Assignment 1

COMP335

Mustafa Alawadi (40217764)

Concordia University

2024-02-05

Q1.

Assignment 1

A) $L_1(L_2 \cap L_3)$ represents the set of strings that is created by combining a string from L_1 with a string that belongs to both L_2 and L_3.

$L_1L_2 \cap L_1L_3$ represents the set of strings that can be created by combining a string from L_1 with a string from L_2 and combining a string from L_1 with a string from L_3.

- The relationship is: The 2 languages are equal. $L_1(L_2 \cap L_3) \subseteq L_1L_2 \cap L_1L_3$

- Proof: If we let r be a string in $L_1(L_2 \cap L_3)$ which suggests that $r = xy$, where $x \in L_1$ and $y \in L_2 \cap L_3$.

Because $y \in L_2 \cap L_3$, it means $y \in L_2$ and $y \in L_3$
Thus, $r \in L_1L_2 \cap L_1L_3$

A counterexample is if r is in $L_1(L_2 \cap L_3)$ which suggests that $r = xy$, where $x \in L_1$, $y \in L_2$ for the first part, and $y \in L_3$ for the L_1L_3, we have $y \in L_2 \cap L_3 \Rightarrow s \in L_1(L_2 \cap L_3)$.

B) $L_1^* \cap L_2^*$ represents the set of strings that are created by combining strings from L_1 n number of times, similarly, we combine strings from L_2 n number of times.

$(L_1 \cap L_2)^*$ represents the set of strings that are created by combining strings from both L_1 and L_2 n number of times.

- The relationship is: $L_1^* \cap L_2^* \subseteq (L_1 \cap L_2)^*$
- Proof: Let r be a string in $L_1^* \cap L_2^*$, thus r is created by forming strings from both L_1 and L_2.

The converse is not needed, there can be scenarios where strings in $L_1^* \cap L_2^*$ that aren't made from strings in $L_1 \cap L_2$. Example: when $L_1 = \{a\}$ and $L_2 \{aa\}$, we have $aa \in L_1^* \cap L_2^*$ but not in $(L_1 \cap L_2)^*$ because $L_1 \cap L_2 = \emptyset$

c) $L_1^* L_2^*$ represents the set of strings that is formed by combining strings from L_1 n number of times and followed by combining strings from L_2 n number of times.

$(L_1 L_2)^*$ represents the set of strings that can be formed by combining strings from L_1 followed by L_2 and n number of times.

