**类SNL语言词法分析器说明文档**

1. **类SNL语言词法分析器的作用**

词法分析器的主要任务是读入源程序的输入字符、将它们组成词素，生成并输出一个词法单元序列（一个token链），每个词法单元对应于一个词素。

1. **类SNL语言词法定义**

**a)语言的字符表**

<字符表> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y |z |

A| B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |

W | X| Y | Z |

0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

+ | - | \* | / | = | == | < | > | < = | > = | < > | ( | ) | [ | ] | { | } | , | . | ; | EOF | 空白字符

注：在程序中，英文字母区分大小写；保留字只能由小写字母组成。

**b)单词的巴科斯范式**

单次符号分类如下：

* 标识符 ID
* 关键字 （标识符的子集含有保留字） int、real、if、then、else、while……
* 数值 （包含无符号整数、小数、实数）
* 单字符分界符 (、)、[、]、{、}、;、,EOF |，空白字符
* 操作符 +、-、\*、/、=、==、<、>、<=、>=、<>

上述各类符号的巴科斯范式如下：

< 标识符 > ::= 字母 { 字母 | 数字 }

< 无符号整数 > ::= 数字 { 数字 }

<小数> ::= 数字 { 数字 } { . }{ 数字 }

<实数> ::= 数字 { 数字 }{ E }{ 数字 }{ 数字 } |

数字 { 数字 }{ E }{ + | - }{ 数字 } |

数字 { 数字 }{ . }{ 数字 }{ E }{ 数字 }{ 数字 }

< 单字符分界符 > ::= ( | ) | [ | ] | { | } | ; | , | EOF | 空白字符

<操作符> ::= + | - | \*| / | = | == | < | > | < = | > = | < >

<字母> ::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t |

u | v | w | x | y |z | A| B | C | D | E | F | G | H | I | J | K | L |

M | N | O | P | Q | R | S | T | U |V | W | X| Y | Z

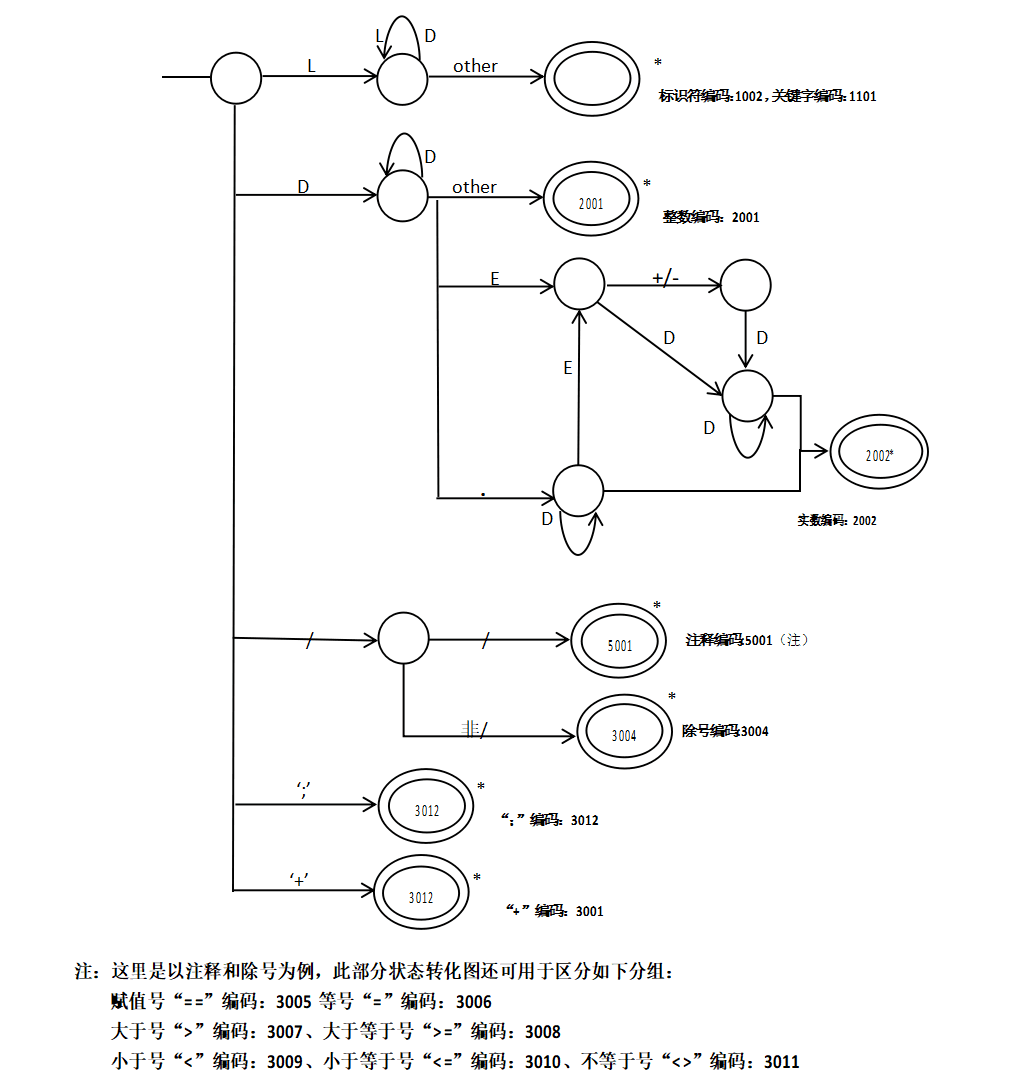
<数字> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

