

第一题:

6. 按 7.4.2 节的办法, 写出布尔式 $A \text{ or } (B \text{ and not } (C \text{ or } D))$ 的四元式序列。

解:

我们对上面的布尔表达式: $A \text{ or } (B \text{ and not } (C \text{ or } D))$

有:

```
L0:  if A goto L.true
      goto L1;
L1:  if B goto L2
      goto L.false;
L2:  if C goto L.false
      goto L3;
L3:  if D goto L.false
      goto L.true;
```

之后我们可以对上面进行逐个的翻译:

对于

```
L0:  if A goto L.true
      goto L1;
```

可以翻译为:

```
100. (jnz, A, -, 0)
101. (j, -, -, 0)
```

对于

```
L1:  if B goto L2
      goto L.false;
```

可以翻译为:

```
102. (jnz, B, -, 0)
103. (j, -, -, 0)
```

对于

```
L2:  if C goto L.false
      goto L3;
```

可以翻译为:

```
104. (jnz, C, -, 0)
105. (j, -, -, 0)
```

对于

```
L3:  if D goto L.false
      goto L.true;
```

可以翻译为：

106. (jnz, D, -, 0)

107. (j, -, -, 0)

之后我们需要修改上面，将其连接起来：

100. (jnz, A, -, 0)

101. (j, -, -, 102)

102. (jnz, B, -, 104)

103. (j, -, -, 0)

104. (jnz, C, -, 0)

105. (j, -, -, 106)

106. (jnz, D, -, 0)

107. (j, -, -, 0)

然后我们注意到其中还有几个四元式的第四个地址还没有填起来，其中 100 和 107 的导向是 L.true，而 103, 104, 106 是导向 L.false。

我们需要建立对应的真链和假链，真链 {107, 100}，假链 {106, 104, 103}。

其中真链链首是 107，假链链首是 106，然后再次填上面的四地址式。

100. (jnz, A, -, 0)

101. (j, -, -, 102)

102. (jnz, B, -, 104)

103. (j, -, -, 0)

104. (jnz, C, -, 103)

105. (j, -, -, 106)

106. (jnz, D, -, 104)

107. (j, -, -, 100)

答：翻译为四元式为：

100. (jnz, A, -, 0)

101. (j, -, -, 102)

102. (jnz, B, -, 104)

103. (j, -, -, 0)

104. (jnz, C, -, 103)

105. (j, -, -, 106)

106. (jnz, D, -, 104)

107. (j, -, -, 100)

其中真链：{107, 100}， 假链：{106, 104, 103}。

第二题：

7. 用 7.5.1 节的办法，把下面的语句翻译成四元式序列：

```
while A < C and B < D do
  if A = 1 then C := C + 1 else
    while A ≤ D do A := A + 2;
```

解:

我们可以将上面的语句翻译成:

```
L0:  if A<C goto L1
      goto Lnext;
L1:  if B<D goto L2
      goto Lnext;
L2:  if A=1 goto L3
      goto L4;
L3:  T1:=C+1;
      C:=T1;
      goto L0;
L4:  if A<=D goto L5
      goto L0;
L5:  T2:=A+2;
      A:=T2;
      goto L4;
```

然后我们根据上面可以生成四元式:

```
L0:  if A<C goto L1
      goto Lnext;
```

翻译成:

```
100. (j<, A, C, 0)
101. (j, -, -, 0)
```

```
L1:  if B<D goto L2
      goto Lnext;
```

翻译成:

```
102. (j<, B, D, 0)
103. (j, -, -, 0)
```

```
L2:  if A=1 goto L3
      goto L4;
```

翻译成:

```
104. (j=, A, '1', 0)
105. (j, -, -, 0)
```

```
L3:  T1:=C+1;
      C:=T1;
      goto L0;
```

翻译成:

```
106. (+, C, '1', T1)
107. (: =, T1, -, C)
108. (j, -, -, 0)
```

```
L4:  if A<=D goto L5
      goto L0;
```

翻译成:

```
109. (j<=, A, D, 0)
110. (j, -, -, 0)
```

```
L5:  T2:=A+2;
      A:=T2;
      goto L4;
```

翻译成:

```
111. (+, A, '2', T2)
112. (:=, T2, -, A)
113. (j, -, -, 0)
```

综上，我们可以有四元式：

```
100. (j<, A, C, 0)
101. (j, -, -, 0)
102. (j<, B, D, 0)
103. (j, -, -, 0)
104. (j=, A, '1', 0)
105. (j, -, -, 0)
106. (+, C, '1', T1)
107. (:=, T1, -, C)
108. (j, -, -, 0)
109. (j<=, A, D, 0)
110. (j, -, -, 0)
111. (+, A, '2', T2)
112. (:=, T2, -, A)
113. (j, -, -, 0)
114. (j, -, -, 0)
```

在最后我们最好再加一个(j, -, -, 0)跳到最开始的位置

然后我们需要将上面的连接起来，其中跳到 Lnext 的只有 101 和 103 两个语句：

```
100. (j<, A, C, 102)
101. (j, -, -, 0)
102. (j<, B, D, 104)
103. (j, -, -, 101)
104. (j=, A, '1', 106)
105. (j, -, -, 109)
106. (+, C, '1', T1)
107. (:=, T1, -, C)
108. (j, -, -, 100)
109. (j<=, A, D, 111)
110. (j, -, -, 100)
```

111. (+, A, '2', T2)

112. (:=, T2, -, A)

113. (j, -, -, 109)

114. (j, -, -, 100)

答：最终结果是：

100. (j<, A, C, 102)

101. (j, -, -, 0)

102. (j<, B, D, 104)

103. (j, -, -, 101)

104. (j=, A, '1', 106)

105. (j, -, -, 109)

106. (+, C, '1', T1)

107. (:=, T1, -, C)

108. (j, -, -, 100)

109. (j<=, A, D, 111)

110. (j, -, -, 100)

111. (+, A, '2', T2)

112. (:=, T2, -, A)

113. (j, -, -, 109)

114. (j, -, -, 100)