2017-2018 学年 第 2 学期

《编译原理》实验报告

学院（系）： 计算机科学与工程学院

班 级： 115030703

学 号： 11503070324

姓 名： 邓芮

指导教师： 黄贤英

时 间： 2018 年 4 月

目录

**[实验二 语法分析器的设计](#_Toc10951_WPSOffice_Level1)** **[3](#_Toc10951_WPSOffice_Level1)**

**[1 实验目的](#_Toc11276_WPSOffice_Level1)** **[3](#_Toc11276_WPSOffice_Level1)**

**[2 实验内容及要求](#_Toc3451_WPSOffice_Level1)** **[3](#_Toc3451_WPSOffice_Level1)**

[2.1文法的构成](#_Toc11276_WPSOffice_Level2) [3](#_Toc11276_WPSOffice_Level2)

[2.2功能设计](#_Toc3451_WPSOffice_Level2) [3](#_Toc3451_WPSOffice_Level2)

**[3 实验方案设计](#_Toc20742_WPSOffice_Level1)** **[3](#_Toc20742_WPSOffice_Level1)**

[3.1界面设计](#_Toc20742_WPSOffice_Level2) [3](#_Toc20742_WPSOffice_Level2)

[3.2读取token文件](#_Toc15315_WPSOffice_Level2) [4](#_Toc15315_WPSOffice_Level2)

[3.3总程序流程图设计](#_Toc21590_WPSOffice_Level2) [6](#_Toc21590_WPSOffice_Level2)

[3.4设计特色](#_Toc21630_WPSOffice_Level2) [7](#_Toc21630_WPSOffice_Level2)

**[4 结果及测试分析](#_Toc15315_WPSOffice_Level1)** **[7](#_Toc15315_WPSOffice_Level1)**

[4.1运行环境及限制](#_Toc14755_WPSOffice_Level2) [7](#_Toc14755_WPSOffice_Level2)

[4.2测试数据说明](#_Toc20811_WPSOffice_Level2) [8](#_Toc20811_WPSOffice_Level2)

[4.3运行结果及功能说明](#_Toc3386_WPSOffice_Level2) [8](#_Toc3386_WPSOffice_Level2)

[4.4测试及分析说明](#_Toc2087_WPSOffice_Level2) [9](#_Toc2087_WPSOffice_Level2)

