词法分析
(2. 手写词法分析器)

魏恒峰

hfwei@nju.edu.cn

2024 年 03 月 08 日 (周五)

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

DragonLexerRules.g4

```
INT : DIGITS ;
// here "2." is an invalid REAL
REAL : DIGITS ('.' DIGITS)? ;
// both "2.99792458E8" and "3E8" are valid SCI
SCI : DIGITS ('.' DIGITS)? ([eE] [+-]? DIGITS)? ;
```

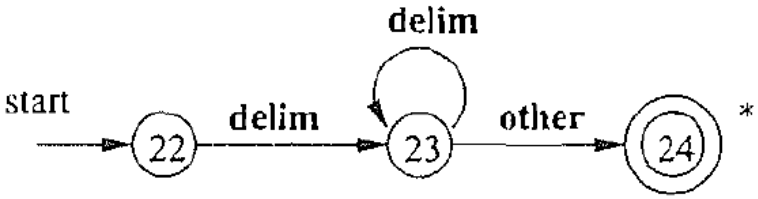
Token.java　　TokenType.java

dragon0.txt

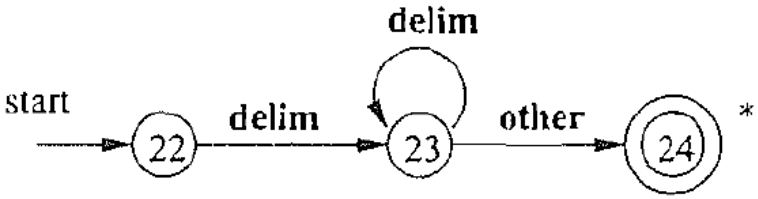
向前看、向前走、调整状态

记录来时最长匹配、无路可走便回头

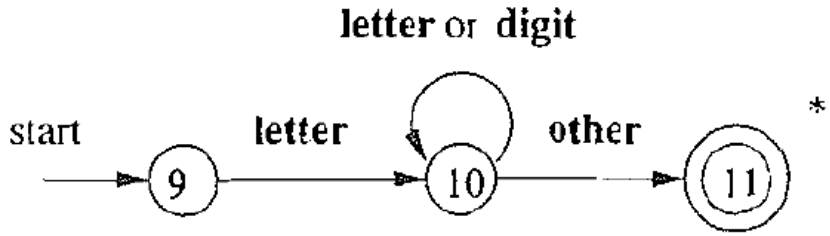
```
nextToken()

while (nextToken())
```



