Homework 1 Questions

Problem 1 (5 points) answer the following questions:

a) If A has a elements and B has b elements, how many elements are in $A\text{x}B$?

- There are $a \times b$ elements in $A\text{x}B$

$A = \{1, 2, 3\}$ *Assume elements in A*

$B = \{x, y, z\}$ *Assume elements in B*

$|A| = \text{a} = 3$ *The length of A is 3, therefore $\text{a} = 3$*

$|B| = \text{b} = 3$ *The length of B is 3, therefore $\text{b} = 3$*

$A\text{x}B = \{(1, x), (1, y), (1, z), (2, x), (2, y), (2, z), (3, x), (3, y), (3, z)\}$

$|A\text{x}B| = 9$ *Listing the elements shows the length of $A\text{x}B$ is 9*

$a \times b = 3 \times 3 = 9$ *Calculating $a \times b$ also shows a length of 9*

b) If C is a set with c elements, how many elements are in the power set of C?

- The power set C is of 2^c

$C = \{0,1\}$ *Assume elements in C*

$|C| = \text{c} = 2$ *The length of C is 2, therefore $c = 2$*

$P(C) = \{\emptyset , 0, 1, (0,1)\}$

$|P(C)| = 4$ *Listing the elements shows the length of $P(C)$ is 4*

$2^c = 2^2 = 4$ *Calculating 2^c also shows a length of 4*

c) Let $R = \{(1, 2), (2, 3), (2, 4)\}$ is a relation on the set $\{1, 2, 3, 4\}$. What is R^* (Reflexive and transitive closure)?

- $R^* = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (4, 4)\}$

$R = \{(1, 2), (2, 3), (2, 4)\}$ *Elements in R*

Find Reflexive Closure of R :

$S = R \cup \{(x, x) \mid x \in x\}$ *Reflexive Closure Formula*

$S = \{(1, 2), (2, 3), (2, 4)\} \cup \{(1, 1), (2, 2), (3,3), (4,4)\}$ *Subtitute Values*

$S = \{(1, 1), (1, 2), (2, 2), (2, 3), (2, 4), (3, 3), (4, 4)\}$ *Combine Elements (Union)*

Find Transitive Closure of R :

$(a,b),(b,c) \in R \Rightarrow (a,c)$ *Transitive Property*

(a,b)	(b,c)	(a,c)
$(1,2)$	$(2,3)$	$(1,3)$
$(1,2)$	$(2,4)$	$(1,4)$
$(1,3)$	$\cancel{(3,1)}$	
$(2,3)$	$\cancel{(3,2)}$	
	$\cancel{(3,3)}$	
	$\cancel{(3,4)}$	
$(1,4)$	$\cancel{(4,1)}$	
$(2,4)$	$\cancel{(4,2)}$	
	$\cancel{(4,3)}$	
	$\cancel{(4,4)}$	
$\cancel{(1,1)}$	$(1,2)$	
$\cancel{(2,1)}$	$(1,3)$	
$\cancel{(3,1)}$	$(1,4)$	
$\cancel{(4,1)}$		

Find Missing Pairs

$R^{+} = \{(1,2),(1,3),(1,4),(2,3),(2,4)\}$ *Include Transitive Pairs*

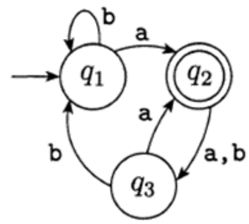
Find Reflexive and Transitive Closure of R :

$R^{*} = R^{+} \cup \{(x,x) \mid x \in S\}$ *Reflexive and Transitive Closure Formula*

$R^{*} = \{(1,2),(1,3),(1,4),(2,3),(2,4)\} \cup \{(1,1),(1,2),(2,2),(2,3),(2,4),(3,3),(4,4)\}$ *Subtitute Values*

$R^{*} = \{(1,1),(1,2),(1,3),(1,4),(2,2),(2,3),(2,4),(3,3),(4,4)\}$ *Combine Elements*

Problem 2 (5 points) Answer the following questions about the state diagrams of a DFA (M_1)



M_1

a) What is the start state?
   - q_1

b) What is the set of accept states?
   - $\{q_2\}$

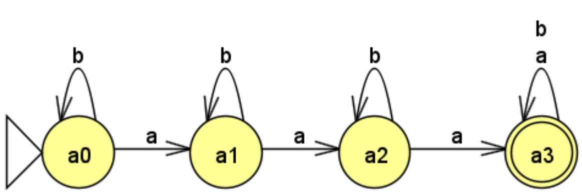
c) What sequence of states does the machine go through on input $aabb$?
   - $\{q_1, q_2, q_3, q_1, q_1\}$

d) Does the machine accept the string $aabb$?
   - No, because string $aabb$ finishes in q_1, which is not an accept state.

e) Does the machine accept the string ε?
   - No, because string ε finishes in q_1, which is not an accept state.

Problem 3 (10 points) Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 46) to give the state diagram of a DFA for the language given. In all parts, $\Sigma = (a, b)$.

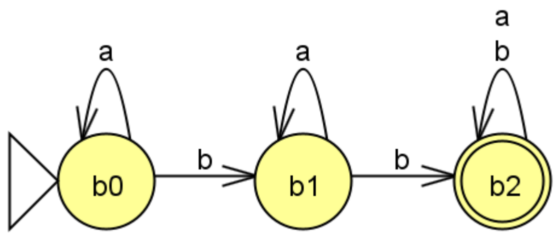
a) $\{w \mid w \text{ has at least three } a\text{'}s \text{ and at least two } b\text{'}s\}$
   - $\delta_1 = \{w \mid w \text{ has at least three } a\text{'}s\}$



Input	Result
aaa	Accept
bbb	Reject
bbabb	Reject
aabaa	Accept
babab	Reject
ababa	Accept
babbabb	Reject
abaabaa	Accept

Figure 1: DFA with Test Input

- $\delta_2 = \{w | w \text{ has at least two } b's\}$



Input	Result
aaa	Reject
bbb	Accept
bbabb	Accept
aabaa	Reject
babab	Accept
ababa	Accept
babbabb	Accept
abaabaa	Accept

