

# 编译原理课程实验报告

## 实验 1：词法分析

姓名	姜思琪	院系	计算机学院	学号	1160300814
任课教师	辛明影	指导教师	辛明影		
实验地点	格物 208	实验时间	2019.4.14 第 3、4 节		
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**为识别以下几类单词：  
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；  
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“；”；  
常数（无符号整数和浮点数等）  
注释（/\*.....\*/形式）
- 除此之外，可以实现一些**额外功能**，如：  
识别字符常数、八进制和十六进制数；  
用户界面可视化 DFA 状态信息和状态转换表，可编辑或文件输入测试用例；  
可识别一些错误，如非法字符，注释不完全，结构不正确等。
- 本次实验基于 DFA 技术设计词法分析器，可以通过用户界面显示并编辑测试用例，也可文本输入测试用例，涵盖各类单词以及错误。  
系统的输出分为两部分：一部分是打印输出词法分析器的符号表,另一部分是打印输出源程序对应的 token 序列，

### 二、文法设计

得分

要求：对如下内容展开描述

- 给出各类单词的词法规则描述（正则文法或正则表达式）

定义  $d \rightarrow 0|1|2|\dots|9$ ,  $w \rightarrow A|B|\dots|Z|a|b|\dots|z$ ,

**标识符**:  $(w|_)(w|_|d)^*$

**关键字**: char | long | short | float | double | const | Boolean | void | null | false | true | enum | int | do | while | if | else | for | then | break | continue | class | static | final | extends | new | return | struct | case | goto | switch | case | default | auto | extern | register | sizeof | typedef | volatile

**运算符**:  $> | >= | < | <= | == | != | | | \& | || | \&\& | ! | ^ | + | - | * | / | \% | ++ | -- | += | -= | *= | /= |$

**界符**:  $, | = | ; | [ | ] | ( | ) | \{ | \} | . | ' | "$

**常数**:

$(0|1|2|\dots|9)(0|1|2|\dots|9)^*((.(0|1|2|\dots|9)(0|1|2|\dots|9)^*)|\epsilon)((E(+|-|\epsilon)((0|1|2|\dots|9)(0|1|2|\dots|9)^*)|\epsilon))|\epsilon$