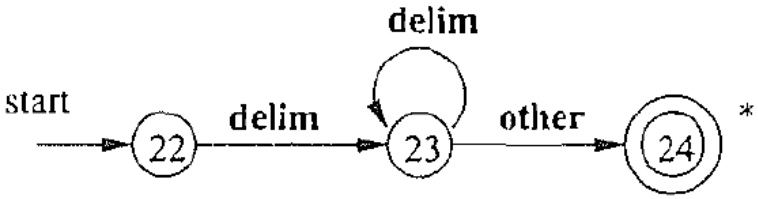
ws: 空白符


delim
start
delim
other
22
23
24
*

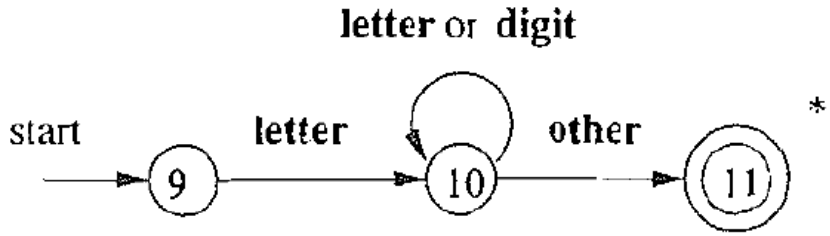
ws: 空白符


letter or digit
start
letter
other
9
10
11
*

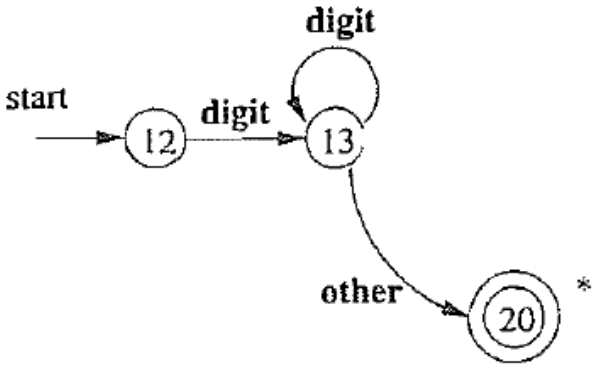
id: 标识符



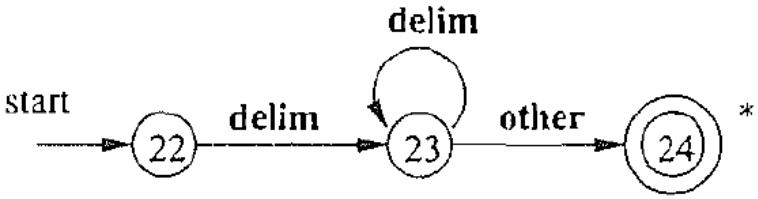
ws: 空白符



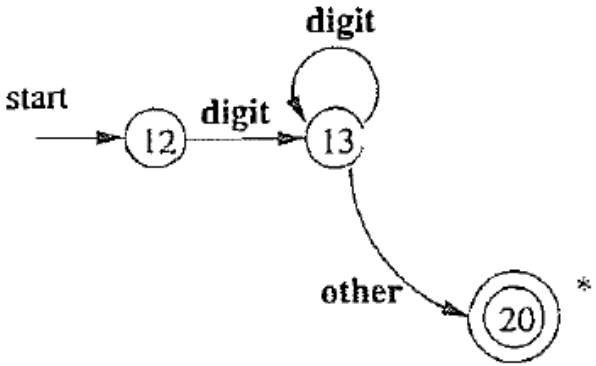
id: 标识符



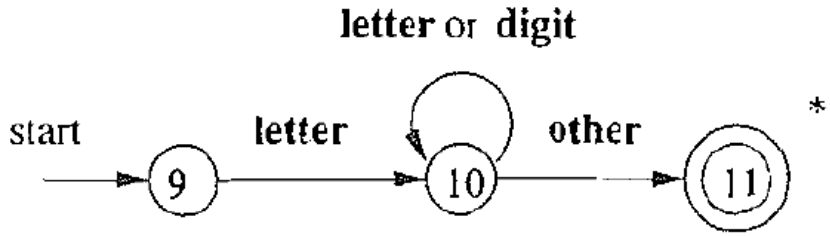
int: 整数



ws: 空白符

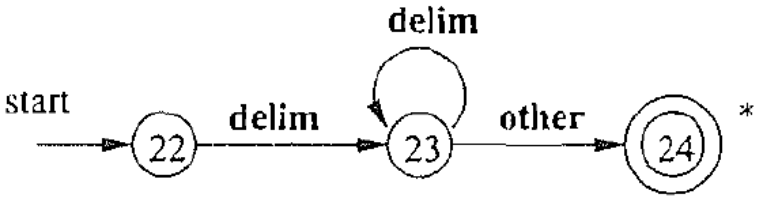


int: 整数

