1) Prove by induction that if S is a finite set, then $|2^S| = 2^{|S|}$.

Base case: when S is an empty set, which means S = {∅}, then $|S|= 0$, $|2^0| = |1| = 1$, $2^{|0|} = 2^0 = 1$. Thus, $|2^S| = 2^{|S|}$ for the base case.

Inductive Assumption: for any finite set S = {1, 2, …, n}, $|2^S| = 2^{|S|}$.

Inductive Step: to prove this, we need to prove that this is true for set S = {1, 2, …, n, n+1}. First, we know that every subset of {1, 2, …, n} are also the subsets of {1, 2, …, n, n+1}. That is to say, if we add (n+1) to each single subset of {1, 2, …, n}, we can get all of the new subsets of {1, 2, …, n, n+1} which do not contained in {1,2,…,n}.

For example: all the subsets of {1,2} = {∅,{1},{2},{1,2}}. If we want to get new subsets of {1,2,3} which do not contained in {1,2}, we can add 3 in each of the single subsets of {1,2}, which is {∅, {1,3},{2,3},{1,2,3}}.

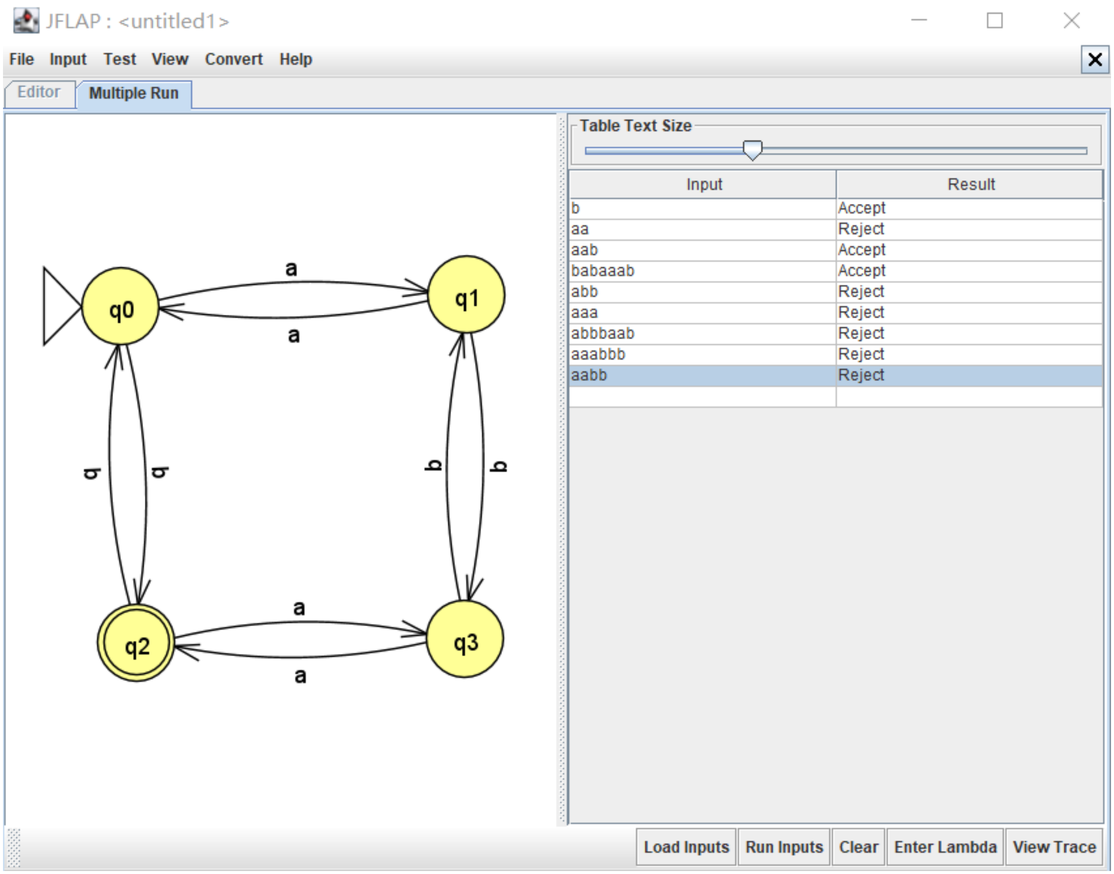
So in this case, when S = {1,2,…,n}, there are $|2^S|$ number of subsets of {1,2,…,n,n+1} which do not combine (n+1) in each single subsets. Also, there are $|2^S|$ number of new subsets of {1,2,…,n,n+1}, which do not contained in {1,2,…,n}. So when S = {1,2,…,n,n+1}, the number of subsets is $|2^S| = |2^{\{1,2,\ldots n,n+1\}}| = 2*|2^{\{1,2,\ldots n\}}|$. Based on the induction assumption, we can write this equation as $2*|2^{\{1,2,\ldots n\}}| = 2 * 2^{|\{1,2,\ldots n\}|}$. So, $|2^{\{1,2,\ldots n,n+1\}}| = 2 * 2^{|\{1,2,\ldots n\}|} = 2 * 2^n = 2^{n+1} = 2^{|\{1,2,\ldots n,n+1\}|}$.

