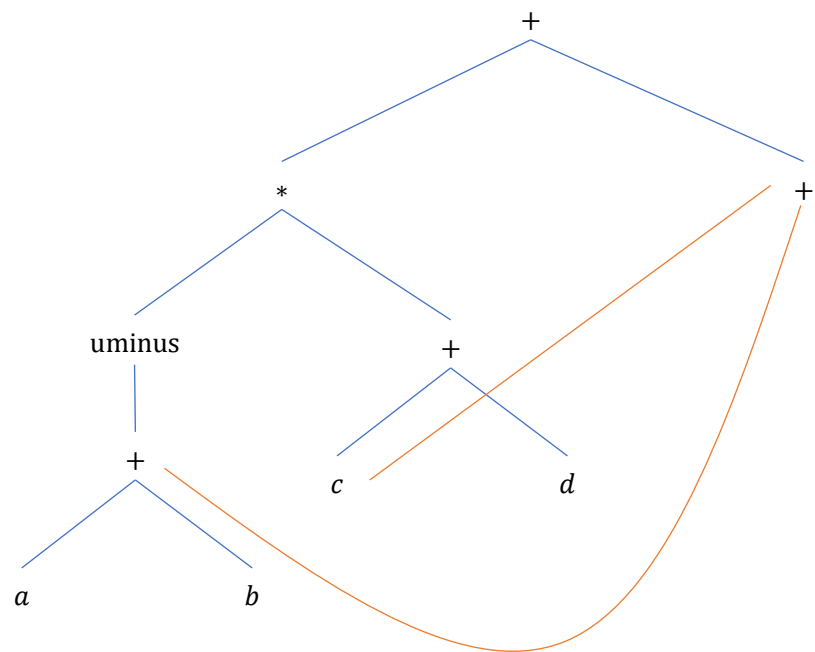


7.1 把算术表达式  $-(a + b) * (c + d) + (a + b + c)$  翻译成：

- (b) 有向无环图 (DAG)
- (d) 三地址代码 (TAC)

解： (b) 将其表现为有向无环图：



(d) 直接写出其三地址代码：

```
t1=a+b
t2=-t1
t3=c+d
t4=t2*t3
t5=t1+c
t6=t4+t5
```

7.2 把下列 C 程序的可执行语句翻译成三地址代码：

```
main() {
    int i;
    int a[10];
    while(i<=10){
        a[i] = 0;
    }
}
```

解： 写出其三地址代码：

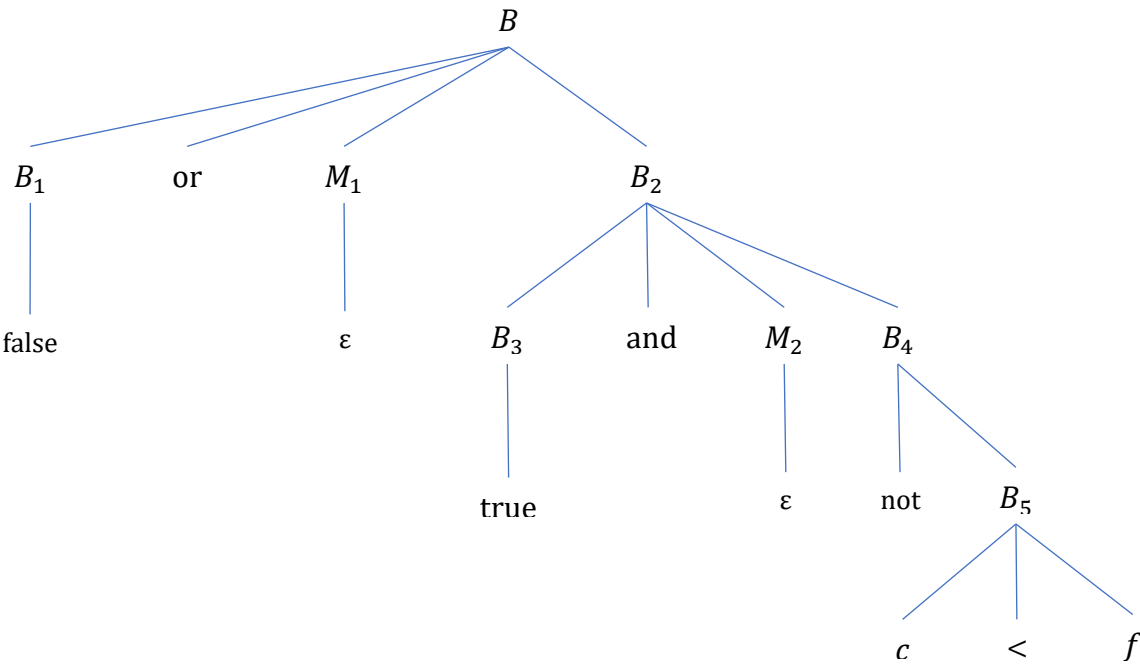
```
1 if i<=10 goto 3
2 goto 5
3 a[i]=0
4 goto 1
5 return
```