Figure 2: DFA with Test Input

- $\{w | w \text{ has at least three } a's \text{ and at least two } b's\}$

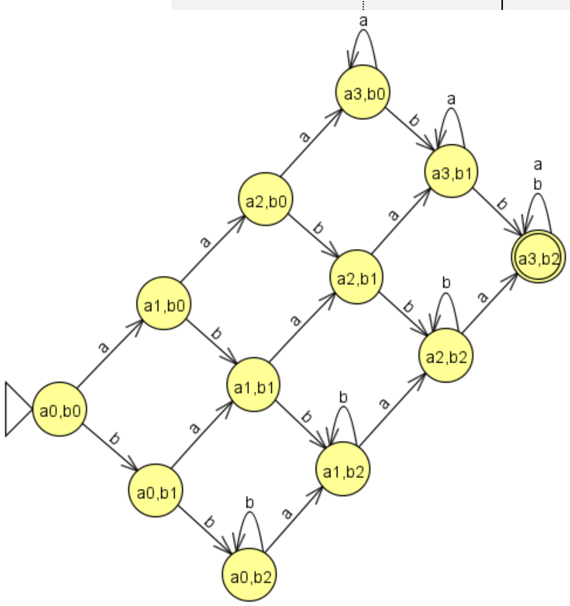
$\delta\big((r_1, r_2), a\big) = \big(\delta_1(r_1, a), \delta_2(r_2, a)\big)$ *The Transition Function*

$(r_1, r_2) \in Q$

$a \in \Sigma$

STATE	***INPUT***	***A***	***B***	***TRANSITION***
$Q = (r_1, r_2)$	a	$\delta_1(r_1, a)$	$\delta_2(r_2, a)$	$(\boldsymbol{\delta_1}, \boldsymbol{\delta_2})$
a_0, b_0	a	$\delta_1(a_0, a) = a_1$	$\delta_2(b_0, a) = b_0$	$\boldsymbol{a_1, b_0}$
a_0, b_0	b	$\delta_1(a_0, b) = a_0$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_0, b_1}$
a_1, b_0	a	$\delta_1(a_1, a) = a_2$	$\delta_2(b_0, a) = b_1$	$\boldsymbol{a_2, b_0}$
a_1, b_0	b	$\delta_1(a_1, b) = a_1$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_1, b_1}$
a_2, b_0	a	$\delta_1(a_2, a) = a_3$	$\delta_2(b_0, a) = b_0$	$\boldsymbol{a_3, b_0}$
a_2, b_0	b	$\delta_1(a_2, b) = a_2$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_2, b_1}$
a_3, b_0	a	$\delta_1(a_3, a) = a_3$	$\delta_2(b_0, a) = b_0$	$\boldsymbol{a_3, b_0}$
a_3, b_0	b	$\delta_1(a_3, b) = a_3$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_3, b_1}$
a_0, b_1	a	$\delta_1(a_0, a) = a_1$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_1, b_1}$
a_0, b_1	b	$\delta_1(a_0, b) = a_0$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_0, b_2}$
a_1, b_1	a	$\delta_1(a_1, a) = a_2$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_2, b_1}$
a_1, b_1	b	$\delta_1(a_1, b) = a_1$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_1, b_2}$
a_2, b_1	a	$\delta_1(a_2, a) = a_3$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_3, b_1}$
a_2, b_1	b	$\delta_1(a_2, b) = a_2$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_2, b_2}$
a_3, b_1	a	$\delta_1(a_3, a) = a_3$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_3, b_1}$

a_3, b_1	b	$\delta_1(a_3, b) = a_3$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_3, b_2}$
a_0, b_2	a	$\delta_1(a_0, a) = a_1$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_1, b_2}$
a_0, b_2	b	$\delta_1(a_0, b) = a_0$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_0, b_2}$
a_1, b_2	a	$\delta_1(a_1, a) = a_2$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_2, b_2}$
a_1, b_2	b	$\delta_1(a_1, b) = a_1$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_1, b_2}$
a_2, b_2	a	$\delta_1(a_2, a) = a_3$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_3, b_2}$
a_2, b_2	b	$\delta_1(a_2, b) = a_2$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_2, b_2}$
a_3, b_2	a	$\delta_1(a_3, a) = a_3$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_3, b_2}$
a_3, b_2	b	$\delta_1(a_3, b) = a_3$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_3, b_2}$

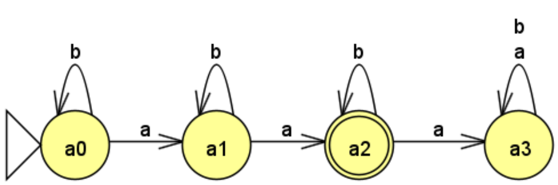


Input	Result
aaa	Reject
bbb	Reject
bbabb	Reject
aabaa	Reject
babab	Reject
ababa	Accept
babbabb	Reject
abaabaa	Accept

Figure 3: DFA with Test Input

b) $\{w \mid w$ has exactly two a's and at least two b's$\}$

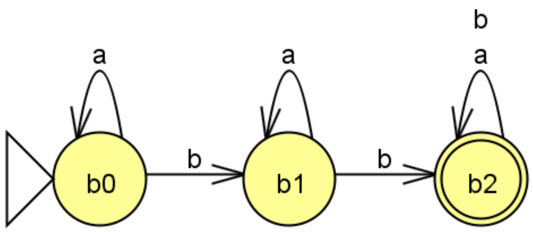
- $\delta_1 = \{w \mid w$ has exactly two a's$\}$



Input	Result
aaa	Reject
bbb	Reject
baa	Accept
abb	Reject
baaa	Reject
aaab	Reject
babab	Accept
ababa	Reject
bbaabb	Accept
aabbaa	Reject

