INDEX

Name Suthesan V.I Class ..... Div. ..... Roll No. 192210542

School Name..... Sub. TOC

S. No.	Date	Particlular	Page No.	Signature
1	13/3/24	DFA to accept string start with 0 end with 1	9	
2	13/3/24	DFA, Accepted or not.	9	
3	13/3/24	NFA which accepts 3 consecutive 0's	9	
4	13/3/24	DFA with even no. of 0's followed with 1	8	
5	13/3/24	NFA with {0,1} accepts "01"	8	
6	13/3/24	DFA which even no. of 0's ends with 1	9	
7	13/3/24	NFA with epsilon regular expression	9	
8	13/3/24	Finite automata for regular expression	8	
9	13/3/24	regular expression for given DFA	9	
10	13/3/24	NFA to DFA Conversion.	9	

1) Design a DFA to accept string which start with 0 & end with 1

Definition:

From each state for each input symbol, there is exactly one transition.

Aim:

To design a DFA to accept string which start with 0 & end with 1.

Procedure:

step 1: Declare 5 tuple and define the input/output values

step 2: Draw rough diagram with the given data.

step 3: With the help of diagram, Draw transition table.

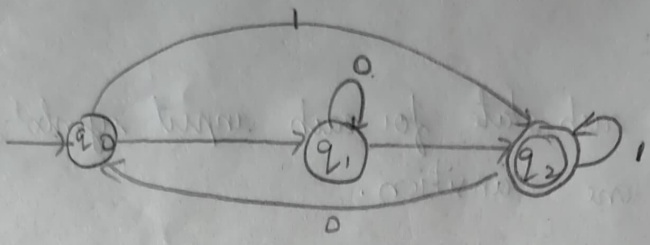
step 4: with the help of transition table draw the circuit and simulate it with the help of software.

step 5: Declare all the required state and select the type which is going to be created and Test whether the given input/output condition is satisfied.

step 6: If yes, then it gets accepted, if no, then its get rejected.

Diagram:

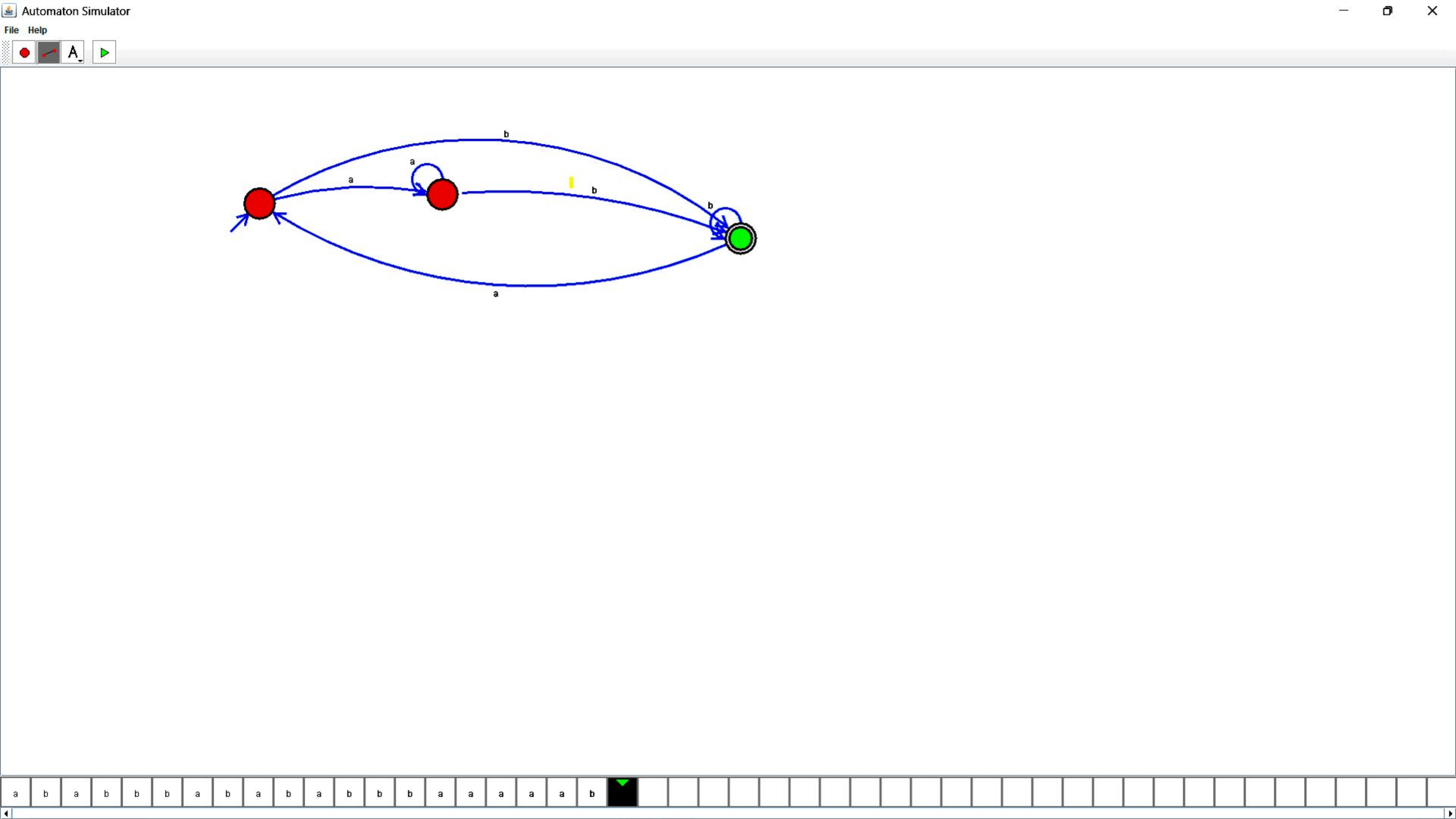
$L = \{01, 001, 101, 0001, 0011 \ldots\}$



