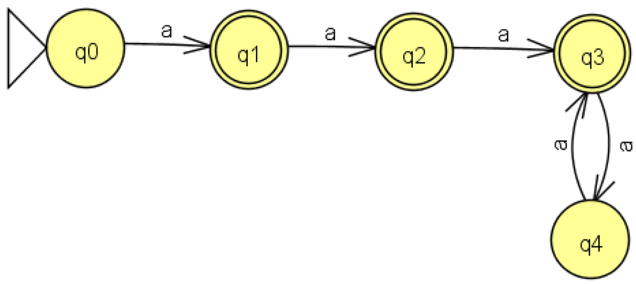
Gretel Rajamoney
April 3, 2022
CS 321 Assignment 2

Section 2.2 (Question 1):



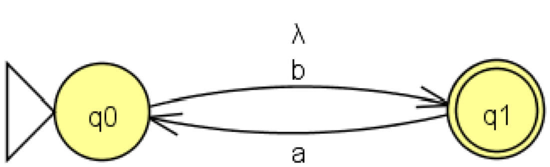
Input	Result
a	Accept
aa	Accept
aaa	Accept
aaaa	Reject
aaaaa	Accept
aaaaaa	Reject
aaaaaaa	Accept
aaaaaaaa	Reject

States: {q0, q1, q2, q3, q4}
Input Alphabet: a
Initial State: q0
Final States: {q1, q2, q3}
Transitions:

δ (q0, a) = q1
δ (q1, a) = q2
δ (q2, a) = q3
δ (q3, a) = q4
δ (q4, a) = q3

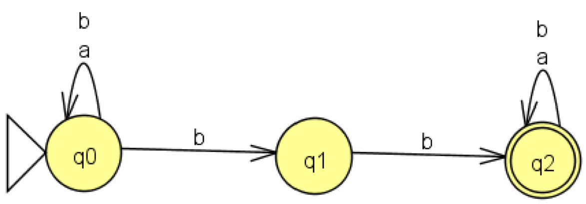
Section 2.2 (Question 2):

Given the states, input alphabet, initial state, final state, and the transitions, the following nondeterministic finite automata can be created.



Input	Result
a	Accept
aa	Accept
b	Accept
bbb	Reject
bb	Reject
abab	Accept
bbab	Reject
ab	Accept

Since this nondeterministic finite automata only accepts strings that do not contain consecutive b's, the complement of this nondeterministic finite automata accepts strings that contain consecutive b's. The complement of the previous nondeterministic finite automata is the following.



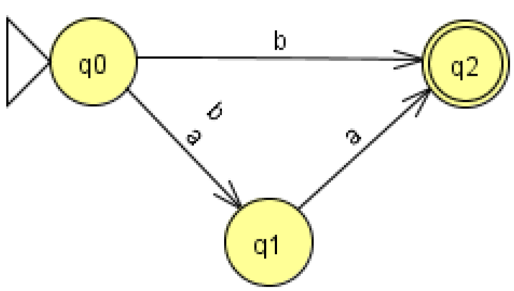
Input	Result
a	Reject
aa	Reject
b	Reject
bbb	Accept
bb	Accept
abab	Reject
bbab	Accept
ab	Reject

States: {q0, q1, q2}
Input Alphabet: {a,b}
Initial State: q0
Final State: q2
Transitions:

δ (q0, a) = q0
δ (q2, a) = q2
δ (q0, b) = {q0, q1}
δ (q1, b) = q2
δ (q2, b) = q2

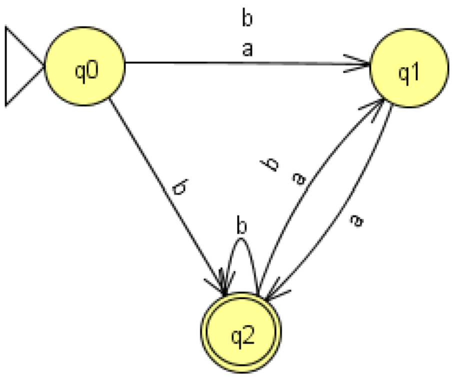
Section 2.2 (Question 3):

Given the states, input alphabet, initial state, final state, and the transitions, the following nondeterministic finite automata that accepts language L can be created.



