

编译系统课程实验报告

实验 1：词法分析

姓名	周牧云	院系	计算学部	学号	1180300315
任课教师	单丽莉	指导教师	单丽莉		
实验地点	格物 207	实验时间	2021/4/16		
实验课表现	出勤、表现得分		实验报告得分		实验总分
	操作结果得分				

一、需求分析

得分

要求：阐述词法分析系统所要完成的功能

设计实现类高级语言的词法分析器，基本功能如下：

(1) 能识别以下几类单词：

- 标识符（由大小写字母、数字以及下划线组成，但必须以字母或者下划线开头）
- 关键字（①类型关键字：整型、浮点型、布尔型、记录型；②分支结构中的 if 和 else；③循环结构中的 do 和 while；④过程声明和调用中的关键字）
- 运算符（①算术运算符；②关系运算符；③逻辑运算）
- 界符（①用于赋值语句的界符，如“=”；②用于句子结尾的界符，如“；”；③用于数组表示的界符，如“[”和“]”；④用于浮点数表示的界符“.”）
- 常数（无符号整数（含八进制和十六进制数）、浮点数（含科学计数法）、字符串常数等）
- 注释（/*.....*/形式）

(2) 能够进行简单的错误处理，即识别出测试用例中的非法字符。程序在输出错误提示信息时，需要输出具体的错误类型（即词法错误）、出错的位置（源程序行号）以及相关的说明文字，其格式为：

Lexical error at Line [行号]: [说明文字].

