Week8 : Automata ¶

TANISHA BISHT RA1911003010259

Q1

Construction of a minimal NFA accepting a set of strings over {a, b} in which each string of the language starts with 'ab'.

In [9]:

```
from automata.fa.nfa import NFA
 1
 2
    nfa = NFA(
        states={'A', 'B', 'C'},
 3
 4
        input_symbols={'a', 'b'},
 5
        transitions={
            'A': {'a': {'B'}},
 6
            'B': {'b': {'C'}},
 7
 8
            'C': {'a':{'C'}, 'b':{'C'}}
 9
10
        initial_state='A',
11
        final_states={'C'}
12
    )
13
    for i in range(2):
14
15
        num = input("Enter the string :")
        if(nfa.accepts_input(num)):
16
17
            print("Accepted :: as the string starts with 'ab'")
        else:
18
            print("Rejected :: as the string does not start with 'ab'")
19
```

```
Enter the string :ababababab
Accepted :: as the string starts with 'ab'
Enter the string :bbbbb
Rejected :: as the string does not start with 'ab'
```

Q2

Construction of a minimal NFA accepting a set of strings over {a, b} in which each string of the language does not start with 'ab'

In [10]:

```
from automata.fa.nfa import NFA
 2
    nfa = NFA(
 3
        states={'A', 'B', 'C'},
 4
        input_symbols={'a', 'b'},
 5
        transitions={
 6
            'A': {'a': {'B'}},
 7
            'B': {'b': {'C'}},
            'C': {'a':{'C'}, 'b':{'C'}}
 8
 9
        },
        initial state='A',
10
11
        final_states={'C'}
12
13
14
    for i in range(2):
        num = input("Enter the string :")
15
16
        if nfa.accepts_input(num):
            print("Rejected :: as the string starts with 'ab'")
17
18
        else:
            print("Accepted :: as the string does not start with 'ab'")
19
```

```
Enter the string :bbbb
Accepted :: as the string does not start with 'ab'
Enter the string :ababa
Rejected :: as the string starts with 'ab'
```

Q3

Construct a DFA for the set of string over {a, b} such that length of the string |w|<=2 i.e, length of the string

In [11]:

```
from automata.fa.nfa import NFA
 2
    nfa = NFA(
 3
        states={'A', 'B', 'C'},
        input_symbols={'a', 'b'},
 4
 5
        transitions={
            'A': {'a':{'B'}, 'b':{'B'}},
 6
 7
            'B': {'a':{'C'}, 'b':{'C'}},
            'C': {'a':{'C'}, 'b':{'C'}}
 8
 9
        },
10
        initial state='A',
11
        final_states={'C'}
12
    )
13
    for i in range(2):
14
15
        num = input("Enter the string :")
        if nfa.accepts_input(num):
16
            print("Accepted :: as the string is greater than or equal to 2")
17
18
        else:
            print("Rejected :: as the string is less than 2")
19
```

```
Enter the string :ababababababbbbbb
Accepted :: as the string is greater than or equal to 2
Enter the string :a
Rejected :: as the string is less than 2
```

In []:

1