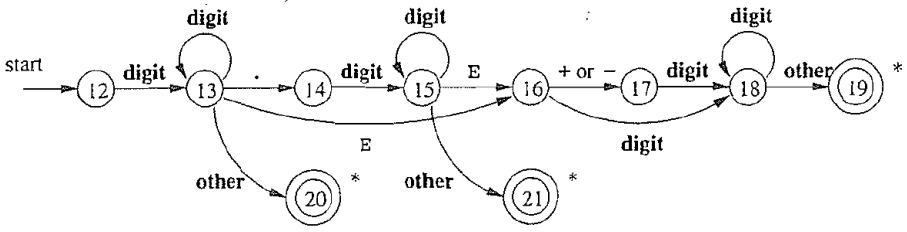
编译原理作业 (1) 参考答案

姓名：王腾　　**邮箱：**171240540@smail.nju.edu.cn

2020 年 12 月 1 日

题目 (手写词法分析器)

根据下面的状态转移图以及课上介绍的识别方法, 给出识别数字 (正整数、不带科学计数法的浮点数以及带科学计数法的浮点数) 的伪代码。



解答：

如果发现问题，希望指出错误发送到我邮箱，谢谢！

Algorithm 1 Parser

```
1:  procedure Parse(str)        ▷ str is input stream and suppose it has a end character EOF
2:      l ← 1, r ← 1                                          ▷ left and right bound of lexeme
3:      state ← 12                                        ▷ simulate the procedure of the diagram
4:      fstate ← 0                                          ▷ to decide final state when goes back
5:      while r ≤ str.length do          ▷ str.length includes EOF, each loop r increases one
6:          Switch : state
7:              Case : 12
8:                  if isdigit(str[r]) then
9:                      state ← 13
10:                 else
11:                     l ← l + 1                                  ▷ skip mysterious character
12:                 end if
13:                 break                                              ▷ begin next loop
14:             EndCase
15:             Case : 13
16:                 if isdigit(str[r]) then
17:                     state ← 13
18:                 else if str[r] ==′ .′ then
19:                     state ← 14
20:                 else if str[r] ==′ E′ then
21:                     state ← 16
22:                 else
23:                     state ← 20
24:                 end if
25:                 fstate ← 20                                ▷ set final state for going back
```

```
26:             break
27:         EndCase
28:         Case : 14
29:             if isdigit(str[r]) then
30:                 state ← 15
31:             else
32:                 state ← fstate
33:                 r ← r − 2
34:             end if
35:             break
36:         EndCase
37:         Case : 15
38:             if isdigit(str[r]) then
39:                 state ← 15
40:             else if str[r] ==′ E′ then
41:                 state ← 16
42:             else
43:                 state ← 21
44:             end if
45:             fstate ← 21
46:             break
47:         EndCase
48:         Case : 16
49:             if isdigit(str[r]) then
50:                 state ← 18
51:             else if str[r] ==′ +′ or str[r] ==′ −′ then
52:                 state ← 17
53:             else
54:                 r ← r − 2
55:                 state ← fstate
56:             end if
57:             break
58:         EndCase
59:         Case : 17
60:             if isdigit(str[r]) then
61:                 state ← 18
62:             else
63:                 r ← r − 3
64:                 state ← fstate
65:             end if
66:             break
67:         EndCase
68:         Case : 18
69:             if isdigit(str[r]) then
70:                 state ← 18
71:             else
```

```
72:                 state ← 19
73:             end if
74:             fstate ← 19
75:             break
76:         EndCase
77:         Case : 19 or 20 or 21
78:             print(str[l − r])                    ▷ print from str[l] to str[r]
79:             l ← r + 1
80:             state ← 12
81:             fstate ← 0
82:             break
83:         EndCase
84:     EndSwitch
85:     r ← r + 1                                    ▷ one loop check one character
86: end while
87: end procedure
```