Figure 4: DFA with Test Input

- $\delta_2 = \{w | w \text{ has at least two } b\text{'}s\}$



Input	Result
aaa	Reject
bbb	Accept
baa	Reject
abb	Accept
baaa	Reject
aaab	Reject
babab	Accept
ababa	Accept
bbaabb	Accept
aabbaa	Accept

Figure 5: DFA with Test Input

- $\{w |\ w \text{ has exactly two } a\text{'s and at least two } b\text{'}s\}$

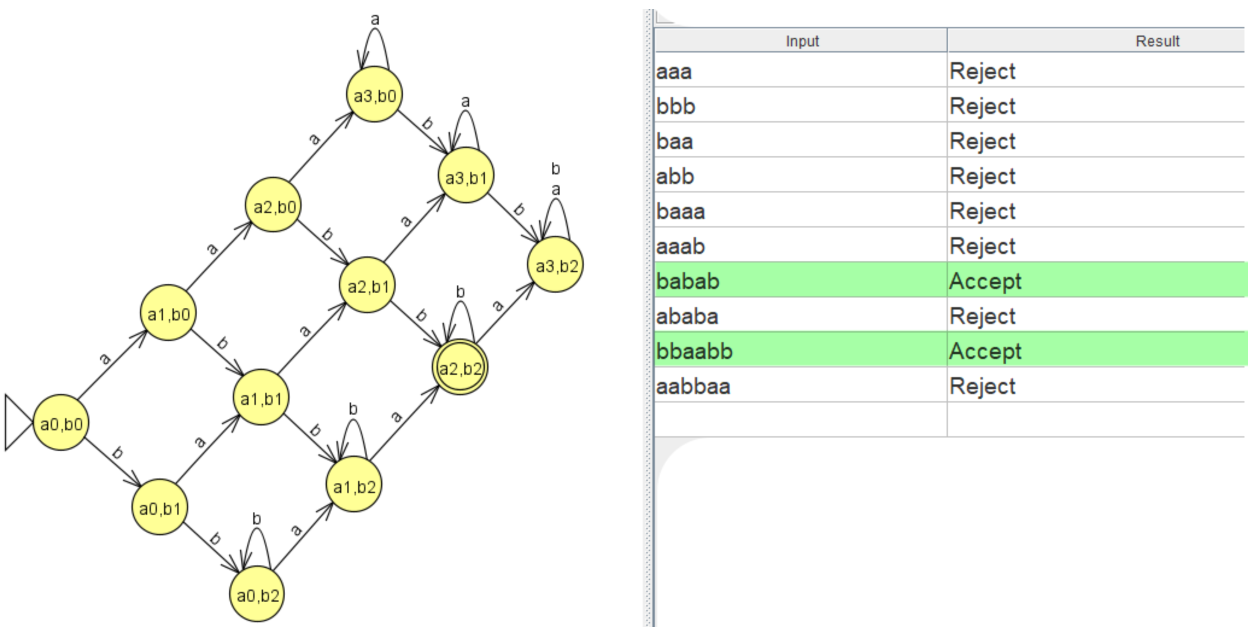
$$\delta\big((r_1, r_2), a\big) = \big(\delta_1(r_1, a), \delta_2(r_2, a)\big)$$ *The Transition Function*

$(r_1, r_2) \in Q$

$a \in \Sigma$

STATE	***INPUT***	***A***	***B***	***TRANSITION***
$Q = (r_1, r_2)$	a	$\delta_1(r_1, a)$	$\delta_2(r_2, a)$	$(\boldsymbol{\delta_1}, \boldsymbol{\delta_2})$
a_0, b_0	a	$\delta_1(a_0, a) = a_1$	$\delta_2(b_0, a) = b_0$	$\boldsymbol{a_1, b_0}$
a_0, b_0	b	$\delta_1(a_0, b) = a_0$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_0, b_1}$
a_1, b_0	a	$\delta_1(a_1, a) = a_2$	$\delta_2(b_0, a) = b_1$	$\boldsymbol{a_2, b_0}$
a_1, b_0	b	$\delta_1(a_1, b) = a_1$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_1, b_1}$
a_2, b_0	a	$\delta_1(a_2, a) = a_3$	$\delta_2(b_0, a) = b_0$	$\boldsymbol{a_3, b_0}$
a_2, b_0	b	$\delta_1(a_2, b) = a_2$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_2, b_1}$
a_3, b_0	a	$\delta_1(a_3, a) = a_3$	$\delta_2(b_0, a) = b_0$	$\boldsymbol{a_3, b_0}$
a_3, b_0	b	$\delta_1(a_3, b) = a_3$	$\delta_2(b_0, b) = b_1$	$\boldsymbol{a_3, b_1}$
a_0, b_1	a	$\delta_1(a_0, a) = a_1$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_1, b_1}$
a_0, b_1	b	$\delta_1(a_0, b) = a_0$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_0, b_2}$
a_1, b_1	a	$\delta_1(a_1, a) = a_2$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_2, b_1}$
a_1, b_1	b	$\delta_1(a_1, b) = a_1$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_1, b_2}$
a_2, b_1	a	$\delta_1(a_2, a) = a_3$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_3, b_1}$
a_2, b_1	b	$\delta_1(a_2, b) = a_2$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_2, b_2}$

a_3, b_1	a	$\delta_1(a_3, a) = a_3$	$\delta_2(b_1, a) = b_1$	$\boldsymbol{a_3, b_1}$
a_3, b_1	b	$\delta_1(a_3, b) = a_3$	$\delta_2(b_1, b) = b_2$	$\boldsymbol{a_3, b_2}$
a_0, b_2	a	$\delta_1(a_0, a) = a_1$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_1, b_2}$
a_0, b_2	b	$\delta_1(a_0, b) = a_0$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_0, b_2}$
a_1, b_2	a	$\delta_1(a_1, a) = a_2$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_2, b_2}$
a_1, b_2	b	$\delta_1(a_1, b) = a_1$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_1, b_2}$
a_2, b_2	a	$\delta_1(a_2, a) = a_3$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_3, b_2}$
a_2, b_2	b	$\delta_1(a_2, b) = a_2$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_2, b_2}$
a_3, b_2	a	$\delta_1(a_3, a) = a_3$	$\delta_2(b_2, a) = b_2$	$\boldsymbol{a_3, b_2}$
a_3, b_2	b	$\delta_1(a_3, b) = a_3$	$\delta_2(b_2, b) = b_2$	$\boldsymbol{a_3, b_2}$

