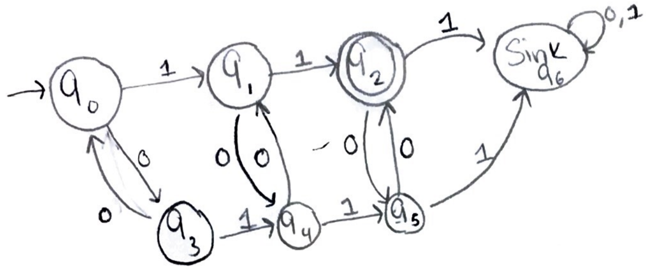
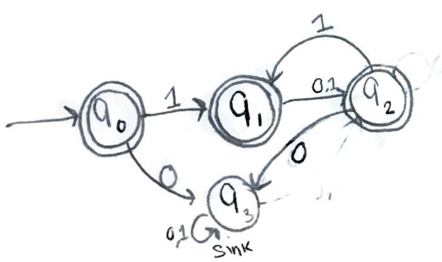
Jonathan Tso
HW01
CS 301

1. a.



b.



c. $Q: q_0, q_1, q_2, q_3$

$\Sigma: \{0,1\}$

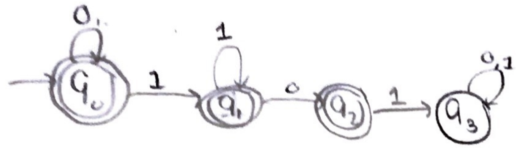
$q_0: q_0$

$F: q_0, q_1$

$\delta:$

	0	1
q_0	q_3	q_1
q_1	q_2	q_2
q_2	q_3	q_1
q_3	q_3	q_3

2. c.



d. Q : q_0, q_1, q_2, q_3

Σ : $\{0,1\}$

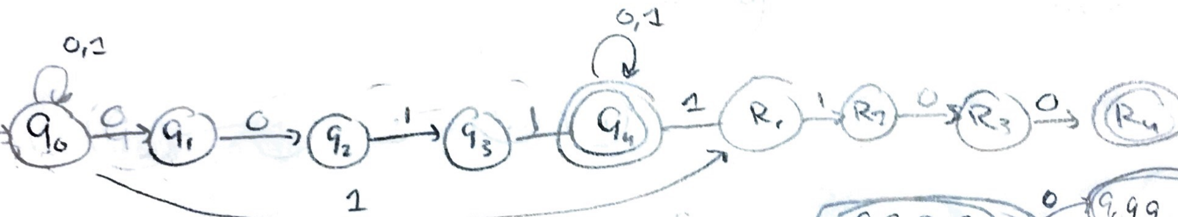
q_0 : q_0

q_F : q_0, q_1, q_2

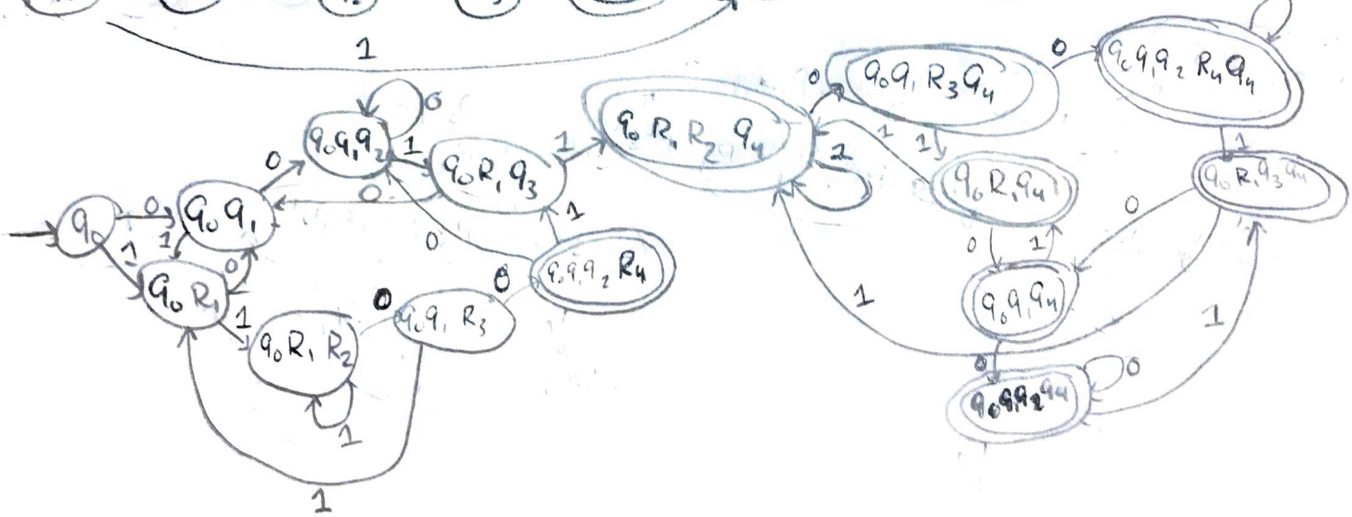
δ :

