****

**课程设计报告**

|  |  |
| --- | --- |
| **课程名称** | **编译原理课程设计** |
| **题 目** | **WHILE循环语句的翻译程序设计（LL（1）法、输出四元式）** |
| **学 院** | **计算机科学与技术** |
| **班 级** | **计算机1701班** |
| **学 号** |  |
| **姓 名** | **张超杰** |
| **指导教师** | **林老师** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2020 | 年 | 1 | 月 | 3 | 日 |

**课程设计任务书**

**学生姓名： 张超杰 专业班级： 计算机1701班**

**指导教师： 林老师 工作单位：计算机科学与技术学院**

**题目: WHILE循环语句的翻译程序设计（LL（1）法、输出四元式）**

**初始条件：**

理论：学完编译课程，掌握一种计算机高级语言的使用。

实践：计算机实验室提供计算机及软件环境。如果自己有计算机可以在其上进行设计。

**要求完成的主要任务:** （包括课程设计工作量及其技术要求，以及说明书撰写等具体要求）

* 1. 写出符合给定的语法分析方法的文法及属性文法。
  2. 完成题目要求的中间代码四元式的描述。
  3. 写出给定的语法分析方法的思想，完成语法分析和语义分析程序设计。
  4. 编制好分析程序后，设计若干用例，上机测试并通过所设计的分析程序。
  5. 设计报告格式按附件要求书写。课程设计报告书正文的内容应包括：

1 系统描述（问题域描述）；

2 文法及属性文法的描述；

3 语法分析方法描述及语法分析表设计；

4 按给定的题目给出中间代码形式的描述及中间代码序列的结构设计；

5 详细的算法描述（流程图或伪代码）；

6 软件的测试方法和测试结果；

7 实践总结（本设计的评价、特点、不足、收获与体会等）；

**时间安排：**

设计安排一周：周1、周2：完成系统分析及设计。

周3、周4：完成程序调试及测试。

周5：撰写课程设计报告。

设计验收安排：设计周的星期五第1节课开始到实验室进行上机验收。

设计报告书收取时间：设计周的次周星期一上午10点。

**指导教师签名： 2019年 10月 1日**

**系主任（或责任教师）签名： 2019年 10月 1日**

# 1系统描述（问题域描述）

## 1.1题目描述

WHILE循环语句的翻译程序设计（LL（1）法、输出四元式）

## 1.2题目要求

1、写出符合给定的语法分析方法的文法及属性文法

2、完成题目要求的中间代码四元式的描述

3、写出给定的语法分析方法的思想，完成语法分析和语义分析程序设计

4、编制好分析程序后，设计若干用例，上机测试并通过所设计的分析程序

# 2问题分析及编译系统的概要设计

编译过程一般分为六个阶段的过程，可以由六个模块完成，它们称为词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、代码优化程序、目标代码生成程序，此外，一个完整编译程序还必须包括“表格管理程序”和“出错处理程序”。

这次实验采用的方法对文法有一定的要求，要求文法必须是LL(1)型文法，由于所用的while语句文法并不符合这一要求，所以要首先对文法进行修改，消除左递归，时文法符合LL(1)型文法的要求。同时，这次实验涉及到词法分析、语法分析、语义分析及表格管理和出错管理。其中，词法分析至少要能识别关键字“while”，标识符（即自定义变量），数字，运算符和界符等等；语法分析要分析程序结构的合法性，即是否为文法的句子；语义分析要能够语法制导翻译出中间代码四元式并将其输出；表格管理是指符号表；出错处理是指在语法分析时，所有非文法句子的错误类型处理。

# 3文法及属性文法的描述

## 3.1文法描述

### 3.1.2 while语句的初始文法描述

while语句： A→w(P){B:}

赋值语句： B→g=C

表达式： C→CHD

项： D→DIE

因子： E→f | g | (C)

加法运算符：H→+|-

乘法运算符：I→\*|/

比较运算符：J→< | > | >= | <=

关系表达式：P→CJC

无符号整数：f

标识符： g

### 3.1.2改进为LL(1)文法之后的文法描述

while语句： A→w(P){B:}

赋值语句： B→g=C

表达式： C→DM

M→HDM | ℇ

项： D→EN

N→IEN | ℇ

因子： E→f | g | (C)