Transition table:

States	0	1
q_0	q_1	q_2
q_1	q_1	q_2
q_2	q_0	q_2

Result: Designing a DFA to accept the string start with 0 and end with 1 is created & executed successfully.


Automaton Simulator
File Help
b
a
a
b
b
a
a b a b b b a b a b a b b b a a a a a b

2) Consider DFA $M = \{Q, \Sigma, \delta, q_0, F\}$ where

$Q = \{A, B, C\}$ $\Sigma = \{0, 1\}$ δ is A is initial C is final

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

$\delta(B, 0) = B$ $\delta(B, 1) = C$

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

i) Draw DFA ii) check string "100110" accepted or not iii) "110" accepted or not.

AIM: To construct DFA & checking strings are accepted or not

DFA: From each state for each & every input symbol there will be exactly one transition.

Procedure:

step 1: Five tuples for DFA $M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{A, B, C\}$. $\Sigma = \{0, 1\}$. $q_0 = A$, $F = C$.

2) Give input 0 from A to B & for input 1 from A to C

3) Give input 0 from B to B & for input 1 from B to C

4) Give input 0 from C to A & for input 1 from C to B

5) To implement this in Automation simulator.

6) Get states A, B, C after opening software

7) Give transitions according to diagram

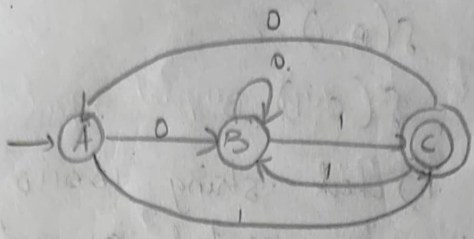
8) Give input string & check whether it goes to destination or not

a) If it goes to destination, that is accepted

b) otherwise rejected.

Diagram:

(i)



(ii) $\delta(A, 100110) = \delta(C, 00110)$

$= \delta(A, 0110)$

$= \delta(B, 110)$

$= \delta(C, 10)$

$= \delta(B, 0)$

$= B$

B is not final states. So the string "100110" is not accepted by the DFA.

iii) $\delta(A, 110) = \delta(C, 10)$

$= \delta(B, 0)$

$= B$

B is not final state. So the string "110" is not accepted by the DFA

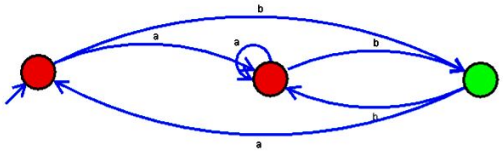
Transition table:

States	0	1
A	B	C
B	B	C
C	A	B

Result: DFA creation and check whether accepted or not is executed successfully.

Automaton Simulator: 1

File Help



a	b	b	b	a	a	a	b	b	b	a	a	a	a	b	a	b

3) Construct NFA which accepts set of all strings with 3 consecutive 0's

Aim: To construct DFA which accepts set of all strings with 3 consecutive 0's.

DFA: From each state & for every input symbol there will be one or more transition.

PROCEDURE:

1) Five tuples for DFA $M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}$, $q_0 = q_0$, $F = q_3$

2) Give input 0 from q_0 to q_0 & q_1 for input 1 from q_0 to q_0.

