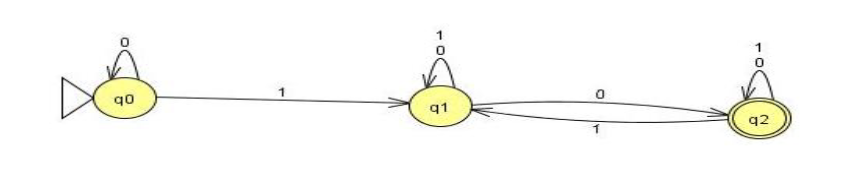
# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Assignment No: 5 (NFA TO DFA & REGULAR EXPRESSION)

**Q) Convert the following NFA to DFA**

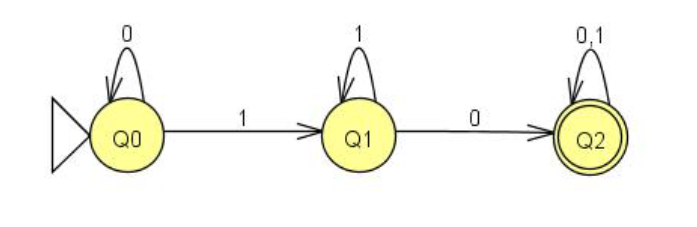
NFA:



Transition Table:

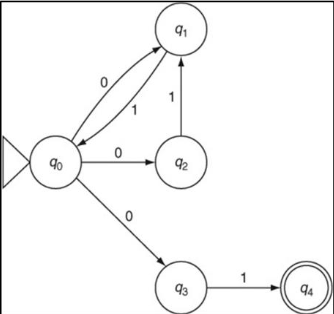
|  |  |  |
| --- | --- | --- |
| Q | Input | |
| 0 | 1 |
| Q0 = {q0} | {q0} = Q0 | {q1} = Q1 |
| Q1 = {q1} | {q1, q2} = Q2 | {q1} = Q1 |
| Q2 = {q1, q2} | {q1, q2} = Q2 | {q1, q2} = Q2 |

DFA:



**Q) Convert the following NFA to DFA**

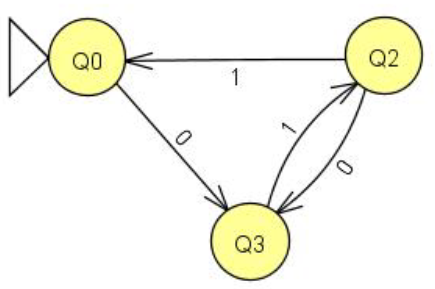
NFA:



Transition Table:

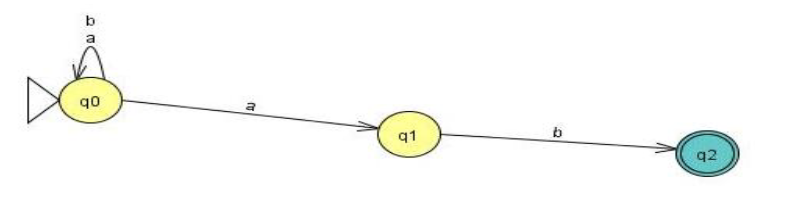
|  |  |  |
| --- | --- | --- |
| Q | Input | |
| 0 | 1 |
| Q0 = {q0} | {q1, q2, q3} = Q3 | - |
| Q3 = {q1, q2, q3} | - | {q0, q1, q4} = Q2 |
| Q2 = {q0, q1, q4} | {q1, q2, q3} = Q3 | {q0} = Q0 |

DFA:



**Q) Convert the following NFA to DFA**

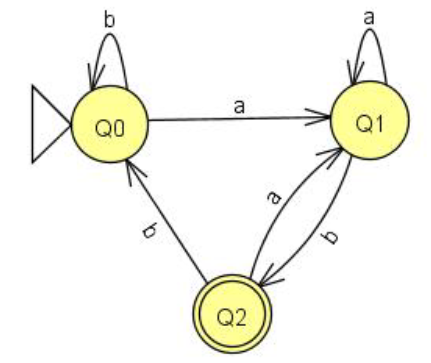
NFA:



Transition Table:

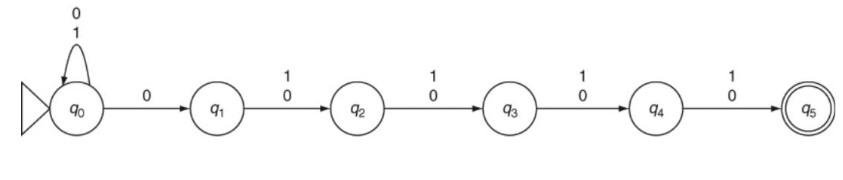
|  |  |  |
| --- | --- | --- |
| Q | Input | |
| a | b |
| Q0 = {q0} | {q0, q1} = Q1 | {q0} = Q0 |
| Q1 = {q0, q1} | {q0, q1} = Q1 | {q0, q2} = Q2 |
| Q2 = {q0, q2} | {q0, q1} = Q1 | {q0} = Q0 |

