**基于MOOC/SPOC翻转课堂预习笔记**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 姓名：葛玉菲 | 班级：软件2003 | 课程名：编译原理 | 指导教师：王曙燕 | |
| 报告时间：2023.3.14 | 预习题目：三元式、四元式 | | 本人得分： |  |
| 本组成员： | 葛玉菲、崔雨、张鑫瑶 | | 小组整体分数： |  |
| 预习内容及重点、难点 | 掌握中间代码常见的几种形式之一：三地址码（三元式、四元式、间接三元式）是什么？优缺点？ | | | |
| 1.预习中遇到的问题有哪些？  2.讲解过程与分工如本次翻转小组如何分工？你负责的部分？你完成了哪些工作？如制作PPT页码。  3.主要讲解点重点、难点及算法思想及实现等。 | 1. 什么是三地址码？各种语句的三地址码形式？  三地址码是一种表示程序中每条指令的中间代码格式。它将每个操作符和操作数表示为一个独立的地址，并使用一个操作码来描述指令的操作。通常，三地址码用于编译器的优化和代码生成阶段。  以下是不同类型语句的三地址码形式：  1. 赋值语句：`x = y + z`  ```  t1 = y + z  x = t1  ```  2. 条件语句：`if (x > y) then z = 1 else z = 0`  ```  if x > y goto L1  z = 0  goto L2  L1: z = 1  L2:  ```  3. 循环语句：`while (x > 0) do x = x - 1`  ```  L1: if x <= 0 goto L2  x = x - 1  goto L1  L2:  ```  4. 函数调用：`y = f(x, z)`  ```  param x  param z  t1 = call f  y = t1  ```  5. 数组访问：`a[2] = a[1] + 3`  ```  t1 = a[1]  t2 = t1 + 3  a[2] = t2  ```   1. 三地址码的三种表现形式：三元式、四元式、间接三元式。三元式、四元式、间接三元式各自优缺点   在编译原理中，语义分析后通常需要生成中间代码来表示程序的语义。三元式、四元式、间接三元式是中间代码的一种形式。  1. 三元式  三元式是一种基本的中间代码表示形式，它由三个元素构成：操作符、操作数1和操作数2。例如，将a+b的结果存储到变量c中，可以使用以下三元式：  ```  add a, b, c  ```  三元式的优点是简洁明了，易于生成和优化。缺点是无法处理复杂的表达式。  2. 四元式  四元式与三元式类似，但添加了一个额外的元素——结果。例如，将a+b的结果存储到变量c中，可以使用以下四元式：  ```  add a, b, t1  move t1, c  ```  其中t1是一个临时变量，用于保存a+b的结果。四元式相比三元式具有更高的表达能力，可以处理更加复杂的表达式。但是其缺点是生成的中间代码较为冗长，可能会影响最终的执行效率。  3. 间接三元式  间接三元式是一种介于三元式和四元式之间的中间代码表示形式。它通过利用临时变量和基本块（basic block）的概念，将复杂的控制流结构转换为简单的顺序执行。例如，以下代码：  ```  if a > b then  c = a + b;  else  c = a - b;  endif;  ```  可以生成以下间接三元式：  ```  (1) if a > b goto (3)  (2) t1 = a - b  (3) t2 = a + b  (4) goto (5)  (5) c = phi(t1, t2)  ```  其中，第一条语句判断是否跳转到第三条语句或者第二条语句，第四条语句用于跳过第二和第三条语句，第五条语句使用phi函数来根据控制流的不同路径选择结果。  间接三元式比四元式更加紧凑，但是其生成和优化较为困难，需要更复杂的算法和数据结构来支持。 | | | |
| 备注： |  | | | |