**注释** ( $/*\dots*/$ 形式):  $/*(\text{其他})*/$

八进制数:  $0(1|2|3|4|5|6|7)(0|1|2|3|4|5|6|7)^*$

十六进制数:  $0x(1|...|9|a|...|f|A|...|F)(0|...|9|a|...|f|A|...|F)^*$

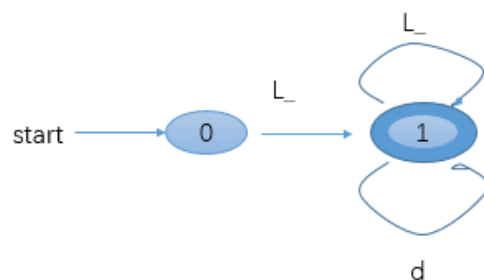
(2) 各类单词的转换图

### 1. 标识符/关键字

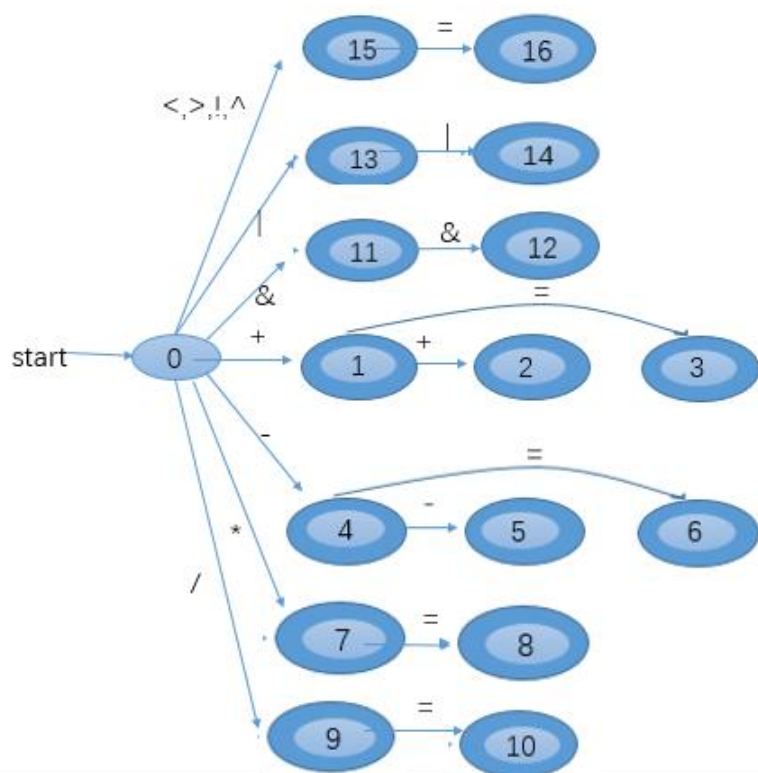
$d \rightarrow 0|1|2|\dots|9$

$l\_ \rightarrow A|B|>|>|Z|a|b|\dots|z|\_$

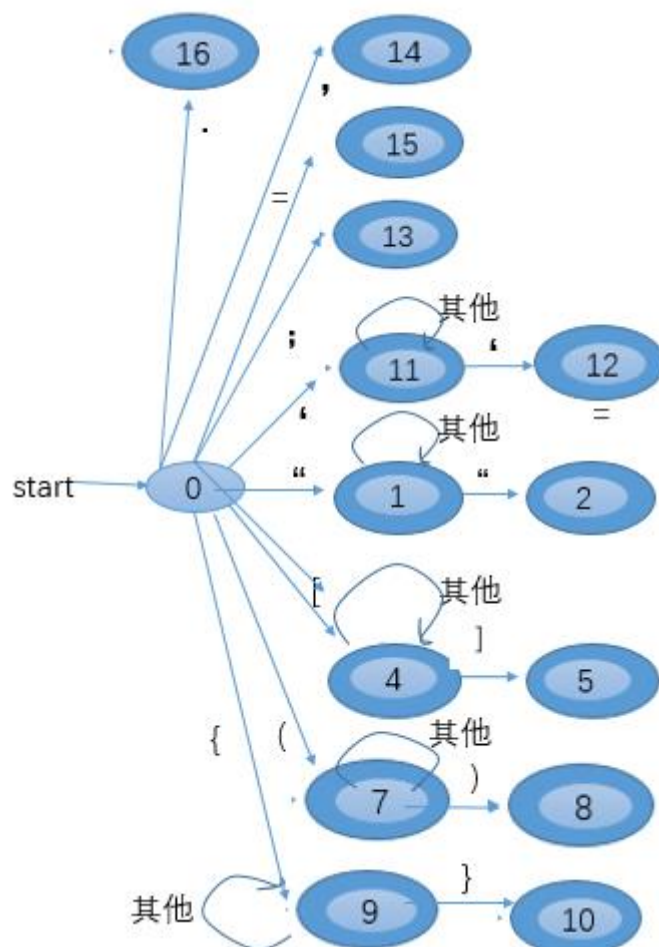
$id \rightarrow l\_ (l\_ | d)^*$



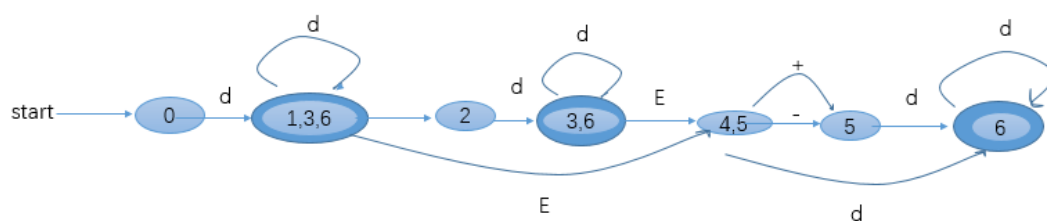