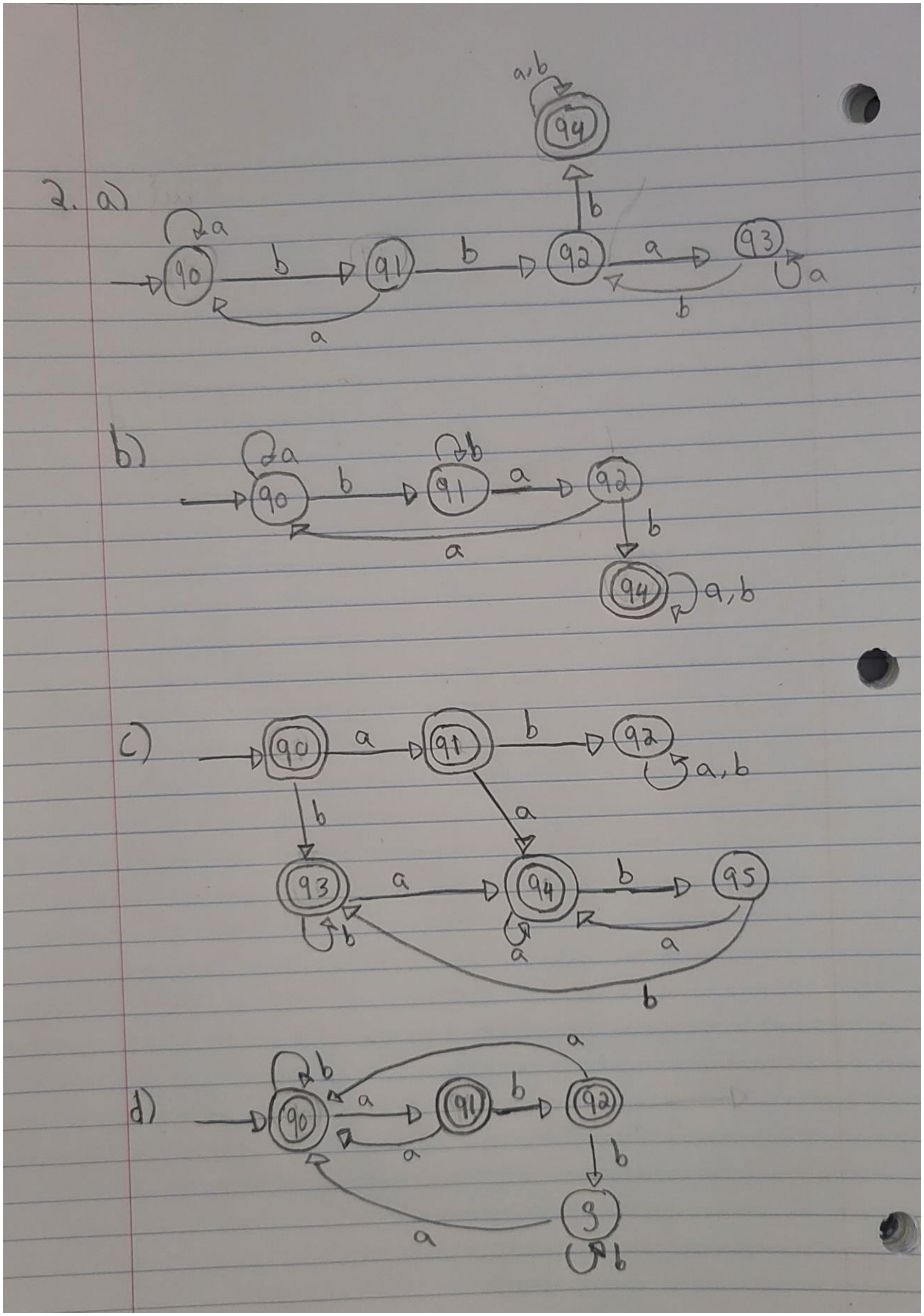
- The relationship is: $L_1^* L_2^* \subseteq (L_1 L_2)^*$
- Proof: If we let r be a string in $L_1^* L_2^*$, then r can be formed by combining strings from L_1 n number of times followed by strings from L_2 n number of times. In $L_1^* L_2^*$, the r string can be formed by combining strings from $L_1 L_2$ in any order n number of times, This claim holds true, but the converse isn't because in $L1^* L2^*$, the $L1$ strings can be selected before the $L2$ strings which isn't mandatory in $(L1 L2)^*$.

D) $L_1^*(L_2L_1^*)^*$ represents the strings starting with any number from the L_1 strings, fallowed by n number of sequences in which each sequence begins with an L_2 string followed by n number of L_1 strings.

$(L_1^*L_2)^*L_1^*$ represents any number of sequences where every sequence begins with n number of L_1 strings followed by a L_2 string. The whole operation can be followed by n number of L_1 strings.

- The relationship is: Languages are equal, $L_1^*(L_2L_1^*)^* \subseteq (L_1^*L_2^*)^*L_1^*$
- Proof: In both expressions, any number of L_1 strings can be placed in the sequence without order. L_2 strings in this case is distributed among them. This pattern can be done n number of times. Therefore, the strings from L_1 and L_2 can be rearranged in a manner that fulfills the condition of both expressions.

Q2.



Q3.

3. States $Q = \{q0, q1\}$

q0 is the initial state and q1 is the accepting state

The alphabet $\Sigma = \{0, 1\}$

Transition function : δ

$$\delta(q0, 0) = q0$$
$$\delta(q0, 1) = q1$$
$$\delta(q1, 0) = q1$$
$$\delta(q1, 1) = q1$$

Thus, we have two states, q0 and q1.
The q0 will be the initial state while q1, will be the accepting state. δ will allow us to determine the next state with the help of the input symbol and the state that is current.