加法运算符：H→+|-

乘法运算符：I→\*|/

比较运算符：J→< | > | >= | <=

关系表达式：P→CJC

无符号整数：f

标识符： g

3.1.3名词说明

以上文法中使用w代替while关键字，M和N分别是为了消除左递归引入的符号，无具体对应含义

## 3.2属性文法

### 3.2.1属性文法定义形式

属性文法是在上下文无关文法的基础上，为每个文法符号（终结符或者非终结符）配备若干相关的“值”（与文法符号相关的属性）。

在一个属性文法中，对应于每个产生式A→a都有一套与之相关联的语义规则，每规则的形式为：b:=f(c1,c2,…，ck)其中f是一个函数，而且或者b是A的一个综合属性并且c1,c2,…，ck是产生式右边文法符号的属性；或者非终结符既可有综合属性也可有继属性，文法开始符号的所有继承属性作为属性计算前的初始值。

### 3.2.2 while语句的属性文法

while语句： A→w {X1 = gotostm} (P) {X2 = gotostm}{B;}

{ backpatch(Ptruelist, M2gotostm);

backpatch(Pfalselist, tacIndex-1);

gen(‘J’, , , X1);

gen( , , ,tacIndex);}

赋值语句： B→g=C{B.val=C.val; gen(‘=’,C.val, ,g.val);}

表达式： C→D{M.i=D.val}M{C.val=M.s}

M→HD{ M1.i=newtemp; gen(H.val,M.i,D.val, M1.i);}

M1{M.s= M1.s}

M→ℇ{ M.s= M.i}

项： D→E{N.i=E.val}N{D.val=N.s}

N→IE{ N1.i=newtemp; gen(I.val,N.i,D.val, N1.i);} N1{N.s= N1.s}

N→ℇ{N.s= N.i}

因子： E→f{ E.val= f.val}

E→g{ E.val= g.val}

E→(C) { E.val= C.val}

加法运算符：H→+|-{H.val=+ | H.val=-;}

乘法运算符：I→\*|/{I.val=+ | I.val=-;}

比较运算符：J→< | > | >= | <={J.val = > | J.val =< |J.val = <= | J.val = >=;}

关系表达式：P→C1JC2{P.val=newtemp;

gen(J.val, C1.val,C2.val,P.val);}

无符号整数：f

标识符： g

### 3.3.3名词说明

上述属性文法中gen(op,par1, par2,result)生成四元式，newtemp产生一个未使用过的名字，val表示符号的值，f和g的值由输入时直接确定，tacIndex为生成四元式的标号。X1和X2是为了后续拉链回填添加的符号，无具体对应含义，backpatch(p,i)为拉链回填，将链表P中的每个元素指向i。

# 4语法分析方法描述及语法分析表设计

## 4.1语法分析方法描述

预测分析法是一种表驱动的方法，它由下推栈，预测分析表和控制程序组成。实际上是一种下推自动机的实现模型。预测分析法的关键为预测分析表的构建，即根据文法中各非终结符FIRST集和FOLLOW集求得SELECT集。预测分析法从开始符号开始，根据当前句型的最左边的非终结符和分析串中的当前符号，查预测分析表确定下一步推导所要选择的产生式，最终得到输入串的最左推导，完成输入串的语法检查。

## 4.2语法分析表设计

利用上文中提到的方法根据消除左递归后的文法求解FIRST集、FOLLOW集和SELECT集，根据SELECT集填写预测分析表，结果如表1-1所示，其中未填写部分表示分析错误。