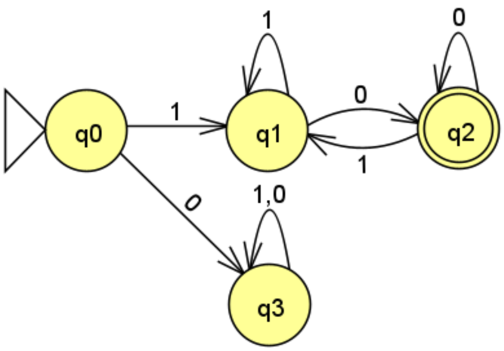


Input	Result
aaa	Reject
bbb	Reject
baa	Reject
abb	Reject
baaa	Reject
aaab	Reject
babab	Accept
ababa	Reject
bbaabb	Accept
aabbaa	Reject

Figure 6: DFA with Test Input

Problem 4 (10 points) 1.6 Give state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is {0,1}.

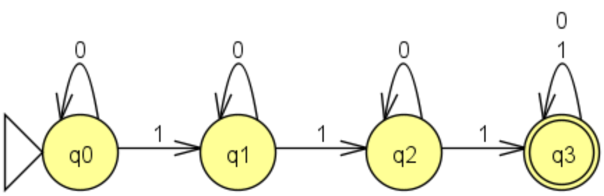
1. $\{w \mid w \text{ begins with a 1 and ends with a 0}\}$



Input	Result
01	Reject
10	Accept
111	Reject
000	Reject
1010	Accept
11110	Accept
01110	Reject
10101	Reject
10001	Reject
100010	Accept

Figure 7: DFA with Test Input

2. $\{w \mid w \text{ begins with at least three 1s}\}$

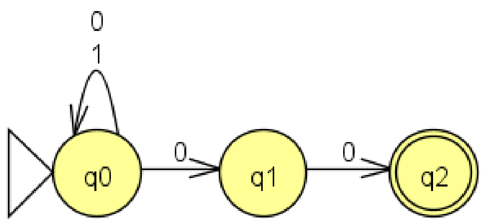


Input	Result
01	Reject
10	Reject
111	Accept
000	Reject
1010	Reject
11110	Accept
01110	Accept
10101	Accept
10001	Reject
100010	Reject

Figure 8: DFA with Test Input

Problem 5 (10 points) 1.7 Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is {0,1}.

a) The language $\{w \mid w \text{ ends with } 00\}$ with three states

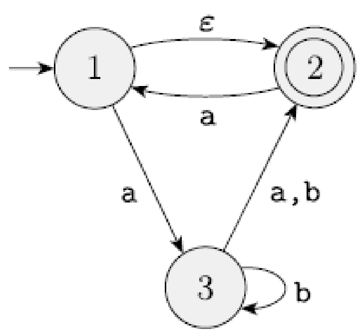


Input	Result
100	Accept
110	Reject
0011	Reject
01001	Reject
10000	Accept
0110100	Accept
101010	Reject
00000	Accept
010	Reject
1100	Accept
00	Accept

Figure 9: NFA with Test Input

- *Since NFAs accept input if any sequence of the possible choices leads to a final state, we can loop all 1's and 0's until the final two 0's appear (aka we do not need to account for every possible appearance of a 1 since NFAs automatically considers that possibility)*

Problem 6 (10 points) 1.16 Use the construction given in Theorem 1.39 to convert the following nondeterministic finite automata to equivalent deterministic finite automata **(show the process).**



(b)

1. Determine *ε-closure* for each state

ε-closure =self-state ∪ ε-reachable states *ε-closure Formula*

	self	*ε-reachable*	
$\varepsilon c\{1\}$	$\{1\}$	$\{2\}$	$\{1,2\}$
$\varepsilon c\{2\}$	$\{2\}$		$\{2\}$
$\varepsilon c\{3\}$	$\{3\}$		$\{3\}$

Name new state A

2. Obtain δ' transition for any New States Found
   - Calculate δ'Transition for A
     $\boldsymbol{A} = \{1, 2\}$ *State A*
     $\delta'(R, a) = \varepsilon\text{-}closure\{\delta(R, a)\}$ *δ' Transition Formula*

     $\delta'(A, a) = \varepsilon\text{-}closure\{\delta((1, 2), a)\}$ *Substitute Values*
     $\delta'(A, a) = \varepsilon\text{-}closure\{\delta(1, a) \cup \delta(2, a)\}$ *Expand, Showing Union*
     $\delta'(A, a) = \varepsilon\text{-}closure\{\{3\} \cup \{1\}\}$ *Find δ Transition for each R*
     $\delta'(A, a) = \{\{3\} \cup \{1, 2\}\}$ *Find ε-closure for each δ*
     $\delta'(A, a) = \{1, 2, 3\}$ *Solve for δ'*
     $\delta'(A, a) = \{1, 2, 3\} = \boldsymbol{B}$ *Name new state B*

     $\delta'(A, b) = \varepsilon\text{-}closure\{\delta((1, 2), b)\}$ *Substitute Values*
     $\delta'(A, b) = \varepsilon\text{-}closure\{\delta(1, b) \cup \delta(2, b)\}$ *Expand, Showing Union*
     $\delta'(A, b) = \varepsilon\text{-}closure\{\{\emptyset\} \cup \{\emptyset\}\}$ *Find δ Transition for each R*
     $\delta'(A, b) = \emptyset$ *Solve for δ'*