**[5 总结及心得体会](#_Toc21590_WPSOffice_Level1)** **[9](#_Toc21590_WPSOffice_Level1)**

**[6 参考文件](#_Toc21630_WPSOffice_Level1)** **[9](#_Toc21630_WPSOffice_Level1)**

# 实验二 语法分析器的设计

## 1 实验目的

（1）理解语法分析的功能、输入/输出接口。

（2）能判断给定的文法能否进行自上而下的语法分析，并能对不满足条件的文法进行改造。

（3）能根据给定的LL(1)文法，编写递归下降的语法分析器。

（4）在语法分析中能处理常见的错误。

## 2 实验内容及要求

### 2.1文法的构成

根据sample语言的文法，设计了算数表达式，逻辑表达式文法，关系表达式。

### 2.2功能设计

导入token文件，进行语法分析，显示错误行号，内容，错误原因。以简单的界面显示，简洁明了，一目了然。

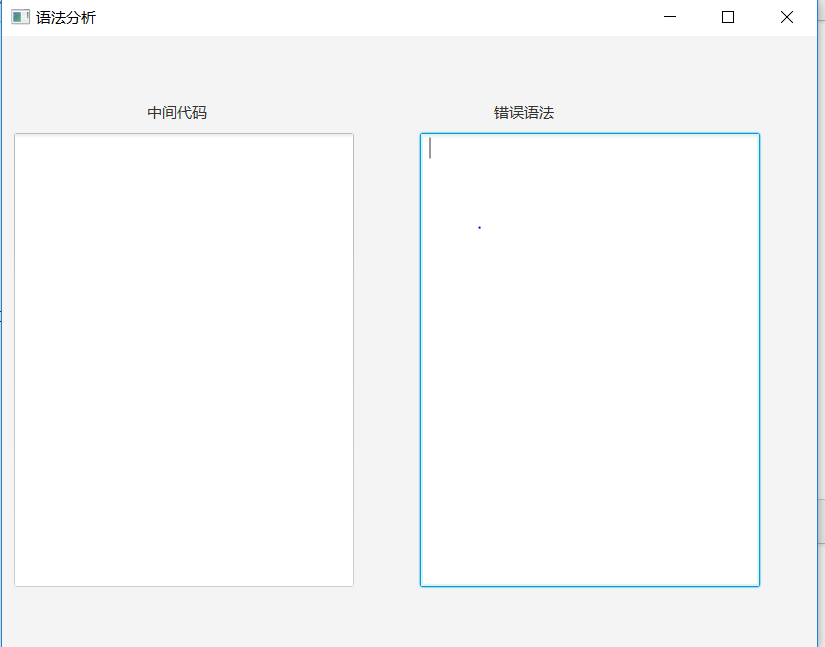
实验要求

功能相对完善，有输入、输出描述，有测试数据，不足之处：系统稳定性不高。

## 3 实验方案设计

### 3.1界面设计

本系统采用Javafxbuilder对界面进行设计，一个好的系统拥有一个良好的交互界面，本系统提供简洁界面，显示简单易懂，如下图所示。

 图3-1 语法分析界面

### 3.2读取token文件

在词法分析的基础上，本系统继续对语法进行分析，首先将词法分析所获得的token文件，存储在一个ArrayList的动态数组里面，这样使得分析语法便利。

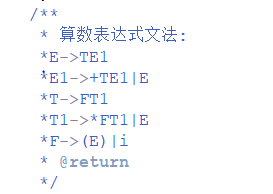


图3-2 获取token文件

文法的设计

在sample语言表达式的定义中，一共分为算术表达式，逻辑表达式和关系表达式，按照递归规则定义，如算术表达式由项进行加减运算构成，项由因子进行乘除运算构成，因子可以认为是单个的标识符，整数或实数，带括号的表达式以及因子取负。

首先，分析算术表达式。算术表达式->算术表达式+或-构成，或仅仅由带+，-号的项或不带符号的项构成。项->项\*因子或项/因子，或仅由因子构成。因子->因子量或带-号的因子或算术表达式加括号构成。因子量->标识符或者整数或者实数构成。由于需要，所以必须对该文法进行左递归以及回溯的消去，最后结果用符号表达即下图所示:

图3-3 算术表达式文法

其次，分析逻辑表达式。逻辑表达式->逻辑表达式or布尔项构成，或仅由布尔项构成。布尔项->布尔项and布尔因子或仅由布尔因子构成。布尔因子->由布尔量或布尔因子取反或逻辑表达式加括号构成。布尔量->标识符或布尔常量或关系表达式构成。关系表达式由算术表达式连接关系运算符再连接算术表达式构成，关系运算符rop为：>,<,<>,<=,>=和==中的任一符号。同理，根据需要需要消除左递归和回溯，最后结果用符号表示即下图所示:

