2016-10-24

词法分析实验报告

邸思诺
141250027

目录

1. 目的

模拟编译器，自己写一个词法分析程序。更深刻的理解词法分析的原理

2. 内容描述

本程序用 java 编写，读取一个文本文件，并对其中内容进行词法分析。本程序基本实现了对 java 程序的词法识别。
可识别内容包括：保留字、标识符、运算符、比较符、分隔符、数字。
输出格式为< token 类型 ， token 值 >的 token 序列。
同时，对未识别出的字符、非法数字进行报错。

3. 思路与方法

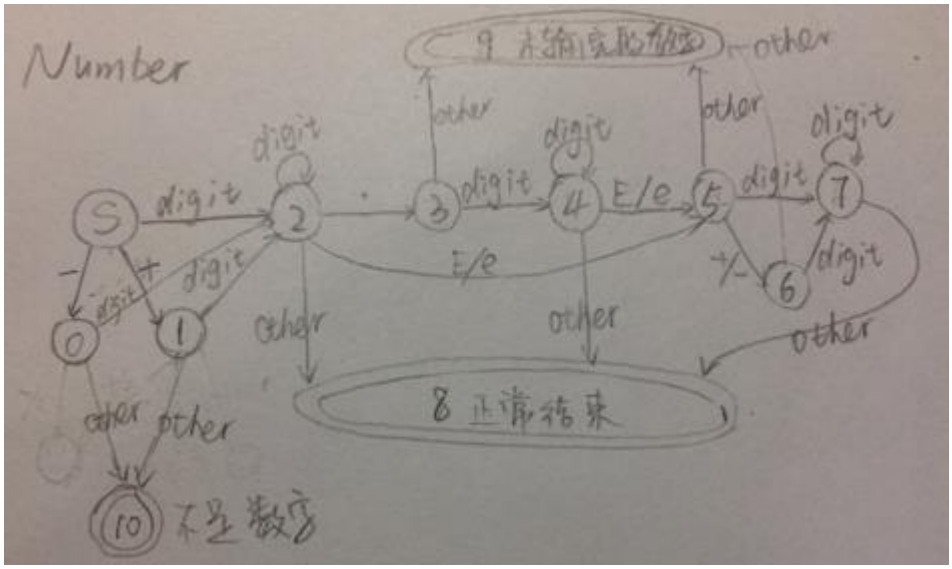
1.针对要识别的 token 写出正则表达式
2.构造出个正则表达式对应的 NFA
3.合并所有 NFA 并化简为 DFA
4.优化 DFA 为 DFA°
5.基于 DFA°编写代码
6.代码中具体的实现：先将文件全部读取，以空白（包括空格、回车、制表符）为分界将读入的内容划分成 string 的 list，再逐一对每个 string 里面的每个 char 进行分析

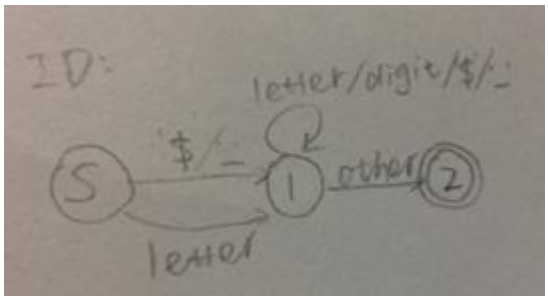
4. 假设

假设输入的文件内容是正常的 java 程序，即包含合法的保留字和运算符。

5. FA 描述

涉及到的主要的 FA 就是判断 Number 和 ID 的，如下。





6. 主要数据结构描述

tokens 用来存放词法分析后的 token 序列，
keyWordsList 存放本程序可识别的保留字，是从枚举类 TokenType 读取的
Token 类的成员变量有类型和值

```
private ArrayList<Token> tokens;
private ArrayList<String> keyWordsList=
```

```
public class Token {
    private TokenType type;
    private String value="";
```

```
public enum TokenType {
    NUMBER(1),
    ID(2),
    SEPERATOR(3),//分隔符      ; { } , . [ ] ( ) :
    RELOP(4), //比较符         > < == >= <= <> !=
    OPERATOR(5),//运算符       + - * / = | & ! ? || && ++ -- += *= /= %=
    UN_KNOW(6),
//保留字们。。。*************************************
IF(0),ELSE(0),WHILE(0),DO(0),CASE(0),SWITCH(0),DEFAULT(0),FOR(0),RETURN(0),C
```

