

# 词法分析

## (2. 手写词法分析器)

魏恒峰

hfwei@nju.edu.cn

2022 年 11 月 14 日 (周一)





## 手写词法分析器

<i>digit</i>	→	[0-9]
<i>digits</i>	→	<i>digit</i> <sup>+</sup>
<i>number</i>	→	<i>digits</i> ( . <i>digits</i> ) ? ( E [ + - ] ? <i>digits</i> ) ?
<i>letter</i>	→	[A-Za-z]
<i>id</i>	→	<i>letter</i> ( <i>letter</i>   <i>digit</i> ) *
<i>if</i>	→	if
<i>then</i>	→	then
<i>else</i>	→	else
<i>relop</i>	→	<   >   <=   >=   =   <>

<i>digit</i>	→	[0-9]
<i>digits</i>	→	<i>digit</i> <sup>+</sup>
<i>number</i>	→	<i>digits</i> ( . <i>digits</i> )? ( E [+-]? <i>digits</i> )?
<i>letter</i>	→	[A-Za-z]
<i>id</i>	→	<i>letter</i> ( <i>letter</i>   <i>digit</i> )*
<i>if</i>	→	if
<i>then</i>	→	then
<i>else</i>	→	else
<i>relop</i>	→	<   >   <=   >=   =   <>

DragonLexerGrammar.g4

识别字符串  $s$  中符合**特定**词法单元模式的**前缀**词素

识别字符串  $s$  中符合**某种**词法单元模式的**所有**词素

识别字符串  $s$  中符合**特定**词法单元模式的前缀词素

**分支:** 先判断属于哪一类, 然后进入特定词法单元的前缀词素匹配流程

识别字符串  $s$  中符合**某种**词法单元模式的前缀词素

识别字符串  $s$  中符合**某种**词法单元模式的所有词素

识别字符串  $s$  中符合**特定**词法单元模式的**前缀**词素

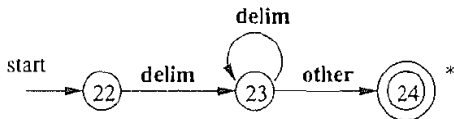
**分支:** 先判断属于哪一类, 然后进入特定词法单元的前缀词素匹配流程

识别字符串  $s$  中符合**某种**词法单元模式的**前缀**词素

**循环:** 返回当前识别出来的词法单元与词素, 继续识别下一个前缀词素

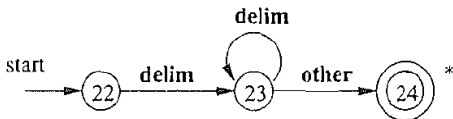
识别字符串  $s$  中符合**某种**词法单元模式的**所有**词素

## 用于识别 ws 的状态转移图



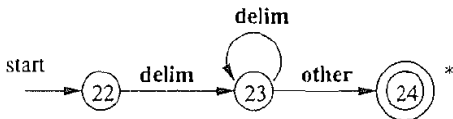


## 用于识别 ws 的状态转移图



\*: 识别出的空白符**不包含**当前 peek 指向的字符

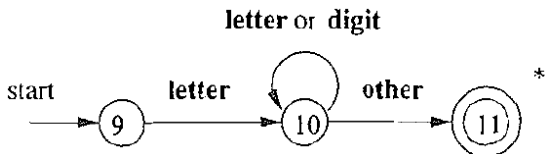
用于识别 ws 的状态转移图



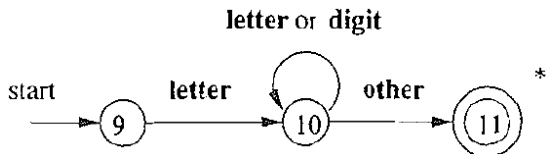
※: 识别出的空白符**不包含**当前 peek 指向的字符

22: 碰到 other 怎么办?

## 用于识别 **id** 的状态转移图

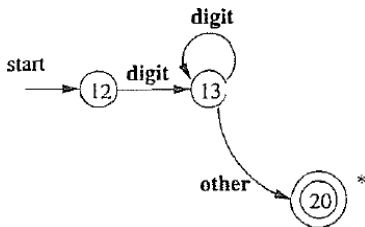


用于识别 **id** 的状态转移图

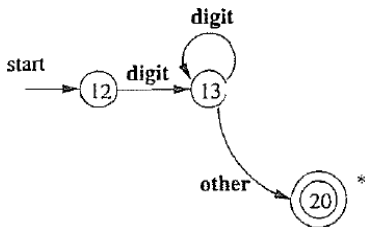


9: 碰到 other 怎么办?

