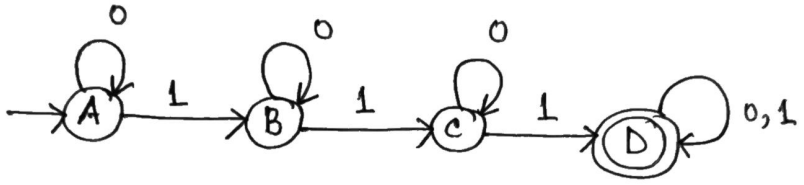
TOC ASSIGNMENT → 2

21.01.2023

Prayas Mazumder

2021CSB071

$L_1 = \{ w \in \{0,1\}^* \mid w \text{ contains atleast three 1's} \}$



DFA $M_1 = \{ Q, A, \Sigma, \delta, F \}$, where $Q = \{ A, B, C, D \}$

$\delta(A,0) = A$ $\quad \delta(C,0) = C$

$\delta(A,1) = B$ $\quad \delta(C,1) = D$

$\delta(B,0) = B$ $\quad \delta(D,0) = D$

$\delta(B,1) = C$ $\quad \delta(D,1) = D$

A = initial state

Σ = alphabet = $\{0,1\}$

F = final state = $\{D\}$ set

where $\delta \rightarrow$ State transition function.

Also, language accepted by DFA $M_1 = L_1 = L(M_1)$.

Some test cases (theory + program)

a) 111 ✓

b) 1011 ✓

c) 010010 ✗

d) 10001000 ✗

State Transition Table

States \ Σ	0	1
A	A	B
B	B	C
C	C	D
D	D	D

* C-file attached

```c
ssignment -->2
r following language
(w ∈ {0, 1}*
ains atleast three 1s)
 program to simulate the DFA M1.*/
 ARE TOTAL 5 STATES

B071 PRAYAS_MAZUMDER  , 21.01.2023 (sat)

 <stdio.h>
 <string.h>

()


 str[71];
 dfa = 'A'; // initially the dfa is in state A , A--> initial state
;

f("\n");
f("enter the binary string : "); // here alphabet = {0,1}

("%s", str);

i = 0; str[i] != '\0'; i++) // '\0' is the NULL terminated string , pointing the end of the string

witch (dfa) // here each case is state

ase 'A':

    if (str[i] == '0')
        dfa = 'A';
    else if (str[i] == '1')
        dfa = 'B';
    break;

ase 'B':

    if (str[i] == '0')
        dfa = 'B';
    else if (str[i] == '1')
        dfa = 'C';
    break;

ase 'C':

    if (str[i] == '0')
        dfa = 'C';
```

```c
        case 'C':

            if (str[i] == '0')
                dfa = 'C';
            else if (str[i] == '1')
                dfa = 'D';
            break;

        case 'D':

            if (str[i] == '0')
                dfa = 'D';
            else if (str[i] == '1')
                dfa = 'D';
            break;
        }
    }

    if (dfa == 'D') // i.e. the final state
    {
        printf("\n");
        printf("Binary string is accepted by dfa M1 \n", dfa);
        printf("\n");
        return 1;
    }
    else
    {
        printf("\n");
        printf("Binary string is not accepted by dfa M1 \n", dfa);
        printf("\n");
        return 0;
    }
```

```
enter the binary string : 100100
Binary string is not accepted by dfa M1
PS C:\Users\ASUS\Downloads> cd "c:\Users\ASUS\Downloads\" ; if ($?) { gcc toc_dfa_atleat_three_1.c -o toc_dfa_atleat_three_
1 } ; if ($?) { .\toc_dfa_atleat_three_1 }

enter the binary string : 100011

Binary string is accepted by dfa M1

PS C:\Users\ASUS\Downloads> cd "c:\Users\ASUS\Downloads\" ; if ($?) { gcc toc_dfa_atleat_three_1.c -o toc_dfa_atleat_three_
1 } ; if ($?) { .\toc_dfa_atleat_three_1 }

enter the binary string : 11000000000000010

Binary string is accepted by dfa M1

PS C:\Users\ASUS\Downloads> cd "c:\Users\ASUS\Downloads\" ; if ($?) { gcc toc_dfa_atleat_three_1.c -o toc_dfa_atleat_three_
1 } ; if ($?) { .\toc_dfa_atleat_three_1 }

enter the binary string : 10000000000001000

Binary string is not accepted by dfa M1

PS C:\Users\ASUS\Downloads> cd "c:\Users\ASUS\Downloads\" ; if ($?) { gcc toc_dfa_atleat_three_1.c -o toc_dfa_atleat_three_
1 } ; if ($?) { .\toc_dfa_atleat_three_1 }

enter the binary string : 100100

Binary string is not accepted by dfa M1

PS C:\Users\ASUS\Downloads>
```