**Kunal Goel**

**AP18110010463**

**CSE-H**

Program-1

Implement a language recognizer which accepts set of all strings over the alphabet

∑={a,b} containing an even number of a’s and an even number of b’s.

Description

The few strings of language are ε (Null string), aa, bb, abab, bbaa, etc.

The Deterministic Finite Automata (DFA) for the given language is: -

**a**

**a**

**b** **b** **b b**

**a**

**a**

A DFA is a five tuple. Let N be the name of DFA,

N= (Q,∑,δ,Q0,F) where,

Q=Set of all states ={Q0,Q1,Q2,Q3},

∑=Input Alphabet={a,b},

Start state is Q0

F=Set of all final States={ Q0} and

δ= Transition Function is as follows:

|  |  |  |
| --- | --- | --- |
| States | a | b |
| Q0 | Q1 | Q2 |
| Q1 | Q0 | Q3 |
| Q2 | Q3 | Q0 |
| Q3 | Q2 | Q1 |

Algorithm

**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.

Method

state=0 //**initial state** while((current=input[i++])!='\0'){

switch(state)

case 0: if(current=='a') state=1;

else if(current=='b') state=2;

else

Print "Invalid token"; exit;

case 1: if(current=='a') state=0;

else if(current=='b') state=3;

else

Print "Invalid token"; exit; case 2: if(current=='a') state=3;

else if(current=='b') state=0;

else

Print "Invalid token"; exit; case 3: if(current=='a') state=2;

else if(current=='b') state=1;

else

Print "Invalid token"; exit;

end switch

end while

}

//**Print** output if(state==0)

Print ”String accepted”

else

Print ”String not accepted”

Code for the given language in C

#include<stdio.h>

#include<stdlib.h>

int main()

{

int state=0,i=0;

char current,input[20];

printf("Enter input string:");

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("String accepted");

else

printf("String not accepted");

return 0;

}

Sample Inputs and their Outputs

|  |  |
| --- | --- |
| Sample Inputs | Outputs |
| aa | String accepted |
| bb | String accepted |
| aaaaa | String not accepted |
| aabddbc | Invalid token |
| aaabbbb | String not accepted |

Conclusion

The above language recognizer has been implemented that recognizes the set of all strings over the alphabets ∑={a,b} containing an even number of a’s and an even number of b’s.

Program-2

Implementation of Language recognizer for set of all strings ending with two symbols of

same type.

Description

Let the alphabet be ∑={a,b}

The few strings of the language are aaa, bbb, ababaabb, baaaa, abbbaa,baabbbabb etc.

The Deterministic Finite Automata (DFA) for the given language is:

**a** **b**

**a**

**b**

**a b a b**

A DFA is a five tuple. Let M be the name of DFA,

M= (Q,∑,δ,Q0,F) where,

Q=Set of all states ={Q0,Q1,Q2,Q3,Q4},

∑=Input Alphabet={a,b},

Start state is Q0

F=Set of all final States={Q2,Q4} and

δ= Transition Function is as follows:

|  |  |  |
| --- | --- | --- |
| States | a | B |
| Q0 | Q1 | Q3 |
| Q1 | Q2 | Q3 |
| Q2 | Q2 | Q3 |
| Q3 | Q1 | Q4 |
| Q4 | Q1 | Q4 |

Algorithm

**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.

Method

state=0 //**initial state** while((current=input[i++])!='\0'){

switch(state)

case 0: if(current=='a') state=1;

else if(current=='b') state=3;

else

Print "Invalid string input"; exit;

case 1: if(current=='a') state=2;

else if(current=='b') state=3;

else

Print "Invalid string input"; exit; case 2: if(current=='a') state=2;

else if(current=='b') state=3;

else

Print "Invalid string input"; exit; case 3: if(current=='a') state=1;

else if(current=='b') state=4;

else

Print "Invalid string input"; exit;

case 4: if(current=='a') state=1;

else if(current=='b') state=4;

else

Print "Invalid string input"; exit;

end switch

end while

}

//**Print** output if(state==2 || state==4)

Print ”String is accepted”

else

Print ”String is not accepted”

Code for the given language in C

#include<stdio.h>

#include<stdlib.h>

int main()

{

char input[100],current;

printf("Enter the input string:");

scanf("%s",input);

int i=0,state=0;

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

{

switch(state)

{

case 0:

if(current=='a')

state=1;

else if(current=='b')

state=3;

else

{

printf("Invalid string input");

exit(1);

}

break;

case 1:

if(current=='a')

state=2;

else if(current=='b')

state=3;

else

{

printf("Invalid string input");

exit(1);

}

break;

case 2:

if(current=='a')

state=2;

else if(current=='b')

state=3;

else

{

printf("Invalid string input");

exit(1);

}

break;

case 3:

if(current=='a')

state=1;

else if(current=='b')

state=4;

else

{

printf("Invalid string input");

exit(1);

}

break;

case 4:

if(current=='a')

state=1;

else if(current=='b')

state=4;

else

{

printf("Invalid string input");

exit(1);

}

break;

}

}

if(state==2 || state==4)

printf("String is accepted");

else

printf("String is not accepted");

return 0;

}

Sample Inputs and their Outputs

|  |  |
| --- | --- |
| Sample Inputs | Outputs |
| bbb | String is accepted |
| aaa | String is accepted |
| abbbaaab | String is not accepted |
| aacbbdd | Invalid string input |
| bababa | String is not accepted |

Conclusion

The above language recognizer has been implemented that recognizes the set of all strings over the alphabets ∑={a,b} ending with two symbols of same type.