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:

State Diagram:

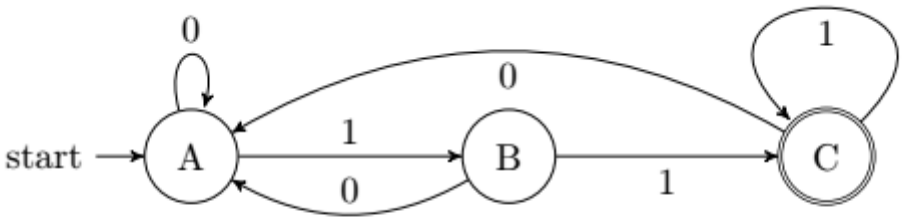

Figure 1: DFA to accept strings ending with 11

State Table:

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

State Function:

$\delta(A, 0) \rightarrow A$
$\delta(A, 1) \rightarrow B$
$\delta(B, 0) \rightarrow A$
$\delta(B, 1) \rightarrow C$
$\delta(C, 0) \rightarrow A$
$\delta(C, 1) \rightarrow C$

Code:

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

using namespace std;

bool is_valid_input(string str) {
  for (char c : str) {
    if (c != '0' && c != '1') {
      return false;
    }
  }
  return true;
}

bool is_accepted(string str) {
  char state = 'A';
  for (char c : str) {
    switch (state) {
      case 'A':
        state = (c == '0') ? 'A' : 'B';
        break;
      case 'B':
        state = (c == '0') ? 'A' : 'C';
        break;
      case 'C':
        state = (c == '0') ? 'A' : 'C';
    }
  }
  return state == 'C';
}

int main() {
  string str;
  char choice;
  do {
    cout << "Enter a string: ";
    cin >> str;
    if (is_valid_input(str)) {
      cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;
    } else {
      cout << "Invalid Input" << endl;
```

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

Question 2

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

Solution:

State Diagram:

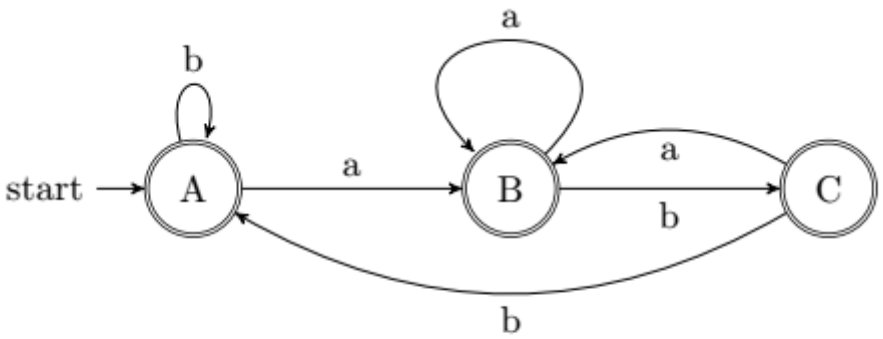

Figure 2: DFA to accept strings not ending with ab

State Table:

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

State Function:

$$\delta(A, 0) \to B$$
$$\delta(A, 1) \to A$$
$$\delta(B, 0) \to B$$
$$\delta(B, 1) \to C$$
$$\delta(C, 0) \to B$$
$$\delta(C, 1) \to A$$

Code:

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

using namespace std;

bool is_valid_input(string str)
{
  for (char c : str)
  {
    if (c != 'a' && c != 'b')
    {
      return false;
    }
  }
  return true;
}

bool is_accepted(string str)
{
  char state = 'A';
  for (char c : str)
  {
    switch (state)
    {
    case 'A':
      state = (c == 'a') ? 'B' : 'A';
      break;
    case 'B':
      state = (c == 'a') ? 'B' : 'C';
      break;
    case 'C':
      state = (c == 'a') ? 'B' : 'A';
      break;
    }
```

```
  }
  return state != 'C';
}

int main()
{
  string str;
  char choice;
  do
  {
    cout << "Enter a string: ";
    cin >> str;

    if (is_valid_input(str))
    {
      cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;
    }
    else
    {
      cout << "Invalid Input" << endl;
    }
    cout << "Do you want to continue? (y/n): ";
    cin >> choice;
  } while (choice == 'y');
  return 0;
}
```

Question 3

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

Solution

State Diagram:

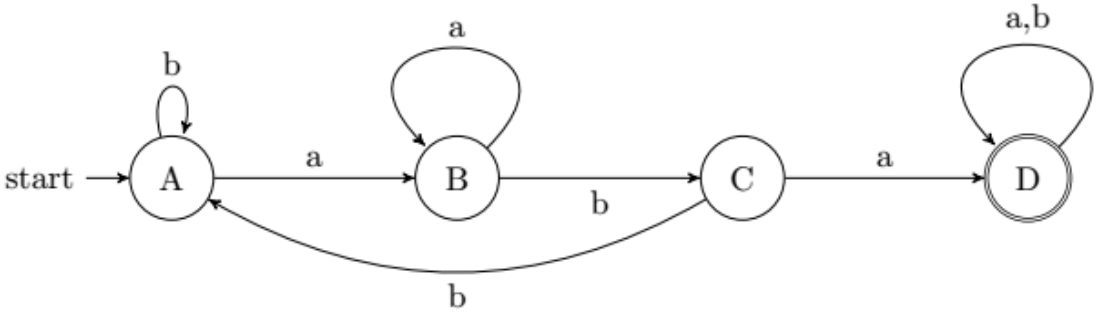

Figure 4: DFA to accept strings having aba as substring

State Table

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

State Function

$\delta(A, a) \rightarrow B$
$\delta(A, b) \rightarrow A$
$\delta(B, a) \rightarrow B$
$\delta(B, b) \rightarrow C$
$\delta(C, a) \rightarrow D$
$\delta(C, b) \rightarrow A$
$\delta(D, a) \rightarrow D$
$\delta(D, b) \rightarrow D$

Code

```
#include<iostream>

using namespace std;

bool check_string(string str){
    for(char c : str){
        if(c != 'a' && c != 'b'){
```

```
            return false;
        }
    }
    return true;
}

bool is_accepted(string str){
    char state = 'A';
    for(char c : str){
        switch(state){
            case 'A':
                state = (c == 'a') ? 'B' : 'A';
                break;
            case 'B':
                state = (c == 'a') ? 'B' : 'C';
                break;
            case 'C':
                state = (c == 'a') ? 'D' : 'A';
                break;
            case 'D':
                state = (c == 'a') ? 'D' : 'D';
                break;
        }
    }
    return state == 'D';
}

int main(){
    string str;
    char choice;
    do{
        cout << "Enter a string: ";
        cin >> str;
        if(check_string(str)){
            cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;
        }else{
            cout << "Invalid Input" << endl;
        }
        cout << "Do you want to continue? (y/n): ";
        cin >> choice;
    }while(choice == 'y');
    return 0;
}
```

Question 4

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

Solution

State Figure

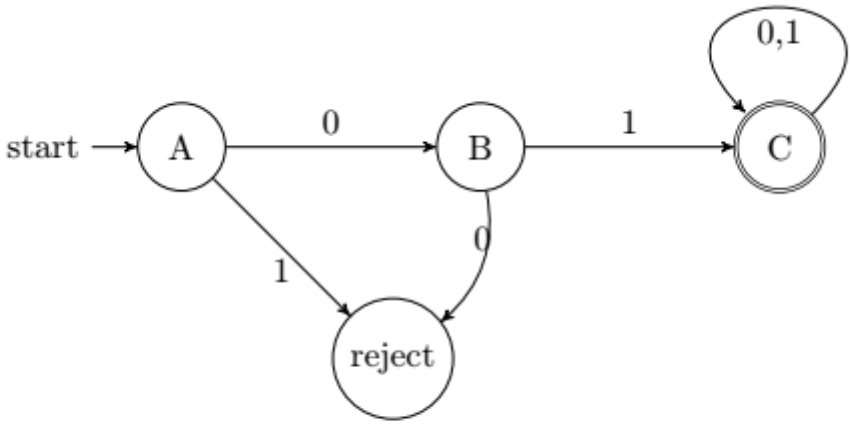

Figure 4: DFA to accept strings starting with 01

State Table

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

State Function

$$\delta(A, 0) \rightarrow B$$
$$\delta(A, 1) \rightarrow Reject$$
$$\delta(B, 0) \rightarrow C$$
$$\delta(B, 1) \rightarrow Reject$$
$$\delta(C, 0) \rightarrow C$$
$$\delta(C, 1) \rightarrow C$$

Code:

```cpp
#include<iostream>

using namespace std;

bool check_string(string str){
    for(char c : str){
        if(c != '0' && c != '1'){
            return false;
        }
    }
    return true;
}

bool is_accepted(string str){
    char state = 'A';
    for(char c: str){
        //state r means the reject state in this case
        switch(state){
            case 'A':
                state = (c == '0') ? 'B' : 'R';
                break;
            case 'B':
                state = (c == '1') ? 'C' : 'R';
                break;
            case 'C':
                state = (c == '0') ? 'C' : 'C';
        }
    }
    return state=='C';
}

int main(){
    string str;
    char choice;
    do{
        cout << "Enter a string: ";
        cin >> str;
        if(check_string(str)){
            cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;
        }else{
            cout << "Invalid Input" << endl;
```

