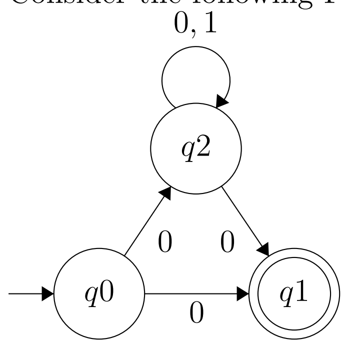
On my honor, I have not given, nor received, nor witnessed any unauthorized assistance on this work.

Print name and sign: ________________________________

Question:	1	2	Total
Points:	18	12	30
Score:			

1. Consider the following FSM:



(a) (1 point) What characteristics make this machine an NFA?

Solution: There are two possible transitions on 0 in state q0 and q2. Also, there are no transitions from state q1. Any of those 3 observations would be enough for full credit on this question.

(b) (2 points) What is the maximum number of states that a DFA equivalent to the given NFA could have? ____8____

Solution: This NFA has 3 states so the maximum possible in the DFA is 2^3 or 8 states. If asked to enumerate them, it would be the powerset of the set of states $\{q0, q1, q2\}$.

(c) (3 points) State whether or not the NFA accepts the following strings.

i. 00 __accepts__

ii. 1000 __rejects__

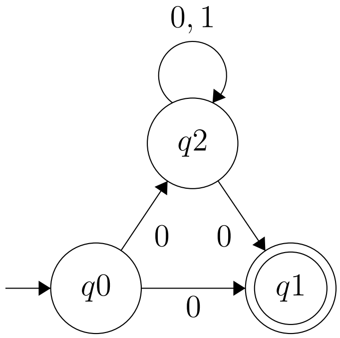
iii. 0001 __rejects__

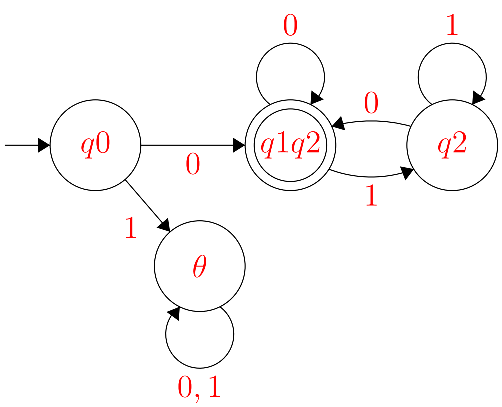
(d) (5 points) Give the formal definition for this NFA.

Solution: $M = (\{q0, q1, q2\}, \{0, 1\}, \delta, q0, \{q1\})$ where δ is represented by the transition table:

state	0	1
q0	{q1, q2}	∅
q1	∅	∅
q2	{q1, q2}	{q2}

(e) (7 points) Convert this NFA to a DFA. Show your work (building the transition table) for partial credit. (*The NFA is reproduced here for easy reference.*)





Solution:

state	0	1
q0	q1q2	∅
q1q2	q1q2	q2
q2	q1q2	q2
∅	∅	∅

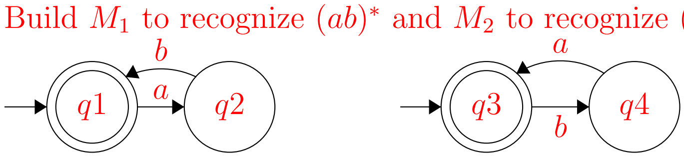
2. (12 points) Construct an NFA for the following regular expression:

$((ab)^*(ba)^*)|(aa)^*$

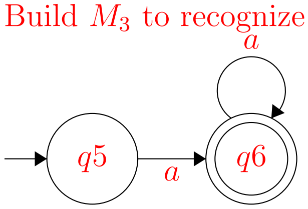
Solution: See pgs. 141-142 in FoC or pgs 59-62 in IToC.

In the steps below, you could also construct DFAs for M_1, M_2, and M_3 because DFAs and NFAs are equivalent. But NFAs are usually simpler with fewer states to keep track of.

Build M_1 to recognize $(ab)^*$ and M_2 to recognize $(ba)^*$:



Build M_3 to recognize aa^*:



Then we can combine those using ϵ transitions and a single new starting state.

$M_{Final} =$