编译原理第七章作业

2251745 张宇

1. 给出下面表达式的逆波兰表示（后缀式）。

解：

(1) a b neg c + *

(2) A not C D not or not or

(3) a b c d e / + * +

(4) A B and C not D or or

(5) a neg b c neg d + * +

(6) A B or C D not E and or and

(7) a b c d e / + * +

(8) A B and C not D or or

3. 请将表达式 **-(a+b) * (c+d) - (a+b+c)** 分别表示成三元式、间接三元式和四元式序列。

解：三元式：

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

间接三元式：

间接代码

（1）

(2)

(3)

(4)

(1)

(9)

(10)

三元式表

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

四元式：

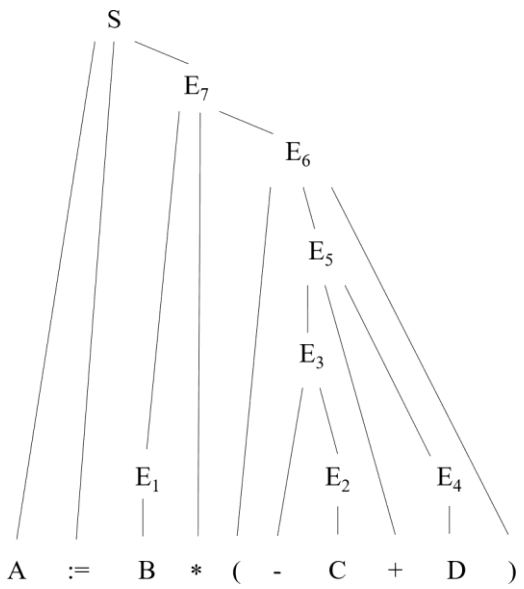
	op	arg1	arg2	result
(0)	+	a	b	T_1
(1)	uminus	T_1		T_2
(2)	+	c	d	T_3
(3)	*	T_2	T_3	T_4
(4)	+	a	b	T_5
(5)	+	T_5	c	T_6
(6)	-	T_4	T_6	T_7

4. 按 **7.3** 节所说的办法，写出下面赋值语句

A:=B*(-C+D)

的自下而上语法制导翻译过程。给出所产生的三地址代码。

解：画出语法分析树如下：



根据语法树，进行如下翻译：

E_1.place = B E_1.code = ‘’
E_2.place = C E_2.code = ‘’
E_3.place = T_1 E_3.code = ‘T_1:=uminus C’
E_4.place = D E_4.code = ‘’
E_5.place = T_2 E_5.code = ‘T_1:=uminus C T_2:=T_1 + D’
E_6.place = T_2 E_6.code = ‘T_1:=uminus C T_2:=T_1 + D’

E_7.place = T_3 E_7.code =‘T_1:=uminus C T_2:=T_1 + D T_3:=B*T_2’
S.code = ‘T_1:=uminus C T_2:=T_1 + D T_3:=B*T_2 A:=T_3’