**c)类SNL语言token设计编码**

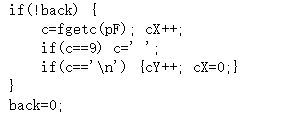
|  |  |  |
| --- | --- | --- |
| **编码** | **语义值** | **token** |
| **1002** | 标识符 | id |
| **1101** | 关键字 | int |
| **1102** | 关键字 | real |
| **1201** | 关键字 | if |
| **1202** | 关键字 | then |
| **1203** | 关键字 | else |
| **1204** | 关键字 | while |
| **1205** | 关键字 | begin |
| **1206** | 关键字 | end |
| **1207** | 关键字 | case |
| **1208** | 关键字 | and |
| **1209** | 关键字 | or |
| **1210** | 关键字 | not |
| **2001** | 整数 | num |
| **2002** | 实数 | num |
| **3001** | 加号 | + |
| **3002** | 减号 | - |
| **3003** | 乘号 | \* |
| **3004** | 除号 | / |
| **3005** | 赋值 | == |
| **3006** | 等于 | = |
| **3007** | 大于 | > |
| **3008** | 大于等于 | >= |
| **3009** | 小于 | ＜ |
| **3010** | 小于等于 | ＜= |
| **3011** | 不等于 | != |
| **3012** | 分号 | ; |
| **3013** | 左括号 | ( |
| **3014** | 右括号 | ) |
| **3015** | 左区号 | { |
| **3016** | 右区号 | } |
| **3017** | 左下标号 | [ |
| **3018** | 右下标号 | ] |
| **3019** | 逗号 | , |

