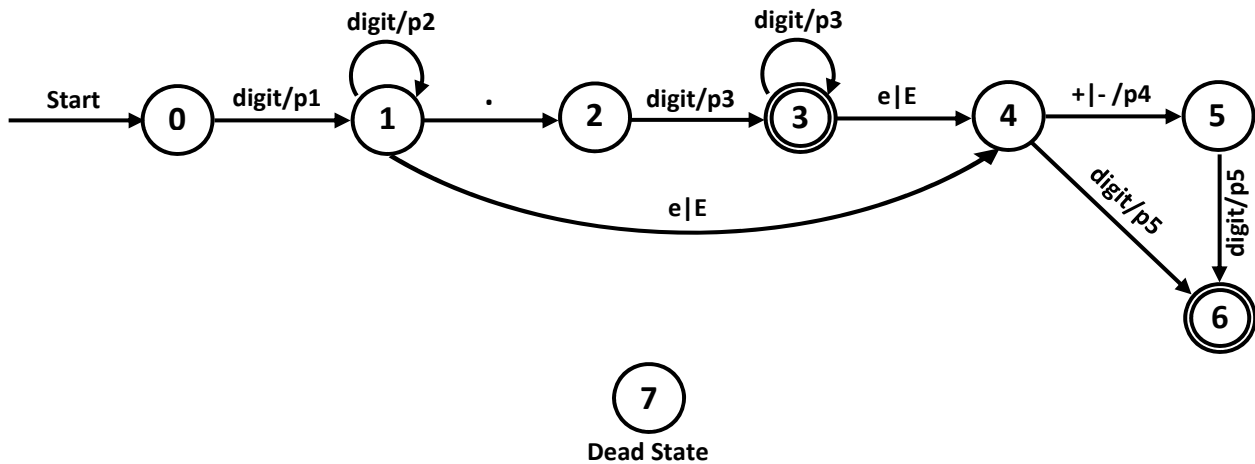


Compiler Design

Computer science dept., 3rd Stage

Exp.3: Floating-point number recognition and evaluation

Step 1: Design the Finite Automata (FA)



Step 2: Build the transition Table

States	Input Symbols				
	digit	dot	e E	+ -	others
0	1/p1	7	7	7	7
1	1/p2	2	4	7	7
2	3/p3	7	7	7	7
3	3/p3	7	4	7	7
4	6/p5	7	7	5/p4	7
5	6/p5	7	7	7	7
6	7	7	7	7	7
7	7	7	7	7	7

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Collections;
6 using System.Threading.Tasks;
7
8 namespace floatingPointRecognition
9 {
10     class Program
11     {
12         public delegate void delgFunc();
13
14         static int index, places, n, sign, exp;
15         static string input;
16
17
18         static int getColNum(char c)
19         {
20             if (Char.IsDigit(c)) return 0; // digit (0-9)
21             switch (c)
22             {
23                 case '.': return 1; // dot
24                 case 'e':
25                 case 'E': return 2; // e|E
26                 case '+':
27                 case '-': return 3; // +|-
28                 default: return 4; // others
29             }
30         }
31
32         static void p1()
33         {
34             n = input[index] - 48; // first digit (convert to numeric)
35         }
36
37
38         static void p2()
39         {
40             n = n * 10 + (input[index] - 48); // next digit (convert to numeric)
41         }
42
43
44         static void p3()
45         {
46             n = n * 10 + (input[index] - 48); // next digit (convert to numeric)
47             places++; // one more place
48         }
49
50
51         static void p4()
52         {
53             if (input[index] == '-')
54                 sign = -1;
55         }
56
57
58         static void p5()
59         {
60             exp = input[index] - 48; // get the value of the exponent
61         }
62
63
64
65
66
67
68
69
70
71
```

```

72
73 static void Main(string[] args)
74 {
75     int state;
76
77     int[,] TT = new int[,] // transition table for FA
78     {
79         //Digit Dot e|E +|- others
80         { 1, 7, 7, 7, 7 }, // state 0
81         { 1, 2, 4, 7, 7 }, // state 1
82         { 3, 7, 7, 7, 7 }, // state 2
83         { 3, 7, 4, 7, 7 }, // state 3
84         { 6, 7, 7, 5, 7 }, // state 4
85         { 6, 7, 7, 7, 7 }, // state 5
86         { 7, 7, 7, 7, 7 }, // state 6
87         { 7, 7, 7, 7, 7 } // state 6
88     };
89
90     delgFunc[,] action = new delgFunc[,] // 2D array for action
91     {
92         //digit Dot e|E +|- others
93         { p1, null, null, null, null }, // state 0
94         { p2, null, null, null, null }, // state 1
95         { p3, null, null, null, null }, // state 2
96         { p3, null, null, null, null }, // state 3
97         { p5, null, null, p4, null }, // state 4
98         { p5, null, null, null, null }, // state 5
99         { null, null, null, null, null }, // state 6
100        { null, null, null, null, null }, // state 7
101    };
102
103    while (true) // infinite loop
104    {
105        state = 0;
106        places = 0;
107        sign = +1;
108        exp = 0;
109        index = 0;
110
111        input = Console.ReadLine();
112
113        while (index < input.Length)
114        {
115            int symbol = getColNum(input[index]);
116            if (action[state, symbol] != null) action[state, symbol](); //if there is an action, call action
117            //In VS2015+, line 116 can be simplified as: action[state, symbol]?.Invoke();
118
119            state = TT[state, symbol]; // get a new state
120            index++;
121
122            if (state == 7) break; // if a dead state, skip the rest of input.
123        }
124
125        if (state == 3 || state == 6)
126            Console.WriteLine("Floating point number recognized: {0}", n * Math.Pow(10, sign*exp-places));
127        else
128            Console.WriteLine("Unrecognized input");
129    } //closing of infinite loop
130 }
131 }
132 }

```