   - Calculate δ'Transition for B
     $B = \{1, 2, 3\}$ *State B*

     $\delta'(B, a) = \varepsilon\text{-}closure\{\delta((1, 2, 3), a)\}$ *Substitute Values*
     $\delta'(B, a) = \varepsilon\text{-}closure\{\delta(1, a) \cup \delta(2, a) \cup \delta(3, a)\}$ *Expand, Showing Union*
     $\delta'(B, a) = \varepsilon\text{-}closure\{\{3\} \cup \{1\} \cup \{2\}\}$ *Find δ Transition for each R*
     $\delta'(B, a) = \{\{3\} \cup \{1, 2\} \cup \{2\}\}$ *Find ε-closure for each δ*
     $\delta'(B, a) = \{1, 2, 3\}$ *Solve for δ'*
     $\delta'(B, a) = \{1, 2, 3\} = B$ *State B*

     $\delta'(B, b) = \varepsilon\text{-}closure\{\delta((1, 2, 3), b)\}$ *Substitute Values*
     $\delta'(B, b) = \varepsilon\text{-}closure\{\delta(1, b) \cup \delta(2, b) \cup \delta(3, b)\}$ *Expand, Showing Union*
     $\delta'(B, b) = \varepsilon\text{-}closure\{\emptyset \cup \emptyset \cup \{2, 3\}\}$ *Find δ Transition for each R*
     $\delta'(B, b) = \varepsilon\text{-}closure\{\{2, 3\}\}$ *Remove Empty Sets*
     $\delta'(B, b) = \{2, 3\}$ *Find ε-closure for each δ, Solve for δ'*
     $\delta'(B, b) = \{2, 3\} = \boldsymbol{C}$ *Name new state C*

- Calculate δ'Transition for C
  $C = \{2,3\}$ *State C*

  $\delta'(C,a) = \varepsilon\text{-}closure\{\delta((2,3),a)\}$ *Substitute Values*

  $\delta'(C,a) = \varepsilon\text{-}closure\{\delta(2,a) \cup \delta(3,a)\}$ *Expand, Showing Union*

  $\delta'(C,a) = \varepsilon\text{-}closure\{\{1\} \cup \{2\}\}$ *Find δ Transition for each R*

  $\delta'(C,a) = \{\{1,2\} \cup \{2\}\}$ *Find ε-closure for each δ*

  $\delta'(C,a) = \{1,2\}$ *Solve for δ'*

  $\delta'(C,a) = \{1,2\} = A$ *State A*

  $\delta'(C,b) = \varepsilon\text{-}closure\{\delta((2,3),b)\}$ *Substitute Values*

  $\delta'(C,b) = \varepsilon\text{-}closure\{\delta(2,b) \cup \delta(3,b)\}$ *Expand, Showing Union*

  $\delta'(C,b) = \varepsilon\text{-}closure\{\emptyset \cup (2,3)\}$ *Find δ Transition for each R*

  $\delta'(C,b) = \varepsilon\text{-}closure\{\{2,3\}\}$ *Remove Empty Sets*

  $\delta'(C,b) = \{2,3\}$ *Find ε-closure for each δ, Solve for δ'*

  $\delta'(C,b) = \{2,3\} = C$ *State C*

3. Determine Final State(s) for DFA

- Identify Final State in Original NFA

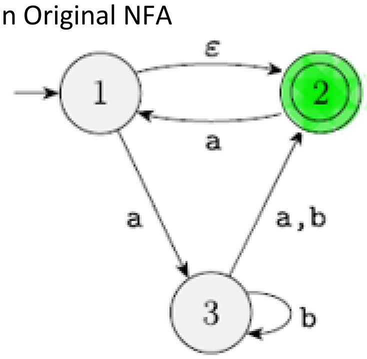


Figure 10: The Final State in the original NFA is 2

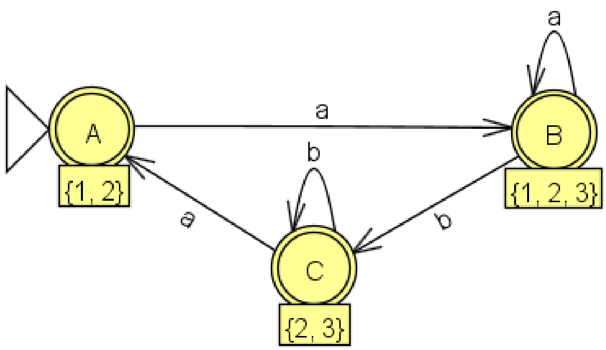
- Identify any DFA states which contain the NFA Final State

DFA States

A	$\{1,2\}$
B	$\{1,2,3\}$
C	$\{2,3\}$

Figure 11: States A, B, C will be Final States in the DFA

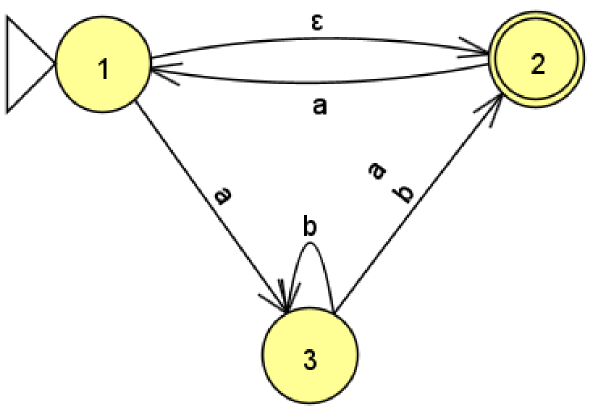
❖ Converted DFA:



Input	Result
aa	Accept
aabb	Accept
abba	Accept
abaaa	Accept
abbba	Accept
b	Reject
bbaa	Reject
baba	Reject
bababb	Reject

Figure 12: DFA with Test Input

By comparing the results of which strings are accepted/rejected, we can see that the DFA above is equivalent to the provided NFA:



Input	Result
aa	Accept
aabb	Accept
abba	Accept
abaaa	Accept
abbba	Accept
b	Reject
bbaa	Reject
baba	Reject
bababb	Reject