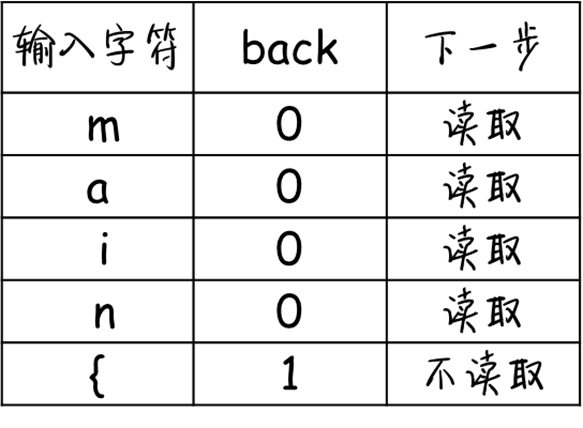
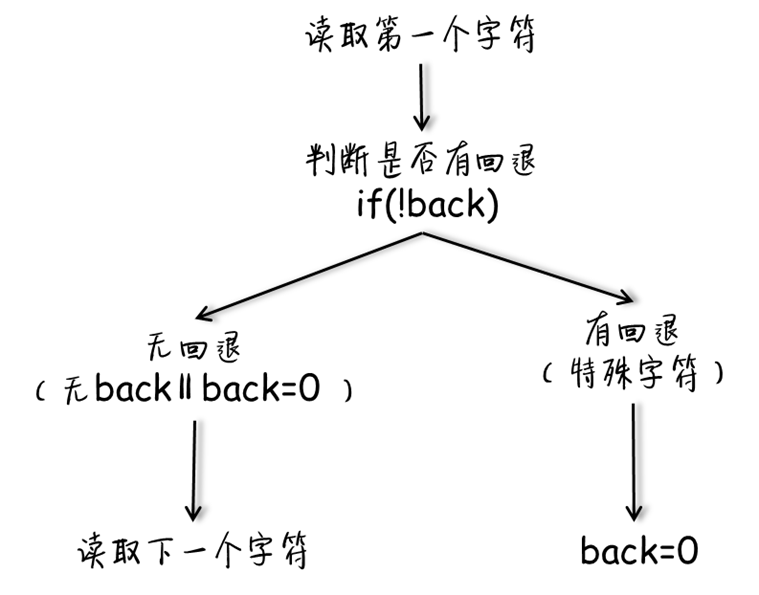
**d）类SNL语言状态转化图**