用于识别 **int** 的状态转移图

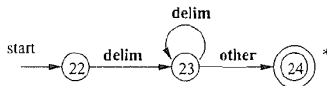


用于识别 `int` 的状态转移图

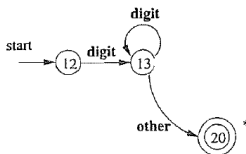


12: 碰到 `other` 怎么办?

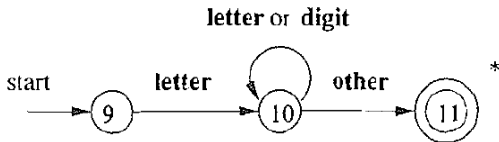
识别字符串  $s$  中符合**某种**词法单元模式的前缀词素 (NEXTTOKEN())



ws: 空白符



int: 整数

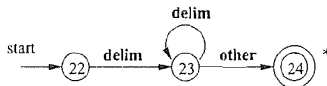


id: 标识符

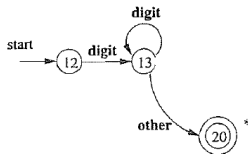


错误处理模块

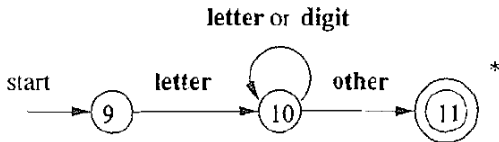
识别字符串  $s$  中符合**某种**词法单元模式的**前缀**词素 (NEXTTOKEN())



ws: 空白符



int: 整数



id: 标识符



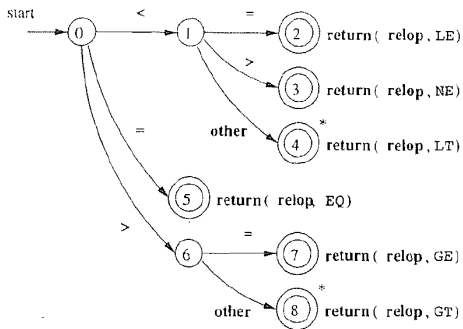
错误处理模块

**关键点:** 合并 22, 12, 9, 根据**下一个字符**即可判定词法单元的类型

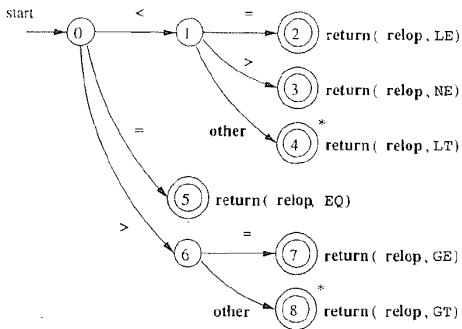
否则, 调用错误处理模块 (对应 other), 报告**该字符有误**, 并忽略该字符



## 用于识别 **relop** 的状态转移图

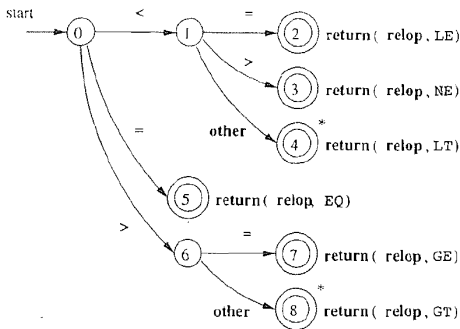


## 用于识别 **relop** 的状态转移图



**“最长优先原则”**: 例如, 识别出 `<=`, 而不是 `<` 与 `=`

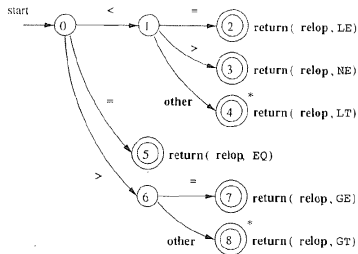
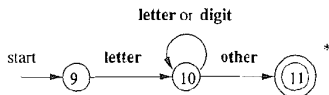
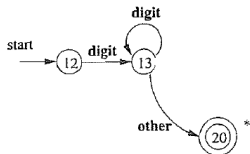
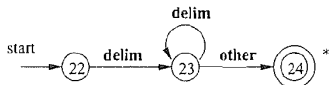
## 用于识别 **relop** 的状态转移图



