

6.6.1: 在图6-36的语法制导定义中添加处理下列控制流构造的规则:

- 1) 一个repeat语句, repeat S while B
- 2) 一个for循环语句, for (S1 ; B ; S2) S3

产生式	语义规则
$P \rightarrow S$	$S.next = newlabel()$ $P.code = S.code \parallel label(S.next)$
$S \rightarrow assign$	$S.code = assign.code$
$S \rightarrow if (B) S_1$	$B.true = newlabel()$ $B.false = S_1.next = S.next$ $S.code = B.code \parallel label(B.true) \parallel S_1.code$
$S \rightarrow if (B) S_1 else S_2$	$B.true = newlabel()$ $B.false = newlabel()$ $S_1.next = S_2.next = S.next$ $S.code = B.code$ $\parallel label(B.true) \parallel S_1.code$ $\parallel gen('goto' S.next)$ $\parallel label(B.false) \parallel S_2.code$
$S \rightarrow while (B) S_1$	$begin = newlabel()$ $B.true = newlabel()$ $B.false = S.next$ $S_1.next = begin$ $S.code = label(begin) \parallel B.code$ $\parallel label(B.true) \parallel S_1.code$ $\parallel gen('goto' begin)$
$S \rightarrow S_1 S_2$	$S_1.next = newlabel()$ $S_2.next = S.next$ $S.code = S_1.code \parallel label(S_1.next) \parallel S_2.code$

