(1)

(a) $\epsilon \in \Sigma^*$ and $L \subseteq \Sigma^*$

Therefore it is possible that $L = \{\epsilon\} \therefore |L| = 1$ (condition 1) $\therefore L^* = \{\epsilon, \epsilon\epsilon, \epsilon\epsilon\epsilon, \dots\}$

but any consecutive concatenation of ϵ is $\epsilon \therefore L^* = \{\epsilon\} \therefore |L^*| < \infty$ (condition 2).

(b) Consider $L = 1, M = 0$

$$LM = \{1\} \circ \{0\} = \{10\}$$

$$(LM)^* = \{10\}^* = \{10, 1010, \dots\}$$

$$L^*M^* = \{1, 11, \dots,\}\{0, 00, \dots\} = \{10, 100, 110, 1100, \dots\}$$

While it might be that $L^*M^* \subset (LM)^*, L^*M^* \notin (LM)^* \therefore (LM)^* \neq L^*M^*$

(c) The NFA generated by such a language would include four states each with a self-loop for 0 and a loop forward for 1 except for the fourth state where it'd loop back to the starting state.

$$Q = \{q_0, q_1, q_2, q_3, q_4\}$$

$$F = \{q_4\}$$

$$q_0 = \{q_0\}$$

$$\delta : \{(q_n, 0) = q_n, (q_n, 1) = q_{n+1}\}\text{for } n < |Q| - 1$$

Apply $\delta(q_4, 1) = q_0$which means during state-elimination after appending all regular expressions created by eliminated states generates$((0^*1)^4)^*$. However, we also have a epsilon on the accepting state q_4 meaning we need append another 0^*due to the zero loop giving us $((0^*1)^4)^*0^*$. This makes sense because we have a some assortment of zeroes appended by $4n|n \geq 1$ ones. But they do not have to be consecutive ones so we apply the Kleene star to generate all possible assortments of $4n$ ones. But we also can have strings that do not have to end with 1 so we'll append the $0^*$to get the language L.

(2) $\because L(M) = L(N) \implies$ the two machines accept the same set of strings.Thus, some strings $w_n, w_m \in L(M), L(N)$ and $w_m, w_n \in L(M), L(N)$. Additionally, two states q_m, q_n are indistinguishable iff $q_m \in Q_m$ and $q_n \in Q_n$. and q_m, q_n lead to an accepting state.

Given that M has no unreachable states, every state $x \in Q_M$ can be accessed from the initial state by some string. Since $L(M) = L(N)$, we can define q_x as the state that is indistinguishable from x for each state x. According to the definition of DFA, such a state must exist because both M and N recognize the same language. Therefore, any string that results in acceptance from x in M must also result in acceptance from q_x in N.

b.

From part a, we can conclude that $A(x) = A(q_x)$ and $A(y) = A(q_y)$. If $x \not\sim y$, there exists a string w that distinguishes x and y in DFA M. By the correspondence between$M$ and N, this implies that w must also distinguish q_x and q_y, which contradicts $q_x \sim q_y$.

If $x \not\sim y$, it follows that $A(x) \neq A(y)$. As stated previously, $A(x) = A(q_x)$ and $A(y) = A(q_y)$. Thus, we can conclude that $A(q_x) \neq A(q_y)$.

c.

The automaton T is derived by applying the minimization algorithm to R, which eliminates any unreachable states. In part (a), we demonstrated that for any state in automaton M, there exists an equivalent state in automaton N, indicating that every state in M is indistinguishable from some state in N. This implies that the number of states in$N$ can only be equal to or greater than that in M.

In part (b), we established that if two states in M are distinguishable, their corresponding states in N must also be distinguishable. This ensures that we do not merge distinguishable states when constructing $N from M$.

Since $T$is the minimized DFA recognizing$L$, no automaton can have fewer states than T while still recognizing L. Therefore, $|Q_T| \leq |Q_S|$, as S could have more states than necessary, while \(T \) possesses the minimal number of states required to recognize the language L.

(3) Consider the language $L = \{w \in \Sigma^* | |w| = \frac{n(n+1)}{2} \text{for some } n \geq 1\}$.

The pumping lemma requires that :

1. $|y| > 0$
2. $|xy| \leq p$, where p is the pumping number
3. $xy^i z \in L$

If we have $w = xyz$, $|w| = |x| + |y| + |z| = \frac{n(n+1)}{2}$, but if we consider that the y string is pumped we can see that:

$$|w'| = |xy^i z| = |x| + i|y| + |z| \geq \frac{n(n+1)}{2}$$

Consider $i = 2$

$$|w'| = |xy^2 z| = |x| + 2|y| + |z| \geq \frac{n(n+1)}{2}$$

$$|w'| = |x| + |y| + |y| + |z| = \frac{n(n+1)}{2} + |y|$$

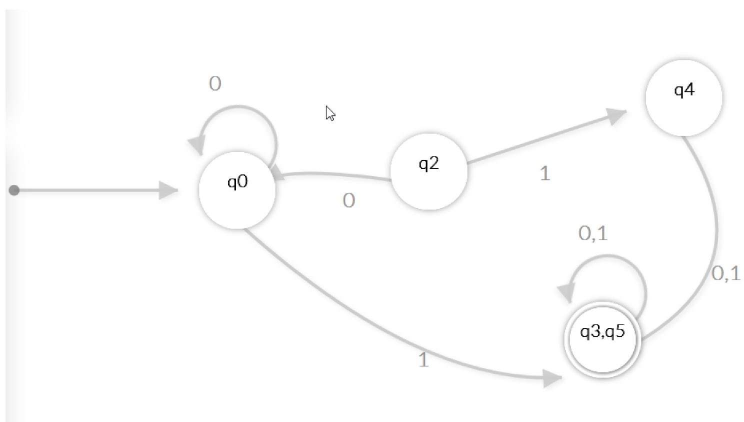
$$|w'| \geq \frac{n(n+1)}{2} + |y| \geq \frac{n(n+1)}{2}$$

However, the only way the LHS = RHS is if $|y| = 0$, which is impossible. Furthermore, the inequality suggests that a triangular number plus $|y|$ would also always be a triangular number. If we consider just the first possible value of $|y| = 1$ for the first triangular number $T_{n=1} = \frac{1(1+1)}{2} = 1$, we see this is false since $|w'| = T_n + |y| \notin L = T_n$.

---

(4)

	0	1	2	3	4	5
0	I	D	D	D	D	
1	D	D	D	D		
2	D	D	D			
3	D	I				
4	D					
5						



---

5. $L = \{w \in \Sigma^* | w = x0^n$,where $n \geq 0; |x| = n$.

For some string w, the first part of the string x is some combination of zeroes and ones of length n. This is followed by a string of zeroes of the same length. Let's consider some string $w' = xy^0z$, where we concatenate some pumping string y zero times (i.e remove it from the string w). If $|y| = s$, then the new length of x will be $n - s$ again followed by a series of n zeroes. This suggests,

$$|w'| = (n - s) + n = 2n$$

$$2n - s = 2n$$

$$s = 0$$

However, s is defined as $|y|$ which is strictly greater than 0. This contradiction means that $w' \notin L \therefore$ L is irregular.