Steven Myers
August 31^{st}, 2017
CCEP - B401 Homework #2

1.1)

	M_1	M_2
a.	q_1	q_1
b.	$F = \{q_2\}$	$F = \{q_1, q_4\}$
c.	$(q_1, q_2, q_3, q_1, q_1)$	$(q_1, q_1, q_1, q_2, q_4)$
d.	No	yes
e.	No	No

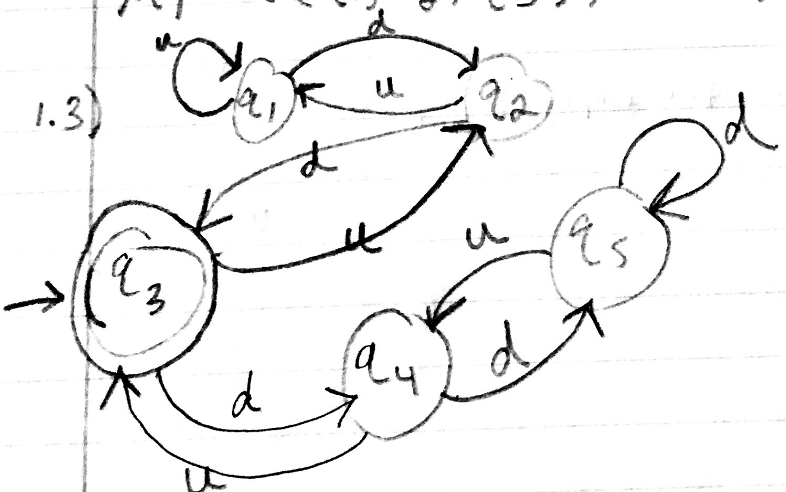
1.2) $M_2 = (\{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta_2, q_1, \{q_1, q_4\})$

δ_2	a	b
q_1	q_1	q_2
q_2	q_3	q_4
q_3	q_2	q_1
q_4	q_3	q_4

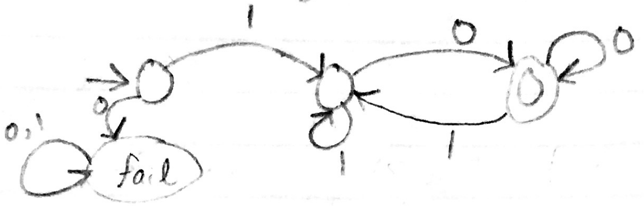
δ_1	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_1

$M_1 = (\{q_1, q_2, q_3\}, \{a, b\}, \delta_1, q_1, \{q_2\})$

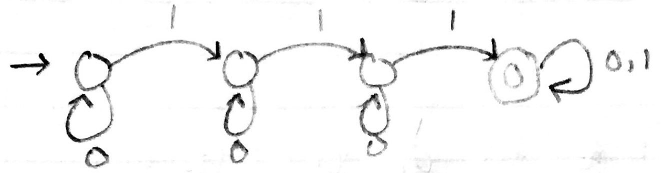
1.3)



1.6) a) $L_a = \{w \mid w$ begins with a 1 and ends in a 0$\}$

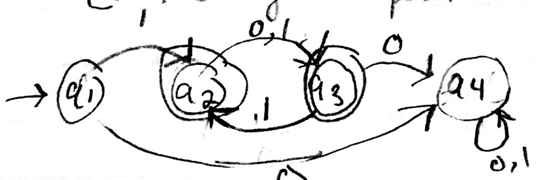


b) $L_b = \{w \mid w$ contains atleast three 1's$\}$

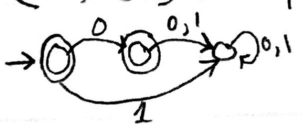


c) $L_c = \{w \mid w$ contains the substring 0101$\}$

i) $L_i = \{w \mid$ every odd position of w is a 1$\}$

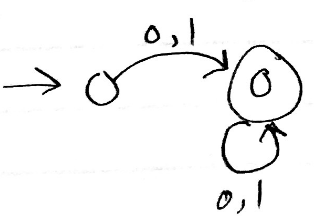


k) $L_k = \{\varepsilon, 0\}$ accepts zero and the empty string



m) $L_m = \{\}$

n) L_n = All strings but the empty string



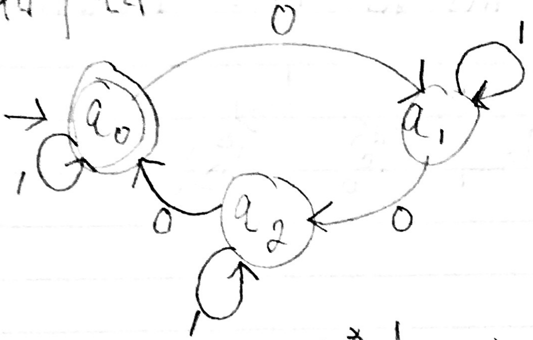
1) Formal set def of DFA in book problem 1.6i

$M_i = (\{q_1, q_2, q_3, q_4\}, \{0,1\}, \delta, q_1, \{q_2, q_3\})$

δ	0	1
q_1	q_4	q_2
q_2	q_3	q_3
q_3	q_4	q_2
q_4	q_4	q_4

$L(M_i) = \{ s \in \{0,1\}^* \mid$ all odd indexed bits of s are $1 \}$.

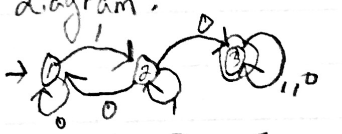
2) (a)



(b) $L(M) = \{ s \in \{0,1\}^* \mid$ s is a string that contains either zero 0's, or a number of 0's that is divisible by 3. $\}$
(3×0 zeros is valid)

3) Claim: For any regular language L, $\neg L$ is also regular, where $L \subseteq \{0,1\}^*$

Let M be a DFA, where $L(M) = L$, and M is described by the following diagram:



$L = \{ s \in \{0,1\}^* \mid s$ is a string containing the substring "10". $\}$

$M = (\{q_1, q_2, q_3\}, \{0,1\}, \delta, q_1, \{q_3\})$

δ	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_3	q_3

Based on our claim, we should be able to find a DFA M' that recognizes $\neg L$ since L is a regular language. We know that in L there are strings like "010" and "11110". In $\neg L$, there are strings such as "101" and "00001". We can easily define a machine M' ~~whose~~ where $L(M') = \neg L$.

3) Pf continued

We can define $\neg L = \{ s \mid s \text{ contains the substring "01"} \}$.
By simply applying the bit-flip operator to the constraints of some language def, we get a regular definition. However, there is a better way to express this. To find a machine M' ~~whose~~ that recognizes $\neg L$, you can ~~flip~~ change only the transition function of M to achieve this. Swap the range values for 0 and 1, leaving the final state and all else the same. Consider δ' below which does this to M's transition function.

δ'	0	1
q_1	q_2	q_1
q_2	q_2	q_3
q_3	q_3	q_3

M'