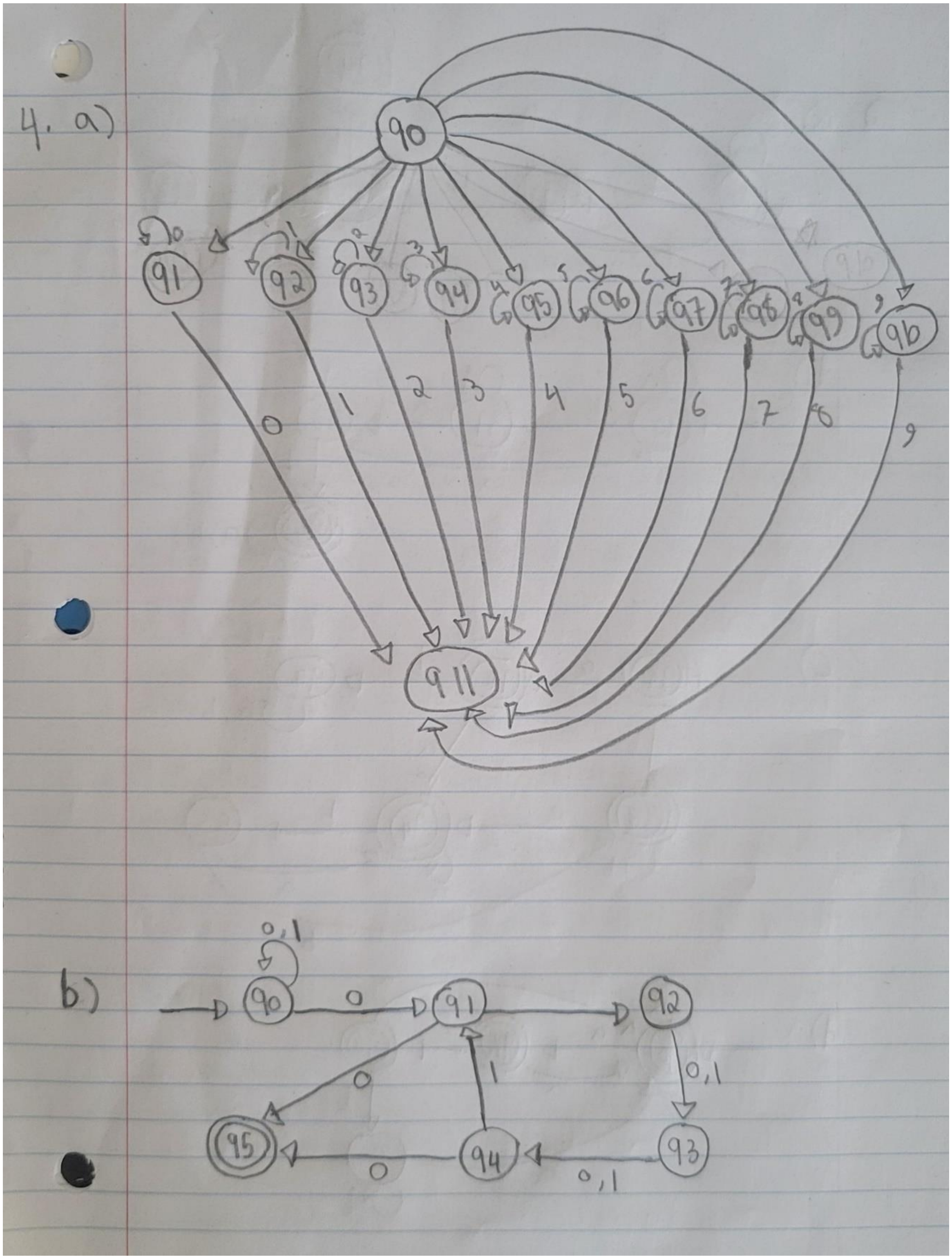
$\underline{L(A) \subseteq L}$

Because q1 is the only accepting state for A, for the string to be accepted, it must have at least one 1
Therefore, strings accepted by A is in L