### 2. 运算符



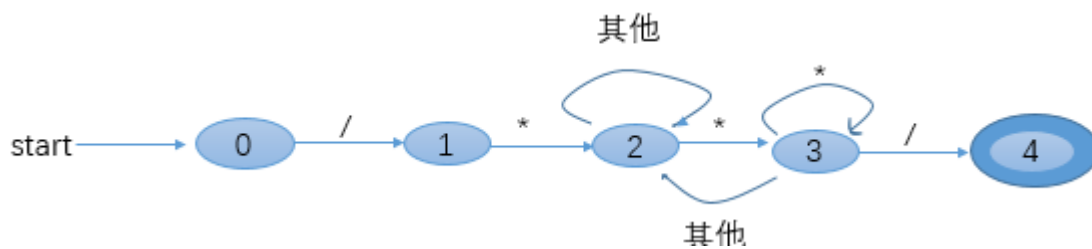
### 3. 界符



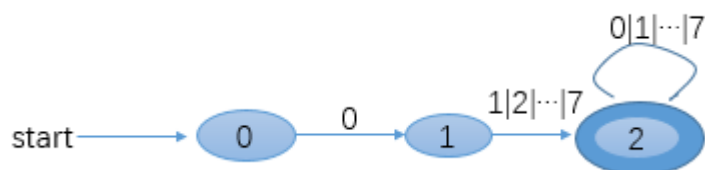
#### 4. 常数



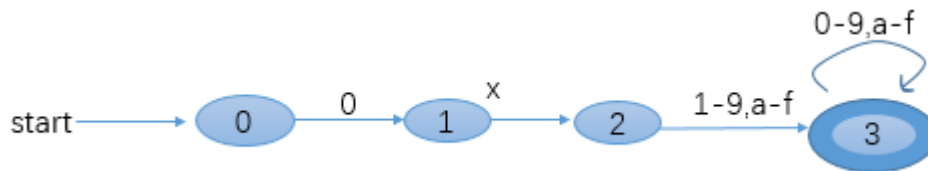
#### 5. 注释



#### 6. 八进制



#### 7. 十六进制



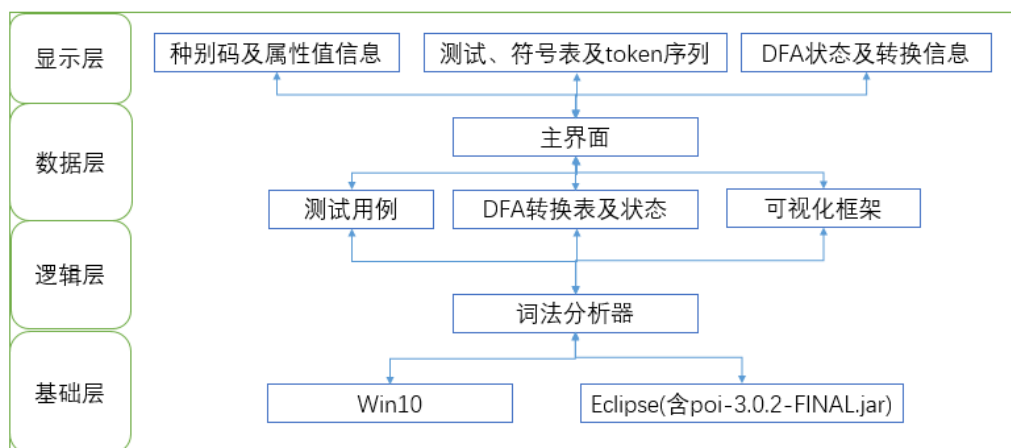
### 三、系统设计

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。

#### 1. 系统框架图

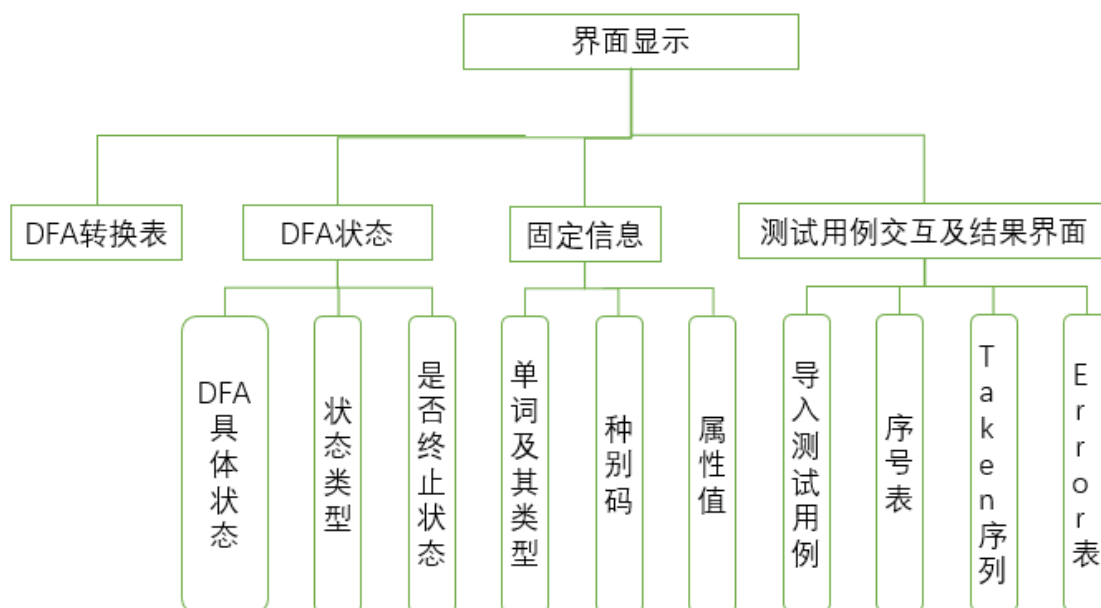


图一 系统框架图

本次实验中，基础层为硬件设施；逻辑层为主要的数据分析，是核心部分，从而实现词法分析；数据层是对应数据的读入；显示层是界面显示，使实验结果和数据可视化，方便观察。