表1-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | w | f | g | ( | ) | { | } | ; | + | - | \* | / | > | < | >= | <= | # |
| A | **w(P){B;}** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| B |  |  | **g=C** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| C |  | **DM** | **DM** | **DM** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| M |  |  |  |  | **ℇ** |  |  | **ℇ** | **HDM** | **HDM** |  |  | **ℇ** | **ℇ** | **ℇ** | **ℇ** |  |
| D |  | **EN** | **EN** | **EN** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N |  |  |  |  | **ℇ** |  |  | **ℇ** | **ℇ** | **ℇ** | **IEN** | **IEN** | **ℇ** | **ℇ** | **ℇ** | **ℇ** | **ℇ** |
| E |  | **f** | **g** | **(C)** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| H |  |  |  |  |  |  |  |  | **+** | **-** |  |  |  |  |  |  |  |
| I |  |  |  |  |  |  |  |  |  |  | **\*** | **/** |  |  |  |  |  |
| J |  |  |  |  |  |  |  |  |  |  |  |  | **>** | **<** | **>=** | **<=** |  |
| P |  | **CJC** | **CJC** | **CJC** |  |  |  |  |  |  |  |  |  |  |  |  |  |

# 5中间代码形式的描述及中间代码序列的结构设计

## 5.1中间代码形式的描述

四元式是一种比较普遍采用的中间代码形式。它的一般形式为：(op,arg1,arg2,result)。其中， op为一个二元 (也可是一元或零元)运算符；arg1,arg2分别为它的两个运算 (或操作)对象，它们可以是变量、常数或系统定义的临时变量名；运算的结果将放入result中。运算对象和运算结果有时指用户自己定义的变量，有时指编译程序引进的临时变量。需要指出的是，每个四元式只能有一个运算符，所以，一个复杂的表达式须由多个四元式构成的序列来表示。例如，表达式A+B\*C可写为序列

（1）\* B C T1

（2）+ A T1 T2

其中，T1，T2是编译系统所产生的临时变量名。当op为一元、零元运算符 (如无条件转移)时，arg2甚至arg1应缺省，即result∶=op arg1或 op result ；对应的一般形式为： (op,arg1,,result)或 (op,,,result)

在实际产生的四元式中，op往往用一整型数表示 (操作符的代码)，它可能附带有不止一种属性。

## 5.2中间代码序列的结构设计

具体四元式结构的设计如图1-1所示，使用vector结构存储生成的所有的四元式。

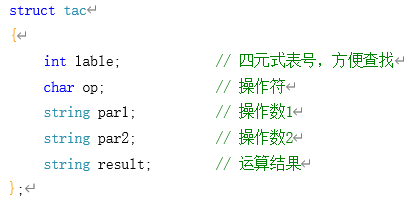


图1-1

# 6.详细的算法描述

## 6.1算法原理描述

本次实验采用的是由语法分析过程来主导语义分析及翻译的过程，即语法制导的翻译程序设计。根据上文中属性文法所规定的翻译方案，即可以预测分析法的方法自顶向下对输入的程序进行语义分析及翻译。

词法分析过程，则依据前面实验中的方法，可以提示错误单词所在位置，内容简单，不再做过多阐述。词法分析正确，则产生语法分析需要的输入串，进行语法分析及语义分析；否则，提示错误，不在进行语法分析及翻译。预测分析法进行翻译程序设计要修改语法分析的符号栈，改进为下推栈。即将属性文法中不同位置对应的翻译方案转化成对应的标号，添加到产生式中，在语法分析式同产生式的符号一同入栈。分析过程中若下推栈栈顶元素为翻译方案对应的标号，则执行标号所对应的动作并出栈，进入下一步分析，直到整个分析过程结束。

对于while语句，由于进行了关系表达式的判断所以要进行语句的跳转，存在三处跳转。第一，while语句的循环条件成立时，则跳转到循环体执行的第一条语句；第二，while语句的循环条件不成立时，则跳转到while语句的下一条语句；第三，循环体执行完毕时，则跳转到循环条件判断处，进行判断。对于跳转，采用拉链回填的方式进行处理。