Input	Result
a	Reject
aa	Accept
b	Accept
bbb	Reject
bb	Reject
abab	Reject
bbab	Reject
ab	Reject

Since the language L is equal to "ba + aa + b", then the nondeterministic finite automata that accepts language L* is equal to "(ba + aa + b)*". The complement of the previous nondeterministic finite automata that accepts language L* is the following.



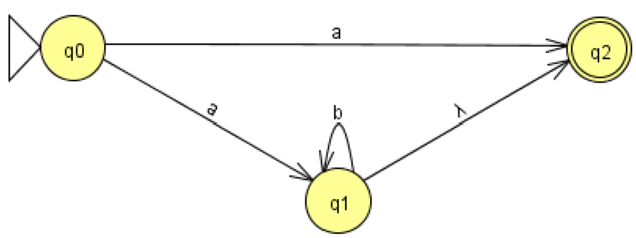
Input	Result
a	Reject
aa	Accept
b	Accept
bbb	Accept
bb	Accept
abab	Reject
bbab	Accept
ab	Reject

States: {q0, q1, q2}
Input Alphabet: {a,b}
Initial State: q0
Final State: q2
Transitions:

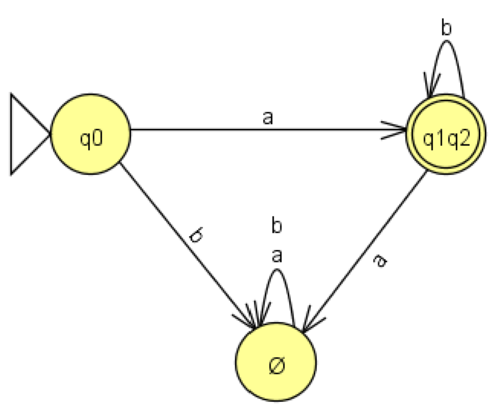
δ (q0, a) = q1
δ (q1, a) = q2
δ (q2, a) = q1
δ (q0, b) = {q1, q2}
δ (q2, b) = {q1, q2}

Section 2.3 (Question 4):

The nondeterministic finite automata prior to following the construction of Theorem 2.2:



The deterministic finite automata after following the construction of Theorem 2.2:



States: {q0, {q1, q2}, ∅}
Input Alphabet: {a,b}
Initial State: q0
Final State: {{q1, q2}}
Transitions:

δ (q0, a) = {q1, q2}
δ ({q1, q2}, a) = ∅
δ (∅, a) = ∅
δ (q0, b) = ∅
δ ({q1, q2}, b) = {q1, q2}
δ (∅, b) = ∅

Section 2.3 (Question 5):
We must first prove that a language that consists of x number of strings is regular for any natural number x, if {w} is regular for any string w. Utilizing induction on the number of strings, we can prove this.
Basis: Let x = 0 for Φ
Hypothesis: Assume that a language L that consists of x strings, is a regular language.
Inductive Step: Since L ∪ {w} is a regular language by definition, we can prove that {w} can be classified as a regular language. Therefore, our first part of our proof is done.
We must now move forward to the rest of our proof. Utilizing induction, we can prove that if w is a string over an alphabet Σ, then {w} is a regular language.
Basis: Through the basis clause of definition of a regular language, {X} and {x} are regular languages for any arbitrary symbol of x and Σ.
Inductive Step: Since {w} is regular for any arbitrary string w over Σ, we can then say that for any symbol x of Σ, {x} is classified as a regular language. Therefore, through the inductive clause of definition of a regular language, {x}·{w} is regular, meaning that {xw} is also regular. Thus concluding our proof.

Section 2.3 (Question 6):
We must prove that if a language L is regular, then so is language L^R. We know that by definition, given a regular language, it is accepted by some finite automaton. Meaning that there is a finite set of states with appropriate transitions. These transitions take us from the initial state to the final state if the input is a string contained in language L. In order to accept the reverse of language L, we must also reverse the direction of all transitions. Meaning that our final state becomes our initial state, and vice versa. Now

that our original machine is reversed, it can accept the language L^R. If there exists a deterministic finite automata or a nondeterministic finite automata for a language, then it must be classified as a regular language. Thus proving that if a language L is regular, then so is language L^R.