

**综合实验报告**

( 2020-2021 年度第 1 学期)

名 称： 编译原理实验

题 目： 词法分析器，语法分析器

院 系： 计算机系

班 级： 软件1802

学 号： 201809000121

学生姓名： 王毅涛

实验课序号：

指导教师： 鲁斌，李继荣

设计周数： 1

成 绩：

日期： 2020年 12月 1 日

**实验一 词法分析器**

**一、目的与要求**

* 1. 构造实现能拆分识别代码得到的程序
  2. 构造TOKEN表
  3. 构造符号表

**二、实验正文**

在本次实验，我使用JAVA编程语言，对于文件形式存放的L语言代码文本，通过JAVA内置的方法类读入字节流，对此进行基于规则的拆解识别。

首先建立判断是否为数字，是否为字母的方法，以及输入的字符串是否为关键字的函数，符号表和TOKEN表内存放的都是TOKEN类的数据，含有原文，类型和符号表地址三个数据段，只有判断为标识符即ID时才会写入符号表文件并且填入符号表地址字段。

从第一个字符开始读取，读取到换行，TAB，空格即可跳过，如果读取到一个字母，则认为是标识符或者关键字，并且持续读入直到遇到分号。此时判断是不是关键字，如果是的话填入TOKEN表进入下一轮，不是的话认定为标识符，首先填写TOKEN表,随后进入符号表检查有无此标识符，如果没有就写入，然后返回序号给TOKEN表中对应项的符号地址字段；如果有就直接写TOKEN表即可。

如果读取到数字，则一致读取直到读取到分号或者空符号，如果读取到小数点，则继续读取直到遇到非小数字符，如果读取到的不是分号或者空符号，则报错。根据是不是小数，写入类型代码。

对于其他符号，如布尔比较运算符号，则读取到第一个符号后试探性读取下一个，如果不能构成完整符号则只保留第一个，能构成则保留全部。

对于符号表，符号表是TOKEN表子集（标识符）派生的表，依照进入先后赋予不同编号，TOKEN表想要找到符号表某一项可以直接根据符号表地址查询。

**三、综合实验总结**

1. 实验难点

对于文本的拆解提取，小数和整数的判定，符号表的构造去重和TOKEN表回填。

1. 心得体会

加强了对于词法分析的理解，提高了JAVA熟练度，为下一步实验奠定基础

**实验二 语法分析器**

**一、目的与要求**

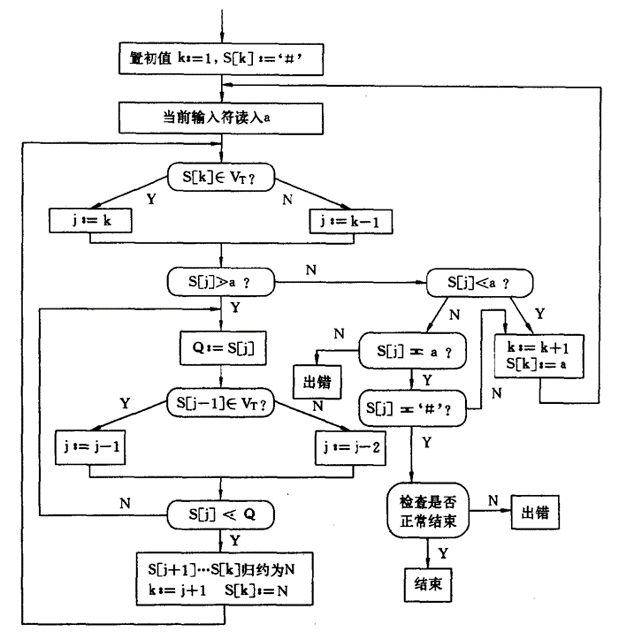
* 1. 算符优先分析法实现
  2. 递归下降分析法实现
  3. 错误提示

**二、实验正文**

在本次实验，我使用JAVA编程语言。调用上一次实验构建的表格生成器，直接可以读取需要使用的表格。词法分析阶段不需要使用到符号表

经过多次失败的尝试，以及向指导老师咨询，我采纳了递归下降和算符优先结合的方案。对于布尔判断，采用算符优先分析法，对于其他部分采用递归下降分析。

出于简化开发流程的考虑，所有的规则都是我手动录入到程序中的，使用时调用即可。对于递归下降部分，输入左部，返回右部字符序列。对于算法优先部分，输入两个字符串，返回数字表示的优先级大小。

使用栈来存储处理部分和待处理部分，处理栈和待处理栈首先都只含有一个#，随后处理栈压入开始符号。之后按照算符优先分析法的算法不断从待处理栈中取出符号和处理栈栈顶规约，或者消除。遇到布尔表达式符号，进入算符优先分析阶段。如果返回了TRUE就继续执行，不是的话立即退出报错。

上图为算法优先算法参考

规约表

优先关系表

为了计算方便，文法中的关键字，标识符等已经全部替换为数字码。

文法：

//递归下降分析

L→SZ

Z->ε|30L

S→18 31 E

E → 23TH | 24TH | TH

H->MH| Ɛ

M → 23T | 24T

T → FW

W->NW| Ɛ

N → 25F |26F

F → 18 | 19 | 20 | 21E22

S→8 B 14 S Q

Q->ε|5 S

S→17 B 4 S

S→2 L 6

S→16 D|ε

D→A 29 K 30 R

R->ε|D

A→18 X

X->ε|28A

K→9 | 3 | 13

------------------------------

上下两个是分离的两个系统

------------------------------

//以下是算符优先分析，外层的B传入时化为X

X->#B#

B->true|false|(C)|G

C->C or H|H

H->H and J|J

J->not J|B

G->E<E|E>E|E<=E|E>=E|E==E|E<>E|E

E → +T | -T | T | E+T | E-T

T → F | T\*F | T/F

F → id | num | (E)

**三、综合实验总结**

1. 实验难点

规则的计算，具体实现，两种分析法的结合

1. 心得体会

反复锤炼计算能力，提高了JAVA熟练度，极大加深课上知识的记忆强度。

您可以在我的GITHUB页面上看到我的代码：

https://github.com/massarfusion/LLanguageCompiler