**深 圳 大 学 实 验 报 告**

**课程名称： 编译原理**

**实验项目名称： 实验五、语法制导翻译和中间代码生成**

**学院： 计算机与软件学院**

**专业： 软件工程**

**指导教师： 尹剑飞**

**报告人：张欣杰 学号：2020151091 班级： 软工02**

**实验时间： 2023年5月20日至6月20日**

**实验报告提交时间： 2023/6/07**

**教务处制**

|  |
| --- |
| **实验目的与要求：**  **目的：**通过设计、实现赋值等语句的中间代码生成程序，熟练掌握语法制导翻译模式及其应用。  **要求:**  **第一部分：（简单）赋值语句的翻译与中间代码生成**  输入文件：由多条（简单）赋值语句组成的源代码，语句间用“；”隔开  输出文件：三地址码（或四元组）序列  例如：输入文件内容：a=b+c\*e/g; d=f\*g …  则输出文件内容：   * + t1=c\*e;   + t2=t1/g;   + t3=b+t2;   + a=t3;   + t4=f\*g;   + d=t4   + ……   **第二部分：（选做）条件语句的翻译与中间代码生成**  输入文件：包含条件语句和赋值语句  输出文件：对应的三地址码序列 |
| **方法、步骤：**  要完成本实验，依据实验要求进行分解，需要完成的实验步骤是：   1. 如何提取输入文件内容的表达式？   首先使用open打开文件，将表达式text逐一读入，然后存放到Analysis\_text变量中，针对该变量使用split(“;”)方法，将该变量中的表达式逐一分割出来并存储到表达式列表中。   1. 如何为赋值语句的中间代码生成做预处理工作   在本次实验中，可以循环遍历提取出来的表达式列表，将表达式进行转化，将原本的中缀表达式按照步骤转化成后缀表达式，方便后续的分析处理工作。   1. 如何将中缀表达式转换为后缀表达式   首先定义一个运算符识别栈，一个运算符栈和一个后缀表达式列表，运算符识别栈用于识别当前中缀表达式中的字符是操作数还是运算符，然后在循环体外定义一个缓冲区buffer作为操作数缓冲区(应对一个变量中含有多个字符的情况)，然后循环遍历中缀表达式，当遇到运算符识别栈中的字符的时候，首先判断缓冲区中是否为空，如果不为空，则将缓冲区的变量入栈到后缀表达式列表中，否则跳过当前步骤，然后将当前字符入栈到运算符栈中；然后再对符号“)”进行判断，如果运算符栈顶元素不为“(”则将当前栈顶元素append到后缀表达式列表中，直到栈顶元素为左括号位置；最后为其余情况，在这种情况中，只需要将当前字符加入缓冲区中即可。在遍历完成之后还需要对缓冲区进行检测，如果不为空则加入到后缀表达式列表中。   1. 如何准确获得当前四元组中的结果符   定义类，在类中增加一个类内变量Analysis\_index作为分析索引，将初始变量值设置为1，每分析得到一个四元组则该索引的值加1。   1. 如何进行赋值语句的翻译和中间代码的生成   在完成后缀表达式的转换之后，循环遍历后缀表达式列表，如果不是运算符，则将列表元素加入到操作数栈中，如果遇到运算符，则对运算符进行判断，如果是双目运算符，则从操作数栈中弹出操作数1 operand1和操作数2 operand2，将结果索引ti加入到操作数栈中，然后将组成的字符串“ti=operand1+operator+operand2”加入到分析结果列表中。如果是“=”，则取出作数1 operand1和操作数2 operand2，然后将组成的字符串“operand1 = operand2”加入到分析结果列表中。 |
| **实验过程及内容：**   1. 提取表达式   打开一个待读取文件，使用read方法写入Analysis\_text变量中，使用该变量创建一个类对象    在类对象中使用split方法分割表达式，形成一个表达式列表     1. 预处理   首先定义一个运算符识别栈，一个运算符栈和一个后缀表达式列表，运算符识别栈用于识别当前中缀表达式中的字符是操作数还是运算符，然后在循环体外定义一个缓冲区buffer作为操作数缓冲区(应对一个变量中含有多个字符的情况)，然后循环遍历中缀表达式    当遇到运算符识别栈中的字符的时候，首先判断缓冲区中是否为空，如果不为空，则将缓冲区的变量入栈到后缀表达式列表中，否则跳过当前步骤，然后将当前字符入栈到运算符栈中；    然后再对符号“)”进行判断，如果运算符栈顶元素不为“(”则将当前栈顶元素append到后缀表达式列表中，直到栈顶元素为左括号位置；    最后为其余情况，在这种情况中，只需要将当前字符加入缓冲区中即可。    在遍历完成之后还需要对缓冲区进行检测，如果不为空则加入到后缀表达式列表中     1. 赋值语句翻译与中间代码生成   在完成后缀表达式的转换之后，循环遍历后缀表达式列表，如果不是运算符，则将列表元素加入到操作数栈中    如果遇到运算符，则对运算符进行判断，如果是双目运算符，则从操作数栈中弹出操作数1 operand1和操作数2 operand2，将结果索引ti加入到操作数栈中，然后将组成的字符串“ti=operand1+operator+operand2”加入到分析结果列表中    如果是“=”，则取出作数1 operand1和操作数2 operand2，然后将组成的字符串“operand1 = operand2”加入到分析结果列表中 |
| **实验结论：**  实验结果：  输入的Analysis.txt文件内容如下：    输出结果：    通过分析结果输出，实验结果正确 |
| **心得体会：**  本次实验，编写了赋值语句的翻译与中间代码生成程序，通过本次实验，对语法制导翻译和中间代码生成有了更深的了解。  语法制导翻译是一种基于文法规则和语法分析树的转换方法。它将源代码中的语法结构与翻译动作相关联，通过在语法分析的同时执行翻译动作，生成目标代码或中间代码。每个文法规则都关联一个或多个翻译动作，这些动作可以进行语义处理、符号表管理、类型检查和中间代码生成等操作。语法制导翻译允许在语法分析的过程中同时进行语义处理，提供了一种方便的方式来将源代码转换为目标代码。  中间代码生成是编译器中的一个重要阶段，将源代码转换为中间表示形式，以便进行进一步的优化和目标代码生成。中间代码是一种与机器无关的表示形式，通常使用类似于三地址代码或四元式的形式来表示。中间代码是对源代码的抽象，更容易进行优化和生成目标代码。中间代码生成阶段通常依赖于语法分析和语义分析的结果，将源代码转换为更高级别的表示，以便后续的优化和转换。  语法制导翻译和中间代码生成程序是编译器中实现语义分析和代码生成的重要组成部分。语法制导翻译通过在语法分析过程中执行翻译动作来处理语义信息，将源代码转换为中间代码。中间代码生成阶段将源代码转换为机器无关的中间表示，为后续的优化和目标代码生成提供基础。这些阶段的设计和实现对于编译器的性能和功能起着关键作用。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：  2022年6月 日 |
| 备注： |