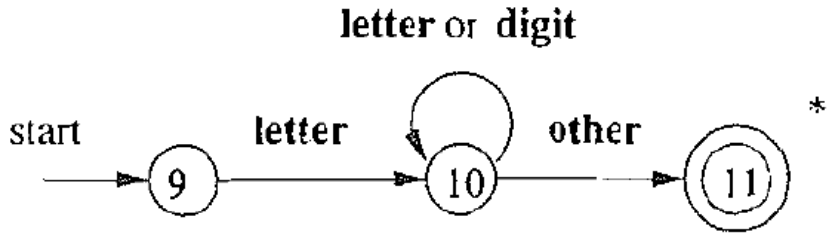


id: 标识符

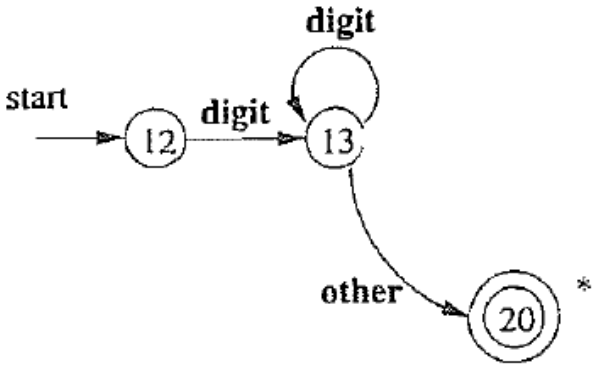
错误处理模块



ws: 空白符



id: 标识符



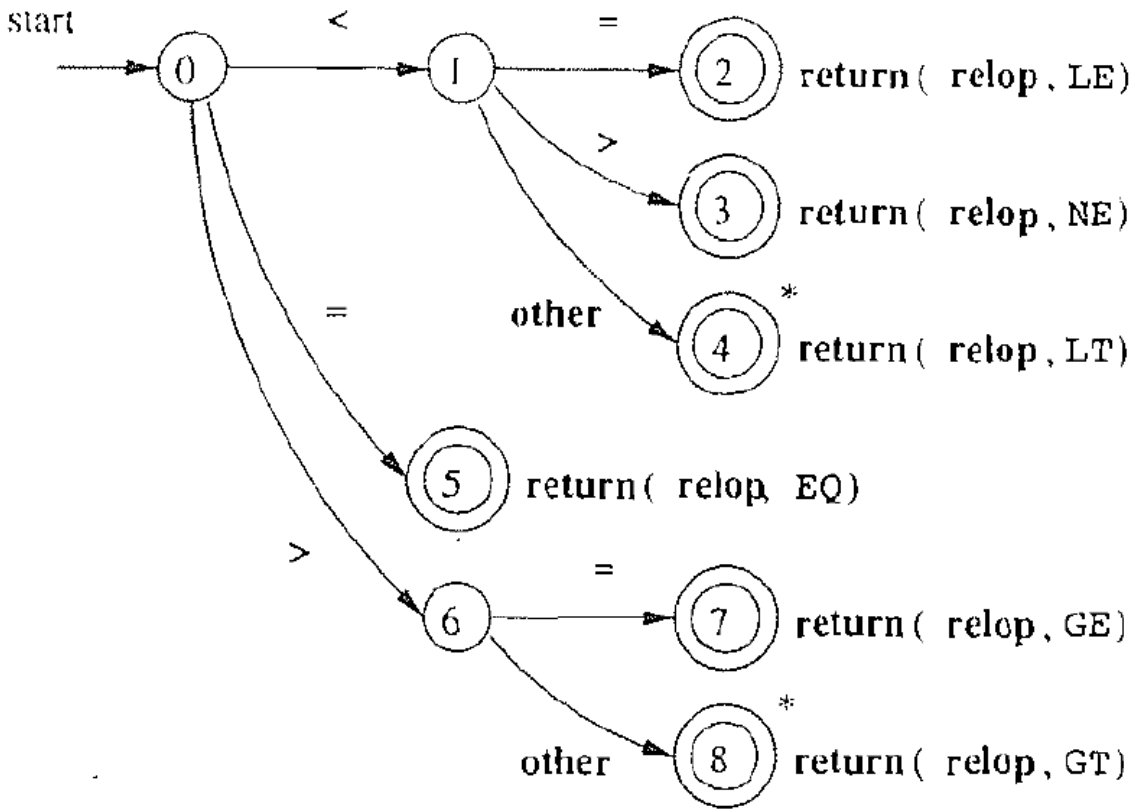
int: 整数

错误处理模块

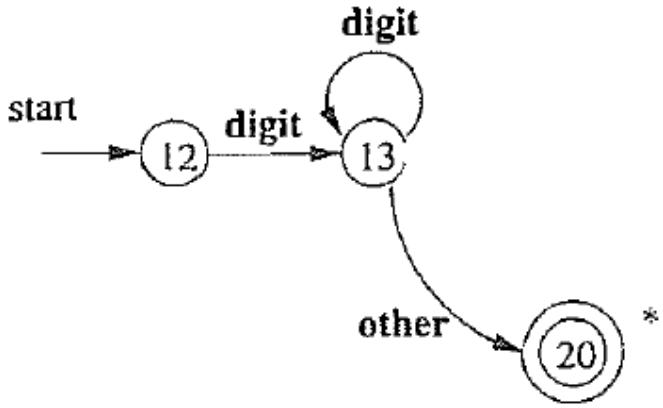
关键点: 合并 22, 12, 9, 根据**下一个字符**即可判定词法单元的类型

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

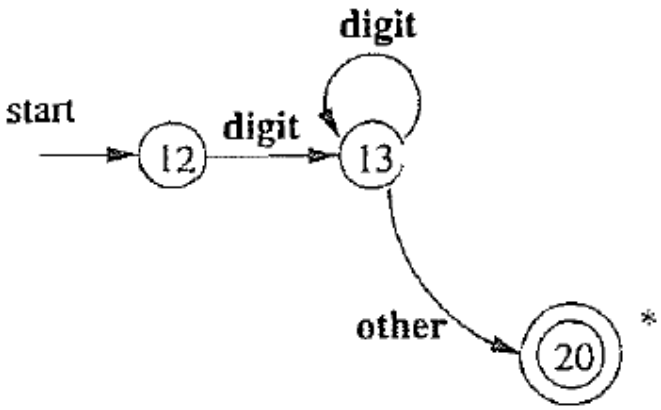
用于识别 relop 的状态转移图



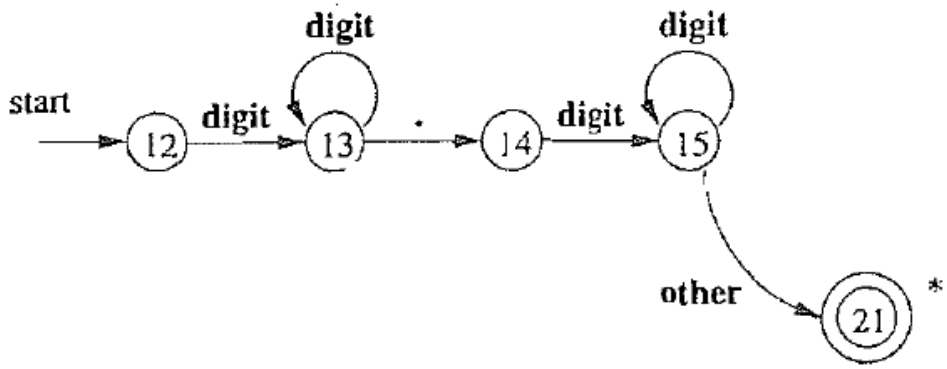
难点: 如何区分 **int**、**real** 与 **sci**?



