

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:

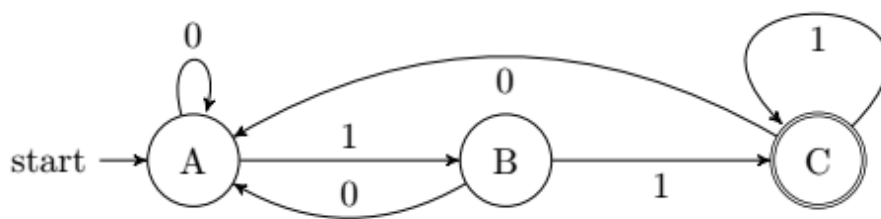


Figure 1: DFA to accept strings ending with 11

2. State Table:

State/input	0	1
A	A	B
B	A	C
C	A	C

3. State Function:

$$\delta(A, 0) \rightarrow A$$

$$\delta(A, 1) \rightarrow B$$

$$\delta(B, 0) \rightarrow A$$

$$\delta(B, 1) \rightarrow C$$

$$\delta(C, 0) \rightarrow A$$

$$\delta(C, 1) \rightarrow C$$

4. Code:

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

```

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

```

Question 2

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

Solution:

1. State Diagram:

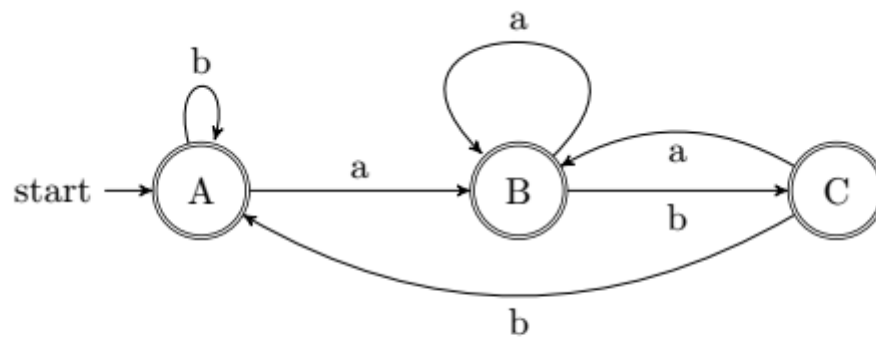


Figure 2: DFA to accept strings not ending with ab

2. State Table:

State/input	a	b
A	B	A
B	B	C
C	B	A

3. State Function:

$$\delta(A, 0) \rightarrow B$$

$$\delta(A, 1) \rightarrow A$$

$$\delta(B, 0) \rightarrow B$$

$$\delta(B, 1) \rightarrow C$$

$$\delta(C, 0) \rightarrow B$$

$$\delta(C, 1) \rightarrow A$$

4. Code:

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

```

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

```

Question 3

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

Solution:

1. State Diagram:

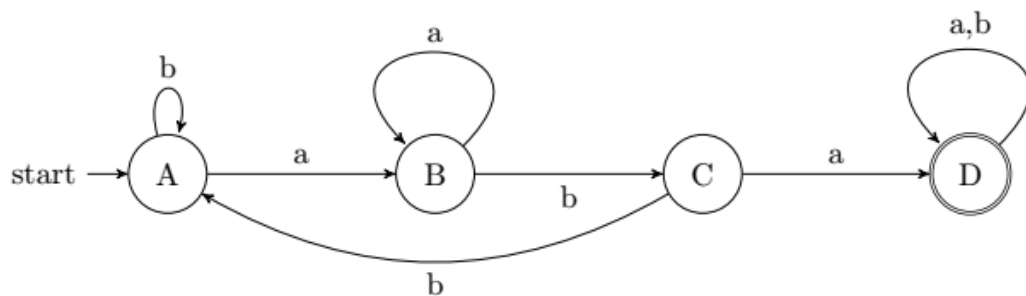


Figure 4: DFA to accept strings having aba as substring

2. State Table

State/Input	a	b
A	B	A
B	B	C
C	D	A
D	D	D

3. State Function

$$\delta(A, a) \rightarrow B$$

$$\delta(A, b) \rightarrow A$$

$$\delta(B, a) \rightarrow B$$

$$\delta(B, b) \rightarrow C$$

$$\delta(C, a) \rightarrow D$$

$$\delta(C, b) \rightarrow A$$

$$\delta(D, a) \rightarrow D$$

$$\delta(D, b) \rightarrow D$$

4. Code

```
1 #include<iostream>
2
3 using namespace std;
4
5 bool check_string(string str){
6     for(char c : str){
7         if(c != 'a' && c != 'b'){
8             return false;
9         }
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 == 'a') ? 'B' : 'A';
20                 break;
```

```

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

```

Question 4

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

Solution:

1. State Figure

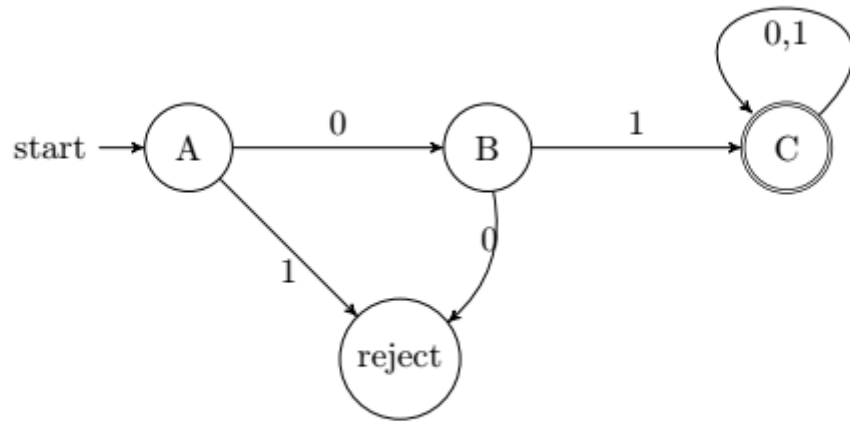


Figure 4: DFA to accept strings starting with 01

2. State Table

State/Input	0	1
A	B	reject
B	reject	C
C	C	C

3. State Function

$$\begin{aligned} \delta(A, 0) &\rightarrow B \\ \delta(A, 1) &\rightarrow \text{Reject} \\ \delta(B, 0) &\rightarrow C \\ \delta(B, 1) &\rightarrow \text{Reject} \\ \delta(C, 0) &\rightarrow C \\ \delta(C, 1) &\rightarrow C \end{aligned}$$

4. Code:

```

1  #include<iostream>
2
3  using namespace std;
4
5  bool check_string(string str){
6      for(char c : str){
7          if(c != '0' && c != '1'){
8              return false;
9          }
10     }
11     return true;

```



```

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

```

Question 5

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

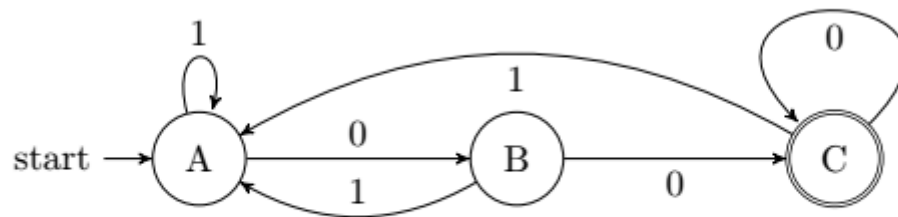


Figure 5: DFA to accept strings ending with 00

2. State Table

State/Input	0	1
A	B	A
B	C	A
C	C	A

3. State Function

$$\delta(A, 0) \rightarrow B$$

$$\delta(A, 1) \rightarrow A$$

$$\delta(B, 0) \rightarrow C$$

$$\delta(B, 1) \rightarrow A$$

$$\delta(C, 0) \rightarrow C$$

$$\delta(C, 1) \rightarrow A$$

4. Code:

```
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          if (c != '0' && c != '1') {
9              return false;
10         }
11     }
```

```

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

```

Question 6

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

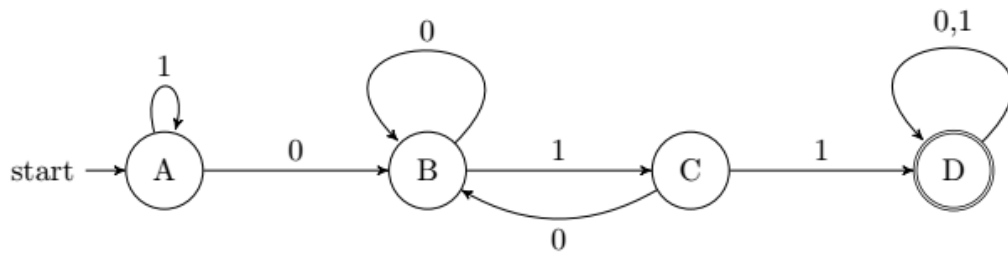


Figure 6: DFA to accept strings having aba as substring

2. State Table

State/Input	0	1
A	B	A
B	B	C
C	B	D
D	D	D

3. State Function

$$\delta(A, 0) \rightarrow B$$

$$\delta(A, 1) \rightarrow A$$

$$\delta(B, 0) \rightarrow B$$

$$\delta(B, 1) \rightarrow C$$

$$\delta(C, 0) \rightarrow B$$

$$\delta(C, 1) \rightarrow D$$

$$\delta(D, 0) \rightarrow D$$

$$\delta(D, 1) \rightarrow D$$

4. Code:

```
1 #include<iostream>
2
3 using namespace std;
4
5 bool check_string(string str){
6     for(char c : str){
```

```

7         if(c != '0' && c != '1'){
8             return false;
9         }
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 == '0') ? 'B' : 'A';
20                 break;
21             case 'B':
22                 state = (c == '0') ? 'B' : 'C';
23                 break;
24             case 'C':
25                 state = (c == '0') ? 'B' : 'D';
26                 break;
27             case 'D':
28                 state = (c == '0') ? 'D' : 'D';
29                 break;
30         }
31     }
32     return state == 'D';
33 }
34
35 int main(){
36     string str;
37     char choice;
38     do{
39         cout << "Enter a string: ";
40         cin >> str;
41         if(check_string(str)){
42             cout << (is_accepted(str) ? "Accepted" : "Not Accepted") <<
endl;
43         }else{
44             cout << "Invalid Input" << endl;
45         }
46         cout << "Do you want to continue? (y/n): ";
47         cin >> choice;
48     }while(choice == 'y');

```

```
49     return 0;  
50 }
```