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

Question 5

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

Solution

State Diagram

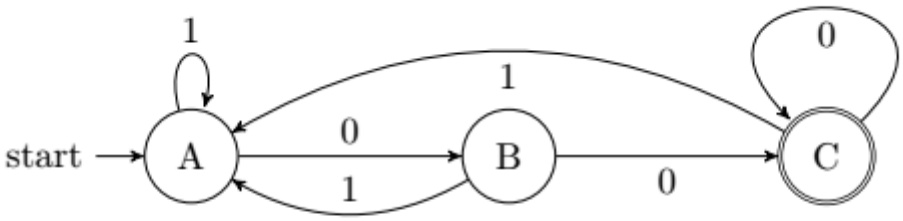

Figure 5: DFA to accept strings ending with 00

State Table

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

State Function

$\delta(A, 0) \rightarrow B$
$\delta(A, 1) \rightarrow A$
$\delta(B, 0) \rightarrow C$
$\delta(B, 1) \rightarrow A$
$\delta(C, 0) \rightarrow C$
$\delta(C, 1) \rightarrow A$

Code:

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

using namespace std;

bool is_valid_input(string str) {
  for (char c : str) {
    if (c != '0' && c != '1') {
      return false;
    }
  }
  return true;
}

bool is_accepted(string str) {
  char state = 'A';
  for (char c : str) {
    switch (state) {
      case 'A':
        state = (c == '1') ? 'A' : 'B';
        break;
      case 'B':
        state = (c == '1') ? 'A' : 'C';
        break;
      case 'C':
        state = (c == '1') ? 'A' : 'C';
    }
  }
  return state == 'C';
}

int main() {
  string str;
  char choice;
  do {
    cout << "Enter a string: ";
    cin >> str;
    if (is_valid_input(str)) {
      cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;
    } else {
      cout << "Invalid Input" << endl;
```

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

Question 6

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

Solution

State Diagram

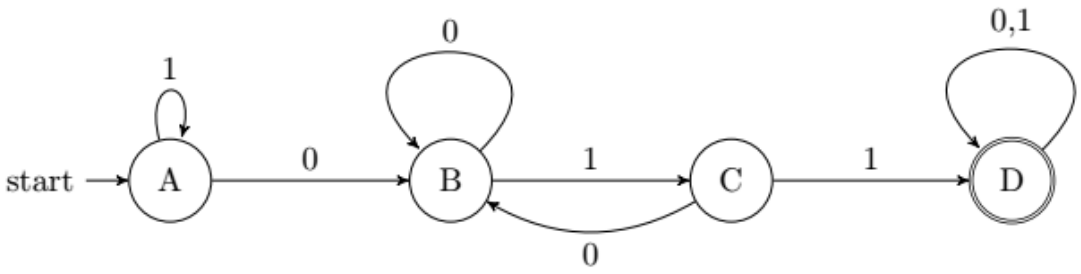

Figure 6: DFA to accept strings having aba as substring

State Table

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

State Function

$$\delta(A, 0) \to B$$
$$\delta(A, 1) \to A$$
$$\delta(B, 0) \to B$$
$$\delta(B, 1) \to C$$
$$\delta(C, 0) \to B$$
$$\delta(C, 1) \to D$$
$$\delta(D, 0) \to D$$
$$\delta(D, 1) \to D$$

Code:

```cpp
#include<iostream>

using namespace std;

bool check_string(string str){
    for(char c : str){
        if(c != '0' && c != '1'){
            return false;
        }
    }
    return true;
}

bool is_accepted(string str){
    char state = 'A';
    for(char c : str){
        switch(state){
            case 'A':
                state = (c == '0') ? 'B' : 'A';
                break;
            case 'B':
                state = (c == '0') ? 'B' : 'C';
                break;
            case 'C':
                state = (c == '0') ? 'B' : 'D';
                break;
            case 'D':
                state = (c == '0') ? 'D' : 'D';
                break;
        }
    }
```

```
    return state == 'D';
}

int main(){
    string str;
    char choice;
    do{
        cout << "Enter a string: ";
        cin >> str;
        if(check_string(str)){
            cout << (is_accepted(str) ? "Accepted" : "Not Accepted") << endl;
        }else{
            cout << "Invalid Input" << endl;
        }
        cout << "Do you want to continue? (y/n): ";
        cin >> choice;
    }while(choice == 'y');
    return 0;
}
```