□考虑布尔表达式

false or true and not c < f

- ❖ 参考本ppt中布尔表达式短路计算、标号回填等翻译技术，生成对应的三地址代码。
- ❖ 假设nextinstr = 200
- ❖ 除了三地址代码外，画出LR分析方法对应的注释分析树（如slide 100），标注出属性和属性值。
- ❖ 结合LR分析方法指出回填的具体细节
  - 在使用哪一个产生式归约时候进行的回填
  - 用哪一个标号，回填了哪一个不完整的goto指令

解：画出其注释分析树，如下图所示：【参考幻灯片 93~100 页】

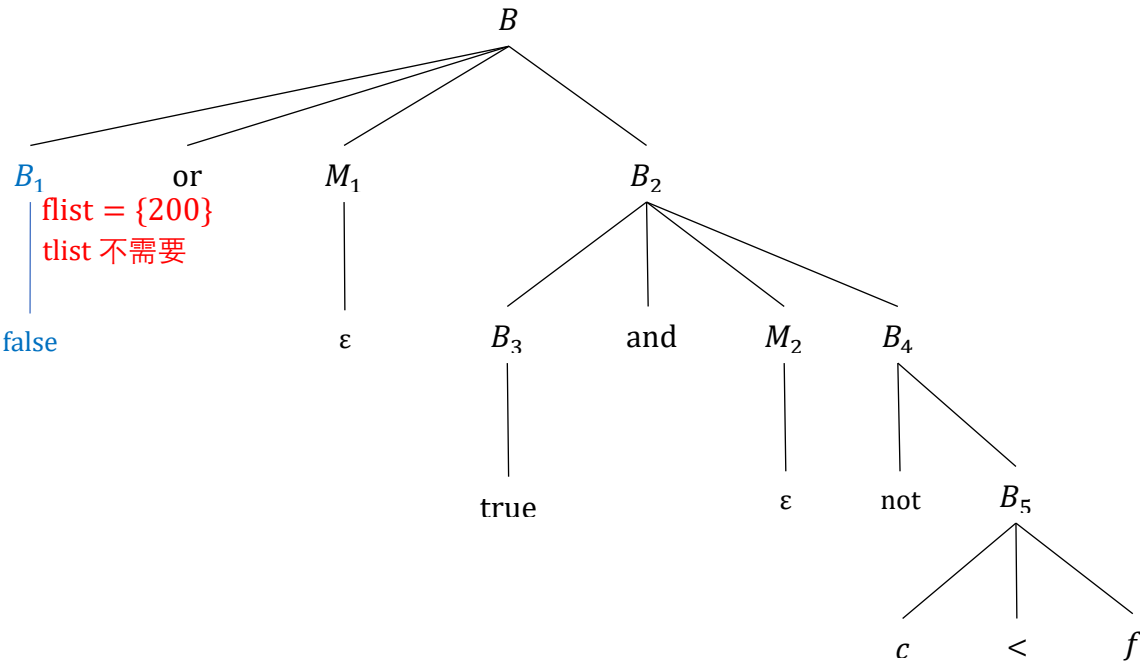


写出其生成的三地址代码，标出哪一些指令需要回填：true、false 其中一个分支不需要回填。

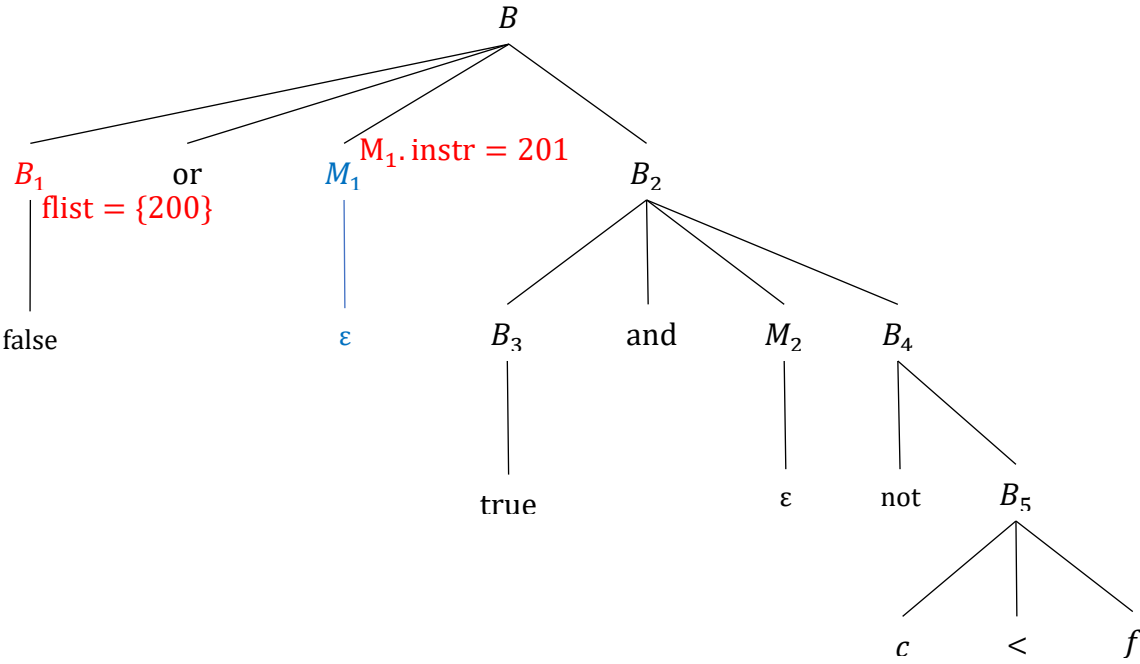
```
200 goto -
201 goto -
202 if c<f goto -
203 goto -
```

