

Exp 1. Deterministic finite Automata

Aim :- To write a C program to simulate a Deterministic finite Automata.

Algorithm

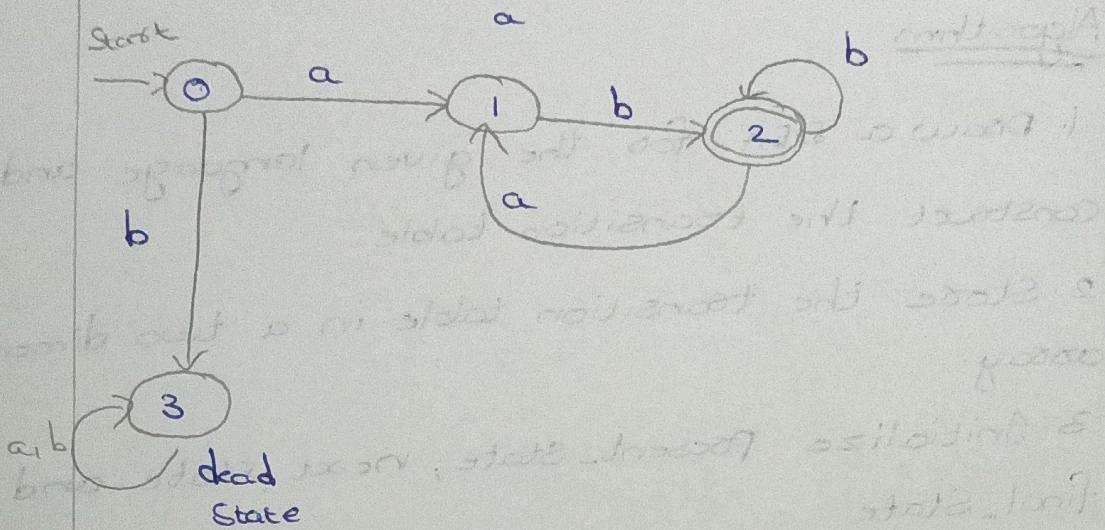
1. Draw a DFA for the given language and construct the transition table.
2. Store the transition table in a two dimensional array.
3. Initialize present_state, next_state and final_state.
4. Get the Input String from the user.
5. find the length of the Input String.
6. Read the Input String character by character.
7. Repeat Step 8 for every character.
8. Refer the transition table for the entry corresponding to present state and the current input symbol.
9. When we reach the end of the input, If the final State is reached, the input is accepted. Otherwise the input is not accepted.

Example

Simulate a DFA for the language $L = \{a^n b^m \mid n \geq 1, m \geq 0\}$ over $\Sigma = \{a, b\}$ that start with a string over $\Sigma = \{a, b\}$ that start with a

end with b

Design of the DFA



Transition Table

State / Input	a	b
0	1	3
1	1	2
2	1	2
3	3	3

Program

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#define max 20
```

```
int main()
```

```
{
```

```
    int trans_table[4][2] = {{1, 3}, {1, 2}, {1, 2}, {3, 3}}
```

```
int final_state = 2;
int present_state = 0;
int next_state = 0;
int invalid = 0;
char input_string[max];
printf("Enter a string");
scanf("%s", input_string);
int l = strlen(input_string);
for(i=0; i < l; i++)
{
    if(input_string[i] == 'a')
        next_state = trans_table[present_state][0];
    else if(input_string[i] == 'b')
        next_state = trans_table[present_state][1];
    else
        invalid = 1;
    present_state = next_state;
}
if(invalid == 1)
{
    printf("Invalid Input");
}
else if(present_state == final_state)
    printf("Accept\n");
else
    printf("Don't Accept\n");
}
```

Output

Enter a string : abaab

Accept.

Enter a String : abbbaaaba

Don't Accept.