$\underline{L \subseteq L(A)}$

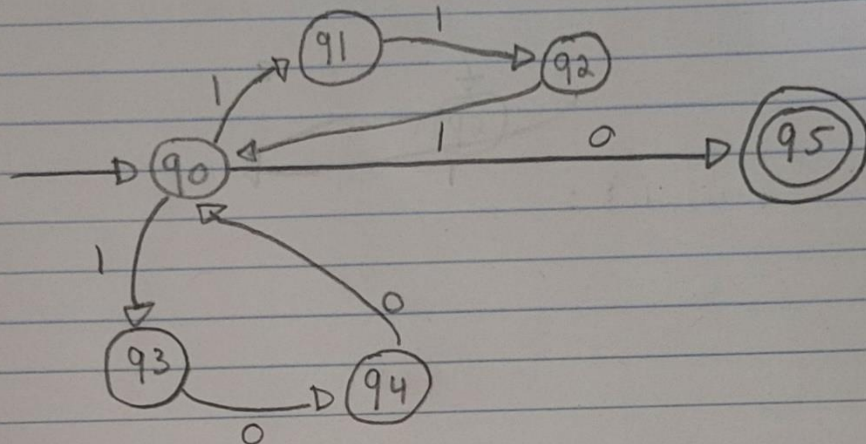
Strings in L have at least one 1. In the DFA, the 1 transitions to state q1 which in this case will accept the state. Thus, any string in L is accepted by A.

Q4.

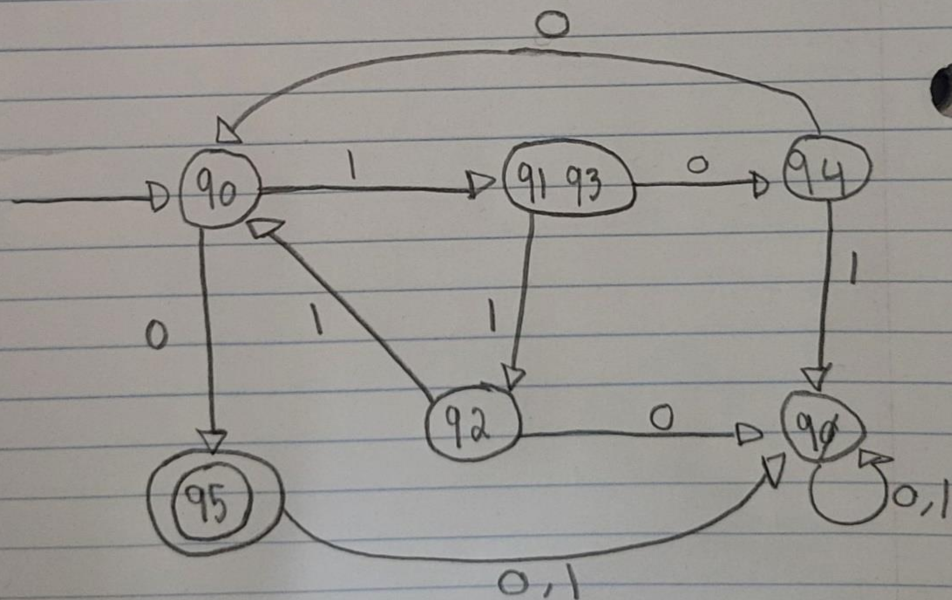


Q5.