说明文字的内容没有具体要求（例如：非法字符），但是错误类型和出错的行号一定要正确，因为这是判断输出错误提示信息是否正确的唯一标准。

(3) 系统的输入形式：要求能够通过文件导入[测试用例](#)。测试用例要涵盖“实验内容”中列出的各类单词。

(4) 系统的输出形式：打印输出测试用例对应的 **token 序列**

二、文法设计

得分

要求：对如下内容展开描述

(1) 给出各类单词的词法规则描述（正则文法或正则表达式）

标识符：`[_][a-z][w*]`

关键字：`r'((auto){1})|(double){1}|(int){1}|(if){1}'\`

`r'(#include){1}|(return){1}|(char){1}|(stdio.h){1}|(const){1})'`

运算符：`r'(\+|\+|\+=|\+|\-|\-|\-|\-|*|/=|/|%=|%)'`

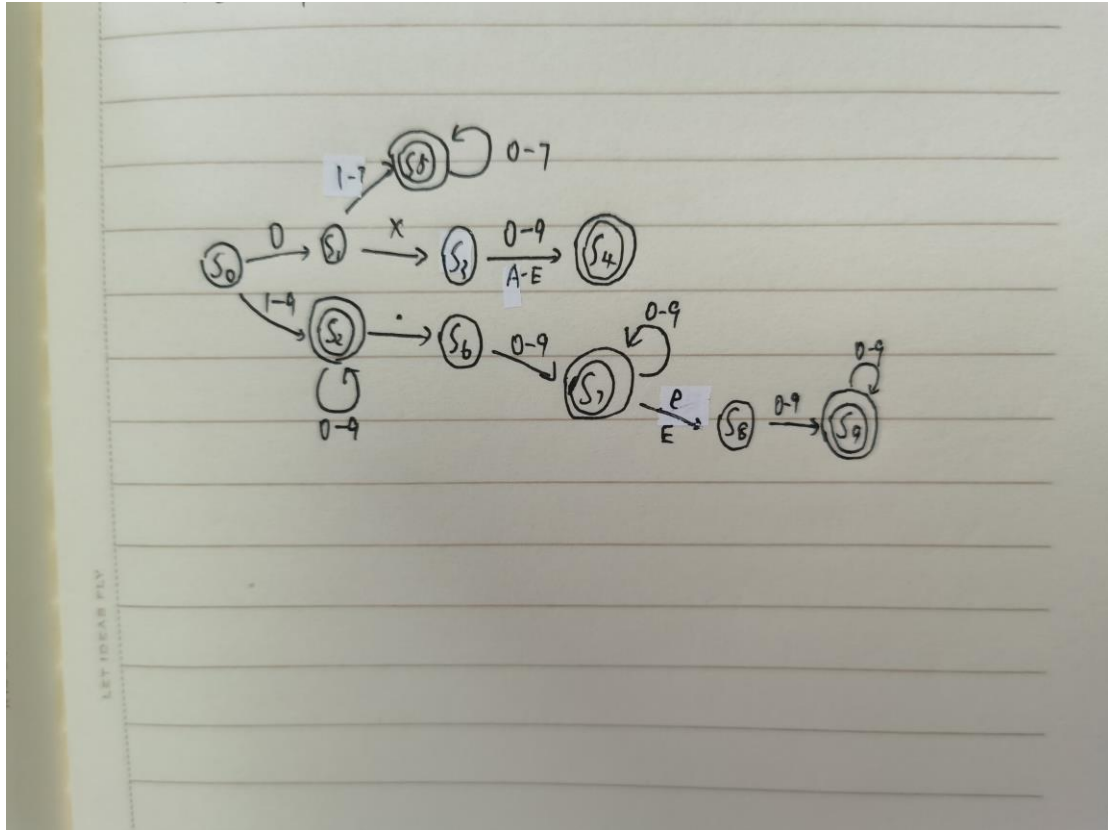
界符：`r'([,:\{\}\<>])'`

常数: $r'(\backslash d+[.]?\backslash d+)$

(2) 各类单词的转换图

其余的单词转换图较为简单

常数:



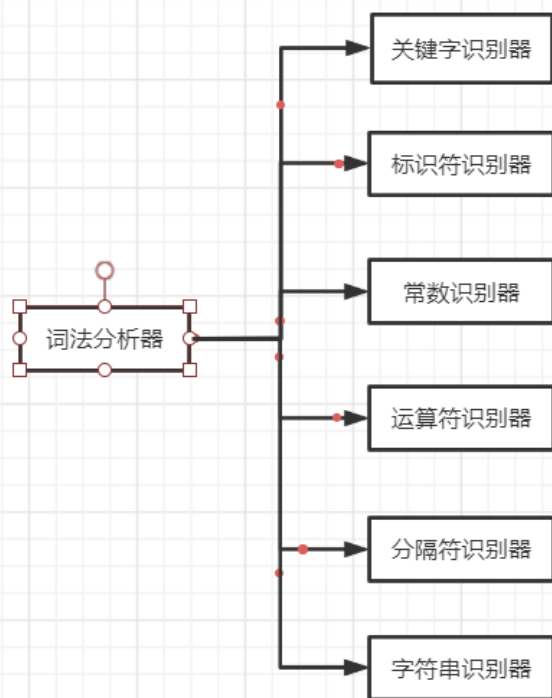
三、系统设计

得分

要求: 分为系统概要设计和系统详细设计。

(1) 系统概要设计: 给出必要的系统宏观层面设计图, 如系统框架图、数据流图、功能模块图等以及相应的文字说明。

功能模块:



(2) 系统详细设计：对如下工作进行展开描述

✓ 核心数据结构的设计

使用 Python 的列表 list[]