图 6-36 控制流语句的语法制导定义

产生式

语义规则

$S \rightarrow repeat S_1$

 $while B$

$B.true = newlabel()$

 $B.false = S.next$

 $S.code = label(B.true) \parallel$

 $S_1.code \parallel B.code$

产生式

语义规则

$S \rightarrow for (S_1 ; B ; S_2) S_3$

$S_1.next = newlabel()$

 $B.true = newlabel()$

 $B.false = S.next$

 $S_2.next = S_1.next$

 $S.code = S_1.code \parallel label(S_1.next) \parallel$

 $B.code \parallel label(B.true) \parallel$

 $S_3.code \parallel S_2.code \parallel$

 $gen('goto', S_1.next)$

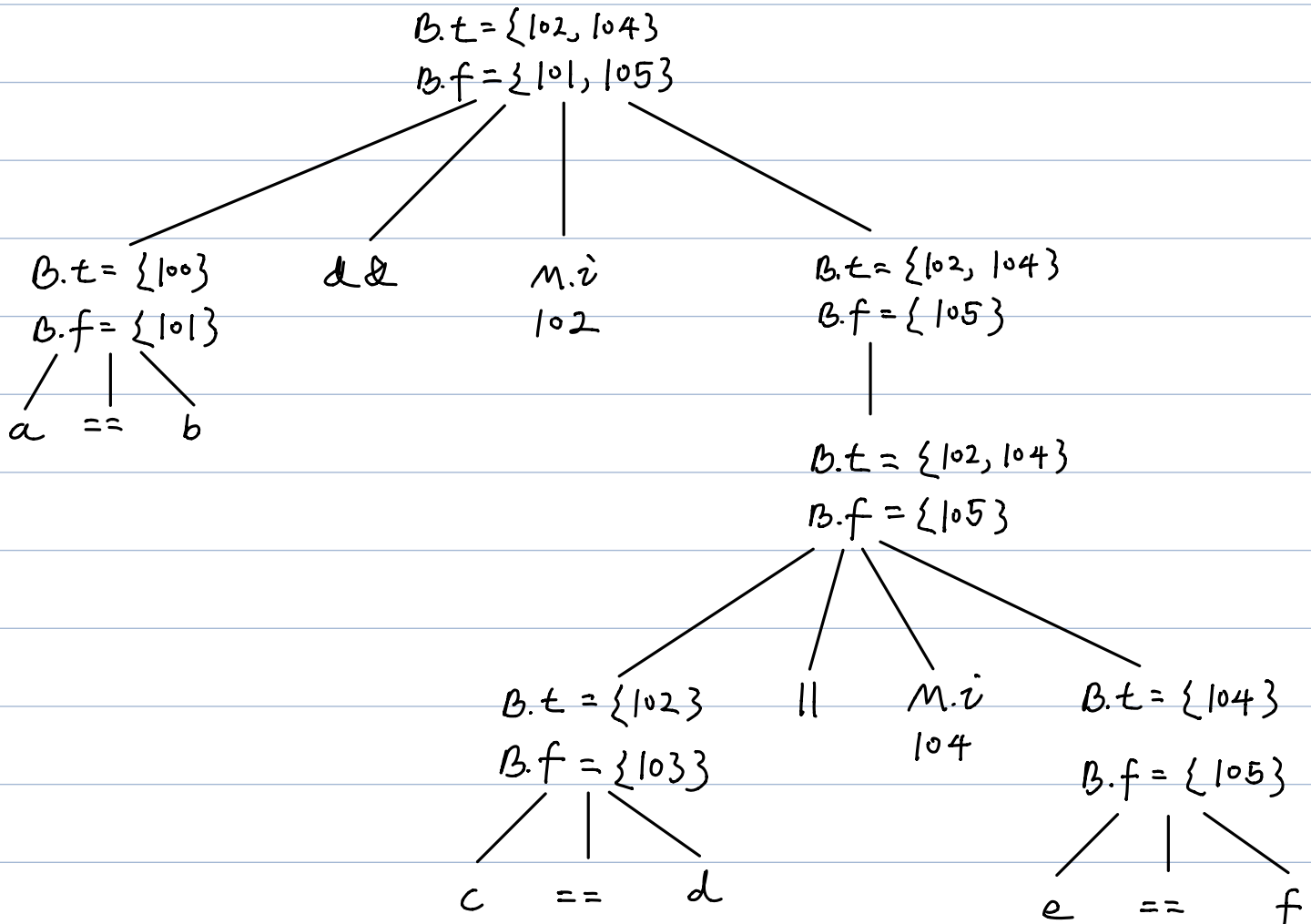
6.7.1: 使用图6-43中的翻译方案翻译下列表达式。给出每个子表达式的truelist和falselist。
你可以假设第一条被生成的指令的地址是100。

- 1) $a == b \ \&\& \ (c == d \ || \ e == f)$
- 2) $(a == b \ || \ c == d) \ || \ e == f$
- 3) $(a == b \ \&\& \ c == d) \ \&\& \ e == f$

