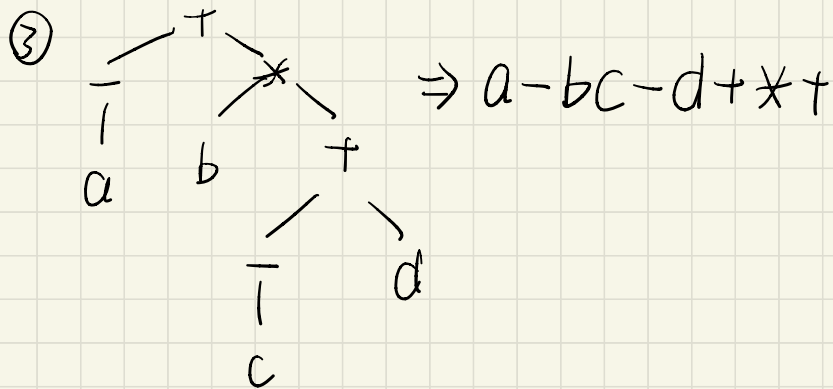
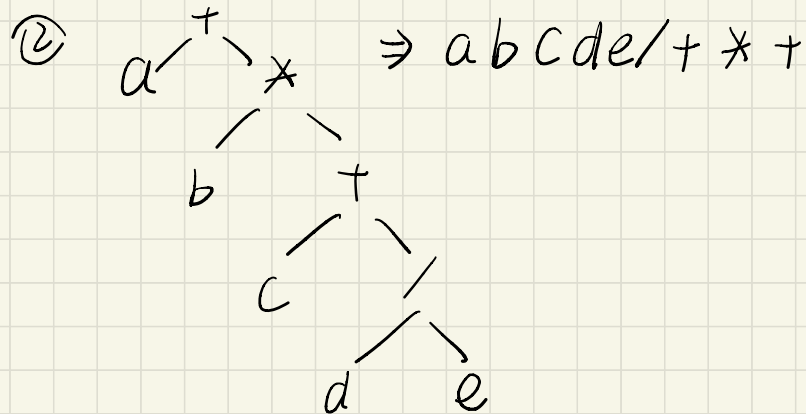
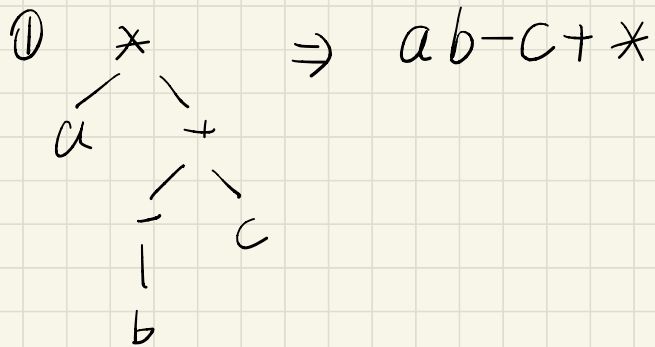
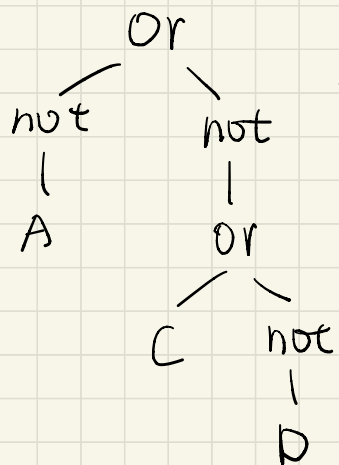


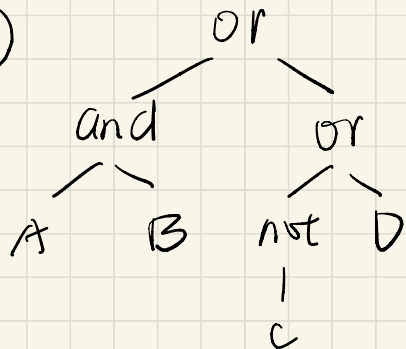
P217.1.