3) Give input to from q_1 to q_2 & for input 1 from q_1 to q_0

4) Give input 0 from q_2 to q_3 & for input 1 from q_2 to q_0

5) Give input 0 from q_3 to q_3 & for input 1 from q_3 to q_3

6) To implement this in Automation simulator

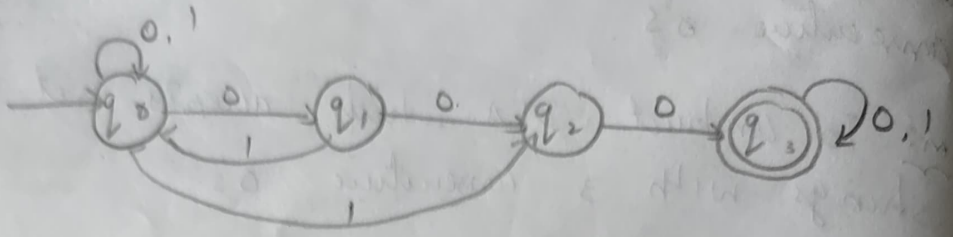
7) Get states q_0, q_1, q_2, q_3 after opening s/w

8) Give transitions according to diagram.

9) Give input string & check whether if goes to destination or not.

10) If it goes to destination, that is accepted otherwise rejected.

Diagram:



Transition table:

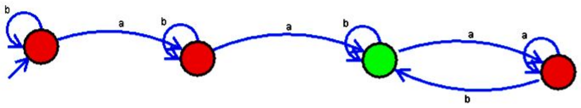
States	0	1
q_0	q_0, q_1	q_0
q_1	q_2	q_0
q_2	q_3	q_0
q_3	q_3	q_3

Result:

To construct NFA which accepts set of all strings with 3 consecutive 0's is created & executed successfully.

Automaton Simulator: 2

File Help



a	a	a	b	b	a	b	b	b	a	a	b	b	b

4) Construct DFA which accepts even no. of 0's followed by single 1

AIM: To construct DFA which accepts even no. of 0's & single 1.

DFA: From each state & for every input symbol there will be exactly one transition.

Procedure:

1) Five tuples for DFA $M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2, q_3\}$. $\Sigma = \{0, 1\}$. $q_0 = q_0$ $F = q_1$.

2) Give input 0 from q_0 to q_1 & for input 1 from q_0 to q_2.

3) Give input 0 from q_1 to q_0 & for input 1 from q_1 to q_3

4) Give input 0 from q_2 to q_3 & for input 1 from q_2 to q_0.

5) Give input 0 from q_3 to q_2 for input 1 from q_3 to q_1.

6) To implemented this in Automata simulator

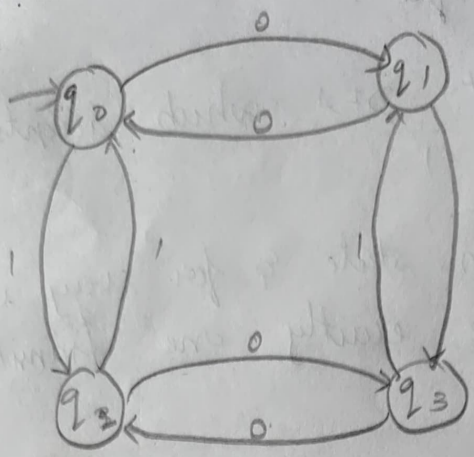
7) Get states q_0, q_1, q_2, q_3 after opening s/w

8) Give transitions according to diagram.

9) Give input string & check whether it goes to destination or not

10) If it goes to destination, that is accepted otherwise rejected.

Diagram:



Transition table:

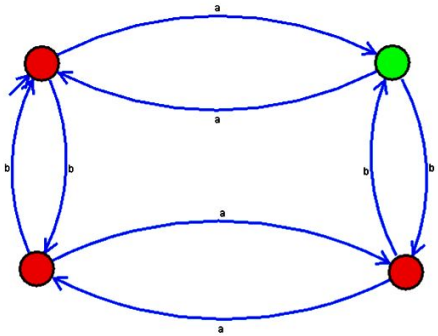
states	0	1
q_0	q_1	q_2
q_1	q_0	q_3
q_2	q_3	q_0
q_3	q_2	q_1

Result:

Construction of DFA which accepts even no. of 0's and single 1 is created & executed successfully

Automaton Simulator: 2

File Help



a	b	a	b	a	b	a	b	a	b	a	b	a

5) Draw NFA with input symbols {0,1} which accepts all the strings "01".

Aim:

To draw NFA with input symbols 0, 1 which accepts all the strings "01".

NFA: From each state for each input symbol we can have 0 or more transitions.

Procedure:

1) Five tuples for NFA $M=\{Q, \Sigma, \delta, q_0, F\}$

$Q=\{q_0, q_1, q_2\}$. $\Sigma=\{0,1\}$. $q_0=q_0$ $F=q_2$.

2) Give input 0 from q_0 to q_0 & q_1, for input 1 from q_0 to q_1

3) Give input 01 from q_1 to q_2

4) Give inputs 0 & 1 from q_2 to q_2 itself.

5) To implement this in Automation Simulator.