下面给出能够识别上述token的状态转化图，如图所示：



1. **类SNL语言词法分析器具体实现过程**
   1. **回退**





* 1. **注释与除号**

对于注释 // 与除号 / 这种类似的符号，会进行判断

在状态图中已经展示

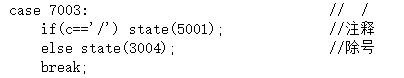
相同情况的还有

< <=

> >=

= ==

判断方法相同，不多做赘述

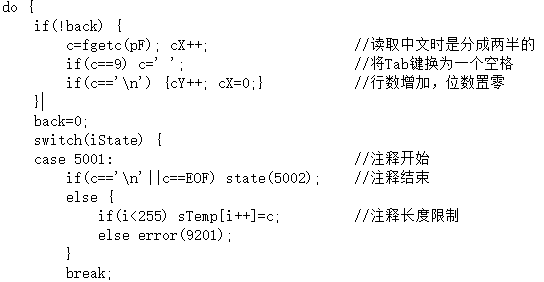


比对后，是除号，进入7003状态

如果继续输入的是/，判定为注释，输出5001，并判断注释结束位置

否则判定为除号，输出3004

* 1. **换行与空格**

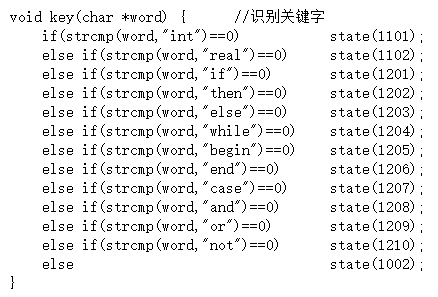


首先获取字符的长度

如果长度为9，则判定为tab键，换为一个空格

如果字符串等于换行，则行数增加，单词位数置零

* 1. **保留字的处理（if/else）**



对于输入判别到的以字母开头的关键字，用自动机识别，判定为保留字的一律处理添加状态，判别非保留字的处理为标识符。

* 1. **行列处理**

**case 1001:**

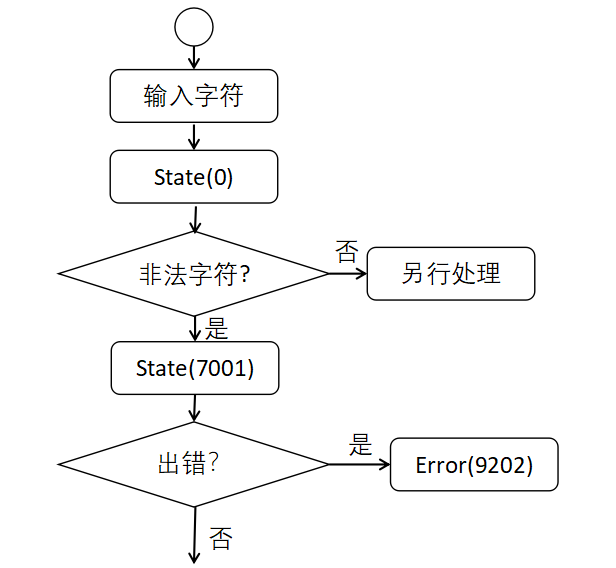
**sTemp[i]='\0'; back=1;**

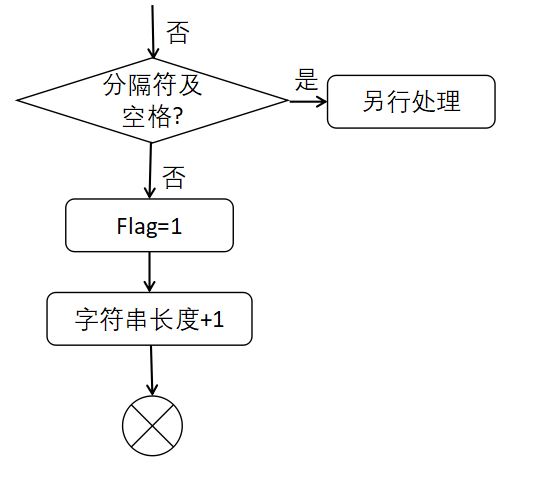
**if(tipsLevel>2) printf("State 1001 @ %d,%d: 获得尚未识别的标识符,cL,cW\n");**