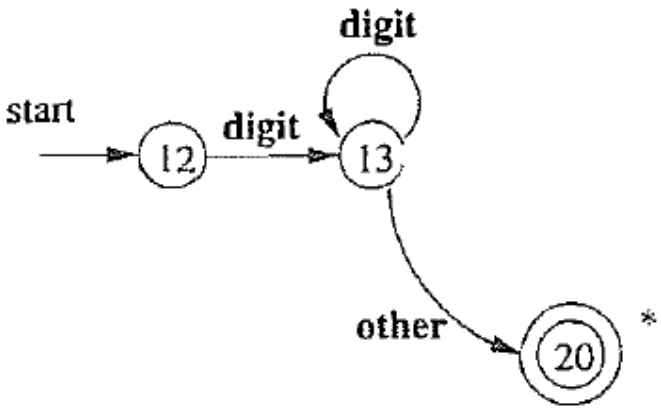
int: 整数



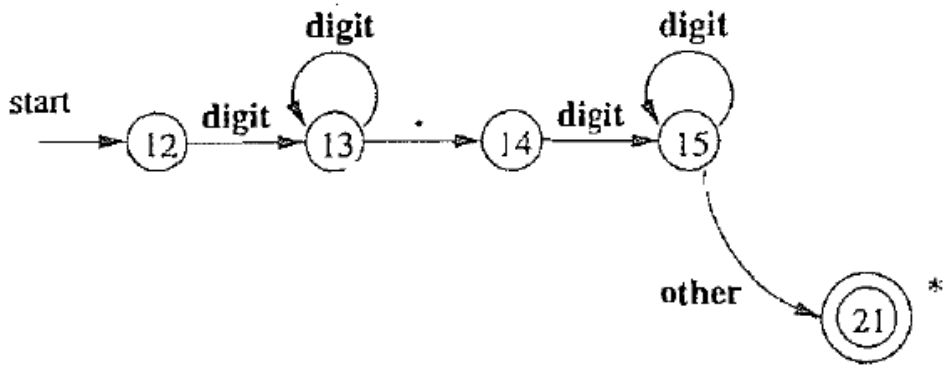
int: 整数



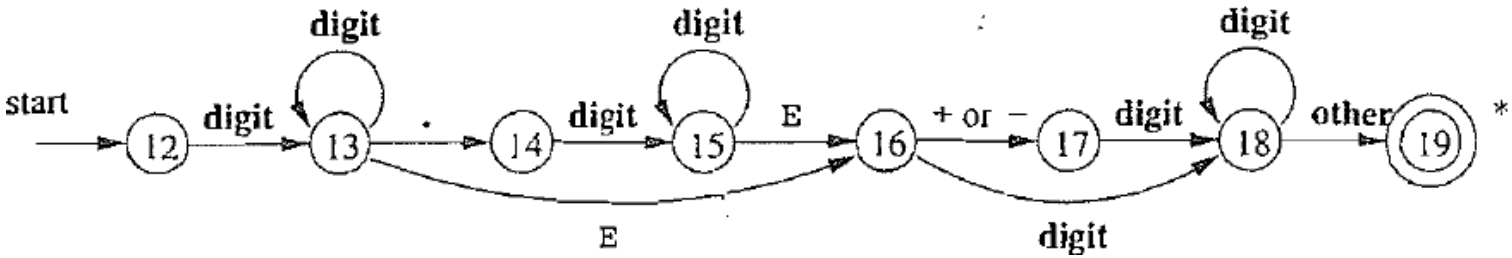
real: 浮点数 (无科学计数法)
(不识别 2.)