6) Get states q_0, q_1, q_2 after opening s/w

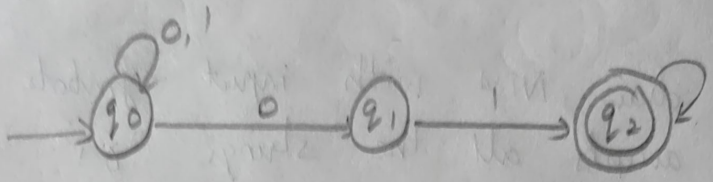
7) Give transitions according to diagram.

8) Give input string & check whether it goes to destination or not.

9) If it goes to destination, that is accepted

10) otherwise rejected.

Diagram:



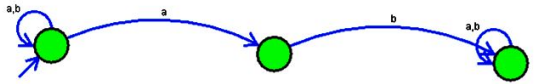
Transition table:

state	0	1
q_0	$q_0 q_1$	q_0
q_1	ϕ	q_2
q_2	q_2	q_2

Result:
NFA with input symbol {0,1} which accepts all strings "01" is created and executed successfully.

Automaton Simulator: 4

File Help



a b a b a

6) Draw the DFA which starts with even no. of 0's & ends with 1.

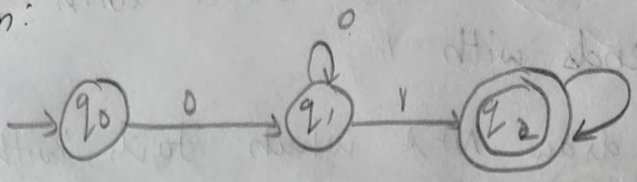
AIM: To draw NFA which starts with even no of 0's & ends with 1.

NFA: From each state for each input symbol we can have 0 or more transitions.

Procedure:

1) Five tuples for NFA $M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{q_0, q_1, q_2\}$. $\Sigma = \{0, 1\}$. $q_0 = q_0$ $F = q_2$.
2) Give input 0 from q_0 to q_1 & for input
3) Give 0 from q_1 to q_1 for input 1 from q_1 to q_2
4) Give input 1 from q_2 to q_2.
5) To implement this in Automation Simulator.
6) Get states q_0, q_1, q_2 after opening the s/w
7) Give transitions according to diagram.
8) Give input string & check whether it goes to destination or not.
9) If it goes to destination that is accepted
10) otherwise rejected.

Diagram:



Transition table:

states	0	1
q_0	q_1	ϕ
q_1	q_2	q_2
q_2	ϕ	q_2

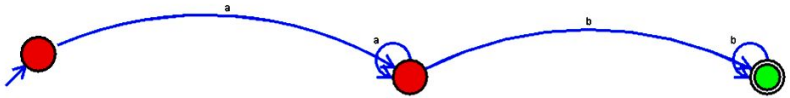
Result:

DFA which starts from even no of 0's and ends with 1 is created & executed successfully.

Automaton Simulator: 5

File Help

NFA which starts with even number of zeros and end with one



a a b

7) Draw NFA with epsilon for the regular expression a) 01+10 b) 0(0+1)* 1

AIM: To draw NFA with epsilon for the RE

NFA: From each state for each input symbol we can have 0 or more transitions.

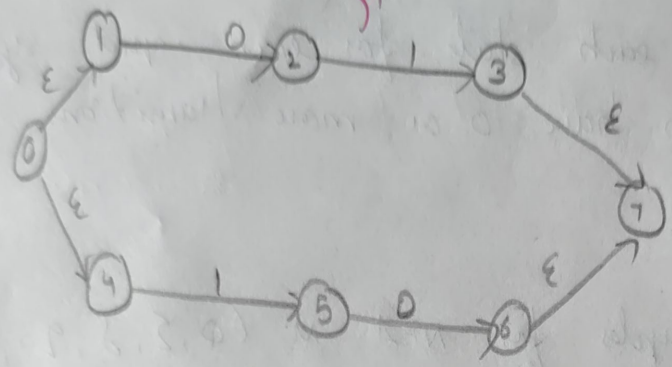
PROCEDURES:

1) Five tuples for NFA $M = (Q, \Sigma, \delta, q_0, F)$
$Q = \{0,1,2,3,4,5,6,7,8,9\}$, $\Sigma = \{0,1\}$ $q_0 = 0$, $F = 9,7$
2) First construct Regular expression for 01 & then for 10
3) Add them two, by showing in the diagram.
4) Give numbering according to diagram
5) To implement this in Automata simulator
6) Get states after opening the s/w
7) Give transitions as shown in diagram.
8) Give input string & check whether it goes to destination or not
9) If it goes to destination that is accepted
10) otherwise Rejected

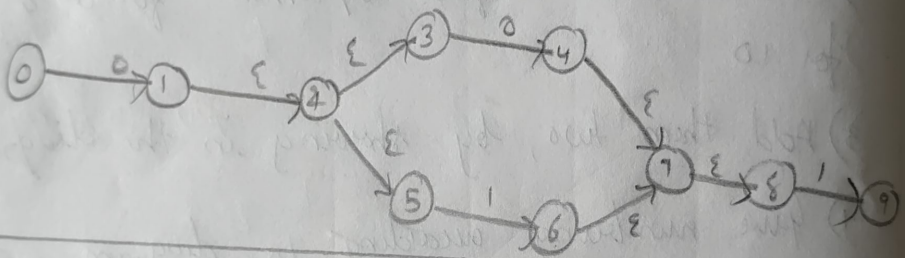
Diagram:

Step by Step?