按照幻灯片 94 页，先对 B<sub>1</sub> 进行分析：

false or true and not e < f

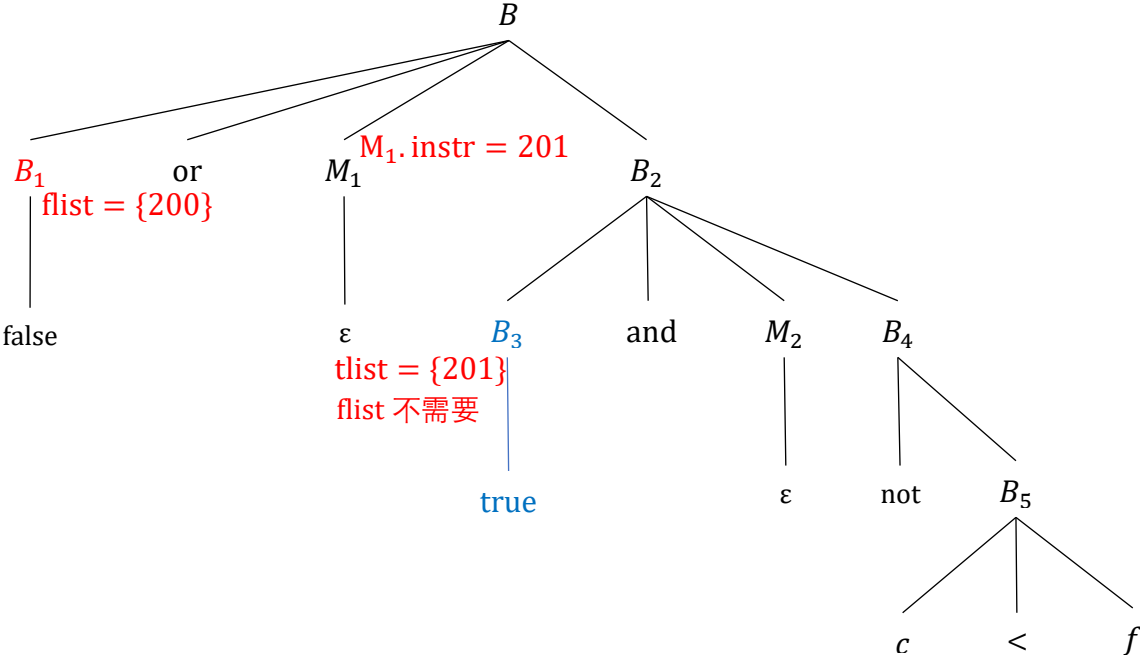


将 B<sub>1</sub> 标记为待回填，接下来分析 M<sub>1</sub>：