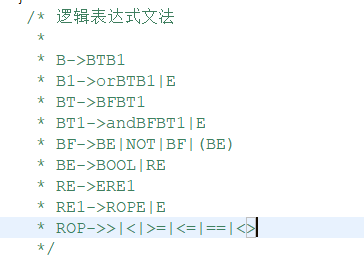


图3-4 逻辑表达式文法

状态转换图的设计

Const识别，首先识别关键字const，const是赋值语句，所有必须形如->标识符:=常量；的规则。状态转换图如下。

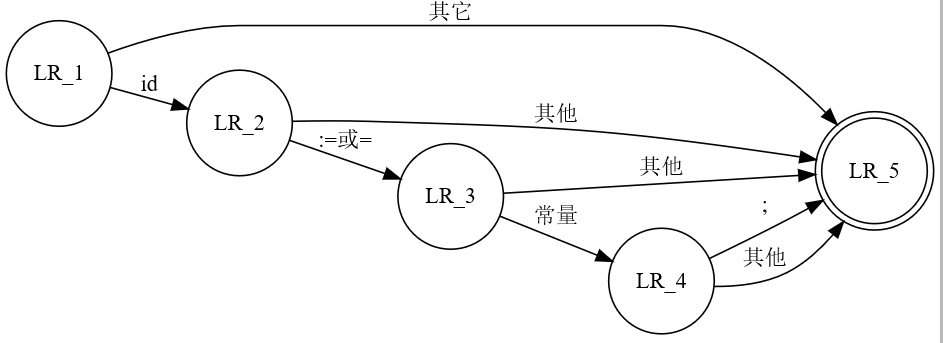


图3-4 const状态转换图

Var识别，识别规则:标识符,标识符....标识符:类型；状态转换图如下。

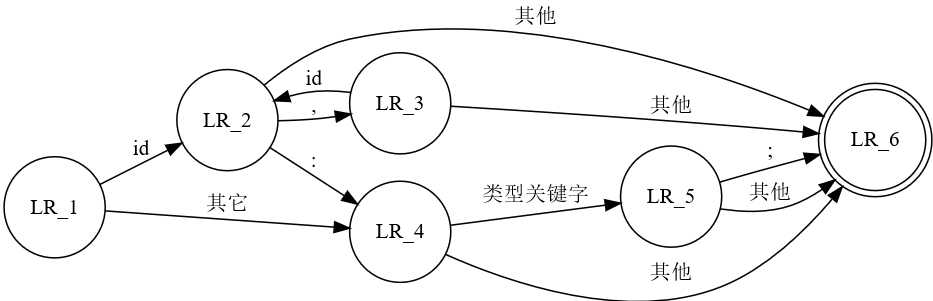


图3-5 var状态转换图

Begin end函数体识别，包含了if，while，for等的语法分析识别，流程图如下。

图3-6 begin-end流程图



### 3.3总程序流程图设计

该过程主要分为五个阶段，分别为:处理程序头部，处理const,var，处理begin，处理赋值语句，while do语句，处理for语句，处理if语句。总体流程图如下所示。