## 6.3算法主要流程图

这里使用的产生式是加入翻译方案之后的产生式，词法分析在课程实验中已经做过多次，这里就不给出其流程图，给出语法分析及翻译程序的主要流程图，如图1-2所示：

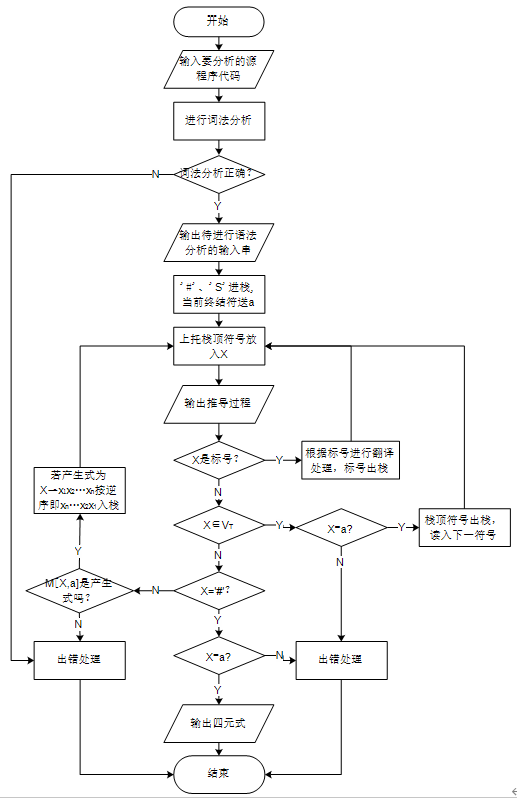


图1-2

# 7.软件的测试方法及结果

## 7.1测试方法描述

黑盒测试，[软件测试](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E6%B5%8B%E8%AF%95)的主要方法之一，也可以称为功能测试、数据驱动测试或基于规格说明的测试。测试者不了解[程序](https://zh.wikipedia.org/wiki/%E7%A8%8B%E5%BA%8F)的内部情况，不需具备应用程序的代码、内部结构和编程语言的专门知识。只知道程序的输入、输出和[系统](https://zh.wikipedia.org/wiki/%E7%B3%BB%E7%BB%9F)的功能，这是从用户的角度针对软件界面、功能及外部结构进行测试，而不考虑程序内部逻辑结构。测试案例是依据应用系统应该做的功能，照规范、规格或要求等设计。测试者选择有效输入和无效输入来验证是否正确的输出。

本次实验采用黑盒测试的方法主要对应用程序的功能进行测试，主要有两大部分：词法分析功能测试和语法制导的翻译程序功能测试。使用正确和错误的测试用例来验证是否正确的输出。

## 7.2软件测试结果

### 7.2.1词法分析测试及结果

词法分析测试结果正确和错误测试用例，输出结果分别如图1-3和图1-4所示:



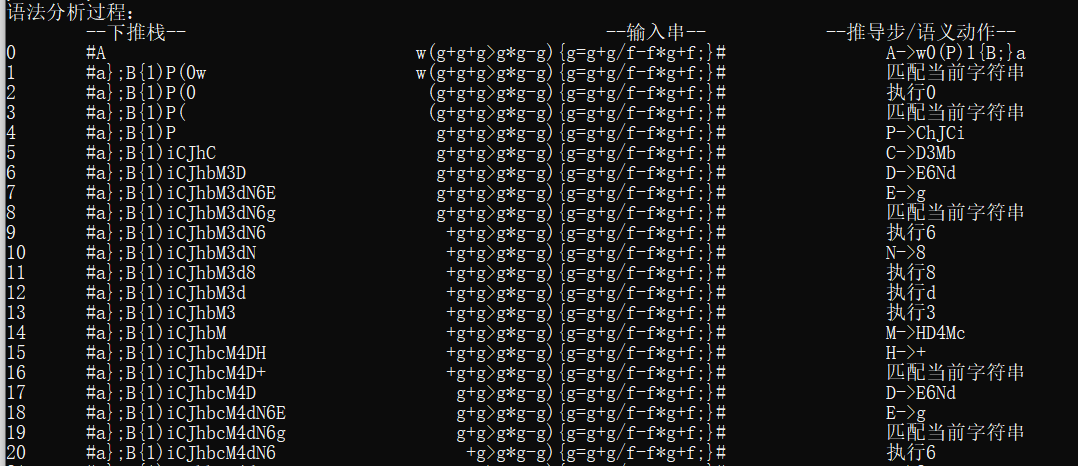
图1-3（词法分析正确之后进行语法分析及翻译）



图1-4（标识符以数字开头报错）

### 7.2.2语法分析及翻译程序测试及结果

语法分析及翻译程序测试结果正确和错误测试用例，输出结果分别如图1-5和图1-6所示:



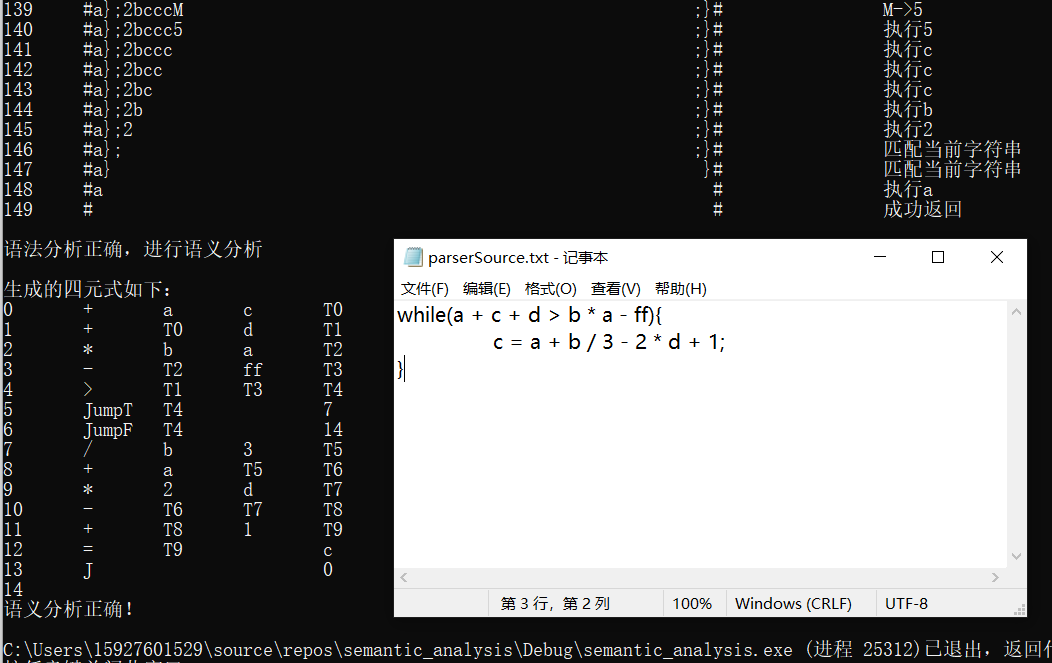


图1-5

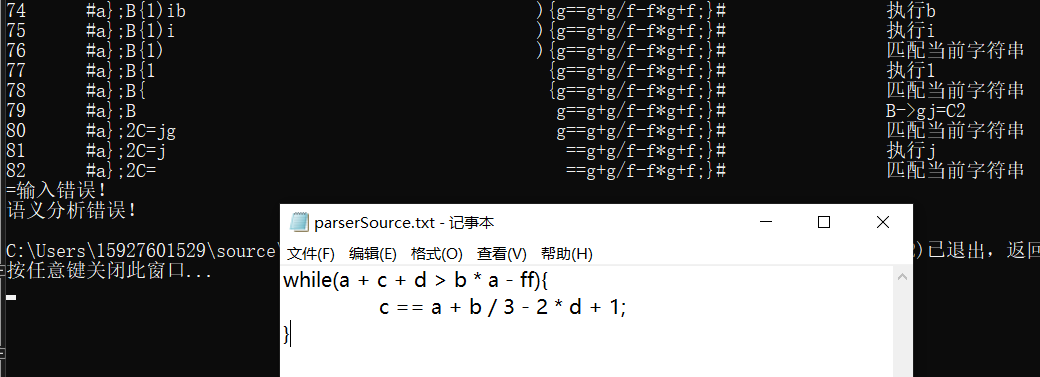


图1-6（赋值语句中使用“==”发生错误，语法分析停止）

# 8.实践总结

## 8.1思考过程

拿到设计题目以后，首先是构造符合题意的文法，对文法进行进一步改进，使其符合LL(1)文法的要求，然后构造文法所对应的预测分析表。完成这两步之后，开始编写源程序。编译程序主要分成三大模块，词法分析模块、语法语义分析模块和中间代码生成和输出。