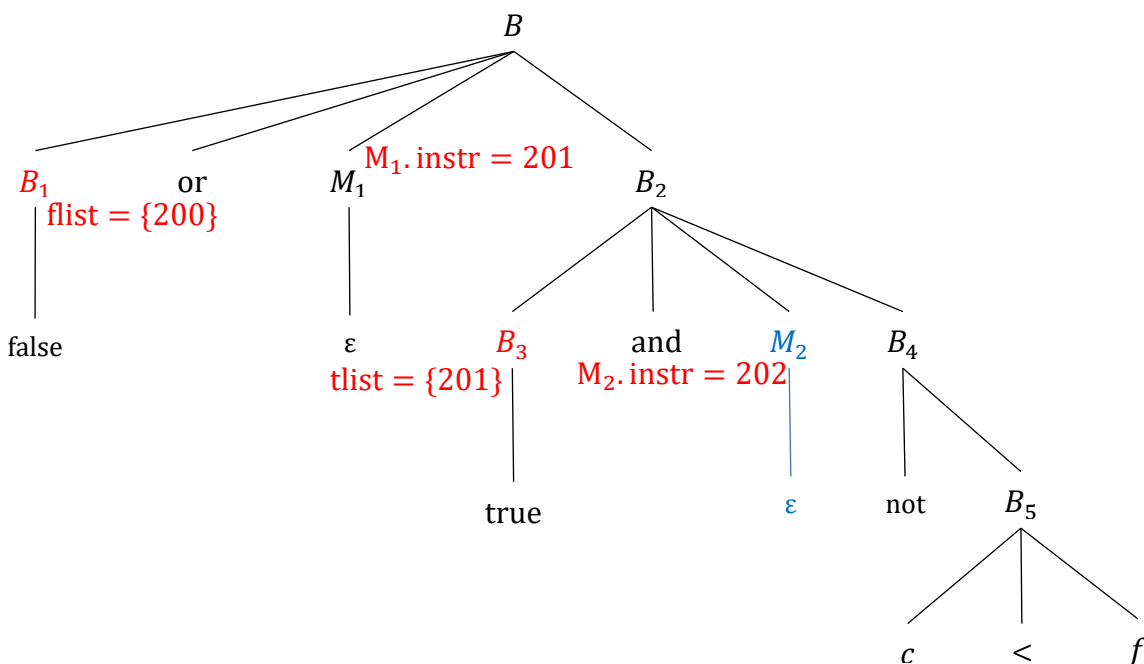


继续向后分析，分析 B<sub>3</sub>：

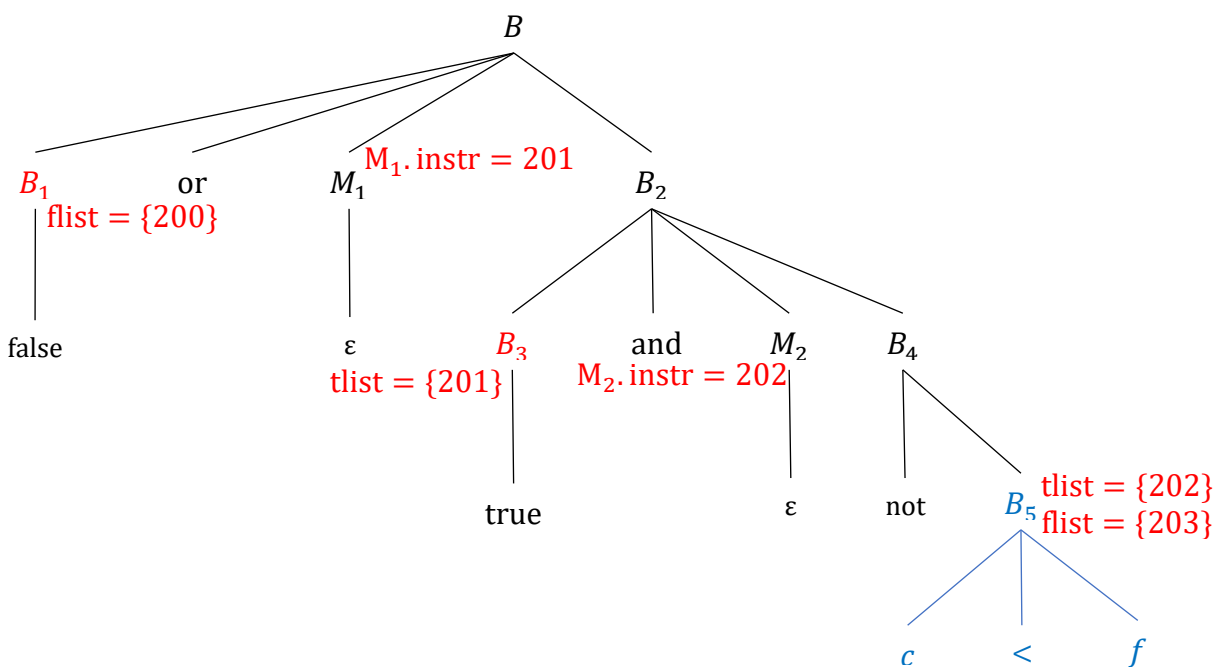