图3-4 总体流程图设计

### 3.4设计特色

本系统给用户提供良好的界面，并将错误的位置锁定显示给用户，简洁明了，简单易懂，快速找到错误之处。

## 4 结果及测试分析

### 4.1运行环境及限制

运行环境：jre.1.8.0\_161,Javafx Scene builder。

限制：本系统主要是对sample语言的语法分析，算术表达式以及逻辑表达式的文法适用于sample语言。

### 4.2测试数据说明

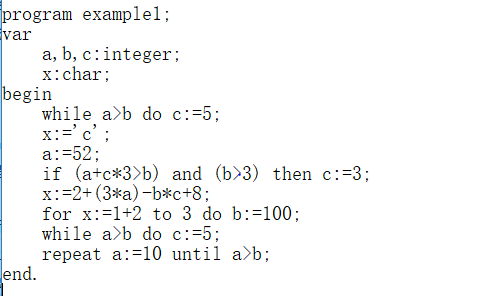


图4-1 测试数据

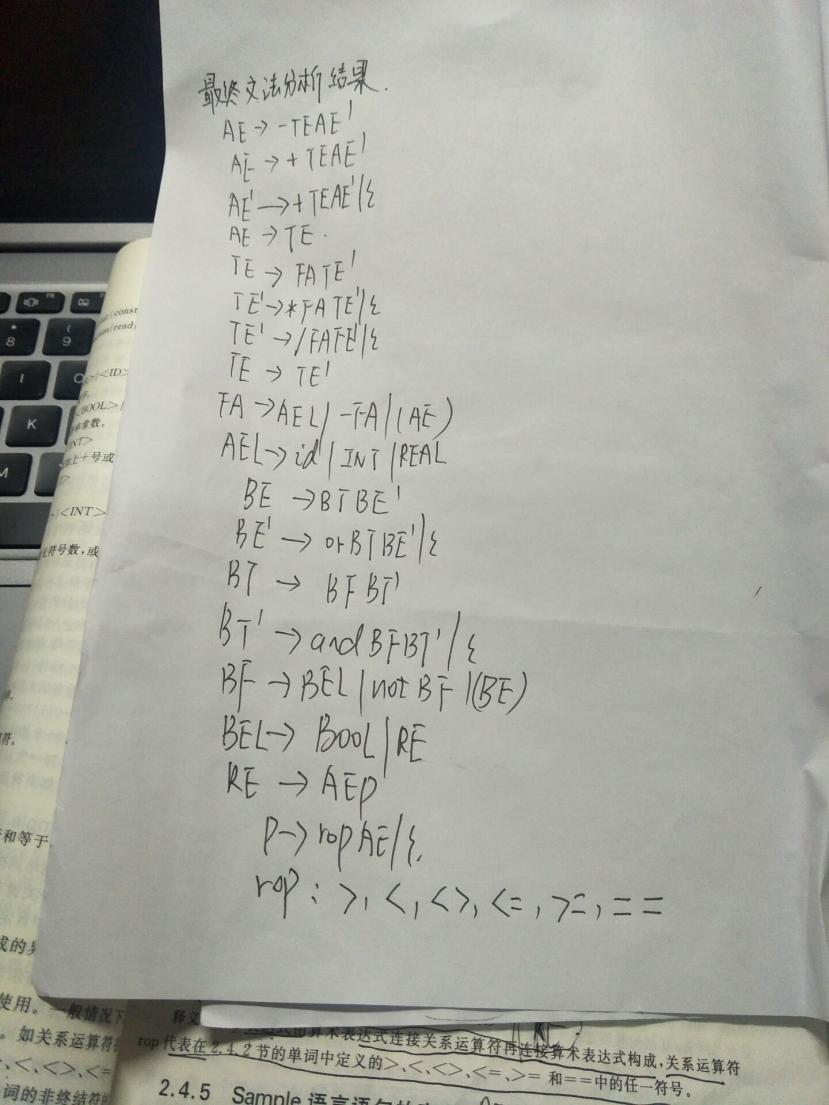


图4-2 消除左递归和回溯的文法

本系统将整个程序的运行步骤通过文法分析，分析出来。

### 4.3运行结果及功能说明

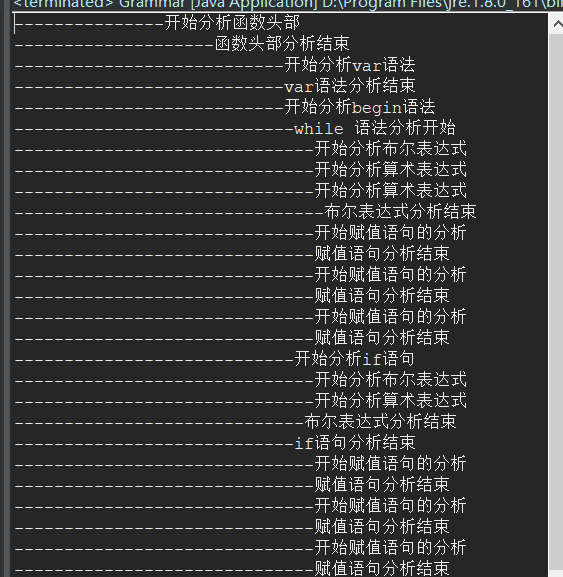


图4-3 结果显示

程序运行结果如图所示。

### 4.4测试及分析说明

本系统实现功能：良好的界面，能识别sample语言的基本语法错误。

## 5 总结及心得体会

在本次的实验中，不仅对于sample语言的语法的定义方式更加了解，而且对于自上而下语法分析中的消除回溯和左递归的方法更加的了解。自上而下分析法是再自左至右扫描输入串，但是由于程序的关键字等不同，所以需要满足特定的语法结构。不仅加深了对于sample语言的文法印象，而且也对于编程能力所有提高，逻辑思维也得到了锻炼，想问题不在从一个方面思考而是较全面的从多个方位思考。最重要的一点是：上课一定要听讲，不然写程序会绕很多弯路，本来简单的程序，却被自己复杂化。

## 6 参考文件

[1] 黄贤英.编译原理及实践教程（第2版）.清华大学出版社, 2012年。