Figure 13: Provided NFA with Test Input

Problem 7 (20 points) 1.18 Give regular expressions generating the following languages. In all parts the alphabet is {0, 1}

a) $\{w|\ w \text{ has at least one } 1\}$
- $0^*1(0 \cup 1)^*$

b) $\{w|\ w \text{ starts and ends with same symbol}\}$
- $(0(0 \cup 1)^*0) \cup (1(0 \cup 1)^*1) \cup (0) \cup (1)$

c) $\{w|\ |w| < 5\}$
- $(\varepsilon \cup 0 \cup 1) \circ (\varepsilon \cup 0 \cup 1) \circ (\varepsilon \cup 0 \cup 1) \circ (\varepsilon \cup 0 \cup 1)$

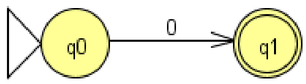
d) $\{w|$ every 3rd position of w is 1$\}$
   - $(0 \cup 1)(0 \cup 1)(1)(\varepsilon \cup 0 \cup 1)(\varepsilon \cup 0 \cup 1)\big((1)(0 \cup 1)(0 \cup 1)\big)^*(\varepsilon \cup 1 \cup 1(0 \cup 1))$

e) $\{w|\ w$ has equal numbers of ‘01’ and ‘10’$\}$
   - $(\varepsilon) \cup (00^*) \cup (00^*1(1 \cup 00^*1)^*00^*) \cup (11^*) \cup (11^*0(0 \cup 11^*0)^*11^*)$

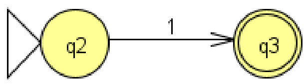
Problem 8 (10 points) 1.19 Use the procedure described in Lemma 1.55 to convert the following regular expressions to nondeterministic finite automata.