false or true and n



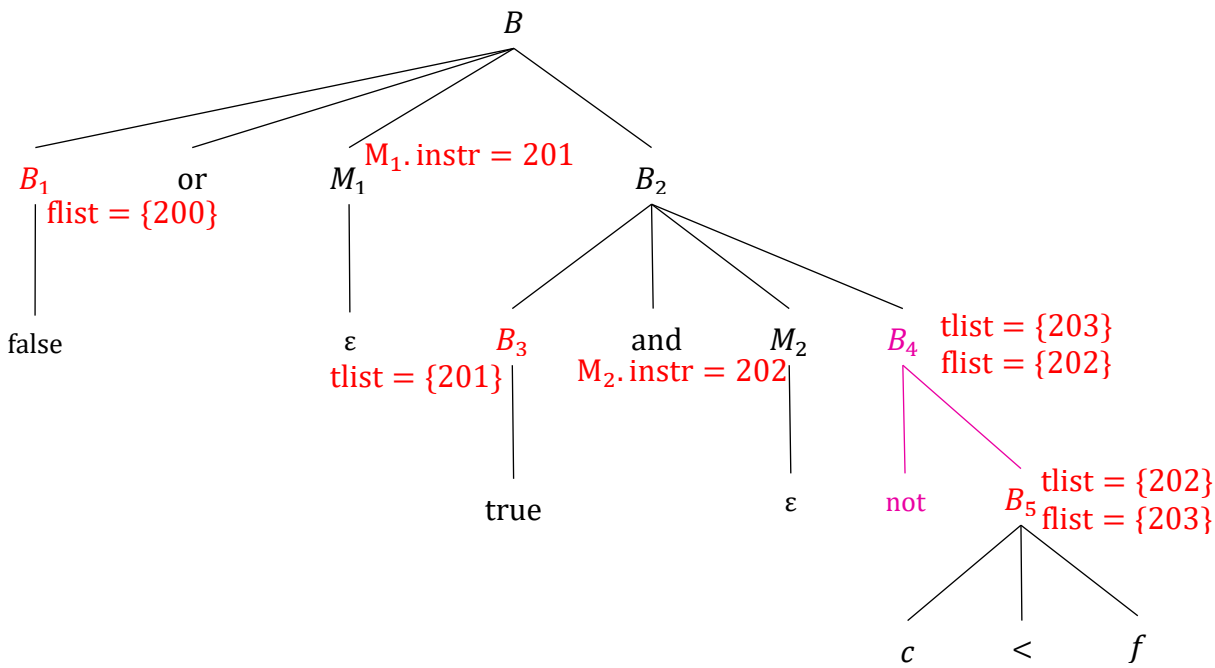
分析  $M_2$ :



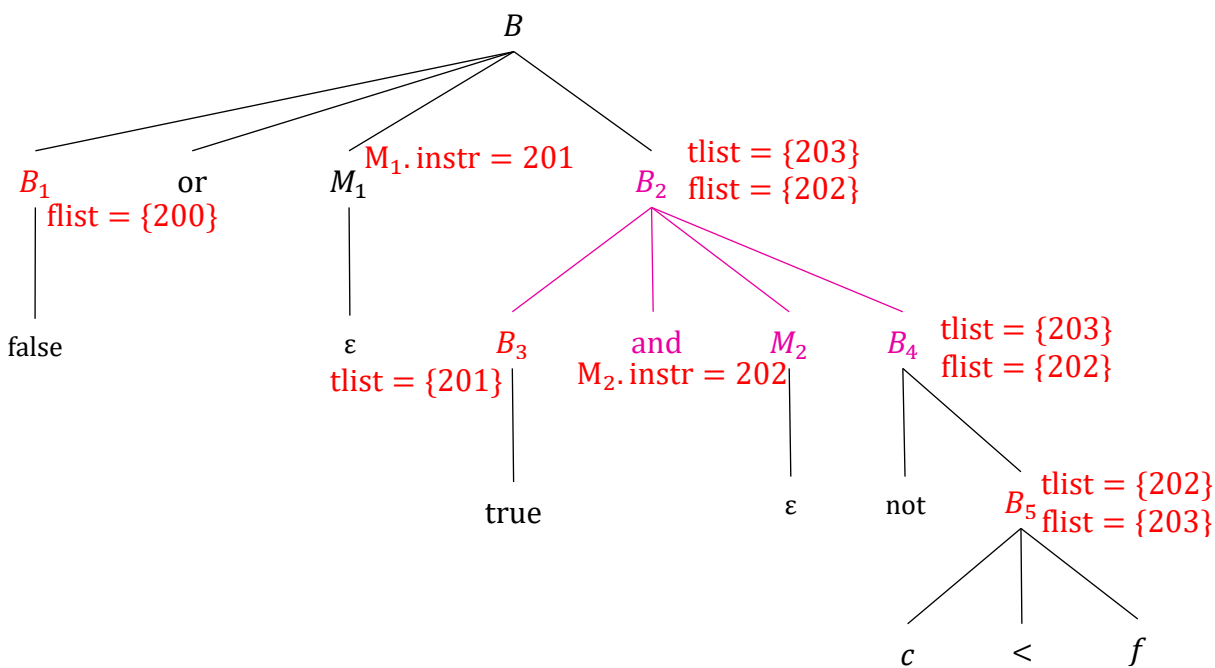
分析  $B_5$ :