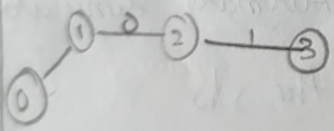
a) 01+10.



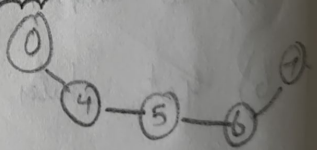
b) 0 (0+1)* 1



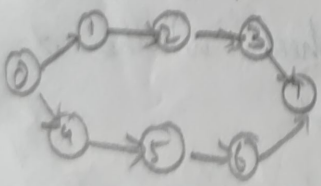
a) step 1



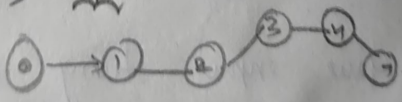
step 2:



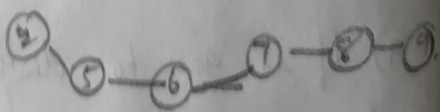
c) step 3



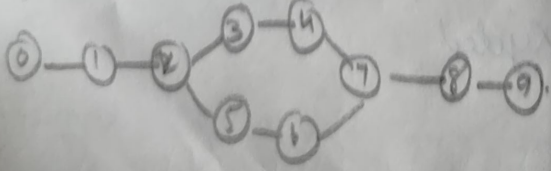
b) step 1



step 2



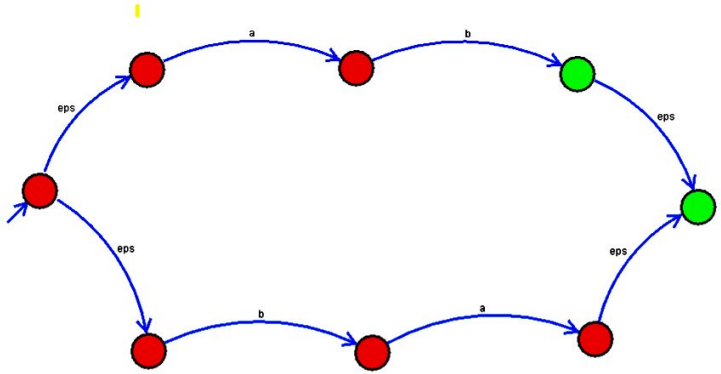
step 3



Result: NFA with epsilon regular expression is created and executed successfully.

Automaton Simulator

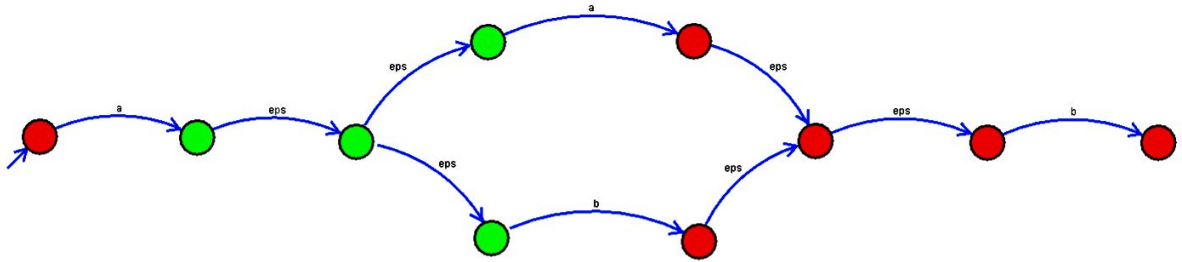
File Help



a b

Automaton Simulator

File Help



a

8) Convert finite Automata for the regular expression .0*1*

Aim: To Convert finite automata for the given regular expression.

Finite Automata:

Its a mathematical model of a system with discrete ills & o/ps.

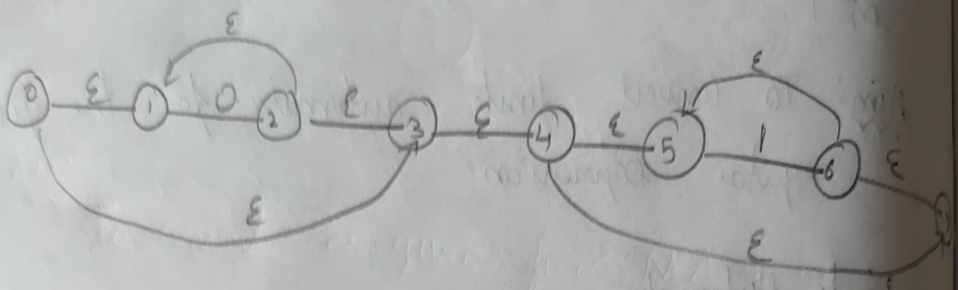
Regular Expression:

The languages accepted by finite automata can described by simple expression called "RE".