Thus, if our claim is true for n, it must also be true for n+1. Since n can be any number, the statement must be true for all n.

2) Give a simple English (not math) description of the language generated by the grammar with the productions: S -> aSa | aa

This grammar will only generate an even number of a as the answer.

3) Create a dfa for Σ = {a, b} that accepts the set of all strings with an even number of 'a's and an odd number of 'b's. Zero is even, so your dfa should accept 'b', but should not accept 'aa'.



states: {q0,q1,q2,q3}
input alphabet: {a,b}
initial state: q0
final states: q2
transitions:

$\delta(q0,a) = q1$

$\delta(q1,b) = q3$

$\delta(q3,a) = q2$

$\delta(q2,b) = q0$

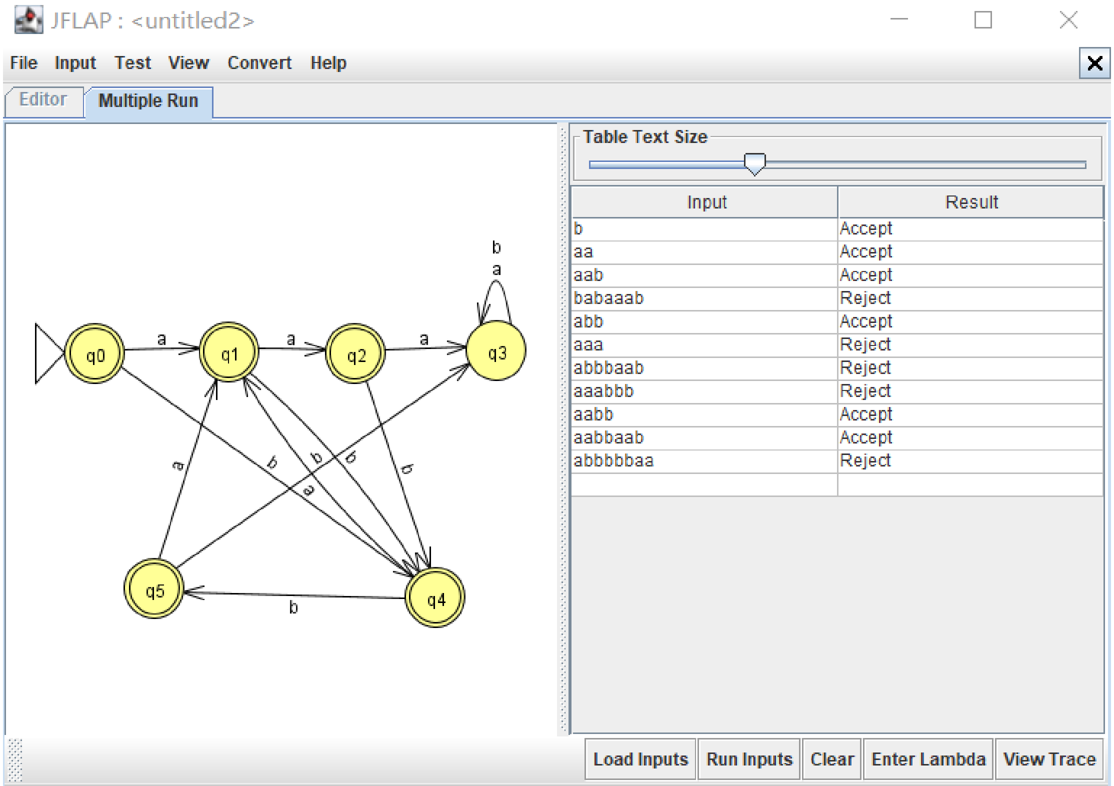
$\delta(q0,b) = q2$

$\delta(q2,a) = q3$

$\delta(q3,b) = q1$

$\delta(q1,a) = q0$

4) Create a dfa for $\Sigma = \{a, b\}$ that accepts the set of all strings in which the same symbol does not occur three or more times in a row. For example it should not accept "aaa" or "aabbbbab".



