 `t` 和 `j` 用于传递属性 `val`。



## 翻译方案

[illegible]

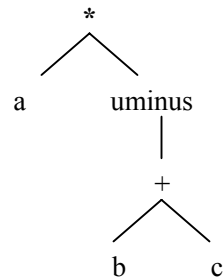
## 问题 8:

翻译算术表达式  $a * -(b+c)$  为

a) 一棵语法树

b) 后缀式

c) 三地址代码



答案:

a) 语法树:

b) 后缀式:

 $a \ b \ c \ + \ uminus \ *$ 

c) 三地址代码:

 $t1 := b + c$  $t2 := - \ t1$  $t3 := a * t2$ 

## 问题 9:

翻译算术表达式  $-(a+b)*(c+d) + (a+b+c)$  为

a) 四元式

b) 三元式

c) 间接三元式

答案:

先写出三地址代码为:

 $t1 := a + b$  $t2 := - \ t1$  $t3 := c + d$  $t4 := t2 * t3$  $t5 := a + b$  $t6 := t5 + c$  $t7 := t4 + t6$ 

a): 对应的四元式为:

	op	arg1	arg2	result
(0)	+	a	b	t1
(1)	uminus	t1		t2
(2)	+	c	d	t3
(3)	*	t2	t3	t4
(4)	+	a	b	t5
(5)	+	t5	c	t6
(6)	+	t4	t6	t7

b): 对应的三元式为:

	op	arg1	arg2
(0)	+	a	b
(1)	Uminus	(0)	
(2)	+	c	d
(3)	*	(1)	(2)
(4)	+	a	b
(5)	+	(4)	c
(6)	+	(3)	(5)

c): 对应的间接三元式为:

	statement
(0)	15
(1)	16
(2)	17
(3)	18
(4)	15
(5)	19
(6)	20

	op	arg1	arg2
15	+	a	b
16	uminus	15	
17	+	c	d
18	*	16	17
19	+	15	c
20	+	18	19

问题 10:

将下列赋值语句译成三地址代码。

$A[i,j] := B[i,j] + C[A[k,l]] + D[i+j]$

答案:

```

t11 := i * 20
t12 := t11 + j
t13 := A - 84;
t14 := 4 * t12
t15 := t13[t14]      //A[i,j]
t21 := i * 20
t22 := t21 + j
t23 := B - 84;
t24 := 4 * t22
t25 := t23[t24]      //B[i,j]
t31 := k * 20
t32 := t31 + l

```

```

t33 := A-84
t34 := 4*t32
t35 := t33[t34]      //A[k,l]
t36 := 4*t35
t37 := C-4
t38 := t37[t36]      //C[A[k,l]]
t41 := i+j
t42 := 4*t41
t43 := D-4
t44 := t43[t42]      //D[i+j]
t1  := t25 + t38
t2  := t1 + t44
t23[t24] := t2

```

**问题 11:**

写出 for 语句的翻译方案

答案:

产生式	动作
$S \rightarrow \text{for } E \text{ do } S1$	$S.begin := \text{newlabel}$ $S.first := \text{newtemp}$ $S.last := \text{newtemp}$ $S.curr := \text{newtemp}$ $S.code := \text{gen}(S.first \text{ ":=" } E.init)$ $\quad \parallel \text{gen}(S.last \text{ ":=" } E.final)$ $\quad \parallel \text{gen}(\text{"if" } S.first \text{ ">" } S.last \text{ "goto" } S.next)$ $\quad \parallel \text{gen}(S.curr \text{ ":=" } S.first)$ $\quad \parallel \text{gen}(S.begin \text{ ":" })$ $\quad \parallel \text{gen}(\text{"if" } S.curr \text{ ">" } S.Last \text{ "goto" } S.next)$ $\quad \parallel S1.code$ $\quad \parallel \text{gen}(S.curr := \text{succ}(S.curr))$ $\quad \parallel \text{gen}(\text{"goto" } S.begin)$
$E \rightarrow v := \text{initial to final}$	$E.init := \text{initial.place}$ $E.final := \text{final.place}$

## 问题 12:

写出说明语句中的名字和类型及相对地址的翻译模式，以允许在形如  $D \rightarrow id : T$  的说明中可用一串名字表来代替单个名字。

## 答案:

产生式	动作
$P \rightarrow D$	{offset := 0}
$D \rightarrow D; D$	
$D \rightarrow id \ L$	{enter(id.name , L.type , offset) offset := offset + L.width}
$L \rightarrow id , L1$	{L.type := L1.type L.width := L1.width enter(id.name , L1.type , offset) offset := offset + L1.width }
$L \rightarrow :T$	{L.type := T.type L.width := T.width}
$T \rightarrow integer$	{T.type := integer T.width := 4}
$T \rightarrow real$	{T.type := real T.width := 8}
$T \rightarrow array \ [num] \ of \ T1$	{T.type:=array(num.val , T1.Type T.width := num.val * T1.Width)}
$T \rightarrow ^T1$	{T.type := pointer(T1.type) T.width := 4}

## 问题 13:

(浙江大学 1997 年) 在一个移入—归约的分析中采用以下的语法制导的翻译模式, 在按一产生式归约时, 立即执行括号中的动作。

$A \rightarrow aB$  {print “0” ;}

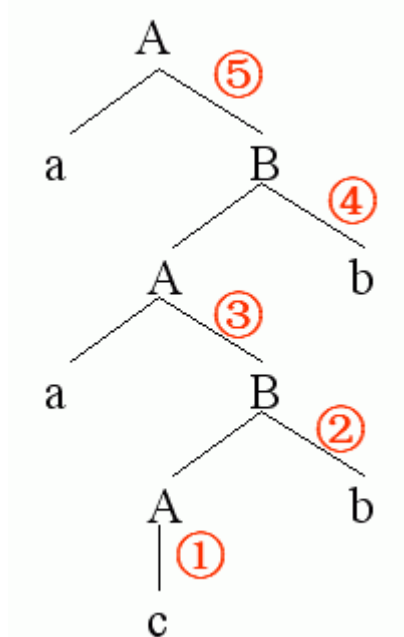
$A \rightarrow c$  {print “1” ;}

$B \rightarrow Ab$  {print “2” }

当分析器的输入为 aacbb 时, 打印的字符串是什么?

答案:

分析器的分析过程如下图所示:



由于分析器采用移入—归约的方式进行分析, 符号串 aacbb 的分析过程将如图中所标的归约顺序进行, 而在按一产生式归约时, 立即执行括号中的动作, 所以分析器打印的字符为 12020。