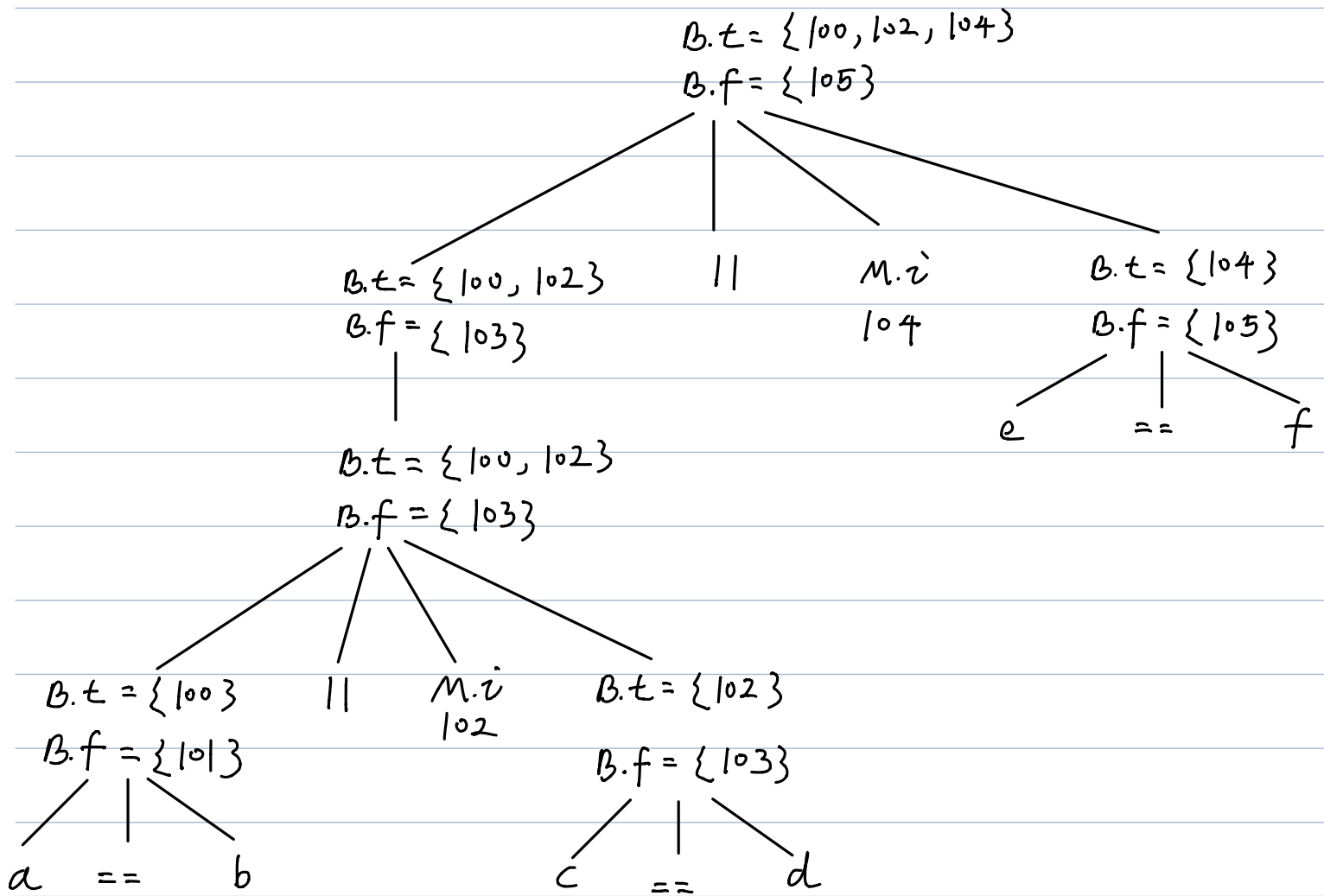
1) $B \rightarrow B_1 \ \ M \ B_2$	{ $backpatch(B_1.falselist, M.instr);$ $B.truelist = merge(B_1.truelist, B_2.truelist);$ $B.falselist = B_2.falselist;$ }
2) $B \rightarrow B_1 \ \&\& \ M \ B_2$	{ $backpatch(B_1.truelist, M.instr);$ $B.truelist = B_2.truelist;$ $B.falselist = merge(B_1.falselist, B_2.falselist);$ }
3) $B \rightarrow ! B_1$	{ $B.truelist = B_1.falselist;$ $B.falselist = B_1.truelist;$ }
4) $B \rightarrow (B_1)$	{ $B.truelist = B_1.truelist;$ $B.falselist = B_1.falselist;$ }
5) $B \rightarrow E_1 \ rel \ E_2$	{ $B.truelist = makelist(nextinstr);$ $B.falselist = makelist(nextinstr + 1);$ $gen('if' E_1.addr \ rel.op \ E_2.addr 'goto -');$ $gen('goto -');$ }
6) $B \rightarrow true$	{ $B.truelist = makelist(nextinstr);$ $gen('goto -');$ }
7) $B \rightarrow false$	{ $B.falselist = makelist(nextinstr);$ $gen('goto -');$ }
8) $M \rightarrow e$	{ $M.instr = nextinstr;$ }

图 6-43 布尔表达式的翻译方案

1) $a == b \ \&\& \ (c == d \ || \ e == f)$



2) $(a == b \parallel c == d) \parallel e == f$



3) $(a==b \ \&\& \ c==d) \ \&\& \ e==f$