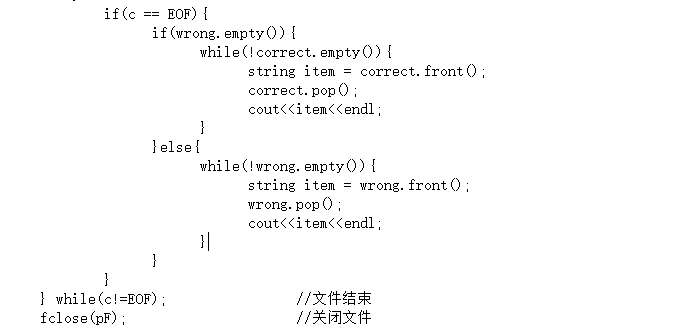
**key(sTemp); state(0); break;**

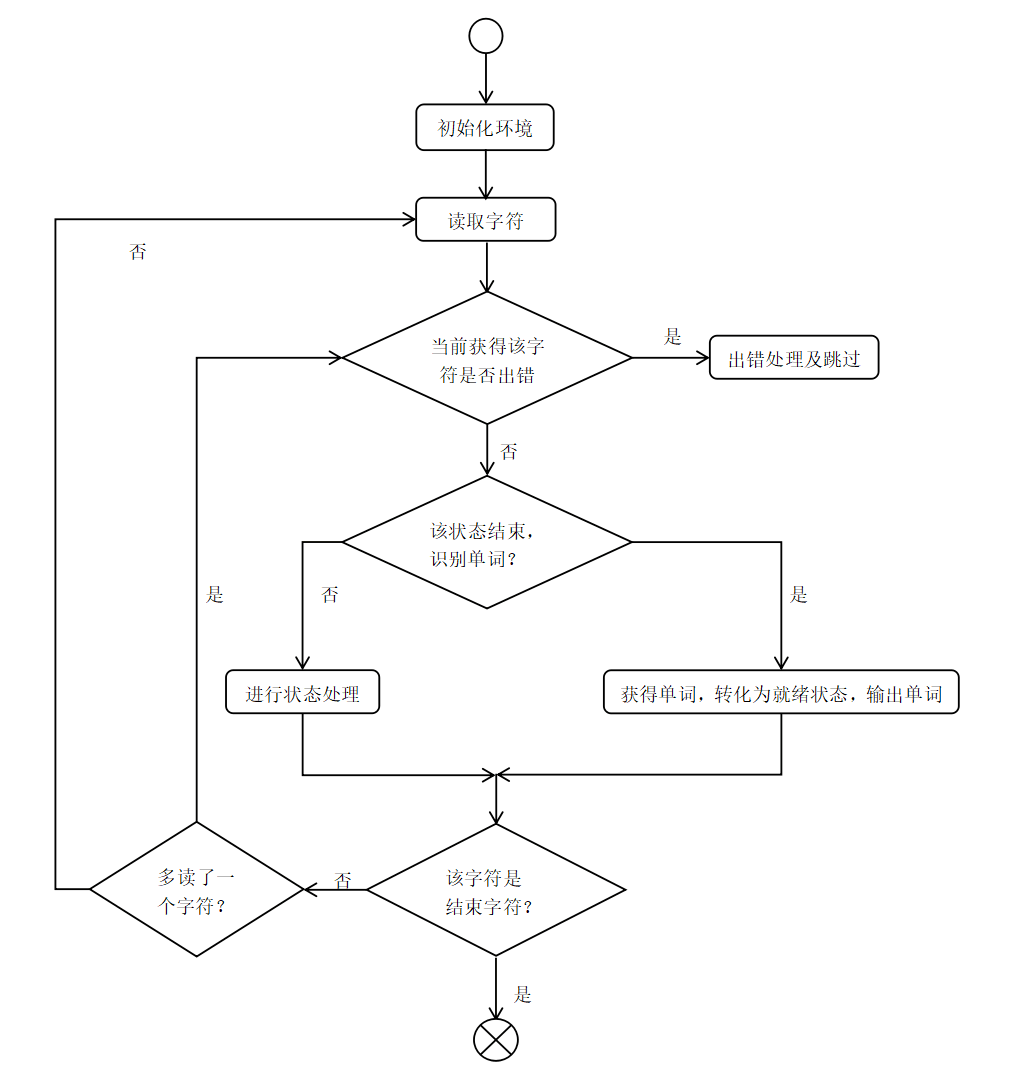
对于以上这一行代码，起始字符前至少有四个空格，每个空格及获取到的字符均会进行判断。获取到tab键则判定为空格，获取到回车自动增加一行，获取到；语句结束，如有回车键，继续增加一行。

* 1. **词法错误判断处理**



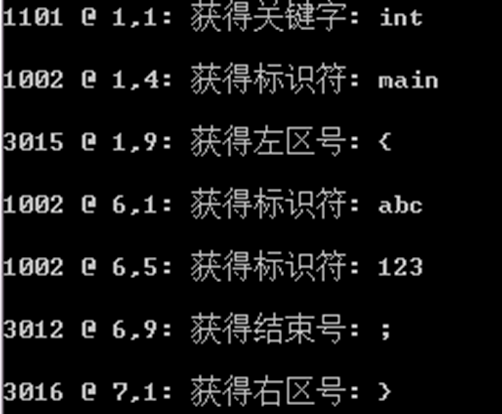


* 1. **结束判断**
  2. **程序流程图**

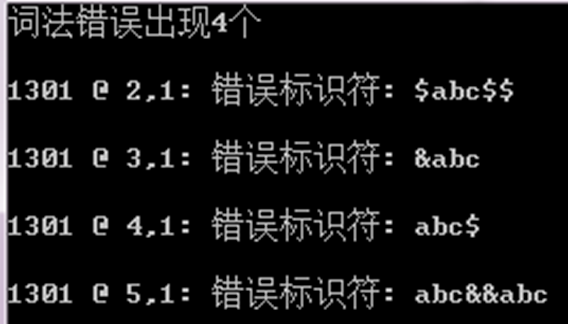


1. **类SNL语言词法分析器测试**

**正确**



**词法错误**



**错误**





1. **总结**

**困难**

小组代码能力略有缺失，想法不能统一，意见有时相悖。制作耗时较长，更新次数较多。

**不足**

由于技术困难问题，并未对词法错误进行处理；对于一些处理还并不是十分完善。

**优点**

小组人员能力不足但胜在认真完成分配任务，积极主动；想法过多但能挑选出比较好的想法，合作过程比较愉快。