	0	1	ε
q_0	q_0	q_1	$\emptyset$
q_1	q_2	q_1	$\emptyset$
q_2	$\emptyset$	q_3	$\emptyset$
q_3	q_3	q_3	$\emptyset$

2a.



b.



3a. Suppose that the language A^R is regular. In consideration of this, we have the observation that language A is regular, indicating that this can be decided by an NFA. We define the NFA that decides A to be M_A and the NFA that decides A^R to be M_R. We also consider that M_A's 5 tuple definition to be the following:

M_A 5-tuple

Q = all $q_0 \ldots q_n$ states

Σ = the alphabet that constitutes the language

δ = $\delta(Q_n \times \Sigma) = Q_n$ s.t. Q_n is a part of Q

q_0 = q_0 s.t. q_0 is the start state of Q states

F = $q_n \ldots$ s.t. each q_n is an accept state of Q states

We introduce the following approach to reverse any and all NFA to prove an NFA exists to decide an A^R if we already know A is decidable:

M_R 5-tuple

Q = all states of M_A

Σ = all alphabet of M_A

δ = the inverse of all δ of M_A

q_0 = new q_{n+1} that is included

F = q_0 of M_A

The way M_R can be generated is to recognize that all final states are changed to non-final states, and the initial state is the new final state. Then, to be able to start from a single start state, we introduce q_{n+1} state, which has ε relation toward the original M_A final states. This lets us have a new start state that goes in every consideration. Finally, provided we have transition δ in M_A, we can invert the direction of the arrow to form the transitions for M_R. The alphabet will be retained. This proves we can have regularity for A^R since M_R can decide the language A^R.

3b. Consider the NFA M that accepts a language A. Assume that NFA N also accepts language A, but only has one accepted state.

The NFA M then has the following 5-tuple:

Q: $q_0 \dots q_n$ where all states are part of the NFA

Σ: the alphabet of the accepted language

δ: $\delta(Q_n \times \Sigma) = Q_n$ s.t. Q_n are part of Q

q_0: Q_0 s.t. ~~that~~ q_0 is a part of Q and is where M starts

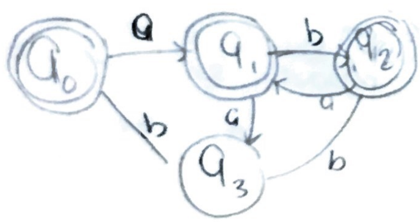
F: Q_n.. s.t. all Q_n included are part of Q and is accepted

We can consider that all values of Q, Σ, δ, q_0 are the same for NFA N. The major varying point is that F only contains one Q_n. This can be surmised by creating a new state Q_{n+1} s.t. Q_{n+1} has a relationship from all previous accept states to Q_{n+1} via ε. This also turns previous accept states into normal states and then Q_{n+1} is the single accept state. This is possible because in NFA's, we only care that we reach one of the multiple paths for a language to be accepted. Thus, any previous NFA M state will lead to NFA N's accept state, leading us to a single accepted state for NFA N.

4. In this problem, we are looking at a language that includes a, b
So the minimum alphabet is $\Sigma = \{a,b\}$

If we consider that A is accepted when it is aaaa and that we also have B is accepted when it is bbbb, we can take a look at the NFA that accepts a where a is odd and b where b is even in position.

Because we are looking at $a_i b_i \ldots$ this will always hold true since we are only considering 2 characters from the alphabet. Thus the NFA:



With this, any language with 2 distinct characters will be able to be decided by the above NFA and as such, we can have A # B always be decided. Because it can always be decided, A # B is regular.