首先进行词法分析，对输入的符号串进行分析，判断出关键字，数字，终结符和非终结符。然后进行语法分析，根据每个产生式的SELECT集，然后得出文法的预测分析表。最后通过语义分析，完成分析过程，并生成相应的中间代码。最后，将生成的中间代码输出。

## 8.2本设计的评价、特点、不足、收获与体会等

本次的课程设计是对学生编译原理学习情况的综合考察，包括词法分析、语法分析、语义分析和中间代码四个基础阶段。实践是检验学习成果的唯一标准，这不仅仅可以检验学生的学习成果，更加可以加深学生对编译原理这门课的学习和理解。相对于开始而言，课程设计可以更好地反映出学生的真实水平，也更加的公平合理。

本次课程设计是while循环语句的翻译程序设计（LL（1）法、输出四元式），使用LL(1)法自顶向下分析，并在语法分析的过程中插入语义分析，使用语法制导的翻译程序设计。本程序能正确实现对while循环语句的翻译工作，能够从文件中读取源程序字符串进行词法分析、语法语义分析以及生成相应的四元式中间代码并输出到文件中，正确完成了实验的要求。并且可以根据不同的错误进行错误提示和处理，功能相对完善，用户体验好。

但程序本身并不是十分完美，还存在一些不足之处。其中词法分析模块相对完善，但是在加入语义动作之后的语法分析模块中没有实现语句的嵌套分析。另外，不足之处就是采取存储加入语义动作之后的预测分析表的数据结构不合理，不应该使用字符串表示产生式，应该使用整形vector类型存储加入语义动作之后的产生式的每一个符号和标号。在使用LL(1)方法进行翻译程序设计时，将语义动作对应的标号添加到下推栈。但是用0~9的数字有限，如果文法相对复杂，则这几个数字不足以用来标记所有的语义动作，所以有待改进。存在不合理之处主要原因是在课设期间期末考试较多，时间不够充分，所以并没有实现语句的嵌套分析以及改进存储预测分析表的数据结构。但整体思路是清晰的，进行完期末考试之后我会逐步改善程序，使程序达到更好的效果。

这次的课程设计使我受益匪浅，对于使用预测分析法进行语法分析、语义分析有了更加深入的理解和掌握。在之前语法分析的课程实验中，我采用的是自底向上的SLR(1)法对赋值语句进行语法分析，而本次课设使用的是自顶向下的LL(1)法进行语法分析，这是我对这两大类型的方法都有了深刻的认识。在我看来，LR方法更适合用来进行语法分析及翻译程序设计，自底向上，每当进行归约的时候执行相应语义动作，相对容易理解和实现，对文法要求相对较低，但项目集的构造是其核心和困难之处。当然使用LL(1)法也可以进行分析，但对文法有较高的要求，要消除回溯（一般情况是：消除左递归；提取左公共因子），使其符合LL(1)文法的要求。各有所长，可根据实际情况进行取舍。总的来说，这次的课设不仅提高了我的编程能力，同时也培养了我的思维能力，也发现了在实际编程过程中的不足。同时也让我注重规范化编程，规范化编程往往可以事半功倍，增强代码的可读性以及便于后期代码的维护和完善。学习计算机专业的课程一定要亲自动手实践，在实践中才能够不断发现问题、解决问题，不断进步、提升自我。

**本科生实践课程成绩评定表**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓 名** | **张超杰** | **班 级** | **计算机1701班** |
| **课程设计题目：**WHILE循环语句的翻译程序设计（LL（1）法、输出四元式） | | | |
| **课程设计答辩或质疑记录：**  Q：LL(1)法对文法要求？  A：  Q：如何等价变换？  A：  Q：LL(1)中语义分析和中间代码如何实现？  A： | | | |
| **成绩评定依据：** | | | |
| **最终评定成绩（以优、良、中、及格、不及格评定）** | | | |

**指导教师签字：**

**年**   **月 日**