ID No 13943
Name Mehran Ali Shah
Subject CC

C++ Program to Perform Finite State Automaton based Search

This is a C++ Program to search a string using finite automata. Given a text txt[0..n-1] and a pattern pat[0..m-1], write a function search(char pat[], char txt[]) that prints all occurrences of pat[] in txt[]. You may assume that n > m.

Here is source code of the C++ Program to Perform Finite State Automaton based Search. The C++ program is successfully compiled and run on a Linux system. The program output is also shown below.

```
#include<stdio.h>
#include<string.h>
#define NO_OF_CHARS 256

int getNextState(char *pat, int M, int state, int x)
{
    // If the character c is same as next character in pattern,
    // then simply increment state
    if (state < M && x == pat[state])
        return state + 1;

    int ns, i; // ns stores the result which is next state

    // ns finally contains the longest prefix which is also suffix
    // in "pat[0..state-1]c"

    // Start from the largest possible value and stop when you find
```

```
    // a prefix which is also suffix
    for (ns = state; ns > 0; ns--)
    {
        if (pat[ns - 1] == x)
        {
            for (i = 0; i < ns - 1; i++)
            {
                if (pat[i] != pat[state - ns + 1 + i])
                    break;
            }
            if (i == ns - 1)
                return ns;
        }
    }

    return 0;
}

/* This function builds the TF table which represents Finite Automata for
a
 given pattern  */
void computeTF(char *pat, int M, int TF[][NO_OF_CHARS])
{
    int state, x;
    for (state = 0; state <= M; ++state)
        for (x = 0; x < NO_OF_CHARS; ++x)
            TF[state][x] = getNextState(pat, M, state, x);
}

/* Prints all occurrences of pat in txt */
void search(char *pat, char *txt)
{
    int M = strlen(pat);
    int N = strlen(txt);

    int TF[M + 1][NO_OF_CHARS];

    computeTF(pat, M, TF);

    // Process txt over FA.
    int i, state = 0;
```

```
    for (i = 0; i < N; i++)
    {
        state = TF[state][txt[i]];
        if (state == M)
        {
            printf("\n pattern found at index %d", i - M + 1);
        }
    }
}

// Driver program to test above function
int main()
{
    char *txt = "AABAACAADAABAAABAA";
    char *pat = "AABA";
    search(pat, txt);
    return 0;
}
```