编译原理实验简报

1.语言定义

(1)标识符

REs:

id–>letter+digit*|letter*

letter –>[a–z][A–Z]

digit–>0|1|2|3|4|5|6|7|8|9

(2)常数

REs:

num–>INT|DOUBLE

INT–>digit dight*

DOUBLE–>dight dight*.digitdight*

digit –>0|1|2|3|4|5|6|7|8|9

(3)关键字

```
"abstract", "boolean", "break", "byte",
        "case", "catch", "char", "class", "continue", "default",
"do",
        "double", "else", "extends", "final", "finally", "float",
"for",
        "if", "implements", "import", "instanceof", "int",
"interface",
        "long", "native", "new", "package", "private",
"protected",
        "public", "return", "short", "static", "super", "switch",
        "synchronized", "this", "throw", "throws", "transient",
"try",
        "void",
"volatile","while","strictfp","enum","goto","const","assert"
```

(4)运算符

分为两类

一类： `'+', '-', '*', '/', '=', '>', '<', '&', '|', '!'`

第二类：+=,–=,*=,/=,>=,<=,==,!=,++, – –,

RE:

operator–>+|–|*|/|……|!|+=|–=| *=| …|– –

(5)分隔符

RE:

separator–> ,|;|{|}| (|)|…|.| “|

```
{ ',', ';', '{', '}', '(', ')', '[', ']', '_', ':', '.', '"' }
```

2.有限状态机描述

(1)

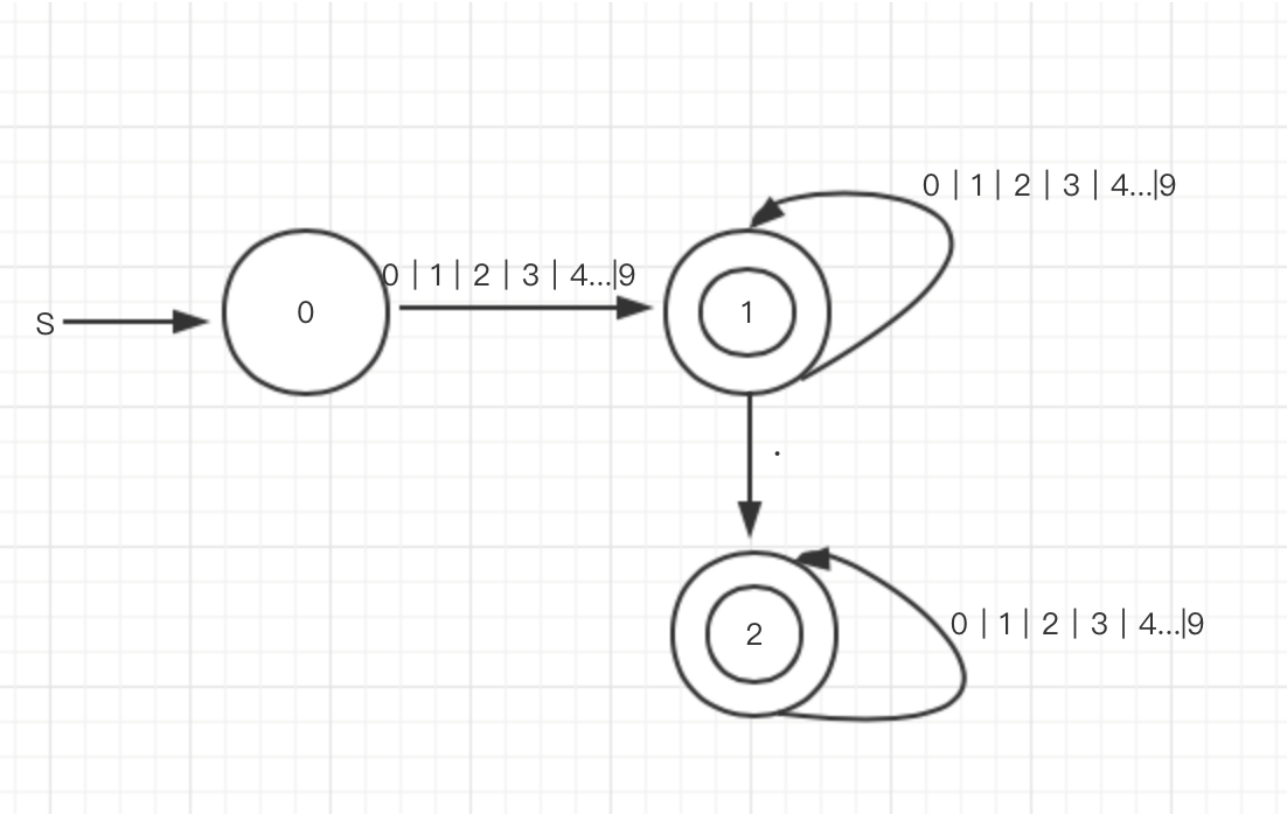
REs:

num–>INT|DOUBLE

INT–>digit dight*

DOUBLE–>dight dight*.digitdight*

digit –>0|1|2|3|4|5|6|7|8|9



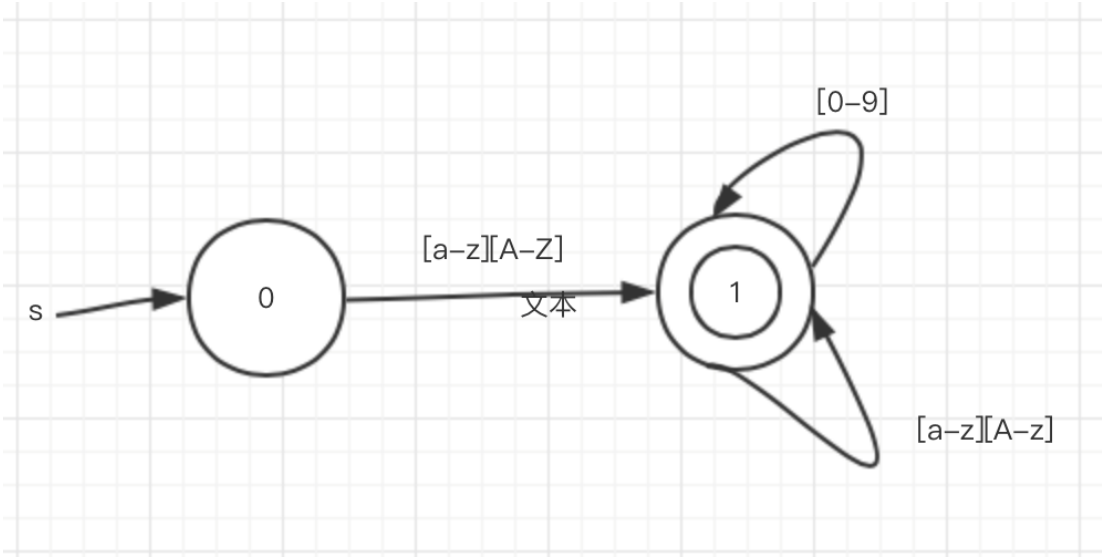
(2)

REs:

id–>letter+digit*|letter*

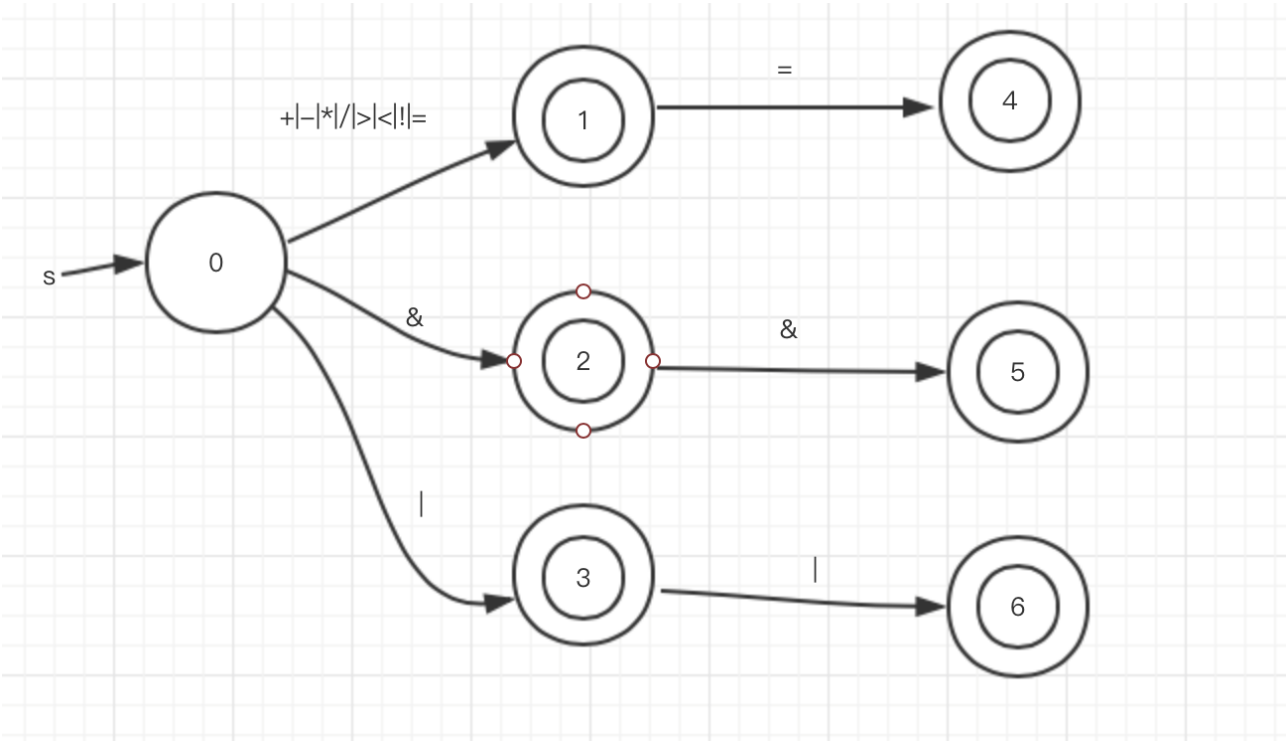
letter –>[a–z][A–Z]

digit–>0|1|2|3|4|5|6|7|8|9



(3)

RE:
operator–>(+|–|*|/|>|<|!|=)(=)?|&(&)?|(\|)(\|)?



3. 实现
具体实现在Lex.java中。
大体思路：扫描input.txt文件，处理空格、制表符、注释等条件。
从第一个有意义的字符开始处理，如果是字母，继续读下一个字符，指针下移，直到指针指向的下一个不是字母也不是数字，将这些字符组成字符串，如果这个字符串是关键字，这标注出为关键字，若不是关键字，则标注为标识符。

指针指向的字符如果是数字，则继续读取接下的字符直到不是数字。如果数字读完后紧接着一个“.”，则说明是小数，则继续读后面的小数部分。

指针指向的字符如果是分隔符，则直接标注为分隔符

指针指向的字符如果是运算符，则首先处理是”/“的情况，将注释删除。处理完注释后，根据读取的运算符，决定接下来的判断，具体见DFA.