

# 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]:

```
1 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'}},
8         'C': {'a': {'C'}, 'b': {'C'}}
9     },
10    initial_state='A',
11    final_states={'C'}
12 )
13
14 for i in range(2):
15     num = input("Enter the string :")
16     if(nfa.accepts_input(num)):
17         print("Accepted :: as the string starts with 'ab'")
18     else:
19         print("Rejected :: as the string does not start with 'ab'")
```

Enter the string :ababababab

Accepted :: as the string starts with 'ab'

Enter the string :bbbbbb

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]:

```

1  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'}},
8          'C': {'a': {'C'}, 'b': {'C'}}
9      },
10     initial_state='A',
11     final_states={'C'}
12 )
13
14 for i in range(2):
15     num = input("Enter the string :")
16     if nfa.accepts_input(num):
17         print("Rejected :: as the string starts with 'ab'")
18     else:
19         print("Accepted :: as the string does not start with 'ab'")

```

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  $|w| \leq 2$  i.e, length of the string

In [11]:

```

1  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'}, 'b': {'B'}},
7          'B': {'a': {'C'}, 'b': {'C'}},
8          'C': {'a': {'C'}, 'b': {'C'}}
9      },
10     initial_state='A',
11     final_states={'C'}
12 )
13
14 for i in range(2):
15     num = input("Enter the string :")
16     if nfa.accepts_input(num):
17         print("Accepted :: as the string is greater than or equal to 2")
18     else:
19         print("Rejected :: as the string is less than 2")

```

Enter the string :abababababbbbbbb  
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	
---	--