



2024 编译原理理论

第 2 次作业

本次作业一共有 2 大题, 每道大题都有若干道小题, 部分小题都会给出一个具体的参考例子, 请你确保你的回答和题目提供的例子的样式保持一致, 否则可能会没有分数。

(1) 数组翻译 (5 分)

(a) 已知数组 A 的元素按行优先 (Row Major) 方式存放在起始于 base 的一片连续单元中, 且 A 的类型表达式 (Type Expression) 为 $\text{array}(3, \text{array}(7, \text{array}(8, \text{float})))$, 每个 float 类型的数据占 4 个字节 (Byte, 则 $\text{address}(A[i][j][k]) = \text{base} + \underline{\hspace{2cm}}$ (将 address 补充完整)。(1 分)

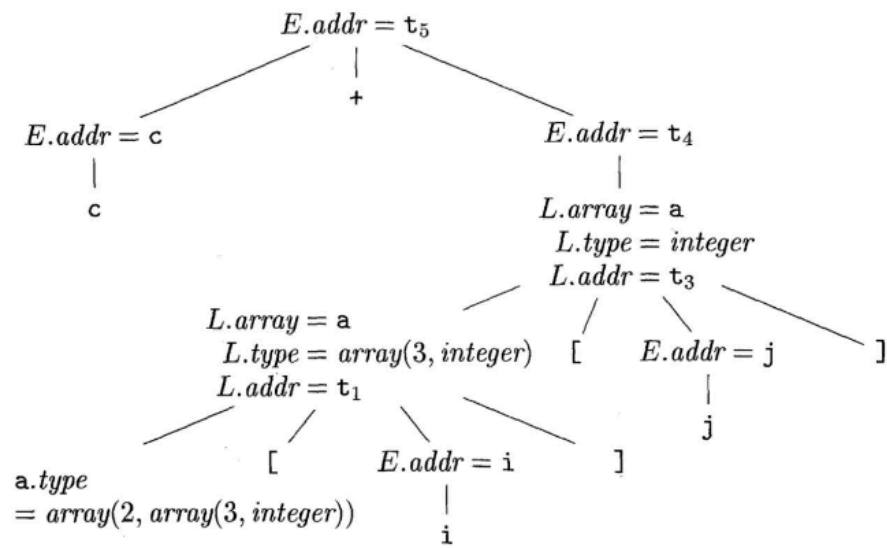
Answer:

$$\begin{aligned}\text{address}(A[i][j][k]) &= \text{base} + i * 7 * 8 * 4 + j * 8 * 4 + k * 4 \\ &= \text{base} + 224 i + 32 j + 4 k\end{aligned}$$

(b) 已知维度为 2×3 的二维整型数组 a 的类型表达式为 $\text{array}(2, \text{array}(3, \text{integer}))$, 图 1 是数组引用的语义动作, 根据该语义动作能够生成 $c + a[i][j]$ 的注释解析树 (Annotated Parse Tree), 如图 2(a) 所示, 请给出 $c + A[i][j][k]$ 的注释解析树 (A 的类型请参考第一小题)。(3 分)