int: 整数

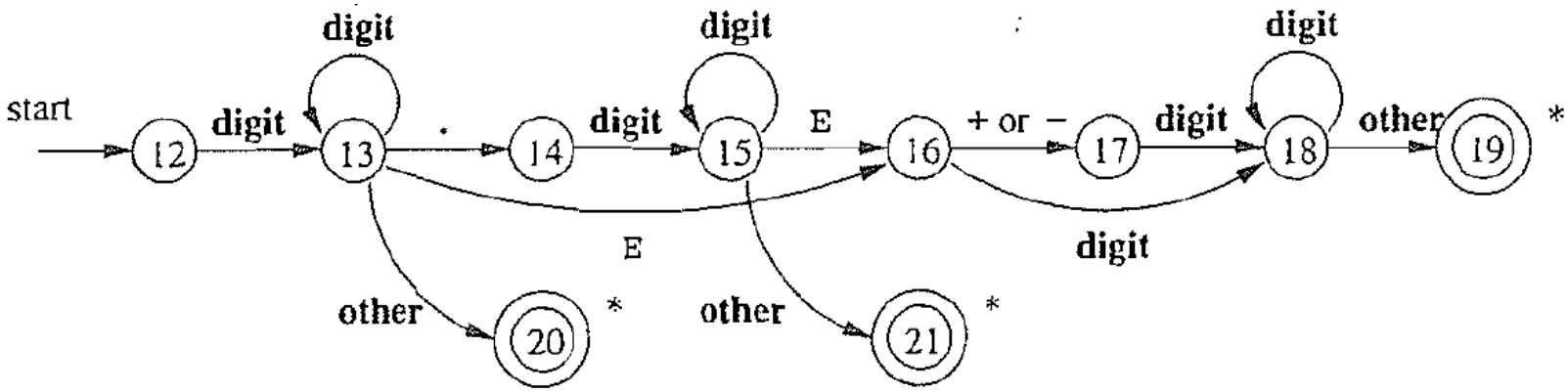


real: 浮点数 (无科学计数法)
(不识别 2.)



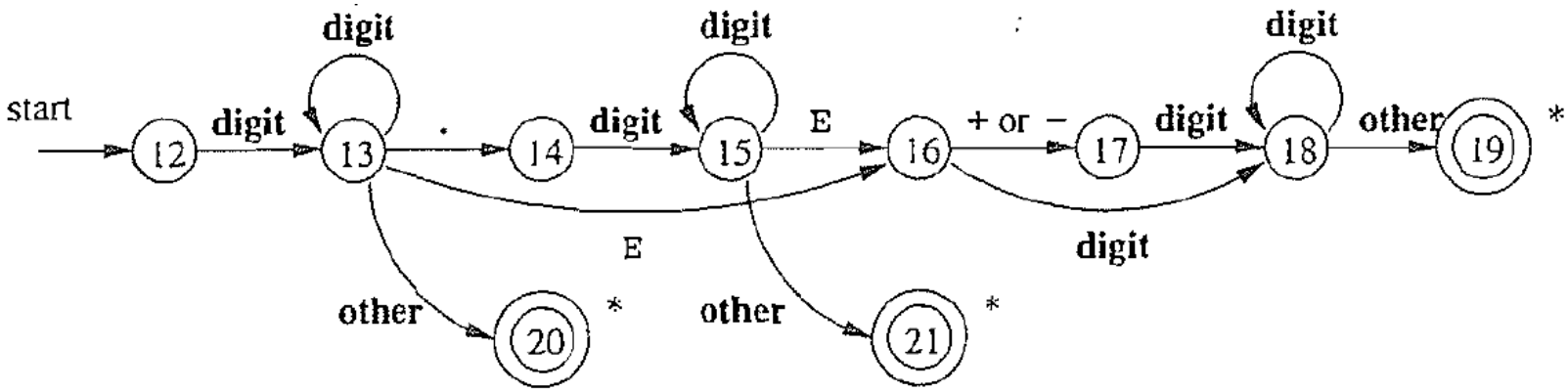
sci: 带科学计数法的浮点数
(2.99792458E8 3E8)

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



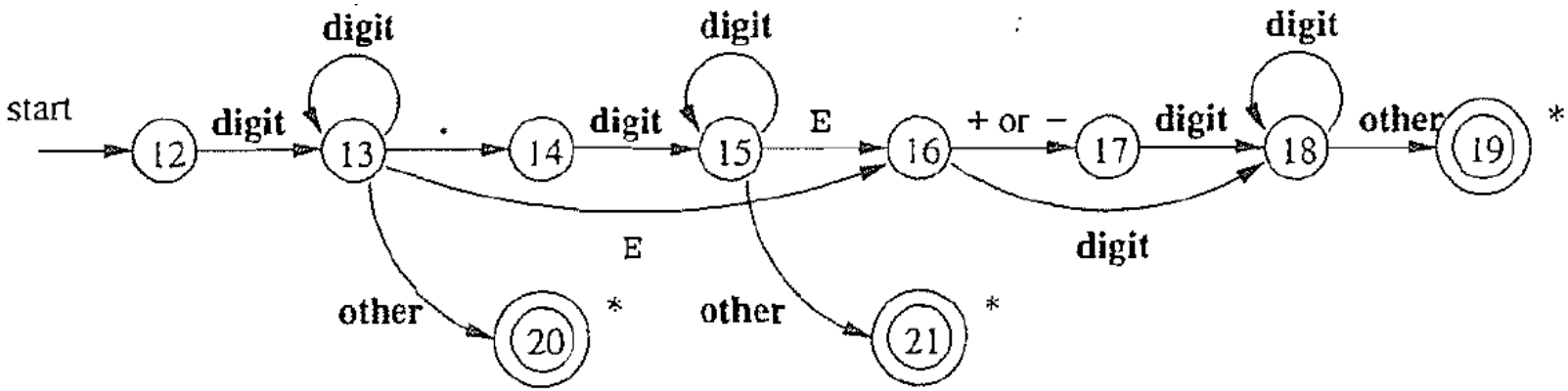
num: 整数部分[. 可选的小数部分][E[可选的 +-] 可选的指数部分]

