

编译原理 lab1 说明文档

161250037 郝睿

1. 实验目的:

使用 java 语言实现一个简单的词法分析器。

2. 内容描述:

使用 java 语言实现一个简单的词法分析器,能够接受外界输入进行词法分析,识别正确的输入,提示错误的输入,输出 token 序列。

3. 想法/方法:

A. 写出正则表达式 RE

B. 将 RE 转换成 NFA

C. 将 NFA 转换成 DFA

D. 基于 DFA 编写代码实现功能

4. 假定:

假定接受的输入是一个正常的简单 java 程序(可以包含常用的合法的保留字、常用的运算符以及顺序、选择、循环语句)

5. 相关 FA 描述:

A. 保留字:

```
public static String[] reservedWords = {"class", "public", "protected", "private", "static",  
"void", "main", "int", "double", "float", "char", "String", "if", "else", "else if", "switch", "case",  
"for", "while", "do", "break", "try", "catch"};
```

B. 操作符:

```
public static String[] operation = {"[", "]", "(", ")", "++", "--", "!", "*", "/", "%", "+", "-", "<<",  
">>", "<>", "<", "=", ">", "==", "!=", "|", "&", "&&", "||", "+=", "-=", "<=", ">="};
```

C. 界符:

```
public static String[] punctuation = {"", "{", "}", ";", "."};
```

D. 标识符:

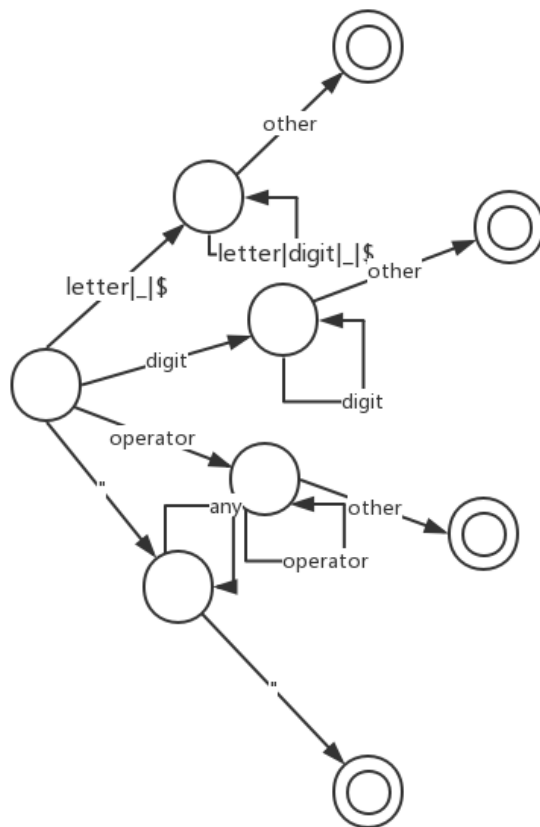
```
letter | _ | $ (letter | digit | _ | $)*
```

E. 数值:

```
digit digit*
```

F. 字符串:

```
"(letter | digit)*"
```



6. 重要数据结构描述:

Token 类: 定义了用于输出的 **token** 的数据结构

ArrayList<Character> input: 接受输入

int index: 输入流指针

ArrayList<Character> word: 用来存储每次扫描中用于判断的单词

7. 核心算法描述:

从“input.txt”文件中获取输入，将其按逐个字符的顺序存储在 **ArrayList<Character> input** 里。

Scanner()方法逐个字符的扫描输入，如果是英文字母或_或\$开头，则可能是保留字或标识符，继续往后读判断其是否是保留字，是则直接输出，否则一直读到不再是英文字母/数字/_/\$，将其作为标识符输出。如果是数字开头，一直读到不是数字的一位，将其作为数字输出。如果是"开头，一直读到下一个"，将其作为字符串输出。其他情况则作为标点符号或者操作符来分别判断。

8. 运行测试用例:



上图为自定义的输入 input 文件以及输出 output 文件

9. 产生的问题以及相应的解决办法:

- A. 一开始只考虑了标识符以英文字母开头的情况，漏掉了用下划线_或者美元符号\$组成标识符的情况。后来在识别标识符/保留字的方法中将其加了进去。
- B. 一开始没有考虑 int 型整数越界的问题，后来在舍友的提醒下增加了判断逻辑。

10. 感想和评价:

通过本次词法分析器的实验，对词法分析器有了更深的了解，巩固复习了编译原理这门课的课堂知识。