华南理工大学

《编译原理》课程实验报告

实验题目： 设计SAMPLE语言的语法、语义分析器

姓名： 张博 学号：

班级： 计科2班 组别：

合作者： 无

指导教师： 董敏

|  |
| --- |
| **实验概述** |
| 【实验目的及要求】  实验目的：  设计一个语法、语义分析器，能生成一系列中间代码，以四元式的方式呈现并且发现一些具体的语法、语义错误。  实验要求：   * 1. 启动程序后，先输出作者姓名、班级、学号（可用汉语、英语或拼音）。   2. 请求输入测试程序名，键入程序名后自动开始编译。   3. 输出四元式中间代码。   4. 能发现程序的语法错误并输出出错信息。   【实验环境】  操作系统：Windows win 10 |
| **实验内容** |
| 【实验过程】   1. 实验主要过程： 2. 熟悉实验要求以及sample语言的保留字和值，并调用之前的词法分析程序生成一系列二元式进行分析。 3. 读入并改写文法，消除文法的左递归。 4. 了解通过递归下降法、预测分析法构造语法分析程序。 5. 了解语义分析的方法，四元式的生成以及如何实现。 6. 对任务目的、实现方式了解后开始写代码，通过c++来实现语法、语义分析程序。   6. 实验源码保存在complier-grammer.cpp中，文法和样例保存在wenfa.txt,example2.txt,example3.txt中。  二、实验结果  1. 能对给出的样例进行词法、语法、语义分析，生成一系列四元式，代表中间代码的生成结果。    2. 能发现一系列语法错误，并找出规约错误位置。     1. 能发现重定义标识符，并将其指出。      1. 能发现未定义就使用的标识符。      1. 能发现表达式语句操作符两端标识符类型不匹配的情况。     三、实验代码及原理分析：  1. 用一个自定义的类WenFa来储存文法信息。left用来储存式子左部分的非终结符，right来存储式子右边的部分。    2. 通过调用词法分析的部分得到token，token储存词法分析后的二元式并通过token\_pos记录当前token读到的位置。      3. 通过哈希表ID\_TYPE和Used\_id记录当前出现标识符的相关信息，包括标识符的值和对应的类型，每个标识符都有一个对应的位置，通过Used\_id来存储，ID\_TYPE用来存储标识符和对应的类型。     1. 通过向量ret储存当前语义分析得到四元式的结果，中间得到的四元式会存储在ret中。      1. 通过递归下降法进行分析，通过P()对token进行规约，如果规约完成后token\_pos的位置在token的末尾，说明成功规约，输出AC并打印出程序中出现的id以及对应的类型。如果token\_pos不在token的末尾，说明规约过程中出现无法规约的部分，即有语法错误，此时输出WA并返回false。     P()是对非终结符P规约的下降函数，读取过程中先判断第一个token是否为program，不是则返回并报错，接下来读入标识符存入ID\_TYPE，并生成四元式作为中间代码，接着对变量定义EP和语句表MSEN进行下降。    EP()函数主要判别有没有var标识符，接着对DF进行下降。    DF()函数是对标识符的信息的记录以及检查有没有重定义标识符，首先读取变量定义中的类型TYPE，如果定义语句没有TYPE，则返回并报错，接着回到token\_pos的位置进行读取，每读到一个标识符ID，就将其类型定义为TYPE的类型并存储在ID\_TYPE中，如果出现重定义，即在ID\_TYPE中对应的ID已经有对应的TYPE,则返回并输出哪个标识符出现重定义。    对语句表的规约，主要判断begin、end、和.这三个保留字是否存在并出现在对应的位置，如果缺少或者位置错误则返回并打印错误语句。    SEN()语句主要通过token目前的种类判断要用哪个语句进行递归下降，如果是if则对IFS进行下降，如果是while则对WHI进行递归下降，如果是repera则对REP进行递归下降，如果是标识符则对赋值语句ASS进行递归下降，如果是begin则对语句表MSEN进行递归下降。    ADD()用于递归下降赋值语句，首先扫描标识符，如果标识符未定义则返回并输出未定义的标识符，接着读入赋值语句，标记第一个标识符的类型type1，接着读入后续的赋值号以及标识符，同时标记第二个标识符的类型type2，如果两个类型不匹配，则返回并输出哪两个标识符类型不匹配。如果都没问题，则生存四元式并存入ret中。    IFS用于规约if语句，主要检查保留字if和then，else，如果扫描到else会继续递归下降一个语句SEN()，同时生成一个等待回填的四元式，在语句执行完后，会根据下一个语句的特性进行回填，具体操作是将回填量压入两个栈中，以此判断要回填至哪个四元式。    WHI()函数与IFS类似，扫描关键字while和do，在扫描前将while开始的语句位置压入栈WHI\_POS中，读完语句后根据这个栈进行回填，最后while句结束把这个while句的位置弹出栈。    REP()函数也与IFS类似，扫描关键字repeat和until，在扫描前将repeat开始的语句位置压入栈REP\_POS中，读完语句后根据这个栈进行回填，最后repeat句结束把这个repeat句的位置弹出栈。    NUMEX1主要是用来读入算术表达式，在读入期间会记录每个标识符及其类型，同时判断标识符有没有出现未定义的情况，另外也会看二元运算符左右俩边的类型匹不匹配，不匹配也会返回并报错，同时用TMP\_CNT记录中间变量的个数，并通过其个数生成对应的四元式，通过栈YuanSu记录中间变量，与TMP\_CNT配合生成赋值语句的四元式。    BOOLL()用来处理布尔常量，如果遇到标识符会先判断是否出现，之后记录类型并判断与其他量类型是否匹配，同时生成一个等待回填的四元式，并返回。     1. 将结果存储在ret中，如果遇到错误，输出WA并指出错误位置，否则输出AC，并输出ID信息以及四元式序列。 |
| **小结** |
| 语法分析、语义分析程序是在词法分析的基础上实现的，相关的内容知识比较重要，我通过网络查找资料，查阅课本、教学PPT等方式，了解到多种实现语法分析以及语义分析的方法，在对比递归下降法和预测分析法后选择了递归下降法，因为递归下降的思路比较清晰，最后终于完成这个艰难但十分有趣的实验任务，这次实验让我学会写代码时思路清晰，注释完整的重要性，没有这些在做项目的过程中很容易慌乱，可以说是很好的一次实践经历。 |
| **指导教师评语及成绩** |
| 评语：  成绩：           指导教师签名：                                                 批阅日期： |