13, 15, 18 : 无论遇到什么字符, 去向明确



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

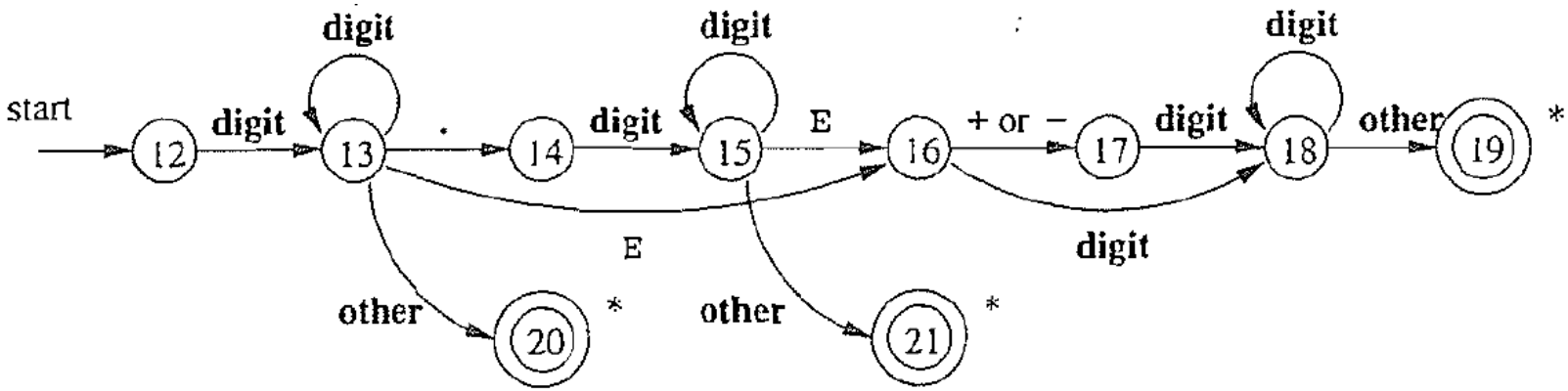
13, 15, 18 : 无论遇到什么字符, 去向明确



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]

13, 15, 18 : 无论遇到什么字符, 去向明确

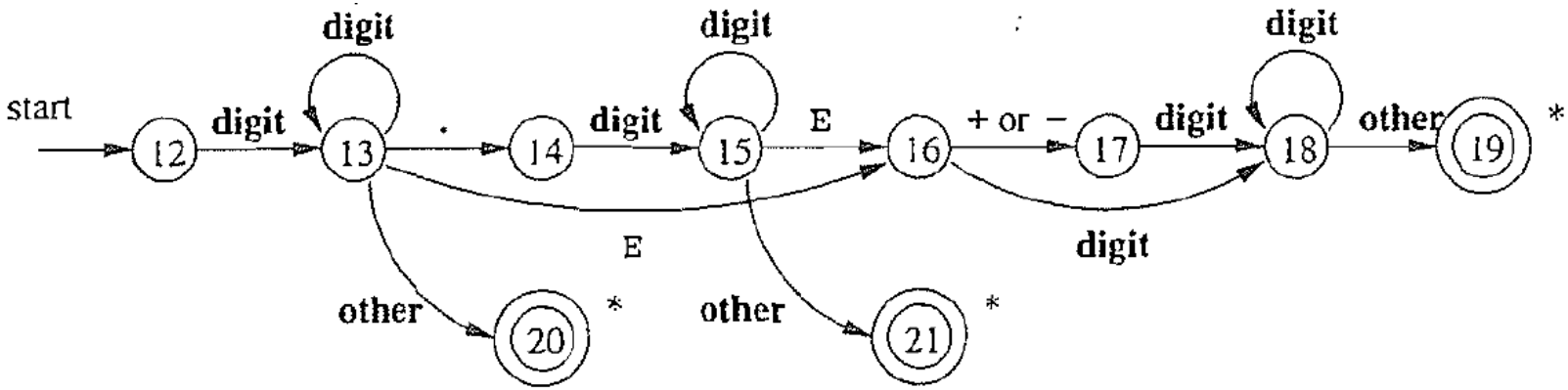


19, 20, 21 : 代表了不同的数字类型

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]