✓ 主要功能函数说明

`def is_blank(self, index):`

判断是否是空白字符

`def skip_blank(self, index):`

跳过空白字符

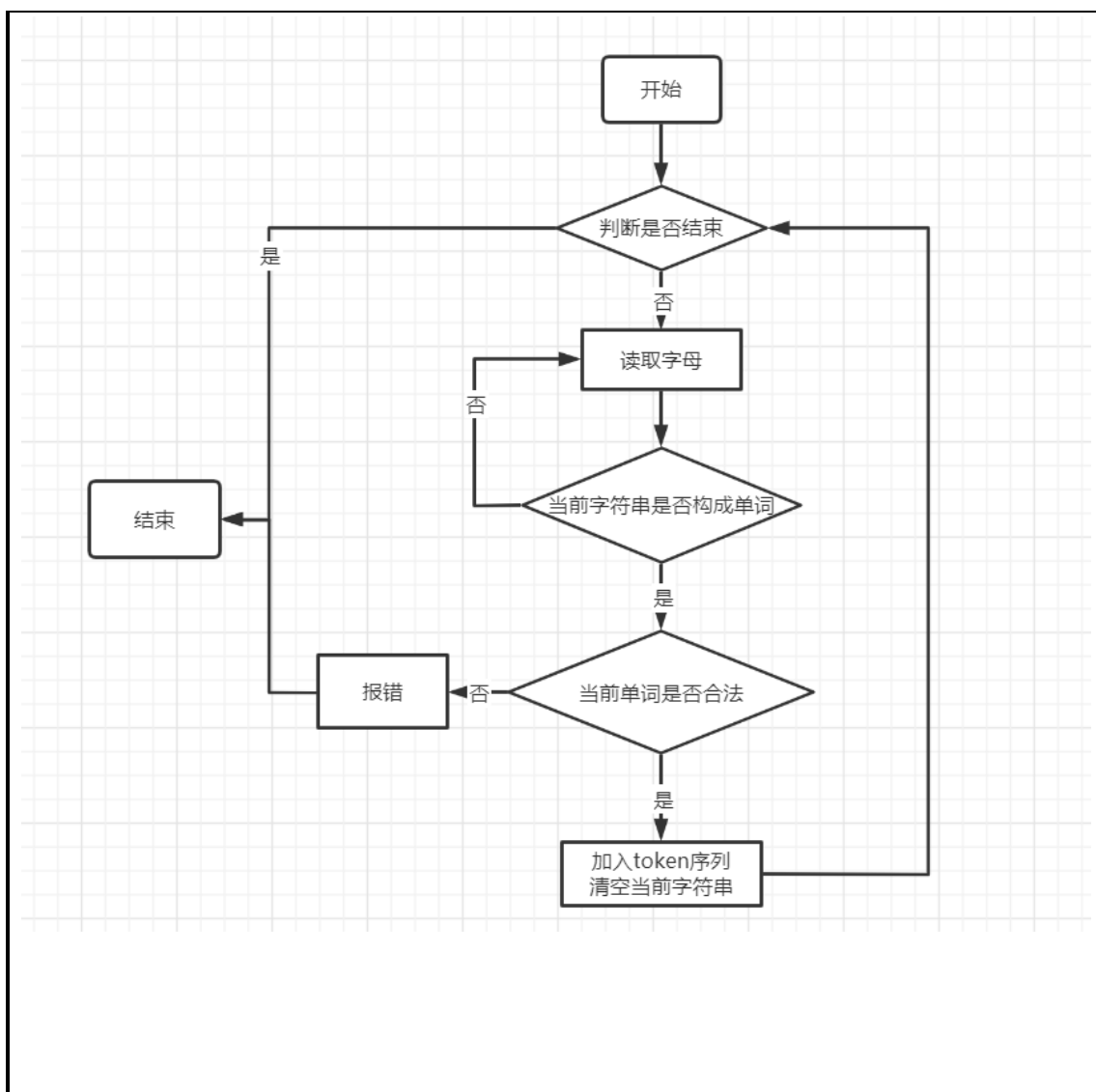
`def is_keyword(self, value):`

判断是否是关键字

`def main(self):`

词法分析的主程序

✓ 程序核心部分的程序流程图



四、系统实现及结果分析

得分

要求：对如下内容展开描述。

(1) 系统实现过程中遇到的问题；

系统对十六进制的数的识别没有考虑进去。

解决方法为在判断常数时如果判断该常数第一个数为 0 时，再判断其后的一个字母是否为 X，如果是，再判断后面的字符串是否为一连串的 0-9 或 A-F 组成，如果是，则认为该单词是常数。