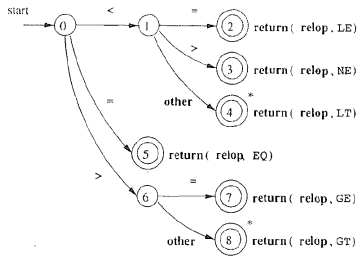
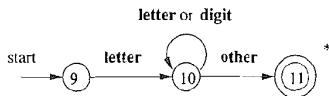
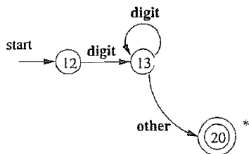
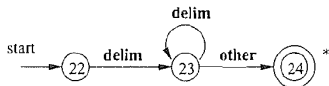
“最长优先原则”：例如，识别出  $\leq$ ，而不是  $<$  与  $=$

0：碰到 other 怎么办？

识别字符串  $s$  中符合**某种**词法单元模式的前缀词素 (NEXTTOKEN())



识别字符串  $s$  中符合**某种**词法单元模式的前缀词素 (NEXTTOKEN())



**关键点:** 合并 22, 12, 9, 0, 根据**下一个字符**即可判定词法单元的类型

否则, 调用错误处理模块 (对应 other), 报告**该字符有误**, 并忽略该字符

但是, 词法分析器的设计并没有这么容易



ws if else id int relop

根据下一个/两个字符即可判定词法单元的类型

每个状态转移图的每个状态要么是接受状态, 要么带有 other 边

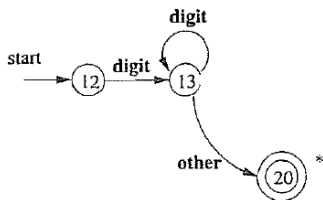
ws if else id int relop

根据下一个/两个字符即可判定词法单元的类型

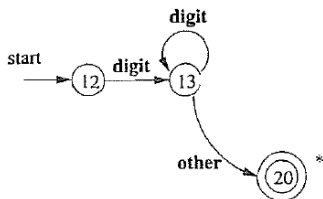
每个状态转移图的每个状态要么是接受状态, 要么带有 other 边

如何同时识别 int、real 与 sci?

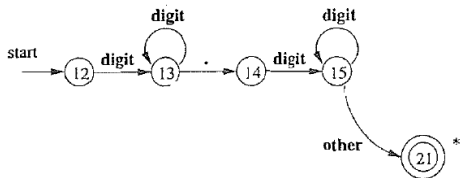




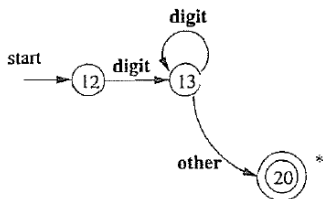
int: 整数



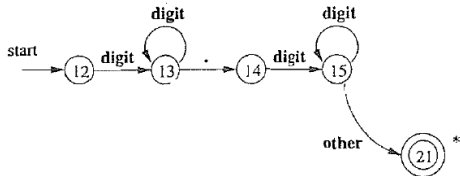
int: 整数



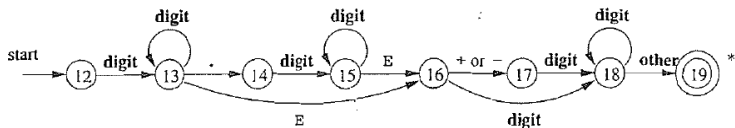
real: 浮点数 (无科学计数法)  
(不识别 2.)



**int:** 整数



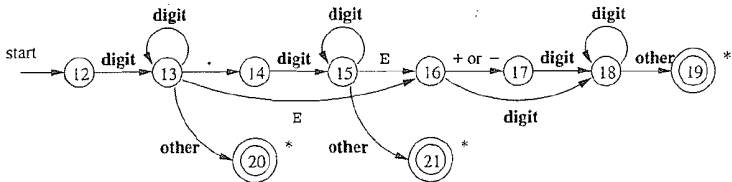
**real:** 浮点数 (无科学计数法)  
(不识别 2.)