分析都已完成，之后进行归约。先对  $B_4$  归约:



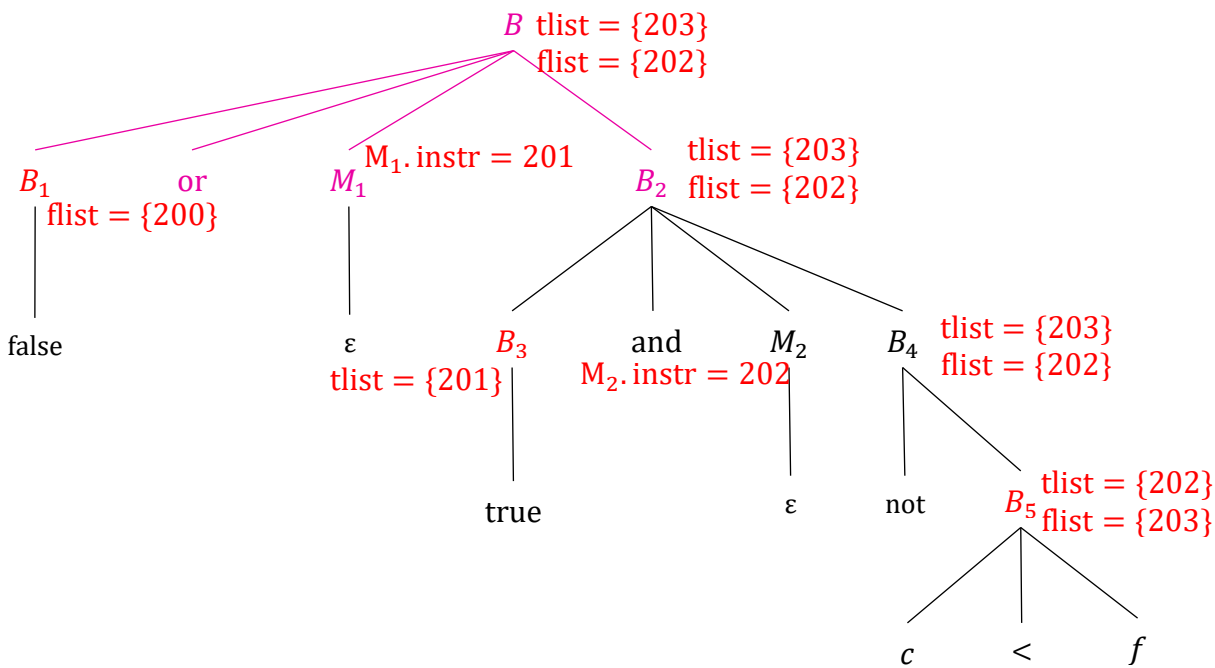
再对上层  $B_2$  归约:



此时  $B_3$  已经可以回填，填入的是  $M_2$  指向的地址 202：

```
200 goto -
201 goto 202
202 if c<f goto -
203 goto -
```

再对  $B$  归约：



至此全部分析回填完毕，形成的三地址码如下所示：

```
200 goto 201
201 goto 202
202 if c<f goto -
203 goto -
```

其他部分的回填要依赖于其他语句的翻译。