

1.

1. $ax(-b+c): ab@c+x$ (这里的@代表一目运算符'-')2. $\text{not } A \text{ or not } (C \text{ or not } D): A \text{ not } C D \text{ not or not or}$ 3. $a+bx(c+d/e): abcde/+x+$ 4. $(A \text{ and } B) \text{ or } (\text{not } C \text{ or } D): AB \text{ and } C \text{ not or or}$ 5. $-a+bx(-c+d): a@bc@d+x+$ (这里的@代表一目运算符'-')6. $(A \text{ or } B) \text{ and } (C \text{ or not } D \text{ and } E): AB \text{ or } CD \text{ not } E \text{ and or and}$ 7. $\text{if } (x+y) \times z = 0 \text{ then } (a+b) \uparrow c \text{ else } a \uparrow b \uparrow c:$

我们将整个条件语句看作一个三目运算表达式

转为逆波兰式为 $xy+zxo=sab+c \uparrow =sab \uparrow c \uparrow = \wedge$ (∧代表三目运算符)3. 表达式: $-(a+b) \times (c+d) - (a+b \uparrow c)$

我们按照中缀表达式求值的顺序构建四元式

三元式如下:

op	arg1	arg2	result	op	arg1	arg2
(0) +	a	b	T ₁	(0) +	a	b
(1) -	T ₁	-	T ₂	(1) -	(0)	-
(2) +	c	d	T ₃	(2) +	c	d
(3) X	T ₂	T ₃	T ₄	(3) X	(1)	(2)
(4) +	a	b	T ₅	(4) +	a	b
(5) +	T ₅	c	T ₆	(5) +	(4)	c
(6) -	T ₄	T ₆	T ₇	(6) -	(3)	(5)

间接三元式如下:

间接三元式

三元式序列	三元式序列
(0) op arg1 arg2	(0)
(1) - (0) -	(1)
(2) + c d	(2)
(3) X (1) (2)	(3)
(4) + (0) c	(4)
(5) - (3) (4)	(5)



... $A := B \times (-C + D)$

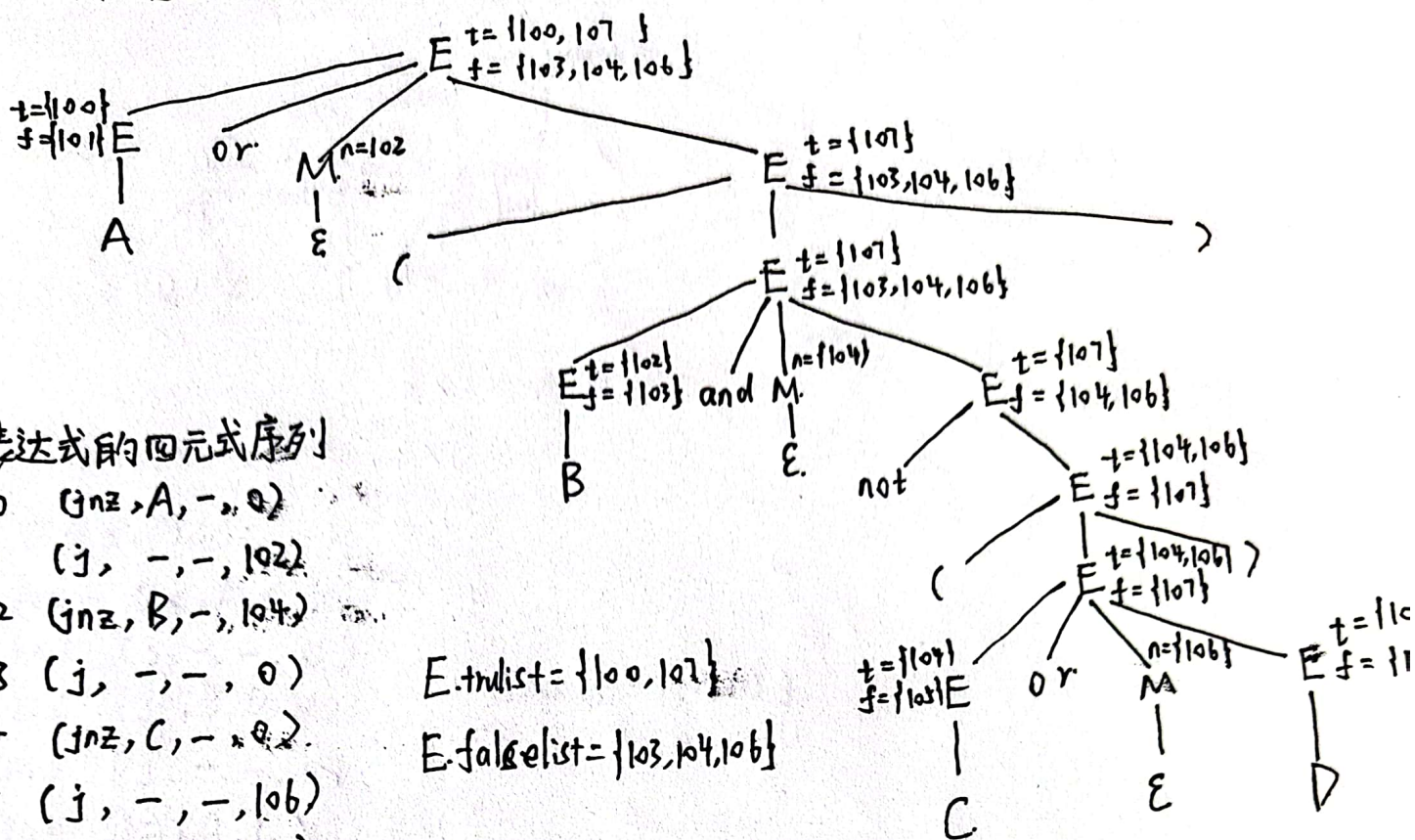
输入	栈	Place	四元式
$A := B \times (-C + D)$			
$:= B \times (-C + D)$	i	A	
$B \times (-C + D)$	i :=	A_	
$\times (-C + D)$	i := i	A_B	
$\times (-C + D)$	i := E	A_B	
$(-C + D)$	i := E x	A_B_	
$-C + D)$	i := E x (A_B_ _	
$C + D)$	i := E x (-	A_B_ _ _	
$+ D)$	i := E x (- i	A_B_ _ _ C	
$+ D)$	i := E x (- E	A_B_ _ _ C	(@, C, -, T ₁)
$+ D)$	i := E x (E	A_B_ _ _ T ₁	
$D)$	i := E x (E +	A_B_ _ _ T ₁ _	
)	i := E x (E + i	A_B_ _ _ T ₁ _ D	
)	i := E x (E + E	A_B_ _ _ T ₁ _ D	(+, T ₁ , D, T ₂)
)	i := E x (E	A_B_ _ _ T ₂	
	i := E x (E)	A_B_ _ _ T ₂ _	
	i := E x E	A_B_ _ T ₂	(x, B, T ₂ , T ₃)
	i := E	A_T ₃	(:=, T ₃ , -, A)
		A	