a) $(0 \cup 1)^*000(0 \cup 1)^*$

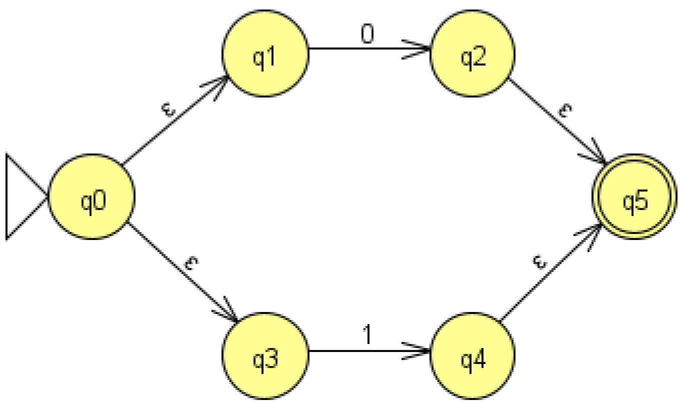
- 0



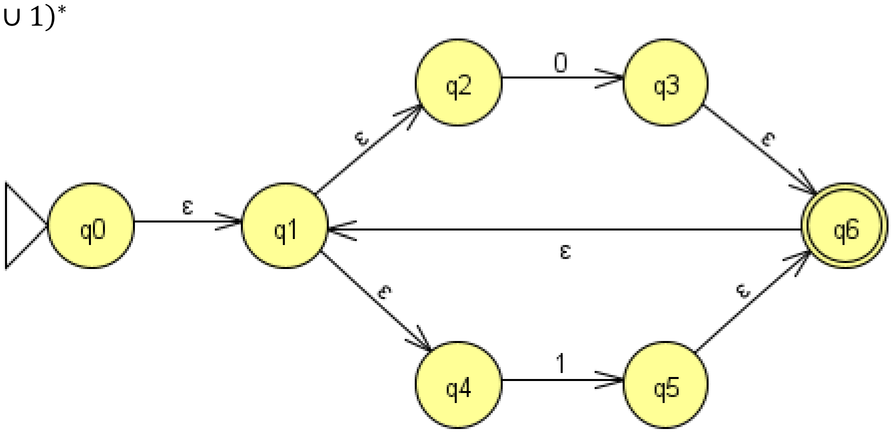
- 1



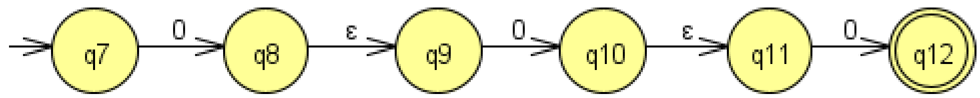
- $0 \cup 1$



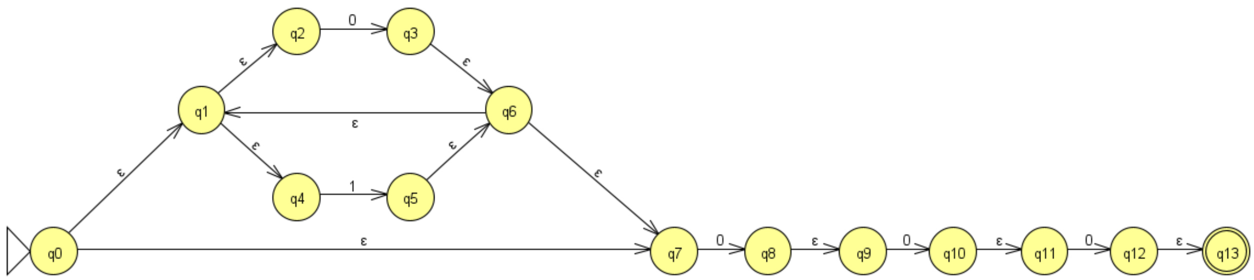
- $(0 \cup 1)^*$



- 000



- $(0 \cup 1)^{*}000$



- $(0 \cup 1)^{*}000(0 \cup 1)^{*}$

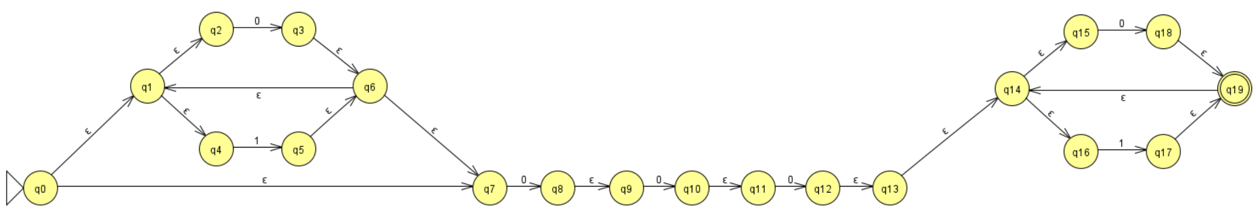


Figure 14: ε -NFA of $(0 \cup 1)^{*}000(0 \cup 1)^{*}$

Additionally, I have converted this ε-NFA to NFA, as shown below

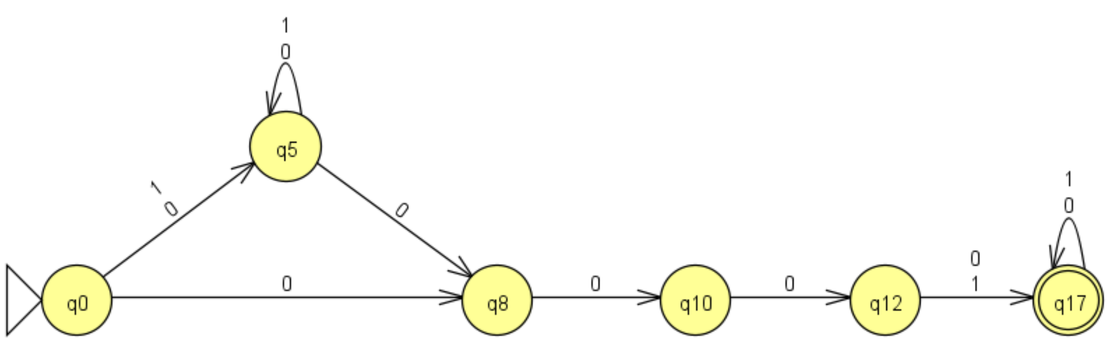


Figure 15: NFA of $(0 \cup 1)^{*}000(0 \cup 1)^{*}$

Problem 9 (20 points) A syntactically valid email address is made up of a username followed by '@' followed by a list of at least two domain names separated by '.'. Assume that user and domain names are made up of letters [a-z] and digits [0-9], and the main domain name, i.e., the last domain name, contains two or three characters.

VALID	INVALID
abc@dsu.edu	a.b.ab
abc@pluto.dsu.edu	ab@ab
11@123.com	ab@ab.abcd

1. (5 points) Use JFLAP design a finite automaton to recognize valid mail address. Enclose your FA's JFLAP file (.jff) in your homework submission.