因此，三地址代码为：

T_1:=uminus C

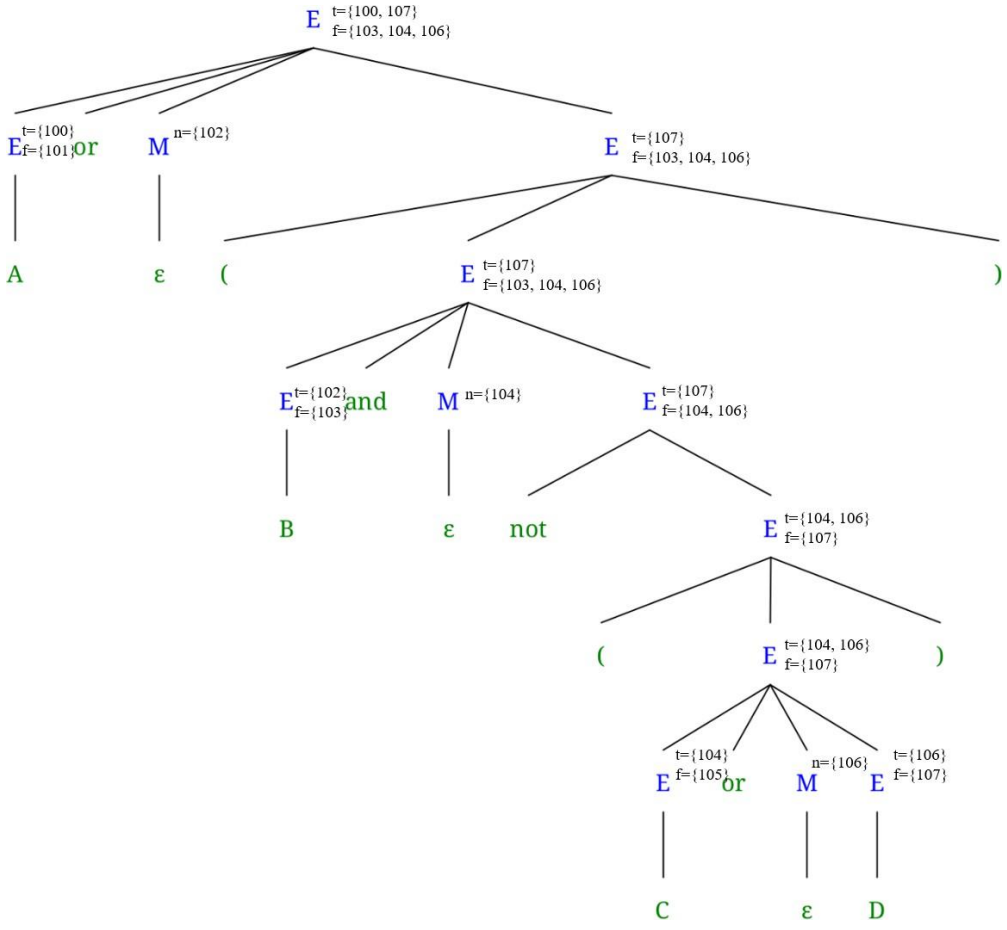
T_2:=T_1 + D

T_3:=B*T_2

A :=T_3

6. 按 **7.4.2** 节的办法，写出布尔表达式 **Aor(B and not (C or D))**的四元式序列。

解：采用一遍扫描的方式，逐渐构建如下的语法树：



因此，布尔表达式 A or (B and not (C or D))的四元式序列为：

100	(jnz, A, -, 0)
101	(j, -, -, 102)
102	(jnz, B, - , 104)
103	(j, -, -, 0)
104	(jnz, C, -, 0)
105	(j, -, -, 106)
106	(jnz, D, -, 0)
107	(j, -, -, 0)