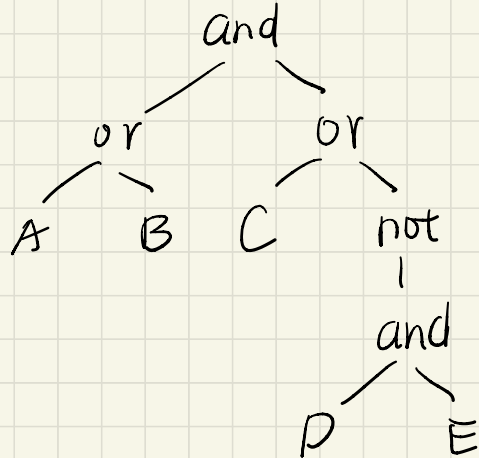
④

 $\Rightarrow A \text{ not } C D \text{ not } \text{or not } \text{or}$

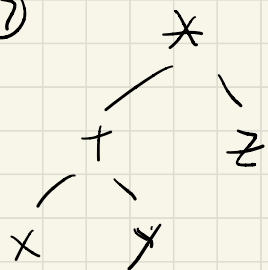
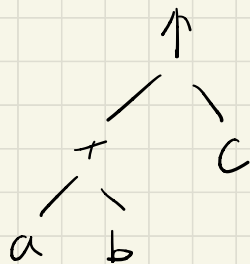
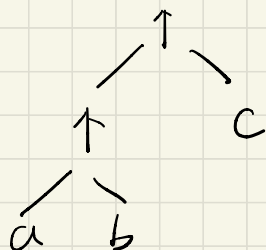
⑤

 $\Rightarrow AB \text{ and } C \text{ not } D \text{ or } \text{or}$

⑥


 $\Rightarrow A B \text{ or } C D E \text{ and not or and}$

⑦


 $\Rightarrow x y + z *$

 $\Rightarrow a b + c \uparrow$

 $\Rightarrow a b \uparrow c \uparrow$

$\therefore \Rightarrow$ if $x \neq y + z$ then $ab \uparrow c \uparrow$ else $ab \uparrow c \uparrow$

问: if, then 可以写为逆波兰吗?

3. (1) 三元式:

- ① $(+, a, b)$
- ② $(+, c, d)$
- ③ $(*, ②, ③)$
- ④ $(-, ③,)$
- ⑤ $(+, a, b)$
- ⑥ $(+, c, ⑤)$
- ⑦ $(-, ④, ⑥)$

(2) 间接三元式: $\textcircled{1} \cong \textcircled{5} \Rightarrow \textcircled{1} \textcircled{2} \textcircled{3} \textcircled{4} \textcircled{1} \textcircled{6} \textcircled{7}$

\therefore 写为 $\textcircled{1} (+, a, b)$

$\textcircled{2} (+, c, d)$

$\textcircled{3} (*, \textcircled{2}, \textcircled{3})$

$\textcircled{4} (-, \textcircled{3},)$

$\textcircled{5} (+, c, \textcircled{1})$

$\textcircled{6} (-, \textcircled{4}, \textcircled{5})$

(3) 四元式: $(+, a, b, R_1)$ R_6 为最终结果.

$(+, c, d, R_2)$

$(*, R_1, R_2, R_3)$

$(-, R_3, , R_4)$

$(+, c, R_1, R_5)$

$(-, R_4, R_5, R_6)$

7.

100 ($<$, A, C, 102)

101 (j, , , 114)

102 ($<$, B, D, 104)

103 (j, , , 114)

104 ($=$, A, 1, 106)

105 (j, , , 109) 跳到else

106 ($+$, C, 1, T_1)

107 ($i=$, T_1 , , C)

108 (j, , , 104)

109 (\leq , A, D, 111)

110 (j, , , 104)

111 ($+$, A, 2, T_2)

112 ($i=$, T_2 , , A)

113 (j, , , 110)

P₂₃₆ · 1.

(1) 符号表是程序中所有标识符和它的声明使用信息的汇总, 是一个数据结构.

(2) 作用 ① 收集符号属性

② 检查上下文语义的合法性的基础.

③ 生成目标代码地址分配的依据.

3. (1) ① 简单方式: 各栏占的存储单元长度一致

② 间接存储: 用一个专用的信息表区存储数组的有关信息, 在符号表的地址栏仅存放入口指针.

(2) 取决于对存储空间利用率的考量: 某些语言要求标识符长度 ≤ 8 , 则用①, 若长度可变, 则用②.