13, 15, 18 : 无论遇到什么字符, 去向明确

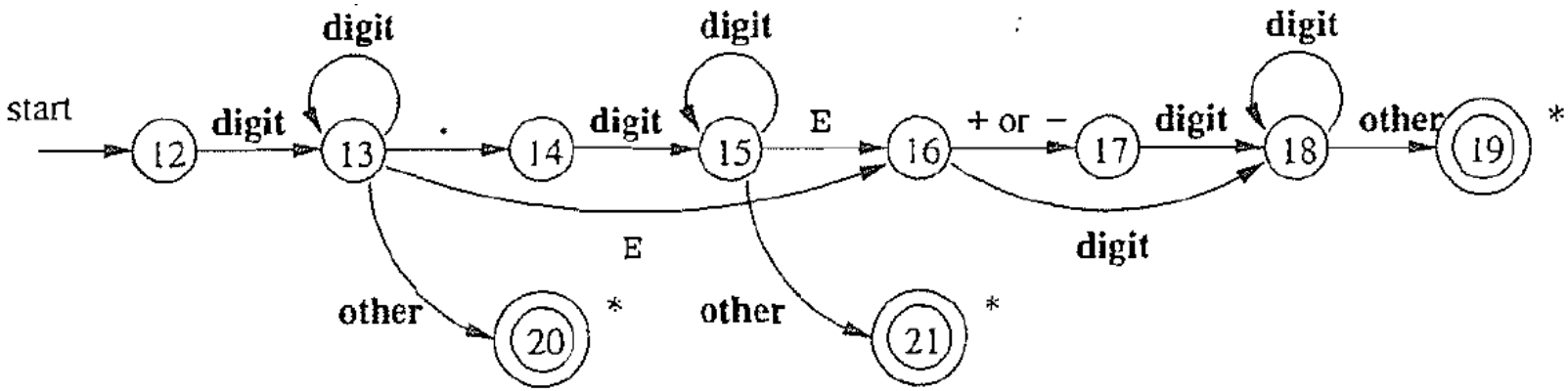


19, 20, 21 : 代表了不同的数字类型

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

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

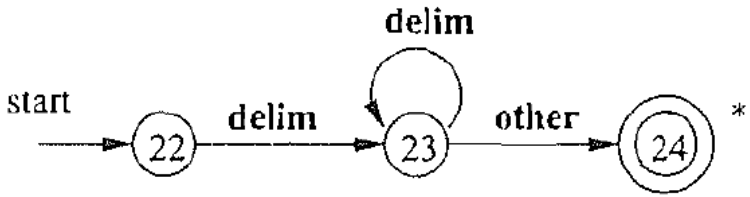
13, 15, 18 : 无论遇到什么字符, 去向明确

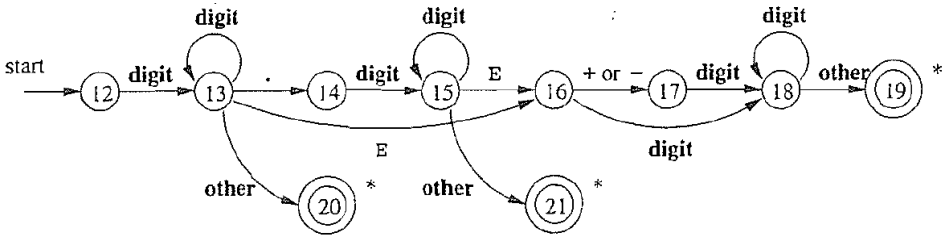


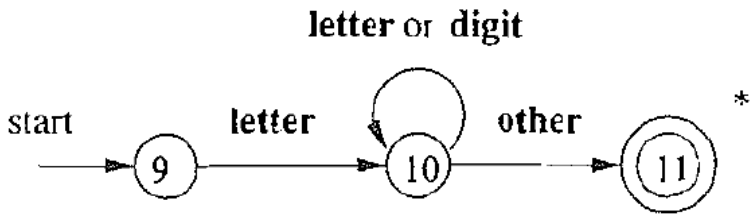
19, 20, 21 : 代表了不同的数字类型

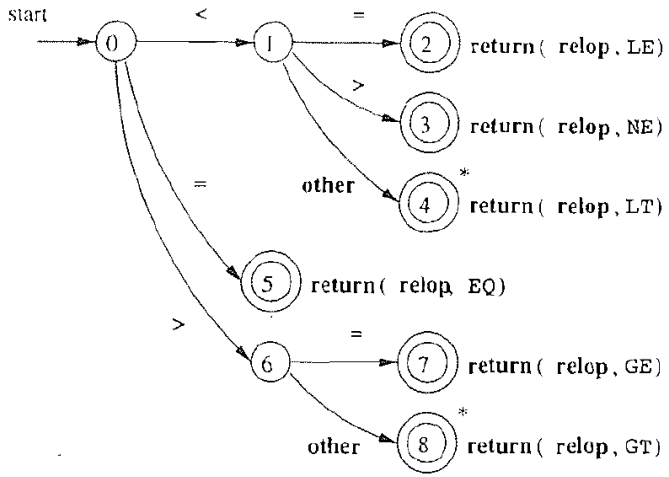
14, 16, 17 : 碰到 other 怎么办? (回退, 寻找**最长匹配**)

14 回到哪里? 16、17 回到哪里?

