Lab Assignment

DFA Lab

Question 1

Give the DFA for the language of string over {0,1} in which each string ends with 11.

Solution:

1. State Diagram:

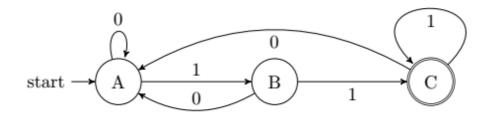


Figure 1: DFA to accept strings ending with 11

2. State Table:

State/input	0	1
A	A	В
В	A	С
С	A	С

3. State Function:

$$egin{aligned} \delta(A,0) &
ightarrow A \ \delta(A,1) &
ightarrow B \ \delta(B,0) &
ightarrow A \ \delta(B,1) &
ightarrow C \ \delta(C,0) &
ightarrow A \ \delta(C,1) &
ightarrow C \end{aligned}$$

```
#include <iostream>
2
   #include <string>
3
   using namespace std;
4
5
6
   bool is_valid_input(string str) {
     for (char c : str) {
7
       if (c != '0' && c != '1') {
8
        return false;
9
        }
10
11
      }
     return true;
12
13
14
15 bool is_accepted(string str) {
16
      char state = 'A';
     for (char c : str) {
17
       switch (state) {
18
         case 'A':
19
            state = (c == '0') ? 'A' : 'B';
20
21
           break;
         case 'B':
22
           state = (c == '0') ? 'A' : 'C';
23
           break;
24
         case 'C':
25
            state = (c == '0') ? 'A' : 'C';
26
27
        }
      }
28
      return state == 'C';
29
30
31
32 int main() {
      string str;
33
      char choice;
34
     do {
35
      cout << "Enter a string: ";</pre>
36
37
       cin >> str;
        if (is_valid_input(str)) {
38
         cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;</pre>
39
40
        } else {
          cout << "Invalid Input" << endl;</pre>
41
```

```
42  }
43     cout << "Do you want to continue? (y/n): ";
44     cin >> choice;
45     } while (choice == 'y');
46     return 0;
47 }
```

Give the DFA accepting the string over {a,b} such that each string does not end with ab.

Solution:

1. State Diagram:

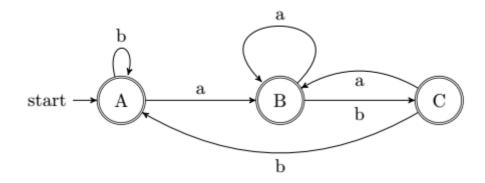


Figure 2: DFA to accept strings not ending with ab

2. State Table:

State/input	a	b
A	В	A
В	В	С
С	В	A

3. State Function:

$$egin{aligned} \delta(A,0) &
ightarrow B \ \delta(A,1) &
ightarrow A \ \delta(B,0) &
ightarrow B \ \delta(B,1) &
ightarrow C \ \delta(C,0) &
ightarrow B \ \delta(C,1) &
ightarrow A \end{aligned}$$

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6
   bool is_valid_input(string str)
7
     for (char c : str)
8
9
       if (c != 'a' && c != 'b')
10
11
       {
12
        return false;
      }
13
     }
14
     return true;
15
16
17
   bool is_accepted(string str)
18
19
     char state = 'A';
20
21
     for (char c : str)
22
       switch (state)
23
24
       case 'A':
25
26
        state = (c == 'a') ? 'B' : 'A';
27
        break;
       case 'B':
28
        state = (c == 'a') ? 'B' : 'C';
29
30
        break;
       case 'C':
31
32
         state = (c == 'a') ? 'B' : 'A';
33
         break;
       }
34
35
     return state != 'C';
36
37
38
39 int main()
40
41
      string str;
```

```
42
      char choice;
      do
43
44
      {
45
        cout << "Enter a string: ";</pre>
46
        cin >> str;
47
        if (is_valid_input(str))
48
        {
49
          cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;</pre>
50
        }
51
52
        else
        {
53
           cout << "Invalid Input" << endl;</pre>
54
        }
55
        cout << "Do you want to continue? (y/n): ";</pre>
56
        cin >> choice;
57
      } while (choice == 'y');
58
59
      return 0;
60
```

Give the DFA for the language of string over {a,b} such that each string contains aba as substring.

Solution:

1. State Diagram:

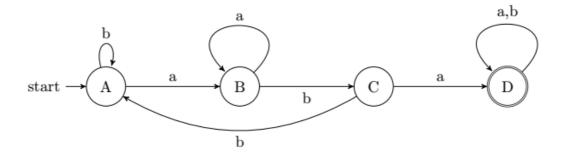


Figure 4: DFA to accept strings having aba as substring

2. State Table

State/Input	a	b
A	В	A
В	В	С
С	D	A
D	D	D

3. State Function

$$egin{aligned} \delta(A,a) &
ightarrow B \ \delta(A,b) &
ightarrow A \ \delta(B,a) &
ightarrow B \ \delta(B,b) &
ightarrow C \ \delta(C,a) &
ightarrow D \ \delta(C,b) &
ightarrow A \ \delta(D,a) &
ightarrow D \ \delta(D,b) &
ightarrow D \end{aligned}$$

```
#include<iostream>
1
2
3
   using namespace std;
4
   bool check_string(string str){
5
        for(char c : str){
6
            if(c != 'a' && c != 'b'){
7
8
                return false;
9
            }
        }
10
        return true;
11
12
   }
13
14
   bool is_accepted(string str){
        char state = 'A';
15
        for(char c : str){
16
            switch(state){
17
                case 'A':
18
                    state = (c == 'a') ? 'B' : 'A';
19
                    break;
20
```