Procedure:

1) Construct regular expression for 0 with *
2) Construct regular expression for 1 with *
3) Combine these two as shown in diagram
4) To implemented this in Automata simulator
5) Goal states after opening the s/w
6) Give transition as shown in diagram
7) Give input string & check whether if goes to destination or not
8) If it goes to destination that is accepted.
9) otherwise rejected

Diagram:

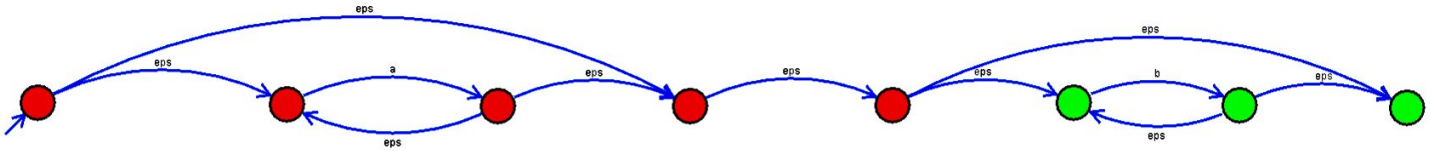


Result:

Conversion of finite Automata is created & executed successfully.

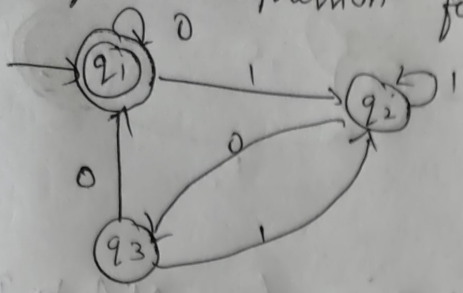
Automaton Simulator

File Help



a b

9) Find regular expression for given DFA



AIM: To find regular expression for given DFA

Arden's theorem:

If P & Q are two RE over input Σ & if P does not contain then the following equation "$R = \phi + RP$" has unique solution $\boxed{R = \phi P^*}$

$$q_1 = q_1 0 + q_3 0 \rightarrow (1)$$

$$q_2 = q_1 1 + q_2 1 + q_3 1 \rightarrow (2)$$

$$q_3 = q_2 0 \rightarrow (3)$$

sub (3) in (1) & (2)

$$q_1 = q_1 0 + q_2 0 0 \rightarrow (4)$$

$$q_2 = q_2 1 + q_1 1 + q_2 0 1 \rightarrow (5)$$

$$q_2 = q_1 1 + q_2 (1 + 01)$$

$R \quad Q \quad R \quad P \quad \Rightarrow \boxed{R = QP^*}$

$$q_2 = q_1 1 (1 + 01)^*$$

sub (6) in (1)

