

Name Zarak khan

Id 13794

Dept BS(CS)

C++ Program to Perform Finite State Automaton based Search

```
Naik.cpp
1  #include<stdio.h>
2  #include<string.h>
3  #define NO_OF_CHARS 256
4
5  int getNextState(char *pat, int M, int state, int x)
6  {
7      // If the character c is same as next character in pattern,
8      // then simply increment state
9      if (state < M && x == pat[state])
10         return state + 1;
11
12     int ns, i; // ns stores the result which is next state
13
14     // ns finally contains the longest prefix which is also suffix
15     // in "pat[0..state-1]c"
16
17     // Start from the largest possible value and stop when you find
18     // a prefix which is also suffix
19     for (ns = state; ns > 0; ns--)
20     {
21         if (pat[ns - 1] == x)
22         {
23             for (i = 0; i < ns - 1; i++)
24             {
25                 if (pat[i] != pat[state - ns + 1 + i])
26                     break;
27             }
28             if (i == ns - 1)
29                 return ns;
30         }
31     }
32
33     return 0;
34 }
35
36 /* This function builds the TF table which represents Finite Automata for a
37    given pattern */
38 void computeTF(char *pat, int M, int TF[][NO_OF_CHARS])
39 {
40     int state, x;
41     for (state = 0; state <= M; ++state)
42         for (x = 0; x < NO_OF_CHARS; ++x)
43             TF[state][x] = getNextState(pat, M, state, x);
44 }
45
46 /* Prints all occurrences of pat in txt */
47 void search(char *pat, char *txt)
48 {
49     int M = strlen(pat);
50     int N = strlen(txt);
51
52     for (int i = 0; i < N - M + 1; i++)
53     {
54         int j = 0;
55         while (j < M && pat[j] == txt[i + j])
56             j++;
57         if (j == M)
58             printf("Found at %d\n", i);
59     }
60 }
```

1 Compile Log 2 Debug 3 Find Results 4 Close

```

52     int TF[M + 1][NO_OF_CHARS];
53
54     computeTF(pat, M, TF);
55
56     // Process txt over FA.
57     int i, state = 0;
58     for (i = 0; i < N; i++)
59     {
60         state = TF[state][txt[i]];
61         if (state == M)
62         {
63             printf("\n pattern found at index %d", i - M + 1);
64         }
65     }
66 }
67
68 // Driver program to test above function
69 int main()
70 {
71     char *txt = "AABAACAADAABAAABAA";
72     char *pat = "AABA";
73     search(pat, txt);
74     return 0;
75 }

```

Output:

```

-----
pattern found at index 0
pattern found at index 9
pattern found at index 13
-----
Process exited after 0.06509 seconds with return value 0
Press any key to continue . . .

```