#### 2. 功能模块图

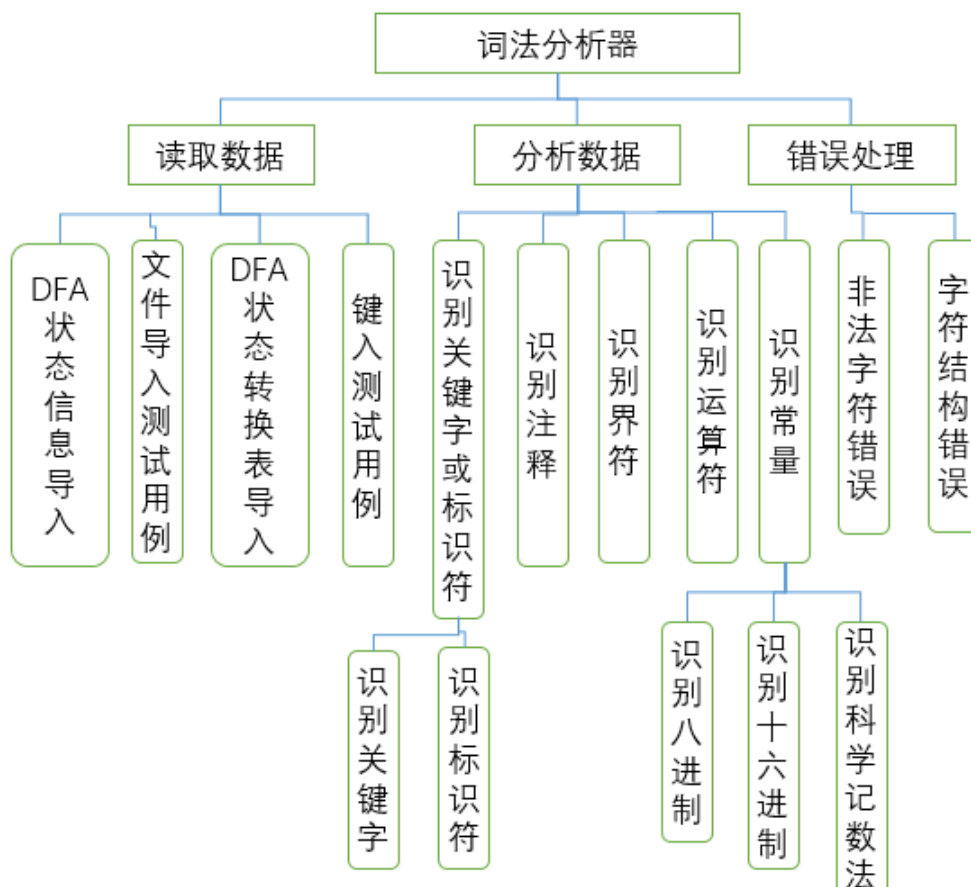
可视化模块：



图二 可视化模块

可视化模块是将实验所用数据，所得结果可视化，方便观察和使用，使系统更加美观，可读性强。

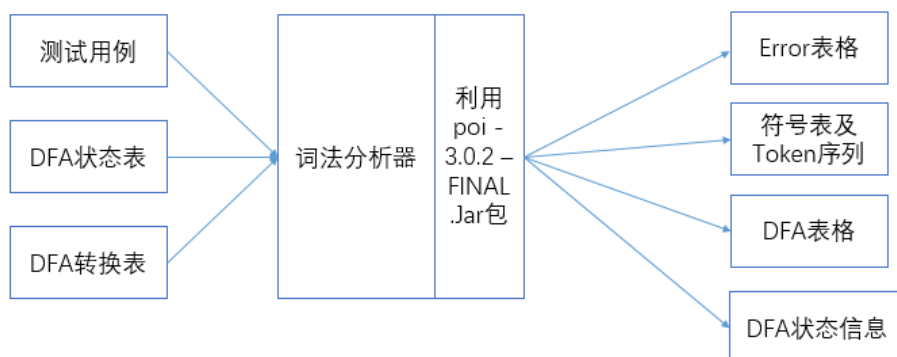
词法分析模块：



图三 词法分析模块

词法分析模块是本实验的核心模块，在此模块实现词法分析，识别关键字、标识符、界符、注释、常量（十六进制、八进制、科学记数法）等，同时识别非法字符输入、结构错误、提示注释未封闭等，通过可视化模块，将结果显示。

### 3. 数据流图



图四 数据流图

本次实验，通过输入测试用例，结合 DFA 状态表和 DFA 转换表进行词法分析，利用 poi-3.0.2-FINAL.jar 包实现结果可视化，显示 Error 表格、符号表、token 序列、