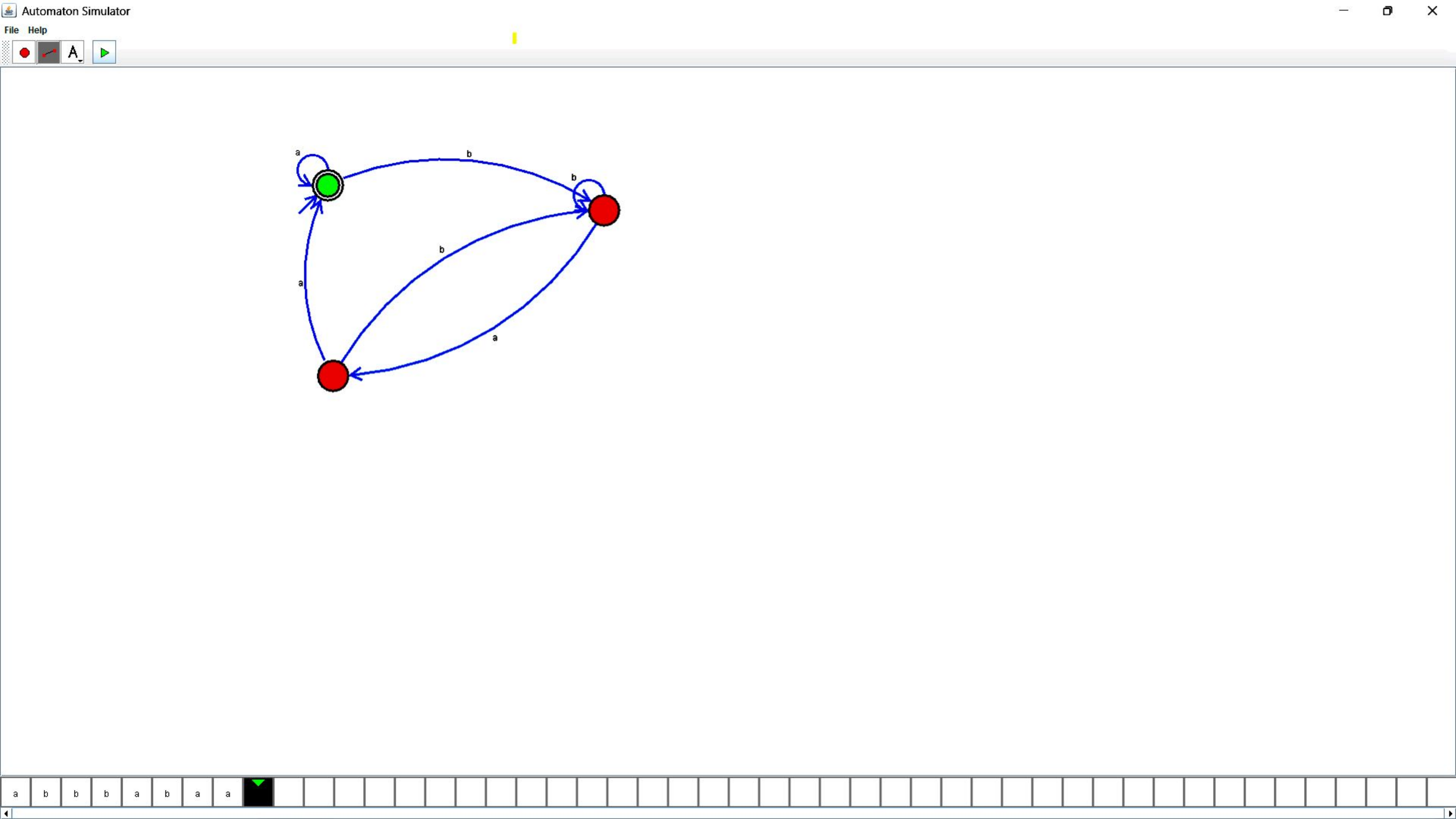
$$q_1 = q_1 0 + q_1 1 (1 + 01)^* 0$$

$$q_1 = q_1 0 (1 + 1(1 + 01)^* 0) + \varepsilon$$

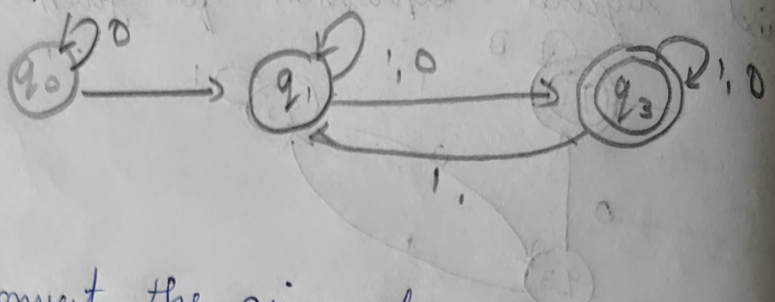
$R \quad Q \quad P \quad a$

$$q_1 = [0 (1 + 1(1 + 01)^* 0)]$$

Result: regular expression for given DFA is created and solved successfully


Automaton Simulator
File Help
a
b
b
b
a
a
a b b b a b a a

10) Convert NFA To DFA.



AIM:

To convert the given from NFA to DFA.

PROCEDURE:

$Q = q_0, q_1, q_2$

$\Sigma = 0, 1$

δ

States	0	1
q_0	q_0	q_1
q_1	q_1, q_2	q_1
q_2	q_2	q_1, q_3

$q_0 = q_0$

$\Phi = q_2$

Let M' be the DFA δ' be the New transition

$\delta'(q_0, 0) = \delta'(q_0, 0) = \{q_0\}$

$\delta'(q_0, 1) = \delta(q_0, 1) = \{q_1\}$

$\delta'(q_0, 0) = \delta(q_1, 0) = \{q_1, q_0\}$

$\delta'(q_1 q_2, 0) = \delta(q_1, 0) \cup \delta'(q_2, 0) = q_1 q_2$

$\delta'(q_1 q_2, 1) = \delta(q_1, q_2, 1) = 2, q_2$

$\delta'(q_1, 1) = \delta'(q_1, 1) = \{q_1\}$

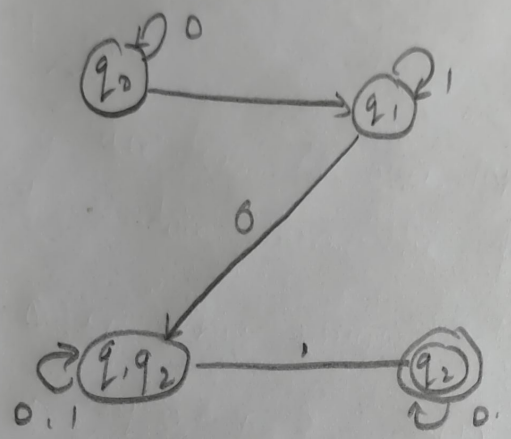
$\delta'(q_2, 0) = \delta'(q_2, 0) = \{q_2\}$