$$\begin{aligned}
S &\rightarrow \mathbf{id} = E ; \quad \{ \text{gen}(\text{top.get}(\mathbf{id.lexeme}) \text{'=' } E.\text{addr}); \} \\
&| \quad L = E ; \quad \{ \text{gen}(L.\text{addr.base} \text{'[' } L.\text{addr} \text{'}]' \text{'=' } E.\text{addr}); \} \\
E &\rightarrow E_1 + E_2 \quad \{ E.\text{addr} = \mathbf{new Temp}(); \\
&\quad \text{gen}(E.\text{addr} \text{'=' } E_1.\text{addr} \text{'+' } E_2.\text{addr}); \} \\
&| \quad \mathbf{id} \quad \{ E.\text{addr} = \text{top.get}(\mathbf{id.lexeme}); \} \\
&| \quad L \quad \{ E.\text{addr} = \mathbf{new Temp}(); \\
&\quad \text{gen}(E.\text{addr} \text{'=' } L.\text{array.base} \text{'[' } L.\text{addr} \text{'}]'); \} \\
L &\rightarrow \mathbf{id} [E] \quad \{ L.\text{array} = \text{top.get}(\mathbf{id.lexeme}); \\
&\quad L.\text{type} = L.\text{array.type.elem}; \\
&\quad L.\text{addr} = \mathbf{new Temp}(); \\
&\quad \text{gen}(L.\text{addr} \text{'=' } E.\text{addr} \text{'*' } L.\text{type.width}); \} \\
&| \quad L_1 [E] \quad \{ L.\text{array} = L_1.\text{array}; \\
&\quad L.\text{type} = L_1.\text{type.elem}; \\
&\quad t = \mathbf{new Temp}(); \\
&\quad L.\text{addr} = \mathbf{new Temp}(); \\
&\quad \text{gen}(t \text{'=' } E.\text{addr} \text{'*' } L.\text{type.width}); \\
&\quad \text{gen}(L.\text{addr} \text{'=' } L_1.\text{addr} \text{'+' } t); \}
\end{aligned}$$