DFA 表格及 DFA 状态信息。

(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

1. Map<Integer, Boolean> **IsFinish**: 存储状态是否是终止状态, true 表示是终止状态, false 表示不是终止状态;

2. Map<Integer, String> **IsType**: 存储每个状态的具体类型;

3. **DFA.java**: DFA 转换表的存储结构, 包含

**int state**: 存储当前状态;

**String[] input**: 存储当前状态的输入符号数组;

**int nextState**: 存储当前状态的下一个状态;

**String type**: 存储当前状态状态类型 (大类);

通过以上三个数据结构, 可以将 DFA 转换表和状态表存储起来, 需要时再取出信息。

4. 种别码和属性值设计

关键字：一词一码

词	单词类型	种别码	属性值
char	关键字	CHAR	-
long	关键字	LONG	-
short	关键字	SHORT	-
float	关键字	FLOAT	-
double	关键字	DOUBLE	-
const	关键字	CONST	-
boolean	关键字	BOOLEAN	-
void	关键字	VOID	-
null	关键字	NULL	-
false	关键字	FALSE	-
true	关键字	TRUE	-
enum	关键字	ENUM	-
int	关键字	INT	-
do	关键字	DO	-
while	关键字	WHILE	-
if	关键字	IF	-
else	关键字	ELSE	-
for	关键字	FOR	-
then	关键字	THEN	-
break	关键字	BREAK	-
continue	关键字	CONTINUE	-
class	关键字	CLASS	-
static	关键字	STATIC	-
final	关键字	FINAL	-
extends	关键字	EXTENDS	-
new	关键字	NEW	-
return	关键字	RETURN	-

signed	关键字	SIGNED	-
struct	关键字	STRUCT	-
union	关键字	UNION	-
unsigned	关键字	UNSIGNED	-
goto	关键字	GOTO	-
switch	关键字	SWITCH	-
case	关键字	CASE	-
default	关键字	DEFAULT	-
extern	关键字	EXTERN	-
register	关键字	REGISTER	-
sizeof	关键字	SIZEOF	-
typedef	关键字	TYPDEF	-
print	关键字	PRINT	-
out	关键字	OUT	-

**标识符：**种别码为 **IDN**，多词一码，属性值用标识符本身来区分，在此就不列出了。

**常量：**种别码为 **CONST**，多词一码，属性值用常量本身来区分，在此就不列出了。

**运算符：**种别码为 **OP**，属性值为数字。

词	单词类型	种别码	属性值
>	运算符	OP	1
>=	运算符	OP	2
<	运算符	OP	3
<=	运算符	OP	4
==	运算符	OP	5
!=	运算符	OP	6
	运算符	OP	7
&	运算符	OP	8
	运算符	OP	9
&&	运算符	OP	10
!	运算符	OP	11
^	运算符	OP	12
+	运算符	OP	13
-	运算符	OP	14
*	运算符	OP	15
/	运算符	OP	16
%	运算符	OP	17
++	运算符	OP	18
--	运算符	OP	19
+=	运算符	OP	20
-=	运算符	OP	21
*=	运算符	OP	22

/=	运算符	OP	23
----	-----	----	----

**界符：**种别码为 BOUND，多词一码，属性值用数字表示。

词	单词类型	种别码	属性值
.	界符	BOUND	1
,	界符	BOUND	2
=	界符	BOUND	3
;	界符	BOUND	4
[	界符	BOUND	5
]	界符	BOUND	6
(	界符	BOUND	7
)	界符	BOUND	8
{	界符	BOUND	9
}	界符	BOUND	10
"	界符	BOUND	11
'	界符	BOUND	12

**十六进制：**种别码为 HEXAD，属性值为本身的形式。

**八进制：**种别码为 OCTAL，属性值为本身的形式。

**注释/\*\*/：**种别码为 NOTE，一词一码。

单词类型	种别码	属性值
注释	NOTE	-
八进制	OCTAL	(w   _)(w   _   d)*
十六进制数	HEXAD	0x(1 ... 9 a ... f A ... F)(0 ... 9 a ... f A ... F)*