$\delta'(q_2, 0) = \delta'(q_2, 0) = q_1 q_2$

δ' =

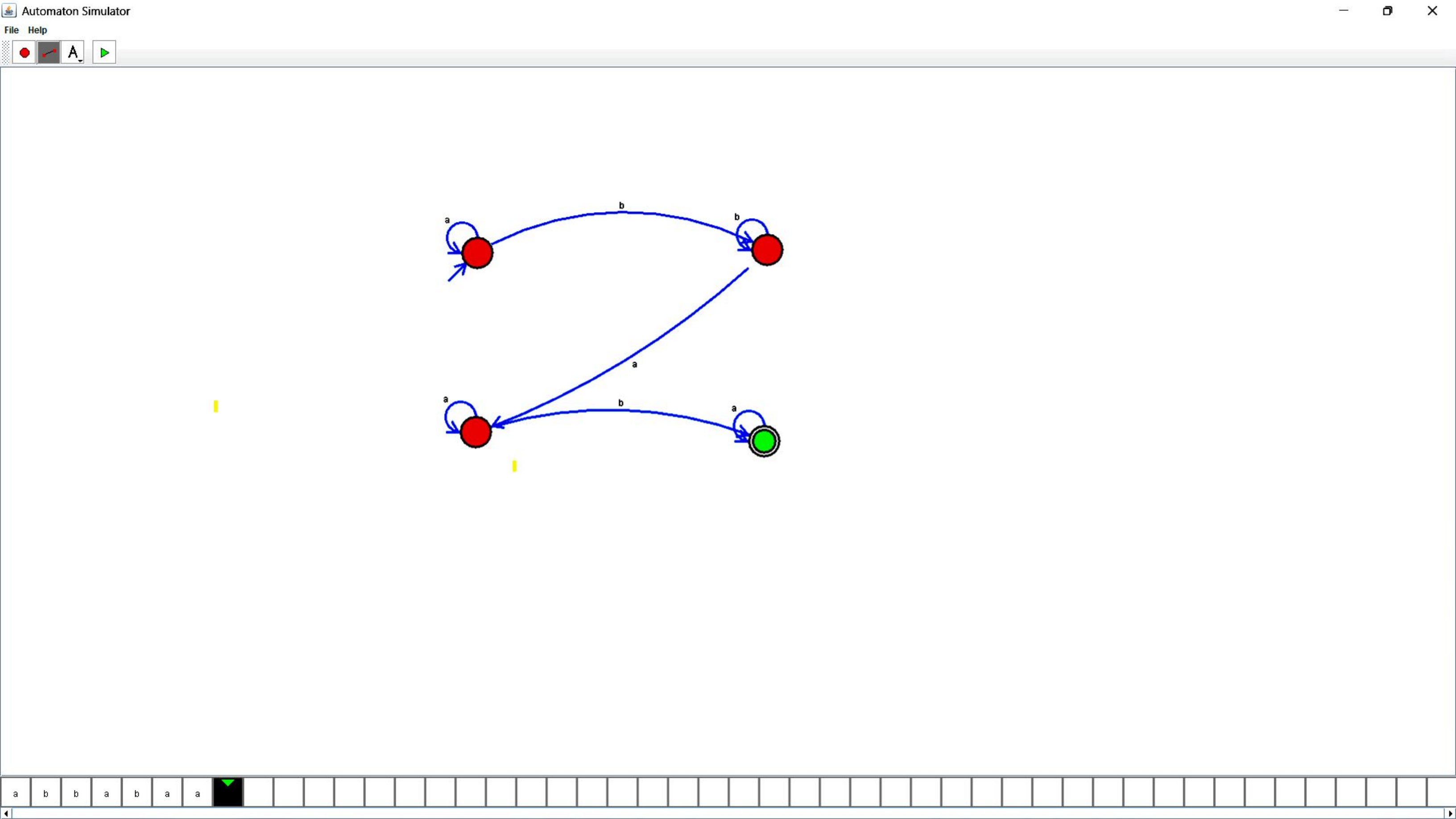
State	0	1
q_0	q_0	q_1
q_1	q_1, q_2	q_1
q_1, q_2	q_1, q_2	q_1, q_2
q_2	q_2	q_1, q_2

DFA:



Incomplete?

Result: Conversion of NFA to DFA has been created and executed successfully.


Automaton Simulator
File
Help
a
b
b
a
b
a
b
a
a
b
a
a
b
b
a
b
a
a