

HARSHINI

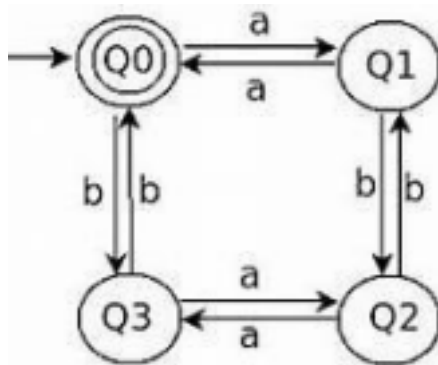
AP19110010442

CSE C

Program 1:

Implement a language recogniser which accepts set of all strings over the alphabet $\Sigma=\{a,b\}$ containing an even number of a's and an even number of b's. The acceptable strings of the language are ϵ (Null string), aa, bb, abba, babbab etc.

Deterministic Finite Automata for the given language is given below:



DFA $M=(Q,\Sigma,\delta,Q_0,F)$ Where

Q =Set of all states $=\{Q_0,Q_1,Q_2,Q_3\}$

Σ =Input Alphabet $=\{a,b\}$,

Start state is Q_0

F =Set of all final States $=\{Q_0\}$

And the transitions are defined in the transition diagram

Algorithm: Language recognizer

Input:

input //input string

Output:

Algorithm prints a message

“String accepted”: If the input is acceptable by the language,

“String not accepted” otherwise,

“Invalid token”: If the input string contains symbols other than input alphabet.

C CODE:

```
#include<stdio.h>

void main(){
int state=0,i=0;
char current,input[20];
printf("Enter input string \t :");
scanf("%s",input);
while((current=input[i++])!='\0'){
switch(state)
{
case 0: if(current=='a')
state=1;
else if(current=='b')
state=2;
else
{
printf("Invalid token");
exit(0);
}
break;
case 1: if(current=='a')
state=0;
else if(current=='b')
state=3;
else
{
printf("Invalid token");
exit(0);
```

```
}  
break;  
case 2: if(current=='a')  
state=3;  
else if(current=='b')  
state=0;  
else  
{  
printf("Invalid token");  
exit(0);  
}  
break;  
case 3: if(current=='a')  
state=2;  
else if(current=='b')  
state=1;  
else  
{  
printf("Invalid token");  
  
exit(0);  
}  
break;  
}  
}  
if(state==0)  
printf("\n\nString accepted\n\n");  
else  
printf("\n\nString not accepted\n\n");  
}
```

PROGRAM 2:

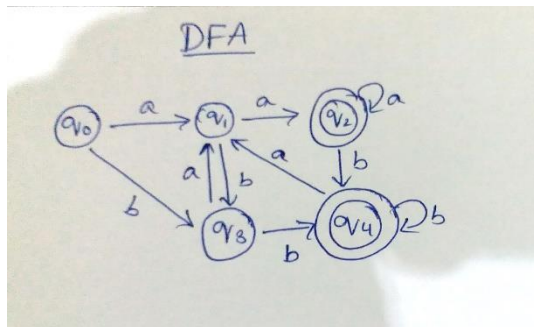
Implementation of Language recognizer for a set of all strings ending with two symbols of the same type.

DFA $M=(Q,\Sigma,\delta,Q_0,F)$ Where Q =Set of all states $=\{Q_0,Q_1,Q_2,Q_3,Q_4\}$ Σ =Input Alphabet $=\{a,b\}$,

The start state is Q_0

F =Set of all final States= $\{Q_2, Q_4\}$

The transitions are described in the Transition diagram.



C CODE:

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

```
#include<stdlib.h>
```

```
int main()
```

```
{
```

```
int state=0,i=0;
```

```
char current,input[20];
```

```
printf("Enter input string \t :");
```

```
scanf("%s",input);
```

```
while((current=input[i++])!='\0'){
```

```
switch(state)

case 0:if(current=='a')

    state=1;

    else if(current=='b')

    state=3;

    else


    { {printf("%d",current);

        printf("Invalid token");

        exit(0);

        }

        break;

case 1:if(current=='a')

    state=2;

    else if(current=='b')

    state=3;

    else

    { printf("Invalid token");

        exit(0);

        }

        break;

case 2:if(current=='a')

    state=2;

    else if(current=='b')

    state=3;

    else

    { printf("Invalid token");

        exit(0);

        }

        break;

case 3:if(current=='a')
```

```
        state=1;
        else if(current=='b')
            state=4;
        else
            { printf("Invalid token");
              exit(0);
            }
        break;
case 4:if(current=='a')
        state=1;
        else if(current=='b')
            state=4;
        else
            { printf("Invalid token");
              exit(0);
            }
    }
}

if(state==2 || state==4)
    printf("\n\nString accepted\n\n");
else
    printf("\n\nString not accepted\n\n");
}
```