(2) 针对某测试程序输出其词法分析结果；

测试样例如下：

```

#include <studio.h>
struct abc{
    int a;
    int b;
}

int c[10];

int main(){
    int a = 1;
    int b = 0;
    char *str = "String123";
    int c[0] = 123;
    int x = 0x213;
    float floatnum = 123.456;
    float floatnum_2 = 123.;
    int intnum = 123;
    bool f = a&&b;
    If(6.4 <= 1){
        printf("Yes\n");}
    return 0;
}

```

词法分析结果如下：

<SHARP, #>

<INCLUDE, include>

<LT, <>

<IDENTIFIER, studio.h>

<GT, >>

<STRUCT, struct>

<IDENTIFIER, abc>

<LP, {>

<INT, int>

<IDENTIFIER, a>

<SEMICOLON, ;>

<INT, int>

<IDENTIFIER, b>

<SEMICOLON, ;>

<RP, }>

<INT, int>

<IDENTIFIER, c>

<LM, [>

<DIGIT_CONSTANT, 10>

<RM,]>
<SEMICOLON, ;>
<INT, int>
<IDENTIFIER, main>
<LL, (>
<RL,)>
<LP, {>
<INT, int>
<IDENTIFIER, a>
<ASSIGN, =>
<DIGIT_CONSTANT, 1>
<SEMICOLON, ;>
<INT, int>
<IDENTIFIER, b>
<ASSIGN, =>
<DIGIT_CONSTANT, 0>
<SEMICOLON, ;>
<CHAR, char>
<MUL, *>
<IDENTIFIER, str>
<ASSIGN, =>
<DOUBLE_QUOTE, ">
<STRING_CONSTANT, String123>
<DOUBLE_QUOTE, ">
<SEMICOLON, ;>
<INT, int>
<IDENTIFIER, c>
<LM, [>
<DIGIT_CONSTANT, 0>
<RM,]>
<ASSIGN, =>
<DIGIT_CONSTANT, 123>
<SEMICOLON, ;>
<INT, int>
<IDENTIFIER, x>
<ASSIGN, =>
<DIGIT_CONSTANT, 0x213>
<SEMICOLON, ;>
<FLOAT, float>
<IDENTIFIER, floatnum>
<ASSIGN, =>
<DIGIT_CONSTANT, 123.456>
<SEMICOLON, ;>
<FLOAT, float>

```

<IDENTIFIER, floatnum_2>
<ASSIGN, =>
<DIGIT_CONSTANT, 123>
<SEMICOLON, ;>
<INT, int>
<IDENTIFIER, intnum>
<ASSIGN, =>
<DIGIT_CONSTANT, 123>
<SEMICOLON, ;>
<IDENTIFIER, bool>
<IDENTIFIER, f>
<ASSIGN, =>
<IDENTIFIER, a>
<ADDRESS, &>
<ADDRESS, &>
<IDENTIFIER, b>
<SEMICOLON, ;>
<IDENTIFIER, If>
<LL, (>
<DIGIT_CONSTANT, 6.4>
<LET, <=>
<DIGIT_CONSTANT, 1>
<RL, )>
<LP, {>
<IDENTIFIER, printf>
<LL, (>
<DOUBLE_QUOTE, ">
<STRING_CONSTANT, Yes\n>
<DOUBLE_QUOTE, ">
<RL, )>
<SEMICOLON, ;>
<RP, }>
<RETURN, return>
<DIGIT_CONSTANT, 0>
<SEMICOLON, ;>
<RP, }>

```

(3) 输出针对此测试程序对应的词法错误报告；
该程序有一处错误，即

```
float floatnum_2 = 123.;
```

此处的 float 型变量的格式有问题，小数点后面没有数字。

报错如下：

float number error!

(4) 对实验结果进行分析。

词法分析的结果大体上都是正确的，但对于&&并没有识别出来，而是识别成了两个&

```
<IDENTIFIER, a>  
<ADDRESS, &>  
<ADDRESS, &>  
<IDENTIFIER, b>
```

指导教师评语：

日期：