```
21
                 case 'B':
                     state = (c == 'a') ? 'B' : 'C';
22
23
                     break;
                 case 'C':
24
                     state = (c == 'a') ? 'D' : 'A';
25
26
                     break;
27
                 case 'D':
                     state = (c == 'a') ? 'D' : 'D';
28
                     break;
29
            }
30
        }
31
32
        return state == 'D';
33
   }
34
   int main(){
35
36
        string str;
37
        char choice;
38
        do{
            cout << "Enter a string: ";</pre>
39
            cin >> str;
40
            if(check_string(str)){
41
                 cout << (is_accepted(str) ? "Accepted" : "Not Accepted") <<</pre>
42
    endl;
            }else{
43
                 cout << "Invalid Input" << endl;</pre>
44
45
46
            cout << "Do you want to continue? (y/n): ";</pre>
            cin >> choice;
47
        }while(choice == 'v');
48
        return 0;
49
50 }
```

Give the DFA for the language of string over $\{0,1\}$ such that each string start with 01.

Solution:

Give the DFA for the language of string over {0,1} such that the set of all string ending in 00.

Solution:

1. State Diagram

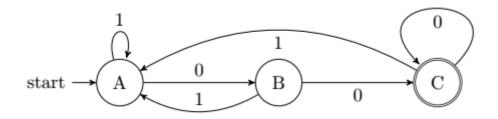


Figure 5: DFA to accept strings ending with 00

2. State Table

State/Input	0	1
A	В	A
В	С	A
С	С	A

3. State Function

$$egin{aligned} \delta(A,0) &
ightarrow B \ \delta(A,1) &
ightarrow A \ \delta(B,0) &
ightarrow C \ \delta(B,1) &
ightarrow A \ \delta(C,0) &
ightarrow C \ \delta(C,1) &
ightarrow A \end{aligned}$$

```
#include <iostream>
#include <string>

using namespace std;

bool is_valid_input(string str) {

for (char c : str) {
```

```
8
        if (c != '0' && c != '1') {
 9
         return false;
        }
10
      }
11
      return true;
12
13
14
   bool is_accepted(string str) {
15
      char state = 'A';
16
     for (char c : str) {
17
        switch (state) {
18
         case 'A':
19
            state = (c == '1') ? 'A' : 'B';
20
           break;
21
         case 'B':
22
            state = (c == '1') ? 'A' : 'C';
23
24
           break;
         case 'C':
25
            state = (c == '1') ? 'A' : 'C';
26
        }
27
      }
28
29
      return state == 'C';
30
31
32 int main() {
33
      string str;
34
      char choice;
     do {
35
      cout << "Enter a string: ";</pre>
36
       cin >> str;
37
        if (is_valid_input(str)) {
38
         cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;</pre>
39
        } else {
40
          cout << "Invalid Input" << endl;</pre>
41
42
        }
        cout << "Do you want to continue? (y/n): ";</pre>
43
44
        cin >> choice;
      } while (choice == 'y');
45
      return 0;
46
47 }
```

Give the DFA for the language of string over {0,1} such that set of strings with 011 as a substring.

Solution:

1. State Diagram

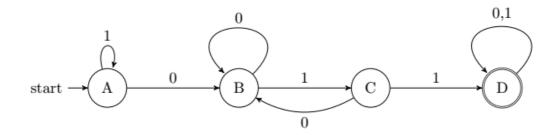


Figure 6: DFA to accept strings having aba as substring

2. State Table

State/Input	0	1
A	В	A
В	В	С
С	В	D
D	D	D

3. State Function

$$egin{aligned} \delta(A,0) &
ightarrow B \ \delta(A,1) &
ightarrow A \ \delta(B,0) &
ightarrow B \ \delta(B,1) &
ightarrow C \ \delta(C,0) &
ightarrow B \ \delta(C,1) &
ightarrow D \ \delta(D,0) &
ightarrow D \ \delta(D,1) &
ightarrow D \end{aligned}$$

```
1 #include<iostream>
2
3
   using namespace std;
4
   bool check_string(string str){
5
6
       for(char c : str){
           if(c != '0' && c != '1'){
7
               return false;
8
            }
9
        }
10
11
       return true;
12
   }
13
   bool is accepted(string str){
14
        char state = 'A';
15
16
       for(char c : str){
            switch(state){
17
18
                case 'A':
                    state = (c == '0') ? 'B' : 'A';
19
                    break;
20
21
                case 'B':
                    state = (c == '0') ? 'B' : 'C';
22
23
                   break;
                case 'C':
24
                    state = (c == '0') ? 'B' : 'D';
25
26
                    break;
27
                case 'D':
                    state = (c == '0') ? 'D' : 'D';
28
                    break;
29
            }
30
        }
31
32
       return state == 'D';
33 }
34
35 int main(){
        string str;
36
       char choice;
37
        do{
38
           cout << "Enter a string: ";</pre>
39
40
           cin >> str;
            if(check_string(str)){
41
```

```
cout << (is_accepted(str) ? "Accepted" : "Not Accepted") <<</pre>
42
    endl;
            }else{
43
                cout << "Invalid Input" << endl;</pre>
44
45
           cout << "Do you want to continue? (y/n): ";</pre>
46
           cin >> choice;
47
        }while(choice == 'y');
48
        return 0;
49
50 }
```