编译原理lab1 Report

——151250192张隽诚

# Motivation/Aim

Lexical Analyzer Programming

根据java文法生成对应的词法分析器（java语言）

# Content description

使用java语言。

包含标识符id，关键字keyword，数字num，运算符operator，分界符separator

以及注释doc的辨别，具体包含内容见[Description of important Data Structures](#_Description_of_important)

包含空白字符的过滤。

包含一些非法输入的处理和注释出错的处理。

Input：stream of characters

Output：sequence of tokens

e.g.

输入： i=2+1;

输出如下：

(id,i)  
(operator,=)  
(num,2)  
(operator,+)  
(num,1)  
(separator,;)

# Ideas/Methods

a) Define some REs by yourself

b) Convert REs into NFAs

c) Merge these NFAs into a single NFA

d) Convert the NFA into a DFAo with minimum states

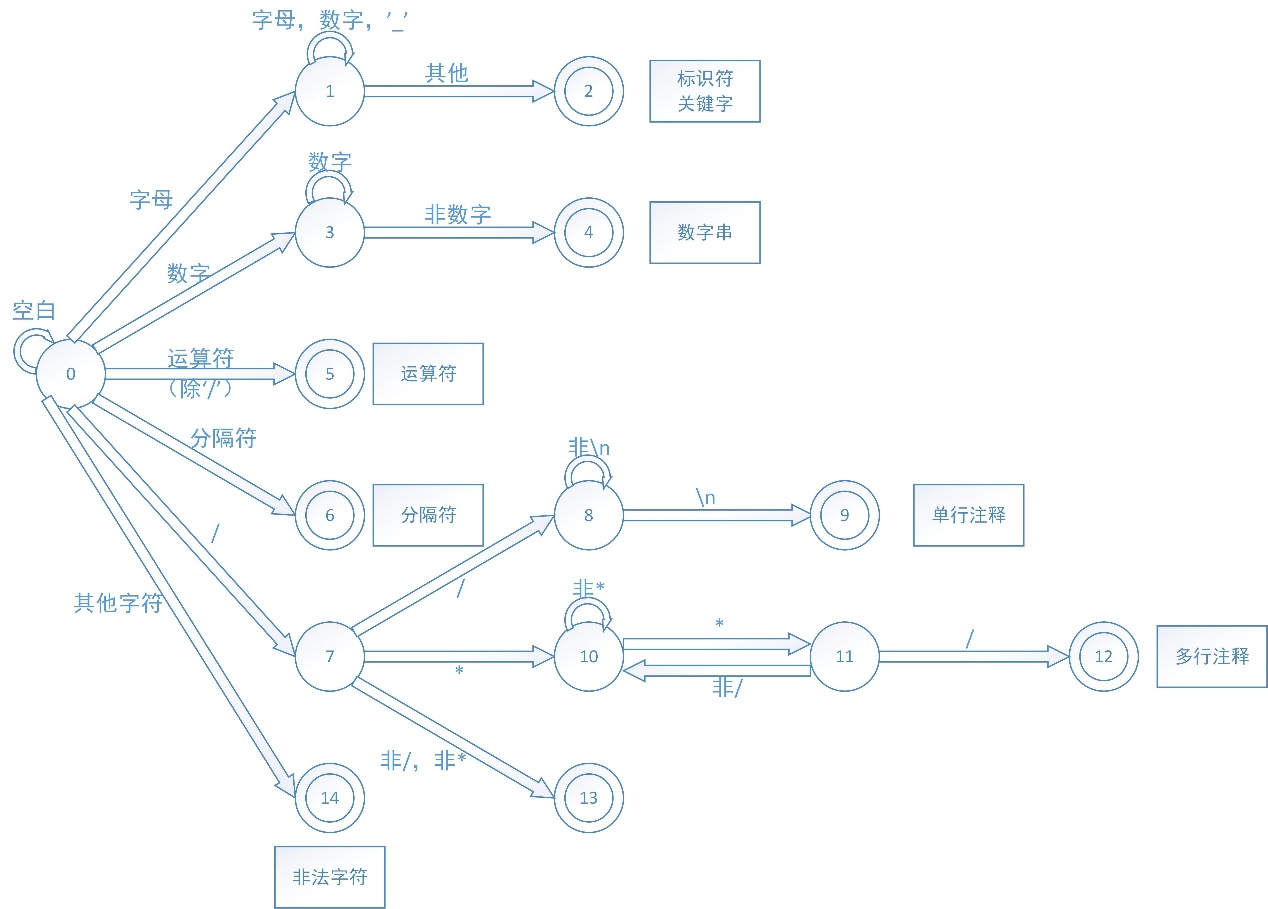
e) Programming based on the DFAo

# Assumptions

输入为一般的以Java语言写成的.java文件对应的字符流文本。

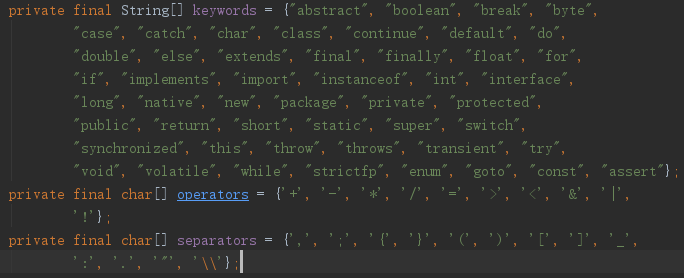
不包含++，--，\*\*，==二连运算符的判定。

# Related FA descriptions



# Description of important Data Structures

关键字，运算符，分界符的集合如下：



# Description of core Algorithms

FileIO类负责文件相关操作，包含读入输入字符流，清空输出文件，写入输出文件

TypeCheck类负责识别token类型

Analyzer类为核心类，继承了TypeCheck类，

其中analyse()方法实现了词法分析，具体状态图参见[Related FA descriptions](#_Related_FA_descriptions)，

generate()方法将识别出的token格式化，打印并写回输出文件。

Start类为普通启动类，创建Analyzer实例，调用analyse()方法进行词法分析。

# Use cases on running

详见input.txt和output.txt

启动器代码Start.java加入一条赋值语句作为一个输入case：

package src;  
  
/\*\*  
 \* @author Bourbon  
 \* @date 2017/10/26  
 \* @description 启动器  
 \*/  
public class Start {  
 public static void main(String[] args) {  
 int i = 2+1;  
 new Analyzer("./input.txt").analyse();  
 }  
}

输出结果：

(keyword,package)  
(id,src)  
(separator,;)  
(doc,/\*\* \* @author Bourbon \* @date 2017/10/26 \* @description 启动器 \*/)  
(keyword,public)  
(keyword,class)  
(id,Start)  
(separator,{)  
(keyword,public)  
(keyword,static)  
(keyword,void)  
(id,main)  
(separator,()  
(id,String)  
(separator,[)  
(separator,])  
(id,args)  
(separator,))  
(separator,{)  
(keyword,int)  
(id,i)  
(operator,=)  
(num,2)  
(operator,+)  
(num,1)  
(separator,;)  
(keyword,new)  
(id,Analyzer)  
(separator,()  
(separator,")  
(separator,.)  
(operator,/)  
(id,input)  
(separator,.)  
(id,txt)  
(separator,")  
(separator,))  
(separator,.)  
(id,analyse)  
(separator,()  
(separator,))  
(separator,;)  
(separator,})  
(separator,})

# Problems occurred and related solutions

Problem：

无法处理结尾为“/\*”型注释情况中，注释中文件结束的情况。

Solution：

每次读入字符都判断是否到底文件末尾，降低效率，提高功能性。

# Your feelings and comments

让人头大。