**sci:** 带科学计数法的浮点数  
(2.99792458E8 3E8)

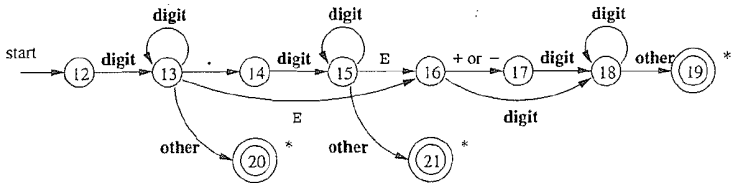
**num:** 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]

num: 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]



19, 20, 21 : 代表了不同的数字类型

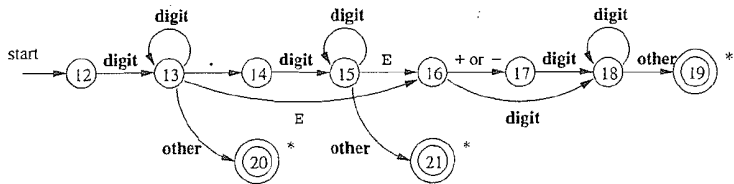
num: 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]



19, 20, 21 : 代表了不同的数字类型

12 : 碰到 other 怎么办?

num: 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]

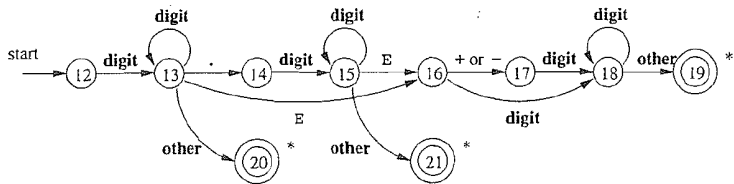


19, 20, 21 : 代表了不同的数字类型

12 : 碰到 other 怎么办?

(尝试其它词法单元或进入错误处理模块)

num: 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]



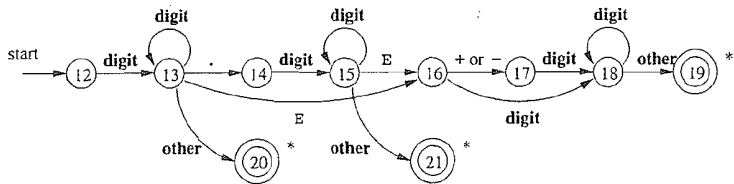
19, 20, 21 : 代表了不同的数字类型

12 : 碰到 other 怎么办? (尝试其它词法单元或进入错误处理模块)

14, 16, 17 : 碰到 other 怎么办?



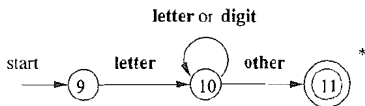
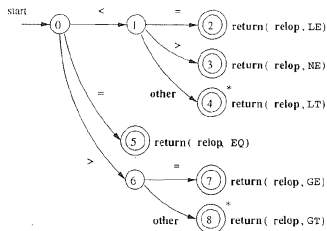
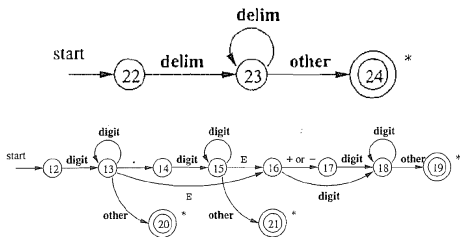
num: 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]

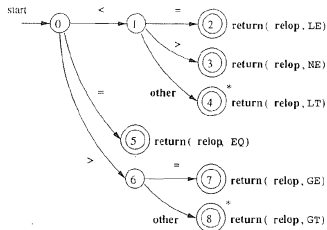
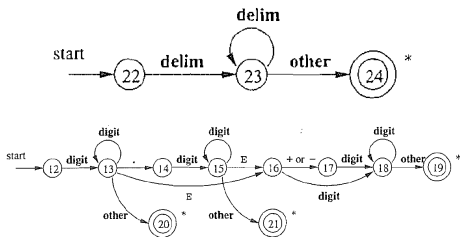


19, 20, 21 : 代表了不同的数字类型

12 : 碰到 other 怎么办? (尝试其它词法单元或进入错误处理模块)

14, 16, 17 : 碰到 other 怎么办? (回退, 寻找**最长匹配**)





**关键点:** 合并 22, 12, 9, 0, 根据**下一个字符**即可判定词法单元的类型  
否则, 调用错误处理模块 (对应 other), 报告**该字符有误**, 忽略该字符。

注意, 在 **real** 与 **sci** 中, 有时需要**回退**, 寻找最长匹配。

Thank  
You!



Office 926

hfwei@nju.edu.cn