states: $\{q0,q1,q2,q3,q4,q5\}$
input alphabet: $\{a,b\}$
initial state: q0
final states: $\{q0,q1,q2,q4,q5\}$
transitions:

$\delta(q0,a) = q1$

$\delta(q0,b) = q4$

$\delta(q1,a) = q2$

$\delta(q1,b) = q4$

$\delta(q2,a) = q3$

$\delta(q2,b) = q4$

$\delta(q3,a) = q3$

$\delta(q3,b) = q3$

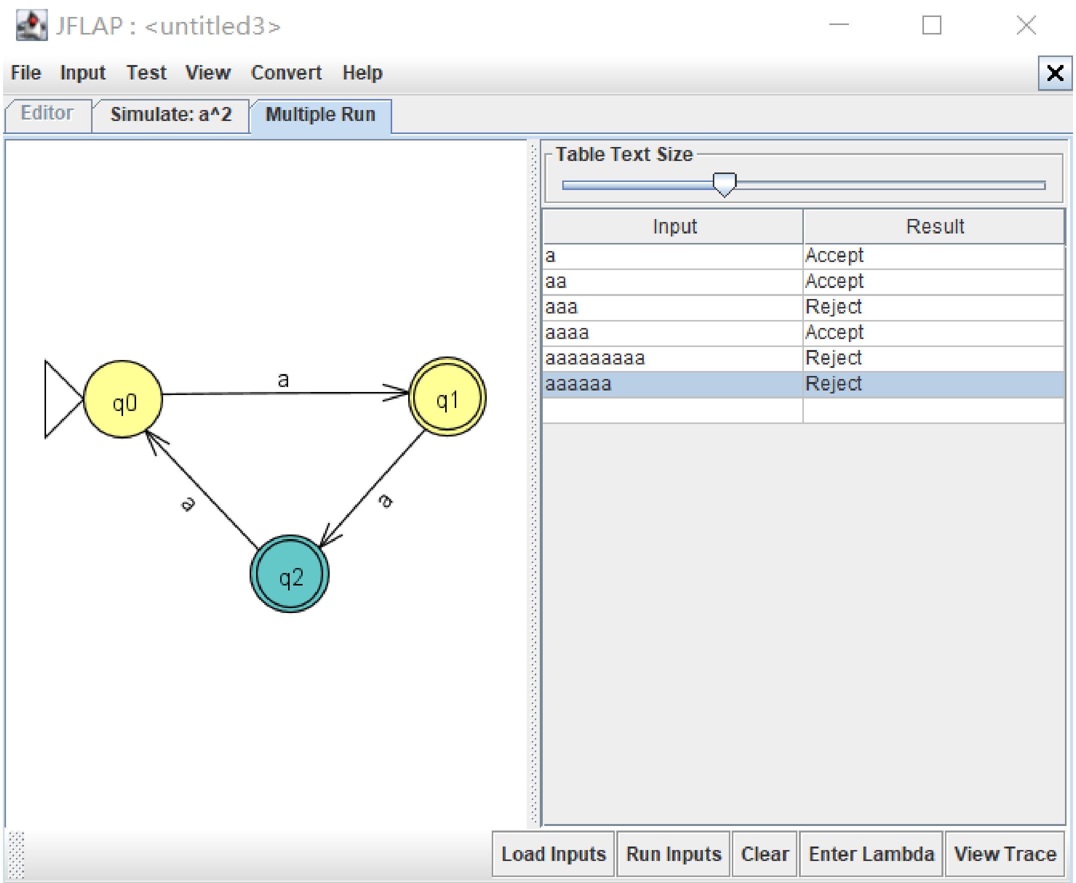
$\delta(q4,a) = q1$

$\delta(q4,a) = q1$

$\delta(q5,a) = q1$

$\delta(q5,b) = q3$

5) Show that the language L = {a^n : n is not a multiple of 3} is regular.



states: {q0,q1,q2 }
input alphabet: a
initial state: q0
final states: {q1,q2}
transitions:

$\delta(q0,a) = q1$

$\delta(q1,a) = q2$

$\delta(q2,a) = q0$