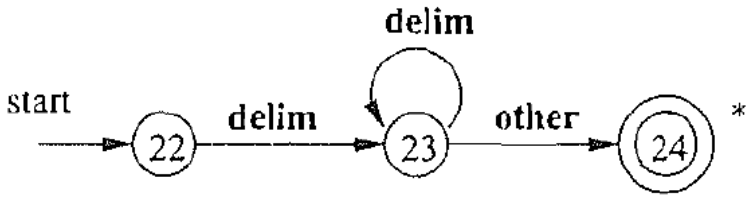

delim
start
22
delim
23
other
24
*

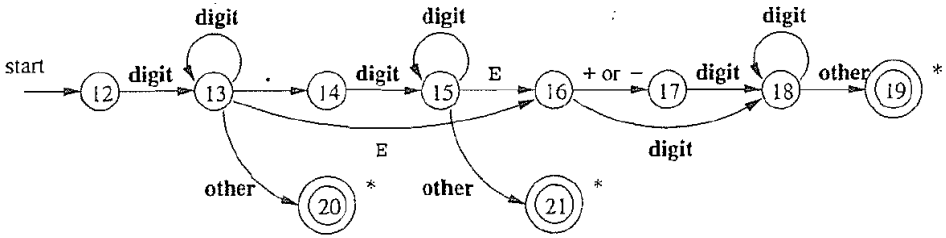

start
12
digit
13
.
14
digit
15
E
16
+ or −
17
digit
18
other
19
*
digit
digit
digit
E
digit
other
20
*
other
21
*

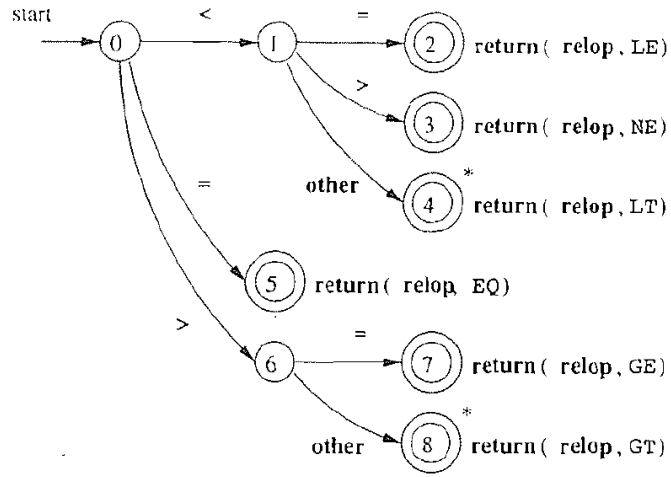

letter or digit
start
9
letter
10
other
11
*

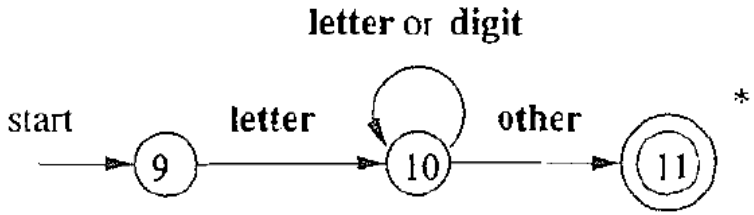

start
0
<
1
=
2
return(relop, LE)
>
3
return(relop, NE)
other
4
return(relop, LT)
*
=
5
return(relop, EQ)
>
6
=
7
return(relop, GE)
other
8
return(relop, GT)
*


ERROR







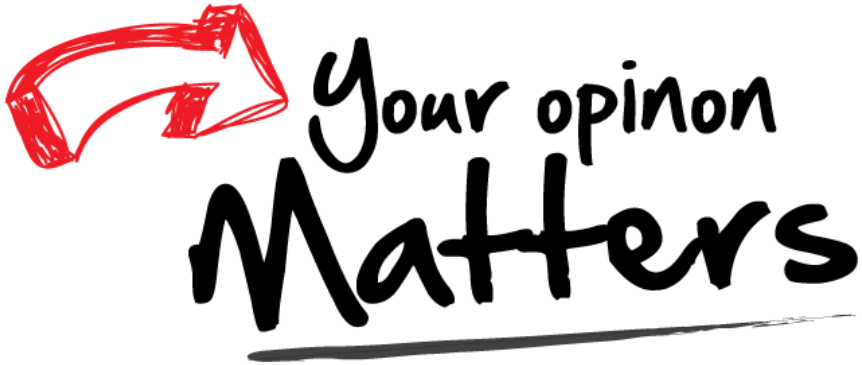


关键点: 合并 22, 12, 9, 0, 根据**下一个字符**即可判定词法单元的类型

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