第十一次作业参考答案

1. 假定图 6-26 中的函数 widen 可以处理图 6-25a 的层
   次结构中的所有类型，翻译下列表达式。假定 c 和 d 是 char 型，s 和 t 是 short 型，i 和 j 是 int 型，x 是 float 型。
   1. `x = s + c`
   2. `i = s + c`
   3. `x = ( s + c ) * ( t + d )`

1.

```
x = s + c
t1 = (int) s
t2 = (int) c
t3 = t1 + t2
x = (float) t3
```

2.

```
i = s + c
t1 = (int) s
t2 = (int) c
i = t1 + t2
```

3.

```
t1 = (int) s
t2 = (int) c
t3 = t1 + t2
t4 = (int) t
t5 = (int) d
t6 = t4 + t5
t7 = t3 * t6
x = (float) t7
```

2. 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. $S \rightarrow \textbf{repeat}\ S_1\ \textbf{until}\ B$，当$B$为真时结束循环
   2. $S \rightarrow \textbf{for}(S_1; B; S_2)S_3$

1.

```
S1.next = newlabel()
B.true = S.next
B.false = newlabel()
S.code = label(B.false) || S1.code
    || label(S1.next) || B.code
```

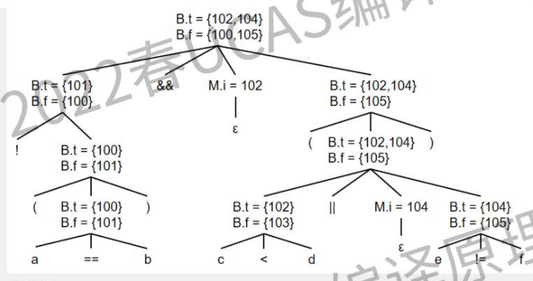
2.

```
S1.next = newlabel()
B.true = newlabel()
B.false = S.next
S2.next = S1.next
S3.next = newlabel()
S.code = S1.code || lable(S1.next) || B.code
    || lable(B.true) || S3.code || label(S3.next)
    || S2.code || gen('goto', S1.next)
```

2. 使用图 6-43 中的翻译方案翻译下列表达式

```
!(a == b) && (c < d || e != f)
```

给出分析过程，给出带 truelist 和 falselist 的注释语法分析树。假设第一条被生成的指令的地址是100。



分析过程：
对 `a == b` 按照$B \rightarrow E_1\ \textbf{rel}\ E_2$的语义动作规约，产生指令：

```
100: if a == b goto _
101: goto _
```

并且$M.i = 102$记录next instr值

用产生式$B \rightarrow (B1)$进行规约，拷贝 truelist 和 falselist，所以有$B.t = \{100\}, B.f = \{101\}$

用产生式$B \rightarrow !B1$进行规约，翻转 truelist 和 falselist，所以有$B.t = \{101\}, B.f = \{100\}$

对于 `c < d` 按照$B \rightarrow E_1\ \textbf{rel}\ E_2$的语义动作规约，产生指令：

```
102: if c < d goto _
103: goto _
```

并且$M.i = 104$

对于 `e != f` 按照$B \rightarrow E_1\ \textbf{rel}\ E_2$的语义动作规约，产生指令：

```
104: if e != f goto _
105: goto _
```

使用$B \rightarrow B_1 || MB_2$规约，调用$backpatch(B_1.f, M.i)$，其中$B_1.f = \{103\}, M.i = 104$，将104填写到103指令，得到：

```
100: if a == b goto _
101: goto _
102: if c < d goto _
103: goto 104
104: if e != f goto _
105: goto _
```

接着执行$B.t = merge(B_1.t, B_2.t)$，所以$B.t = \{102, 104\}$；$B.f = B_2.f$，所以$B.f = \{105\}$。

用产生式$B \rightarrow (B_1)$进行规约，拷贝 truelist 和 falselist，所以有$B.t = \{102,104\}, B.f = \{105\}$

使用$B \rightarrow B_1 \&\& MB_2$规约，调用$backpatch(B_1.t, M.i)$，其中$B_1.t = \{101\}, M.i = 102$，将 102 填写到 101 指令，得到：

```
100: if a == b goto _
101: goto 102
102: if c < d goto _
103: goto 104
104: if e != f goto _
105: goto _
```

接着执行$B.t = B_2.t$，所以$B.t = \{102, 104\}$，

接着执行$B.f = merge(B_1.f, B_2.f)$，所以$B.f = \{100, 105\}$
最终生成的指令如下：

```
100: if a == b goto _
101: goto 102
102: if c < d goto _
103: goto 104
104: if e != f goto _
105: goto _
```

整个表达式为真当且仅当控制流到达102, 104的goto，
整个表达式为假当且仅当控制流到达100, 105的goto。
在后续的编译过程中，已知或同分别应该做什么的时候，这些指令的目标将会被填写完整。

4. 假设下面的 C 语言表达式合法且不存在隐式类型转换，推导各个符号可能类型的最一般情况

```
(p(L) ? init : f(init, L[0])) + 3
```

提示：C 语言中数组的下标运算实际上存在从数组到指针的隐式转换，参考这里。

本题考察类型合一，来自这个课件最后的例子，我们把它改写成 C 式的语法。

符号	类型
p	bool(T*)
L	T*
init	int
f	int(int, T)