图 1: 数组引用的语义动作



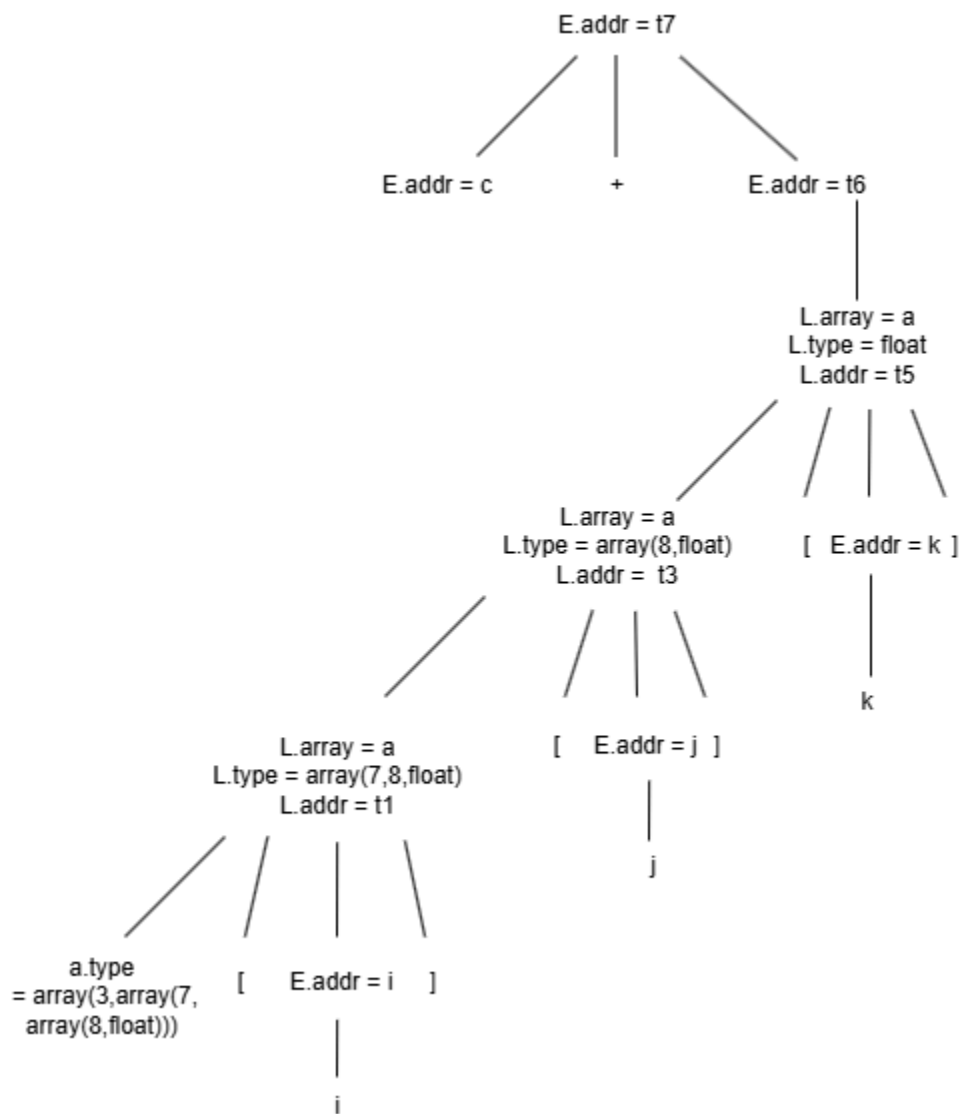
(a) 注释语法树

$t_1 = i * 12$
 $t_2 = j * 4$
 $t_3 = t_1 + t_2$
 $t_4 = a[t_3]$
 $t_5 = c + t_4$

(b) 三址码

图 2: $c + a[i][j]$ 的注释解析树与三址码

Answer:



(c) 已知每个 int 类型的数据占 4 个字节，根据图 1 的语义动作可以写出 $c + a[i][j]$ 的三址码，如图 2 (b) 所示，请你写出 $c + A[i][j][k]$ 的三址码（A 的类型请参考第一小题）。（1 分）

Answer:

$$t_1 = i * 224$$

$$t_2 = j * 32$$

$$t_3 = t_1 + t_2$$

$$t_4 = k * 4$$

$$t_5 = t_4 + t_3$$

$$t_6 = a[t_5]$$

$$t_7 = c + t_6$$

(2) 跳转回填 (5 分)

(a) 图 3 为布尔表达式的翻译方案, 根据该翻译方案可以得到布尔表达式 $x < 100 \parallel x > 200 \&\& x! = y$ 的注释语法树, 如图 4(a) 所示, 图中的 t , f 和 i 分别是 truelist, falselist 和 instr 的缩写, 请你给出布尔表达式 $a == b \&\& (c == d \parallel e == f)$ 的注释语法树。(3 分)

- 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) $B \rightarrow E_1 \ rel \ E_2$ { $B.truelist = makelist(nextinstr);$
 $B.falselist = makelist(nextinstr + 1);$
 $emit('if' \ E_1.addr \ rel.op \ E_2.addr \ 'goto \ -');$
 $emit('goto \ -');$ }
- 6) $B \rightarrow true$ { $B.truelist = makelist(nextinstr);$
 $emit('goto \ -');$ }
- 7) $B \rightarrow false$ { $B.falselist = makelist(nextinstr);$
 $emit('goto \ -');$ }
- 8) $M \rightarrow \epsilon$ { $M.instr = nextinstr; \}$

