Gretel Rajamoney
March 29, 2022
CS 321 Assignment 1

Section 1.1 (Question 1):

Hypothesis: Let $P(n)$ be the hypothesis for a set S of size n, meaning $n = |S|$ and $|2^S| = 2^{|S|}$.

Base Case: Since the set is finite, we must start with a set of size 0. Let $n = 0$ in order to examine an empty set. When $n=0$, the cardinality of the empty set is equal to 1, since $|2^S| = 1$ when $S = \text{null}$. The cardinality of the null set is 0, therefore $|S| = 0$, meaning that $2^{|S|}$ equals 2^0 which results in 1. Thus proving that $|2^S| = 2^{|S|}$ when $n = 0$.

Inductive Step: Assume $P(x)$, we must prove $P(x+1)$.

Assuming $P(x)$, $|2^S| = 2^{|S|}$ holds for any set S of size x. This means that $x = |S|$, and $|2^S| = 2^x$. Therefore the power of set S contains 2^x elements in total. Because it can be concluded that the power set is the set of all sub-sets of S, we know that set S has 2^x distinct sub-sets. We can then move forward to analyzing a set S' with a size of $x+1$. This means that the power set $2^{S'}$ is the set of all sub-sets. The sub-sets of S' equals the sub-sets that do include the added element x and the sub-sets that do not include the added element x. This means that the total number of sub-sets for S', is equal to $2^x + 2^x$, which is simplified to 2^{x+1}. Since for a set S' of size $x+1$, the size of the power set is 2^{x+1}, $|S'| = x+1$, and $2^{|S'|} = 2^{x+1}$, we can conclude that $|2^{S'}| = 2^{|S'|}$ for any set S' with an element size of $x+1$.

continue

Since we have proven that $|2^S| = 2^{|S|}$ for an arbitrary set S with x elements, we can imply that $|2^{S'}| = 2^{|S'|}$ for an arbitrary set S' with an element size of x+1. Thus completing our inductive step.

We have proven that $|2^S| = 2^{|S|}$ using induction.

Section 1.2 (Question 2):

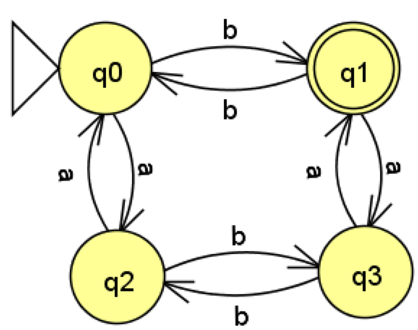
Some examples of the language generated by the grammar include: "aa", "aaaa", "aaaaaa", ... continuing in increments of 2.

It can be inferred from analyzing the language that the output will always contain an even number of 'a's.

The simple English description of the language can be an automata that only accepts an even number of 'a's.

Section 2.1 (Question 3):

Input	Result
a	Reject
b	Accept
ab	Reject
abb	Reject
aab	Accept
aaab	Reject
aabbb	Accept
aaaabb	Reject
aaabb	Reject
aaaabbb	Accept

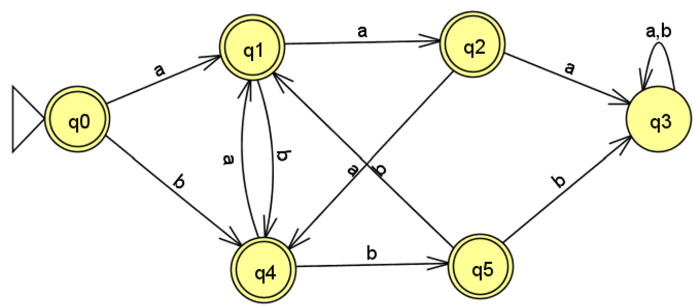


States: {q0, q1, q2, q3}
Input Alphabet: {a, b}
Initial State: q0
Final State: q1
Transitions:

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

Section 2.1 (Question 4):

Input	Result
aaa	Reject
bbb	Reject
aabbbbab	Reject
abab	Accept
aabb	Accept
aaab	Reject
ababab	Accept
aaabbb	Reject
baaa	Reject
ab	Accept
a	Accept
b	Accept



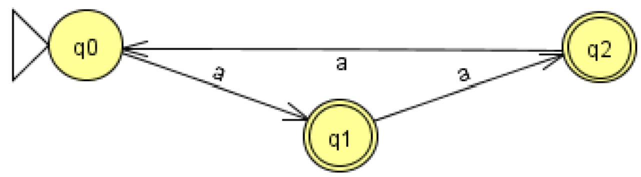
States: {q0, q1, q2, q3, q4, q5}
Input Alphabet: {a, b}
Initial State: q0
Final States: {q0, q1, q2, q4, q5}
Transitions:

δ (q0, a) = q1
δ (q1, a) = q2
δ (q2, a) = q3
δ (q3, a) = q3
δ (q4, a) = q1
δ (q5, a) = q1

δ (q0, b) = q4
δ (q1, b) = q4
δ (q2, b) = q4
δ (q3, b) = q3
δ (q4, b) = q5
δ (q5, b) = q3

Section 2.1 (Question 5):

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



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

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