最终规约出的 E 有属性：

E.truelist = {100, 107}

E.falselist = {103, 104, 106}

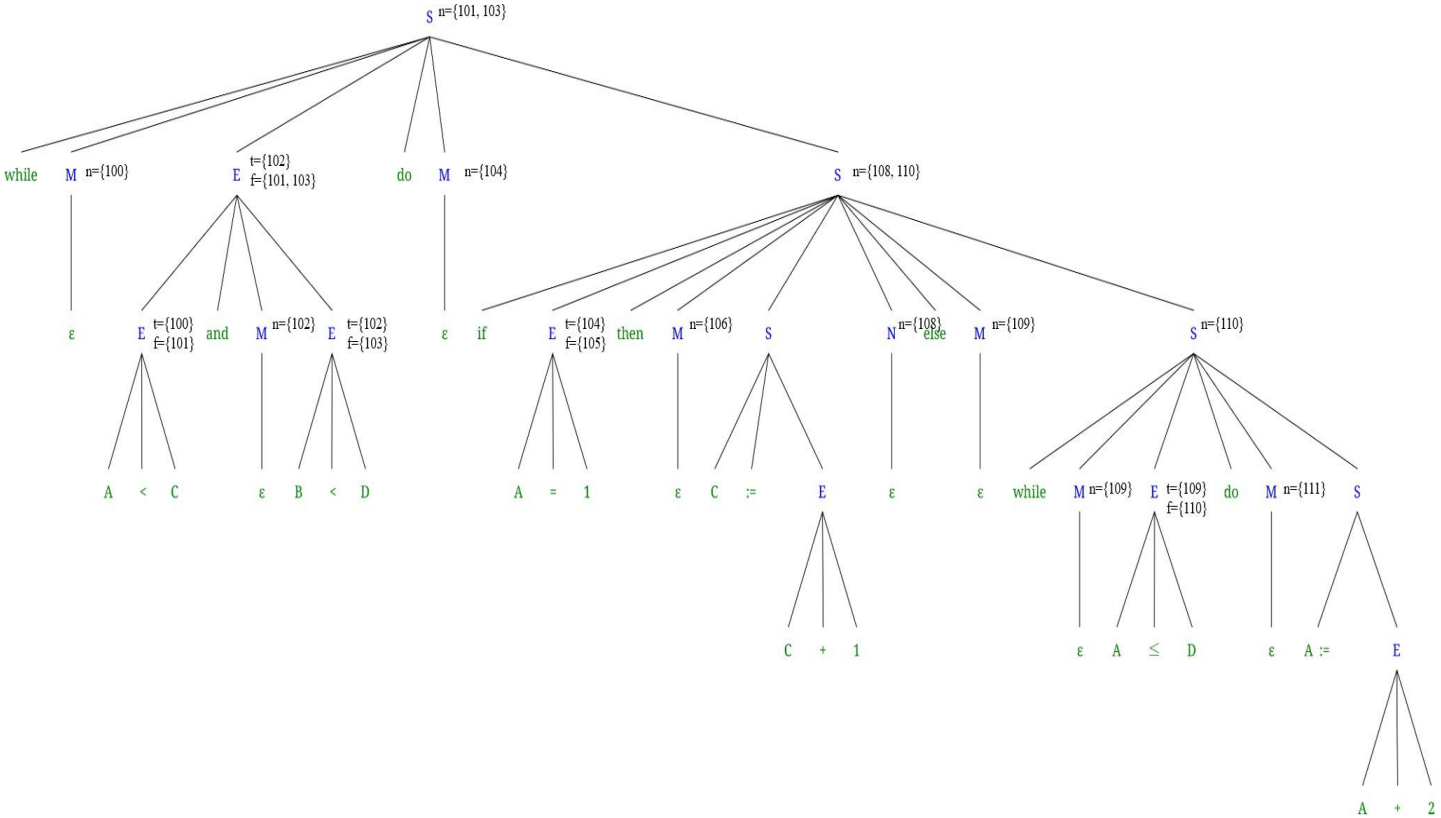
6. 用 **7.5.1** 节的办法，把下面的语句翻译成四元式序列：

while A < C and B < D do

if A = 1 then C:=C+1 else

while A≤D do A:=A+2;

答：采用一遍扫描的方式，逐渐构建如下的语法树：



因此，题目中所描述的 while 语句的四元式序列为：

100 (j<, A, C, 102)
101 (j, -, -, 0)
102 (j<, B, D , 104)
103 (j, -, -, 0)
104 (j=, A, 1, 106)
105 (j, -, -, 109)
106 (+, C, 1, T_1)
107 (:=, T_1, -, C)
108 (j, -, -, 100)
109 (j≤, A, D, 111)
110 (j, -, -, 100)
111 (+, A, 2, T_2)
112 (:=, T_2, -, A)
113 (j, -, -, 109)
114 (j, -, -, 100)

最终规约出的 S 有属性：

S.nextlist = {101, 103}