……

}

7. 核心算法

本程序主要方法有

```
public static ArrayList<String> getInputWord(String src);//读取文件
private String numberAnalyser(String word,int st);//对 number 的词法分析
private void analyse(String word);//主要词法分析
```

1. getInputWord：

   先对文件的空白（空格、换行符、制表符）做处理，将每处空白替换成一个空格，然后再用 Split 将文件内容按照空白划分成 String 的一个 list。

2. numberAnalyser：

   是分析是否是 number 的，思路如前面 5.FA 描述提到的

3. analyse：

   本程序的算法主体。

   逐一对读取的每个 sring 里面的每个字符进行词法分析。

   若第一个字符为字母，则不断读取下一个字符并检查是否为保留字（与保留字 list 比对），直到出现不是字母、数字、$、_的字符出现则停止继续读取。此期间若遇到保留字匹配且下一位不是字母或数字，则创建一个 Token 对象并保存。否则就认作是 ID

   然后是判断分隔符（单字符并且不用读取下一个字符）

   之后是数字，调用 numberAnalyser 方法。

   接下来是一系列的比较符、运算符的判断，都需要读取下一位字符。

   以上每一个判断的算法块，最后都需要判断这一个 string 在产生了这个 token 之后是否还有剩余，若有剩余则继续调用 analyse 方法。

8. 运行截图

输入文件：

```
public static char hhh() {

        int intX=-4.5e3;
        double y=+3.298;
        char $holder_ ;
        if(x!=y&&y>=0){
            x++;
        }else y/=6;
        x>>2;
        return '';
}
```

输出结果：

```
< 保留字PUBLIC , >
< 保留字STATIC , >
< 保留字CHAR , >
< ID , hhh >
< 分隔符 , ( >
< 分隔符 , ) >
< 分隔符 , { >
< 保留字INT , >
< ID , intX >
< 运算符 , = >
< 数字 , -4.5e3 >
< 分隔符 , ; >
< 保留字DOUBLE , >
< ID , y >
< 运算符 , = >
< 数字 , +3.298 >
< 分隔符 , ; >
< 保留字CHAR , >
< ID , $holder_ >
< 分隔符 , ; >
< 保留字IF , >
< 分隔符 , ( >
< ID , x >
< 比较符 , != >
< ID , y >
< 运算符 , && >
< ID , y >
< 比较符 , >= >
< 数字 , 0 >
< 分隔符 , ) >
< 分隔符 , { >
< ID , x >
< 运算符 , ++ >
< 分隔符 , ; >
< 分隔符 , } >
< 保留字ELSE , >
< ID , y >
< 运算符 , /= >
< 数字 , 6 >
< 分隔符 , ; >
< ID , x >
< 运算符 , >> >
< 数字 , 2 >
< 分隔符 , ; >
< 保留字RETURN , >
< 分隔符 , ' >
< 分隔符 , ' >
< 分隔符 , ; >
< 分隔符 , } >
```

9. 问题&解决方法

1. **问题：**

   如果将整个输入文件空白都去掉，形成 char 数组，然后进行词法分析的话，会导致如：in tx 被解析成 int x

   int intX 被解析成 int int X 等类似错误

   解决方法：

   将输入文件的空白都替换成一个空格（空白行则删掉），然后用 split 方法将输入文件的内容划分成 String 数组，再交给词法分析程序对每个字符串逐字符的分析

2. **问题：**

   保留字判断失误，

   如：char_x 被解析成 保留字 char 和 ID _x，正确结果应该是 ID char_x

   解决方法：

改进保留字判断的 FA，在确认字符串与保留字匹配之后，要再读一位字符，确保其不是字母或数字。

10.个人总结

这次的实验总体来说就是看起来容易写起来难。真的开始写代码的时候才意识到有很多细节都要考虑。总之这次实验让我更深刻的理解了词法分析程序的运作原理。~还是挺开心哒