5.A) $L_1L_2 = \{11, 110\}^*0$

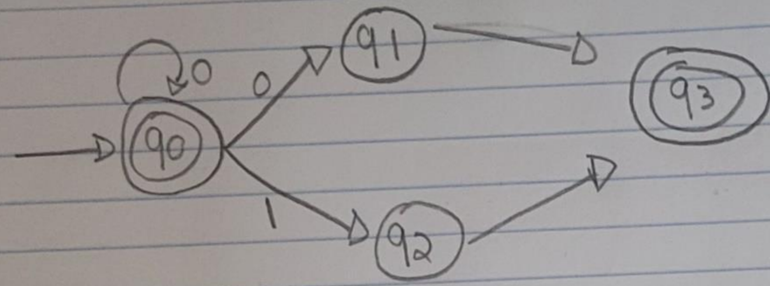


B)



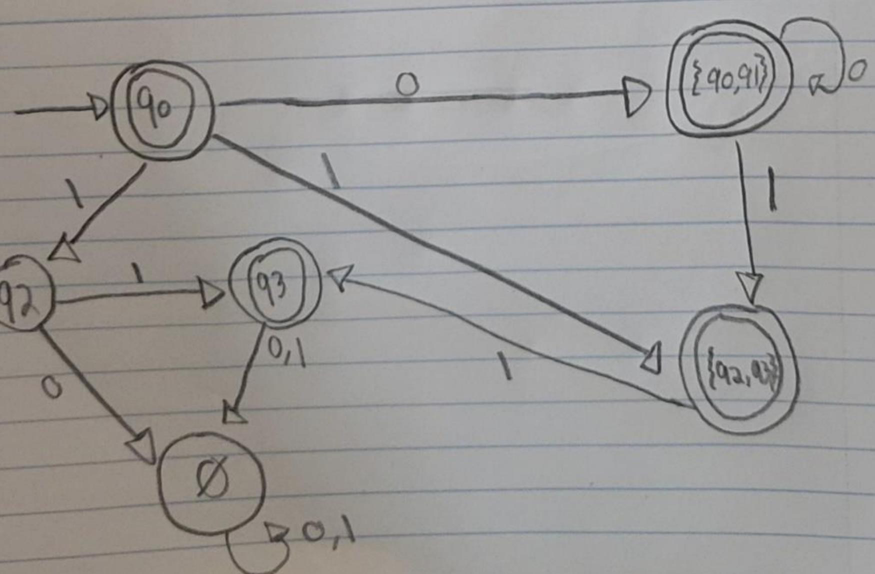
Q6.

6.A) NFA accepting $L_1 L_2$



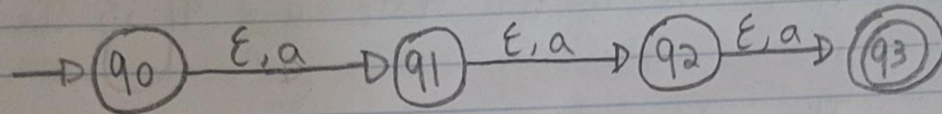
b)

States	0	1
{q0}	{q0,q1}	{q2}
{q0,q1}	{q0,q1}	{q2,q3}
{q2}	∅	{q3}
{q2,q3}	∅	{q3}
{q3}	∅	∅
∅	∅	∅



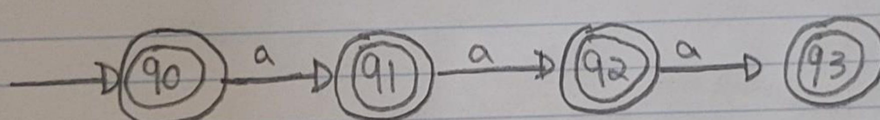
Q7.

7. ε-NFA A



ε-free NFA B

$q_0 = \varepsilon close(q_0) = \{q_0, q_1, q_2, q_3\}$
$q_1 = \varepsilon close(q_1) = \{q_1, q_2, q_3\}$
$q_2 = \varepsilon close(q_2) = \{q_2, q_3\}$
$q_3 = \varepsilon close(q_3) = \{q_3\}$



DFA C

States	a
$q_0 = \{q_0\}$	$\{q_0, q_1\}$
$q_1 = \{q_0, q_1\}$	$\{q_0, q_1, q_2\}$
$q_2 = \{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$
$q_3 = \{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$