三地址代码如下:

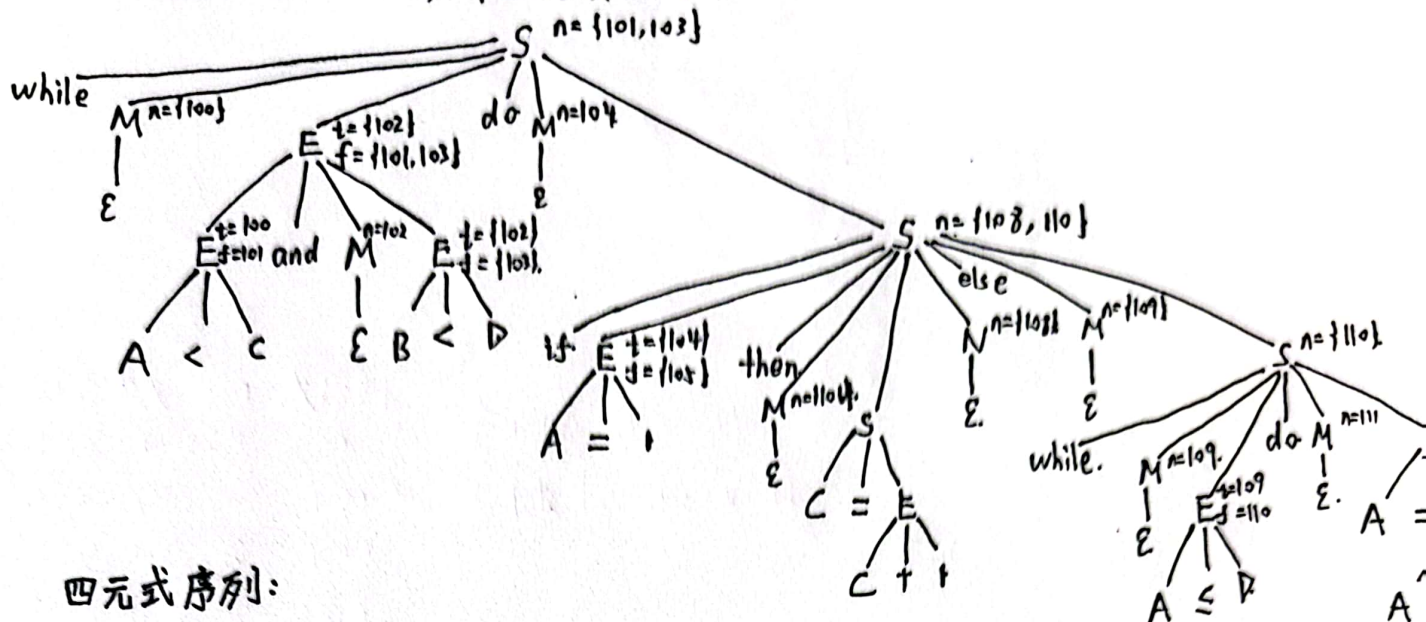
$T_1 := -C$
 $T_2 := T_1 + D$
 $T_3 := B \times T_2$
 $A := T_3$



6. 采用一遍扫描的方式递归构造构建出语法树



7. 利用遍扫描的方法,构建语法树如下:



四元式序列:

- 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, T₁)
- 107 (:=, C, T₁, -)
- 108 (j, -, -, 100)
- 109 (j ≤, A, D, 111)
- 110 (j, -, -, 100)
- 111 (+, A, 2, T₂)
- 112 (:=, A, T₂, -)
- 113 (j, -, -, 109)
- 114 (j, -, -, 100)

S.nextlist = {101, 103}