图 3: 布尔表达式的翻译方案

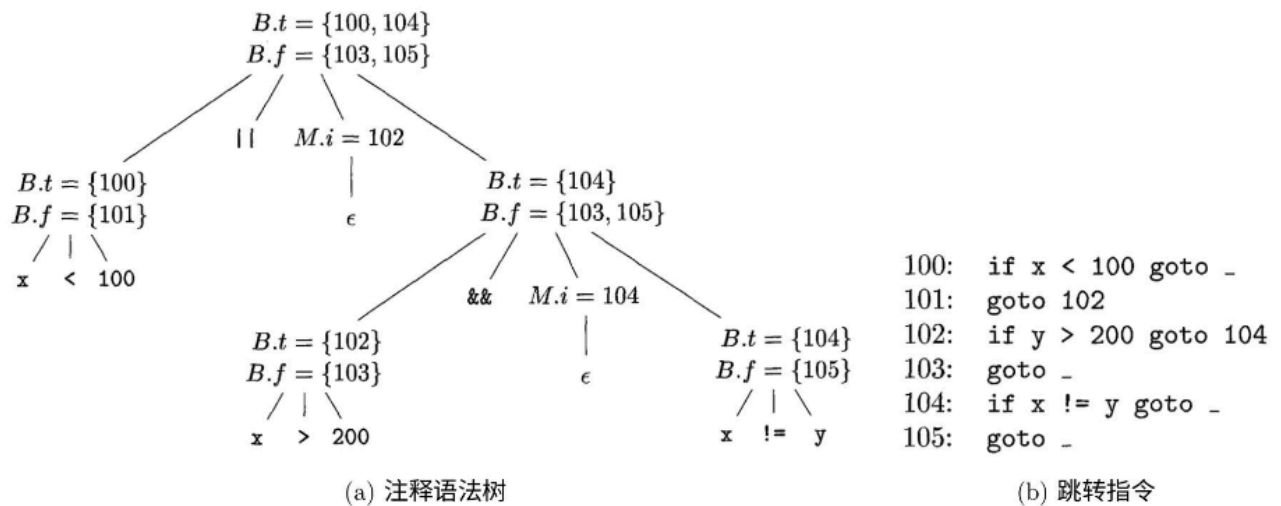
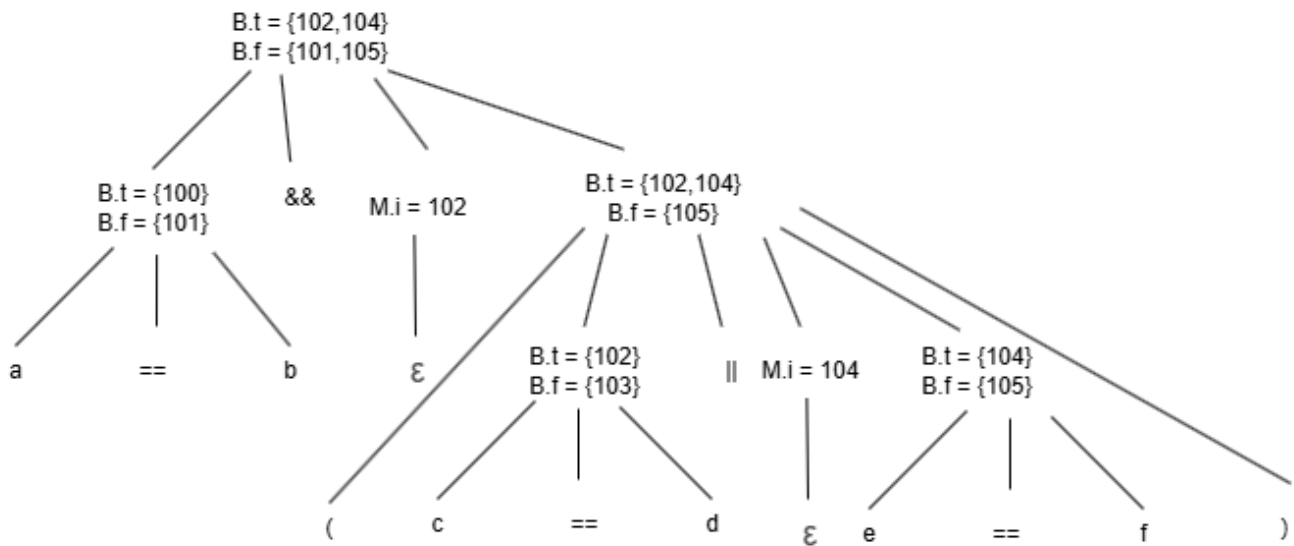


图 4: $x < 100 \ || \ x > 200 \ \&\& \ x \neq y$ 的注释语法树与跳转指令

Answer:



(b) 根据图 3 的布尔表达式的翻译方案可以将布尔表达式 $x < 100 \parallel x > 200 \&\& x \neq y$ 翻译成图 4(b) 的跳转指令, 该翻译过程先将跳转目标留空, 再通过对应的语义动作回填跳转目标, 请你以同样的方式给出 $a == b \&\& (c == d \parallel e == f)$ 的跳转指令 (假设第一条指令的地址为 100)。(2 分)

Answer:

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