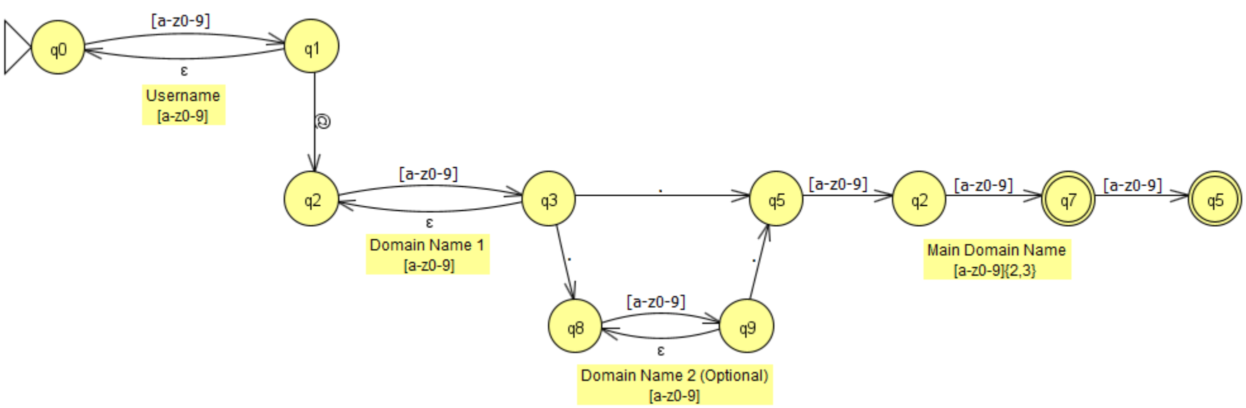


Figure 16: ε-NFA for Email Address Strings

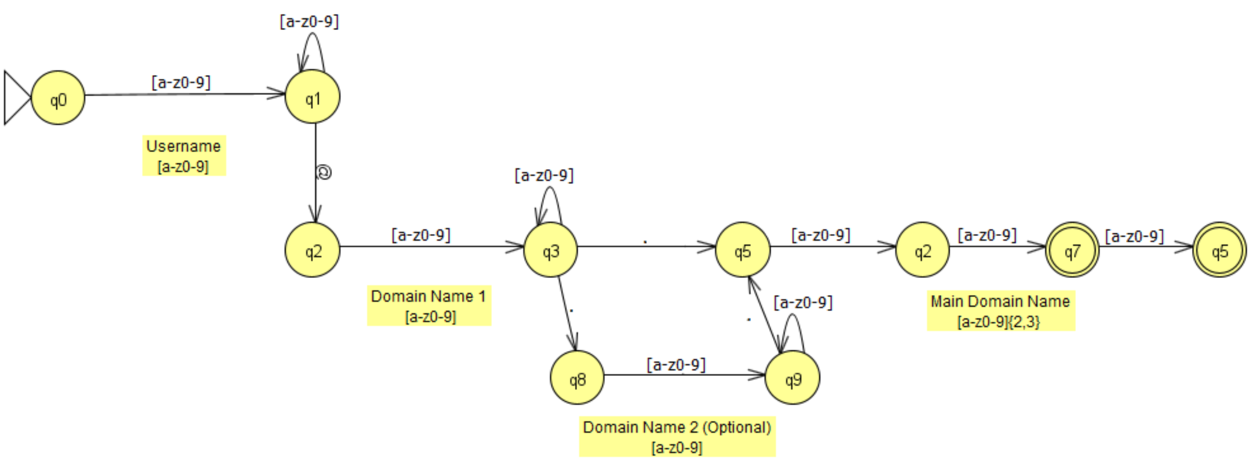


Figure 17: NFA for Email Address Strings

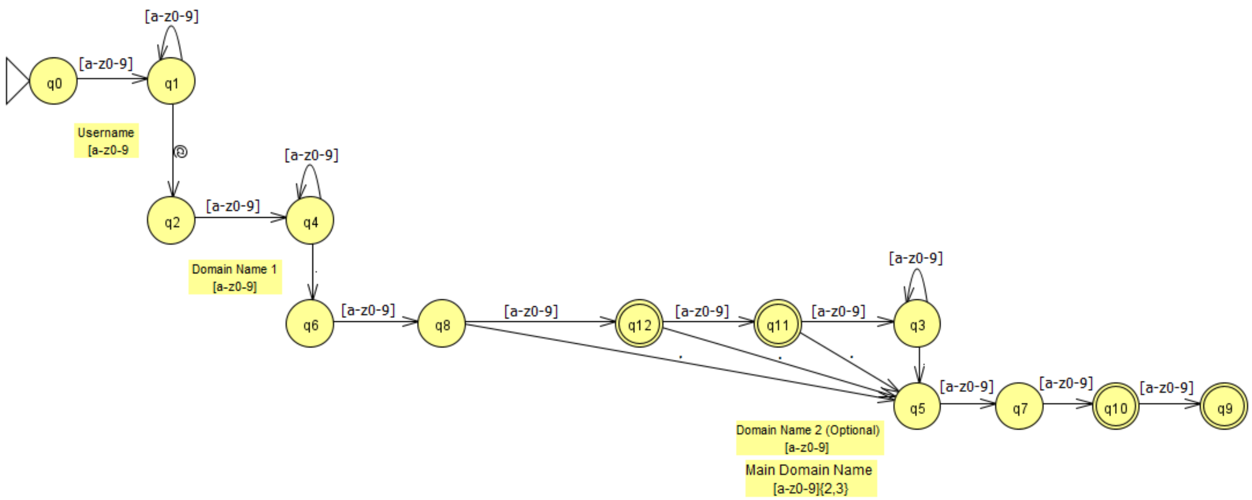
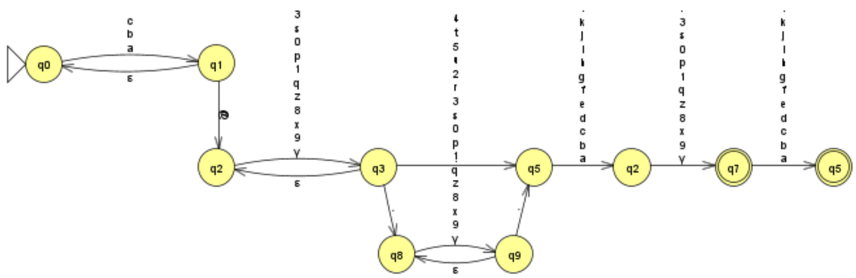


Figure 18: DFA for Email Address Strings

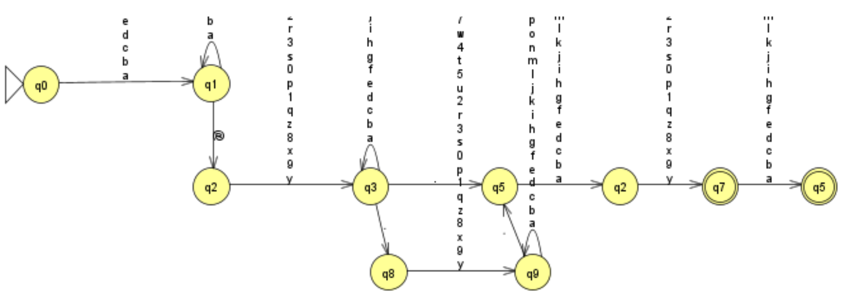
JFLAP `.jff` *files are also attached in homework submission as* `HW1-Q9eNFA[Final].jff, HW1-Q9DFA[Final].jff`

2. (5 points) Test the above six testing cases using JFLAP Multiple Run function. You can include more testing cases if you want. Include a screenshot of your testing results in your homework submission as below:



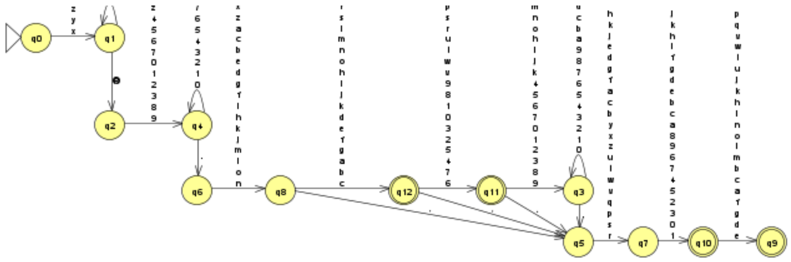
abc@dsu.edu	Accept
abc@pluto.dsu.ed	Accept
11@123.com	Accept
a.b.ab	Reject
ab@ab	Reject
ab@ab.abcd	Reject

Figure 19: Test Input for Email ε-NFA



abc@dsu.edu	Accept
abc@pluto.dsu.ed	Accept
11@123.com	Accept
	Reject
a.b.ab	Reject
ab@ab	Reject
ab@ab.abcd	Reject

Figure 20: Test Input for Email NFA



abc@dsu.edu	Accept
abc@pluto.dsu.ed	Accept
11@123.com	Accept
	Reject
a.b.ab	Reject
ab@ab	Reject
ab@ab.abcd	Reject

Figure 21: Test Input for Email DFA

3. (10 points) Use a programming language at your choice to implement the FA designed in Step 1. Submit your source code and also write a readme file to show how to compile your program, how to run your program, and some testing results you have. **(Note: The program should implement the FA based on your design and simulate the way how FA works. There is no credit for the program if the implementation is based on the use of regular expressions.)**

- Will be included in HW folder