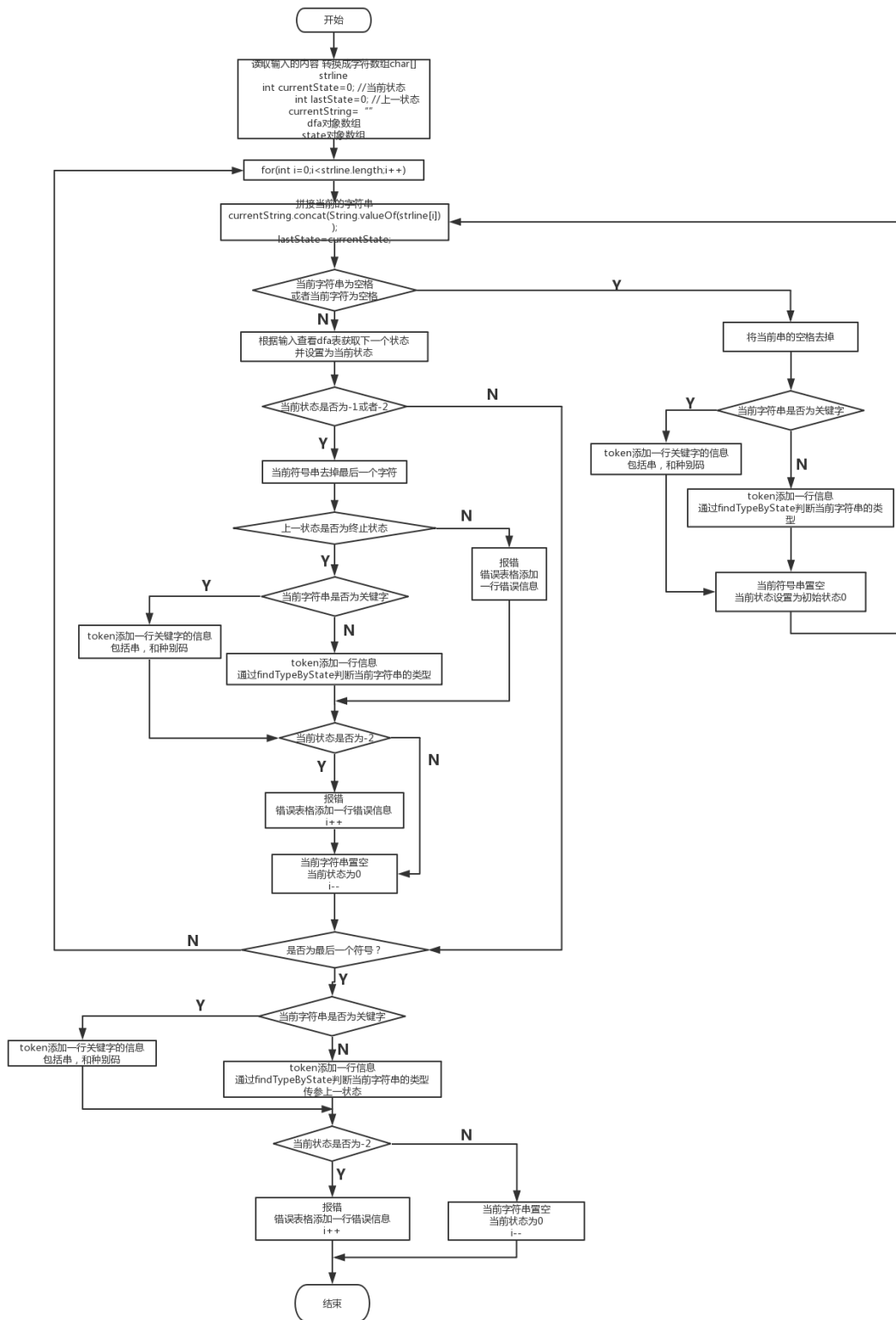
#### ✓ 主要功能函数说明

1. **void Analysis(String s):** 词法分析的主要函数，在这里进行识别各种词类，基于 DFA 的转换表和状态实现；一些要求：注释里可以含非法字符和换行，将注释中换行识别为空格等。
2. **int ChangeState(char currentChar, int currentState, DFA[] dfa):** DFA 状态跳转函数，实现从一个状态转移到另一个状态；
3. **String TakenID(String a, String b)/ String AttrID(String b, String c):** 此函数返回种别码/属性值；
4. **boolean IsKeyWord(String a):** 判断是否是关键字，是返回 true，不是返回 false；
5. **DFA[] addDFA():** 此函数存储 DFA 转换表，设置状态类型（大类）；
6. **void initialize()/void initialize2()/void initialize3()/void initialize4():** 可视化函数，生成四个图表；
7. **String[][] getData(File file, int ignoreRows):** 读取 Excel 的内容，第一维数组存储的是一行中格列的值，二维数组存储的是多少个行。file 读取数据的源 Excel ignoreRows 读取数据忽略的行数。
8. **String getType(int t):** 已知状态，返回此状态的具体类型（详细类型）；
9. **boolean isFinish(int t):** 已知状态，返回此状态是否是终止状态，是返回 true，不是返回 false。



✓ 程序核心部分的程序流程图

参考文献流程图画法如下:



要求：对如下内容展开描述。

(1) 系统实现过程中遇到的问题；

1. 种别码设计问题

显而易见，将种别码设计为数字最简单，我最后改进用字母表示的，再结合属性值即可。用许多 if-else 或者 case 是非常麻烦的，我通过设计一个数组来存储所有的种别码，利用循环即可，要简便许多。

2. 词法分析器实现过程的方法

我在搜索许多资料发现，很多人用 if-else 或 case 来实现词类识别，类似于分类讨论的穷举，这是一个不完善的方法。通过资料学习，我找到了存储 DFA 转换表和状态表的方法来实现词法分析，这样更简便，涵盖的内容更广。

3. 存储 DFA 状态采用何种方法

JAVA 中，对于二元组存储很便利，比如 Map, List 等，对于多元组，就要通过新建一个对象来实现存储，这样更方便。

4. 无法读取 excel 获得数据

通过学习别人的 demo 和博客，才解决。

5. 最后一个字符不显示

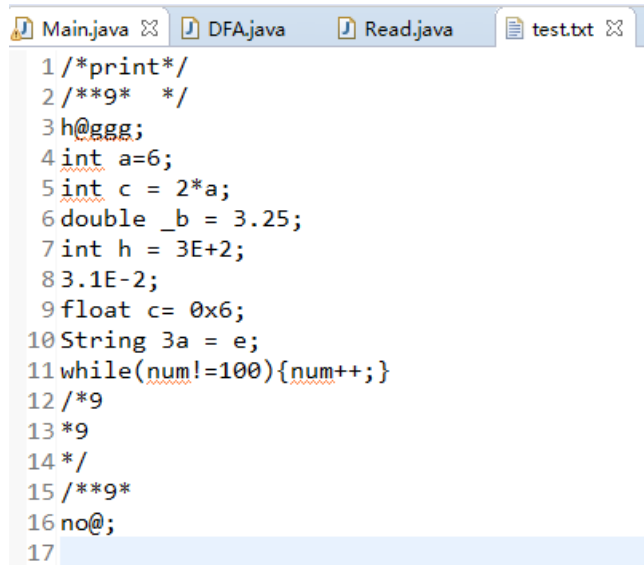
增加一个判断是否为最后一个字符，根据上一状态判断最后一个字符串的类型就可以解决。

6. 图表画不出来或者很奇怪

这个 jar 包第一次接触，需要自行导入 eclipse，通过一些 demo 熟悉一些写法，然后画出满意的图。

(2) 针对某测试程序输出其词法分析结果：

测试内容如下：



```
1 /*print*/
2 /**9* */
3 h@ggg;
4 int a=6;
5 int c = 2*a;
6 double _b = 3.25;
7 int h = 3E+2;
8 3.1E-2;
9 float c= 0x6;
10 String 3a = e;
11 while(num!=100){num++;}
12 /*9
13 *9
14 */
15 /**9*
16 no@;
17
```

符号表及 token 序列结果如下图：

# TOKEN TABLE

字符串	类别	种别码	value
/*print*/	注释	NOTE	-
/**g**/	注释	NOTE	-
h	标识符	IDN	h
@	非法字符	无	
ggg	标识符	IDN	ggg
,	界符	BOUND	4
int	关键字	INT	-
a	标识符	IDN	a
=	界符	BOUND	3
6	整数	CONST	6
,	界符	BOUND	4
int	关键字	INT	-
c	标识符	IDN	c
=	界符	BOUND	3
2	整数	CONST	2
*	运算符	OP	15
a	标识符	IDN	a
,	界符	BOUND	4
double	关键字	DOUBLE	-
_b	标识符	IDN	_b
=	界符	BOUND	3
3.25	小数	CONST	3.25
,	界符	BOUND	4
int	关键字	INT	-
h	标识符	IDN	h
=	界符	BOUND	3
3E+2	科学计数法常数	CONST	3E+2
,	界符	BOUND	4
3.1E-2	科学计数法常数	CONST	3.1E-2
,	界符	BOUND	4
float	关键字	FLOAT	-
c	标识符	IDN	c
=	界符	BOUND	3
0x6	十六进制数	HEXAD	0x6
,	界符	BOUND	4
String	标识符	IDN	String
3a	格式不正确	无	
	初始		
=	界符	BOUND	3
e	标识符	IDN	e
,	界符	BOUND	4
while	关键字	WHILE	-
(	界符	BOUND	7

num	标识符	IDN	num
!=	运算符	OP	6
100	整数	CONST	100
)	界符	BOUND	8
{	界符	BOUND	9
num	标识符	IDN	num
++	运算符	OP	18
;	界符	BOUND	4
}	界符	BOUND	10
/*9*9*/	注释	NOTE	-
/*9* no@;	注意注释未封闭	无	