DFA:



**Q) Draw NFA with string over {0,1} with 0 as the fifth last symbol.**

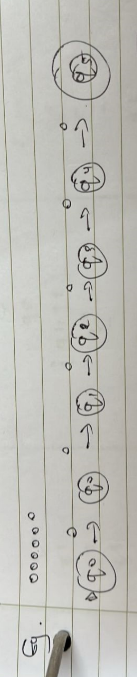
NFA:



Transition Table:

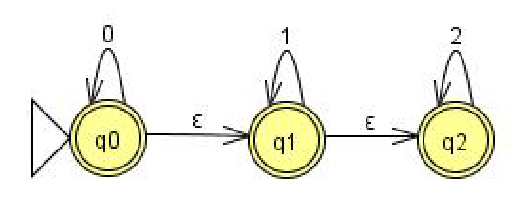
|  |  |  |
| --- | --- | --- |
| Q | Input | |
| 0 | 1 |
| Q0 | {q0,q1} | Q1 |
| Q1 | Q2 | Q2 |
| Q2 | Q3 | Q3 |
| Q3 | Q4 | Q4 |
| Q4 | Q5 | Q5 |
| Q5 | - | - |

Example:



**Q) Construct NFA with Ɛ moves for “Zero or more number of 0’s followed by zero or more number of 1’s followed by zero or more number of 2’s . Convert this DFA**

NFA:



E - Closure :- (q0) = {q0, q1, q2}

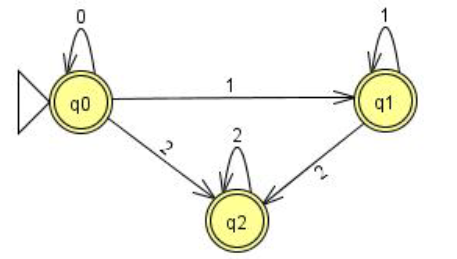
(q1) = {q1, q2}

(q2) = {q2}

Transition Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Q | Closure | Input | | |
|  |  | 0 | 1 | 2 |
| Q0 = {q0} | {q0, q1, q2} | q0, q1, q2} = Q0 | {q1, q2} = Q1 | {q2} = Q2 |
| Q1 = {q1, q2} | {q1, q2} | - | {q1, q2} = Q1 | {q2} = Q2 |
| Q2 = {q2} | {q2} | - | - | {q2} = Q2 |

Example:



Q Explain the following Regular expression in your own words.

a. (0+1)\* 011(0+1)\*

Matches any binary string containing the substring 011.

b. [(0+1)\* 000 (0+1)\* 010 (0+1)\*]

Matches binary strings containing both 000 and 010 as substrings.

c. (aa)\*(bb)\*+a(aa)\*b(bb)\*

Matches strings with an even number of a's followed by an even number of b's, or strings starting with a, followed by an even number of a's, then b, and finally an even number of b's.

Q. Give applications of regular expressions and FA

Regular Expression:

* **Pattern Matching**
* **Input Validation**.
* **Syntax Highlighting**
* **Database Queries**

Finite Automata:

* **Text Processing**.
* Network protocols.
* **String Matching Algorithms**

Q. Give Regular Expressions for:

a. For all strings over a,b which contains even number of a’s followed by odd number of b’s

(aa)\*a(b(bb)\*)

b. For all strings over 0,1 that start and end with different letters.

0(0+1)\*1 + 1(0+1)\*0

c. Language consisting of all strings ∑= {a,b} with at least one consecutive ‘a’ ie “aa” .

(a+b)\*aa(a+b)\*

d. For all strings over {0, 1} having no consecutive 1s.

(0+10)\*(ε+1)

e. For all strings over {0, 1} containing the sequence 011

(0+1)\*011(0+1)\*

f. For all strings over {a, b} whose length is a multiple of 3.

((a+b) (a+b) (a+b))\*

g. For all strings over {0, 1} containing no more than two 0s.

1\*(0+ε)1\*(0+ε)1\*

h. Binary even number

(0+1)\*0

i. All strings that begin with a’s and end with b’s and are of even length ,∑= {a,b}

a(a+b)\*b

j. All strings with exactly two a’s , ∑= {a,b}

b\*ab\*ab\*

k. All string with at least one ‘a’ and one ‘b’

((a+b)\*a(a+b)\*b(a+b)\*+(a+b)\*b(a+b)\*a(a+b)\*)

l. All strings over ∑= {a,b}, whose length is multiple of 3

((a+b) (a+b) (a+b))\*