DFA 转换表如下：

DFA转换表																			
1	abcd...	ghijk...x	0	89	1234...	E	.	/	*	+	-	=	&		><!^				
0	1	1	1	30	2	1	28	8	21	13	16	19	23	25	27				
1	1	1	1	1	1	1													
2	-2	-2	-2	2	2	2	5	3											
3				4	4	4													
4				4	4	4	5												
5				7	7	7				6	6								
6				7	7	7													
7				7	7	7													
8									9			12							
9	9	9	9	9	9	9	9	9	10	9	9	9	9	9	9	9			
10	9	9	9	9	9	9	9	9	11	10	9	9	9	9	9	9			
11																			
12																			
13										14		15							
14																			
15																			
16											17	18							
17																			
18																			

DFA 状态表如下：

DFA状态		
状态	类型	是否终止状态
0	初始	true
1	标识符	true
2	整数	true
3	常数	false
4	小数	true
5	常数	false
6	常数	false
7	科学计数法常数	true
8	运算符	true
9	注释	false
10	注释	false
11	注释	true
12	运算符	true
13	运算符	true
14	运算符	true
15	运算符	true
16	运算符	true
17	运算符	true
18	运算符	true

固定的词类信息如下：

固定信息			
词	单词类型	种别码	属性值
unsigned	关键字	UNSIGNED	-
goto	关键字	GOTO	-
switch	关键字	SWITCH	-
case	关键字	CASE	-
default	关键字	DEFAULT	-
extern	关键字	EXTERN	-
register	关键字	REGISTER	-
sizeof	关键字	SIZEOF	-
typedef	关键字	TYPDEF	-
volatile	关键字	VOLATILE	-
NULL	关键字	NULL	-
>	运算符	OP	1
>=	运算符	OP	2
<	运算符	OP	3
<=	运算符	OP	4
==	运算符	OP	5
!=	运算符	OP	6
	运算符	OP	7
&	运算符	OP	8

(3) 输出针对此测试程序对应的词法错误报告；

## ERROR TABLE

出错符号	出错地方	出错原因
@	第21字符	非法字符
3a	第111字符	格式不正确
/**9* no@;	第162字符	注意注释未...

(4) 对实验结果进行分析。

此次实验错误信息全部检测出：

@为非法字符；

/\*\*9\* no@;注释不完全；

而且，科学记数法常数 3E+2、十六进制数 0x6、注释/\*print\*/等也均识别出；  
关键字 int、while，运算符++、\*，以及各种常量均识别正确。

此次实验结果全对。

整体运行结果效果如下：

选择文件

测试

```
/*print*/
print("Hello,World");
@
int a=6;
int c = 2*a;
double _b = 3.25;
int h = 36*2;
float c = (a);
while(num=100){num++}
/*a*/
```

字符串	类别	种别码	value
/*print*/	注释	NOTE	-
print	关键字	PRINT	-
@	符号	BOUND	7
int	符号	BOUND	11
a	标识符	IDN	Hello
=	符号	BOUND	2
Word	标识符	IDN	Word
3	运算符	OP	11
2	符号	BOUND	11
h	符号	BOUND	8
2	符号	BOUND	4
@	非法字符	IL	-
int	关键字	INT	-
a	标识符	IDN	a
=	符号	BOUND	3
5	整数	CONST	5
4	符号	BOUND	4
int	关键字	INT	-
c	标识符	IDN	c
=	符号	BOUND	3
2	整数	CONST	2
*	运算符	OP	15

出错符号	出错地方	出错原因
@	第30字符	非法字符
/*a*/	第121字符	非法字符

固定信息

词	单词类型	种别码	属性值
char	关键字	CHAR	-
long	关键字	LONG	-
short	关键字	SHORT	-
float	关键字	FLOAT	-
double	关键字	DOUBLE	-
const	关键字	CONST	-
boolean	关键字	BOOLEAN	-
void	关键字	VOID	-
null	关键字	NULL	-
false	关键字	FALSE	-
true	关键字	TRUE	-
enum	关键字	ENUM	-
int	关键字	INT	-
do	关键字	DO	-
while	关键字	WHILE	-
if	关键字	IF	-
else	关键字	ELSE	-
for	关键字	FOR	-
when	关键字	WHEN	-

DFA转换表

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

DFA状态

状态	类型	是否终止状态
0	初始	true
1	标识符	true
2	整数	true
3	小数	false
4	小数	true
5	整数	false
6	整数	false
7	科学计数法整数	true
8	运算符	true
9	注释	false
10	注释	false
11	注释	true
12	运算符	true
13	运算符	true
14	运算符	true
15	运算符	true
16	运算符	true
17	运算符	true
18	运算符	true

指导教师评语:

日期: