Combining Automata

Motivation

It is always possible to start and build up an recognizer from scratch, but if we come up with procedures to combine automata, it is possible to use existing machines to complete tasks more rapidly. This module will example the construction of the following machines, and how to use JFLAP to determine the resulting machine.

1) How to take the complement of a machine: a machine which accepts anything which is not accepted by the original machine.

2) How to take the union of a machine: a machine which accepts that which is accepted by **any** of two or more machines.

3) How to take the intersection of a machine: a machine which accepts that which accepted by **all** of two or more machines.

   By using 1,2 and 3, we can define:

4) The subtraction of two machines: a machine which accepts that which is accepted by a given machine, but not the other.

Questions to think about

1) What is the complement of a machine consisting of a single initial state which accepts nothing?

2) What is the union of any machine with a single initial state which accepts nothing?

3) What is the intersection of any machine with a machine which accepts everything?

4) What is the result of subtraction of a machine from itself?

Complement of a Machine

Consider a very simple machine, using the alphabet *{a,b}*, for example a machine which accepts the symbol *a.*
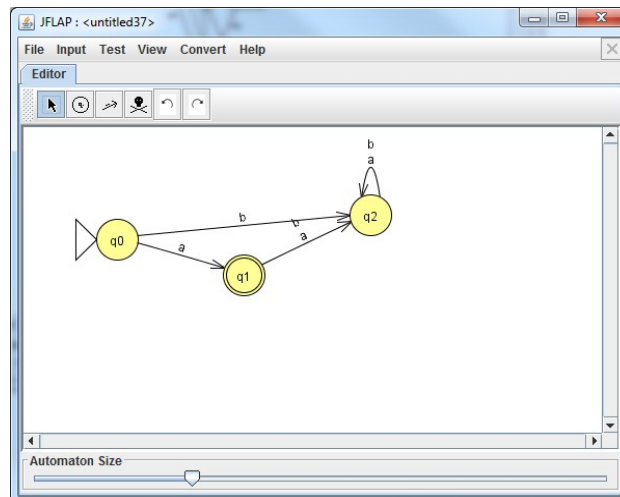
The complement of this machine, would be a machine which accepts anything other than the single symbol *a* , including accepting nothing.
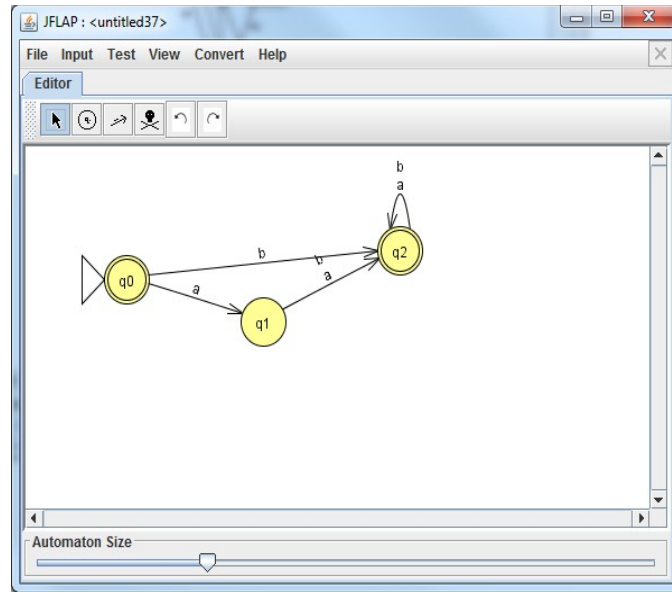
To correctly take the complement of this machine, it is necessary to provide a transition to an accept state on any additional input. Since the input alphabet is *{a,b}* both of these symbols must cause the transition to an accepting state.

Questions and Practice

1) If the alphabet is increased to *{a,b,c}* what would have to be done to the above example machine to take the complement?

2) What is the complement of the complement of the above machine?

Union of two Machines

The union of two machines accepts the constructs which each of the two machines accept. The input sequence is fed to both machines at the same time, and the state of the union of the machine is the pair (or cross product) of the state each machine is at a given moment in time.   The combined machine is said to have accepted the input, if either of the individual machines is in an accept state.

Lets consider the really simple machines on the inputs *{a,b}* one machine accepting the construct *a* and another the construct *b.*
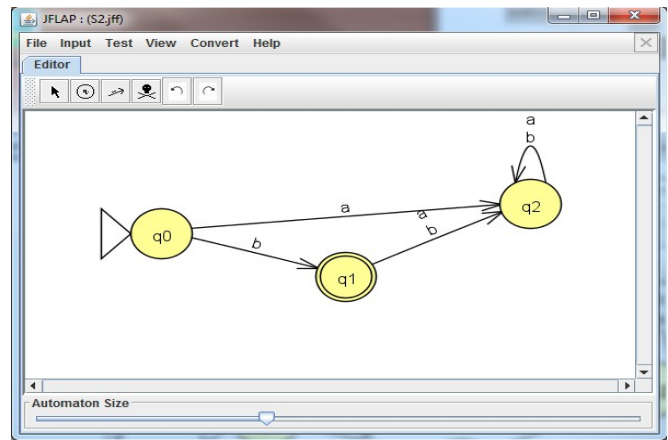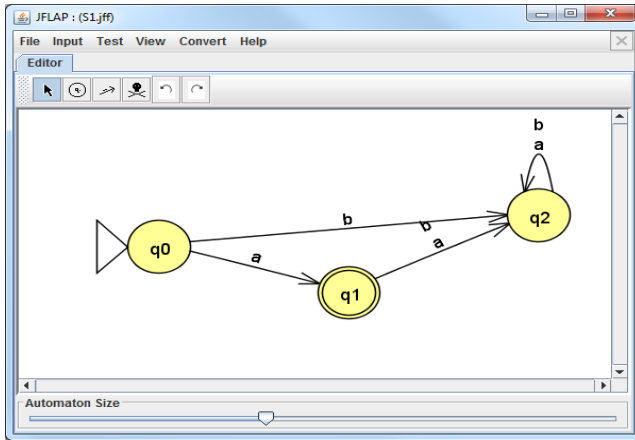
At any moment in time, these two machines, operating independently and on **separate** inputs can be in the states [{q0,q0}, {q1,q0},.{q0,q1},{q1,q1}]. When we combine these, and have the two machines operating on the same inputs the set of **reachable** states are [{q0,q0}, {q1,q0},.{q0,q1}]. Why?

To use JFLAP to construct the DFA corresponding to the union we will define an NFA to model the joint states of the machines, and then convert the NFA to a DFA and minimize the DFA.

Step 1: Create a NFA from the two DFAs. Open one of the DFAs and use the Combine Two option on the Convert menu to select the other machine.

Step 2: Add an initial state with null transitions to the initial states of the two machines. *Test to see if it accepts both **a** and **b***

Step 3: Using JFLAP, convert to a DFA by using the convert>DFA tool, and Complete and press Done. *Can you identify the resulting states with the corresponding pairs of the original machine?*

Step 4: Minimize the resulting system. Click on a node in the right most window, click on Complete Subtree in the Minimization tool, and then finish. Press complete in the left window and then Done
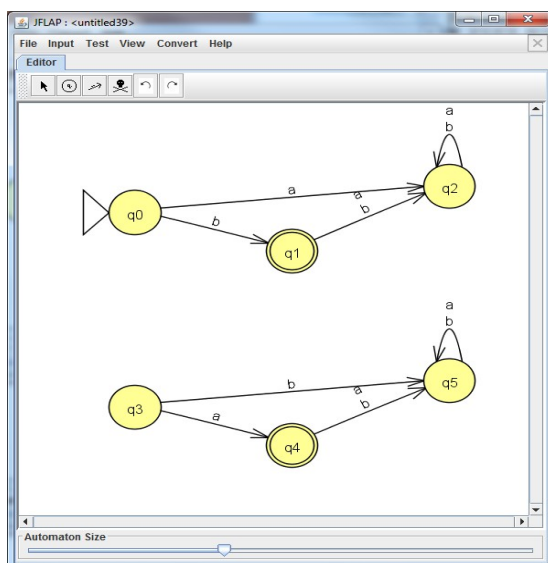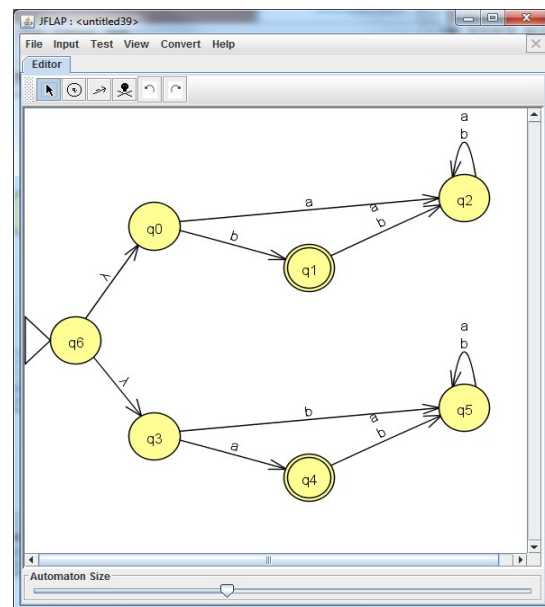

Illustration 1: Result of Step 1
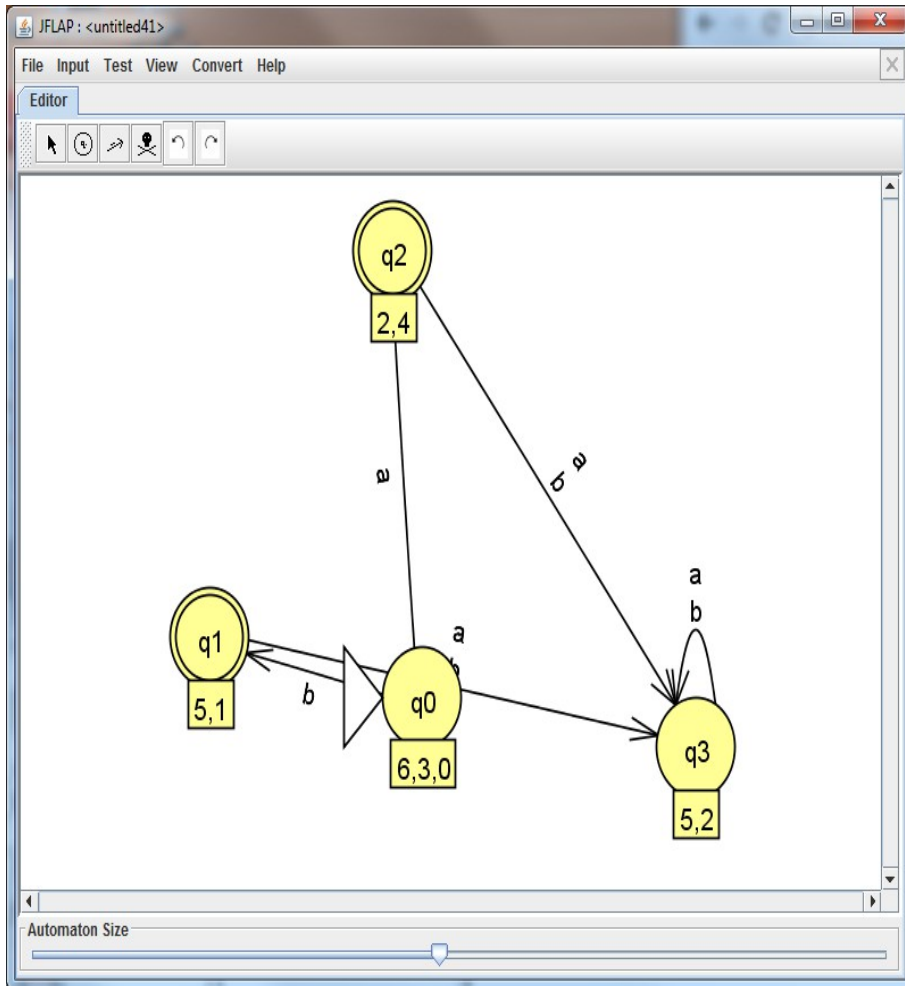

Illustration 2: STEP 2
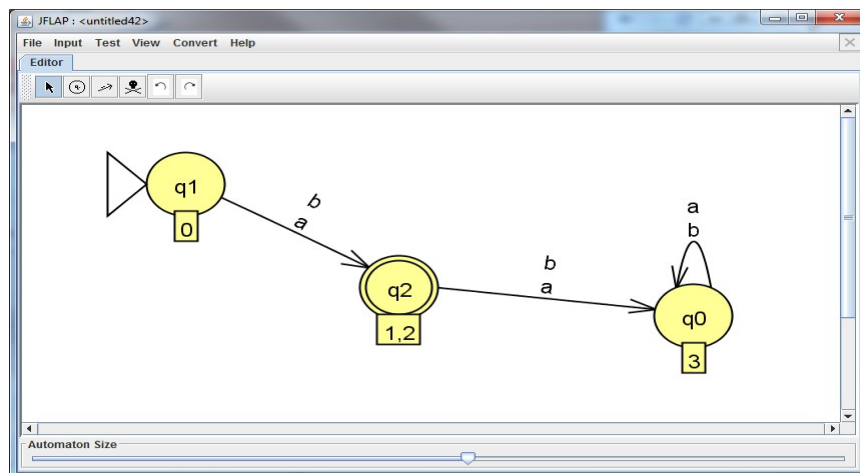
Illustration 3: Result of Converting to a DFA

Illustration 4: Result of Step 4

Questions and Practice:

1) See if you can identify the pairs (cross product) of the states in the final machine
2) Repeat the procedure and demonstrate the union of a machine with it self is the machine it self
3) Repeat the procedure with a machine which accepts nothing is also the original machine.
4) Repeat the procedure with a machine which accepts everything and discuss the result.

Intersection of Two Machines

To produce the intersection of two machines, we will use DeMorgan's theorem so we can use the procedure for union. Recall, DeMorgan's theorem states, that the complement of the union is equal to the intersection of the complement of the individual machines.

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

To find

$$A \cap B$$

We use

$$\overline{\overline{A} \cup \overline{B}} = A \cap B$$

Questions and Practice:

1) Review DeMorgan's theorem and use a truth table to demonstrate it.
2) Form the Complement of the two machines above

As an example of the procedure, lets form the intersection of the simple machines above.

Step 1 : Take the complement of the two machines

Step 2: Take the union of the two complemented machines

Step 3: Convert to DFA and Minimize

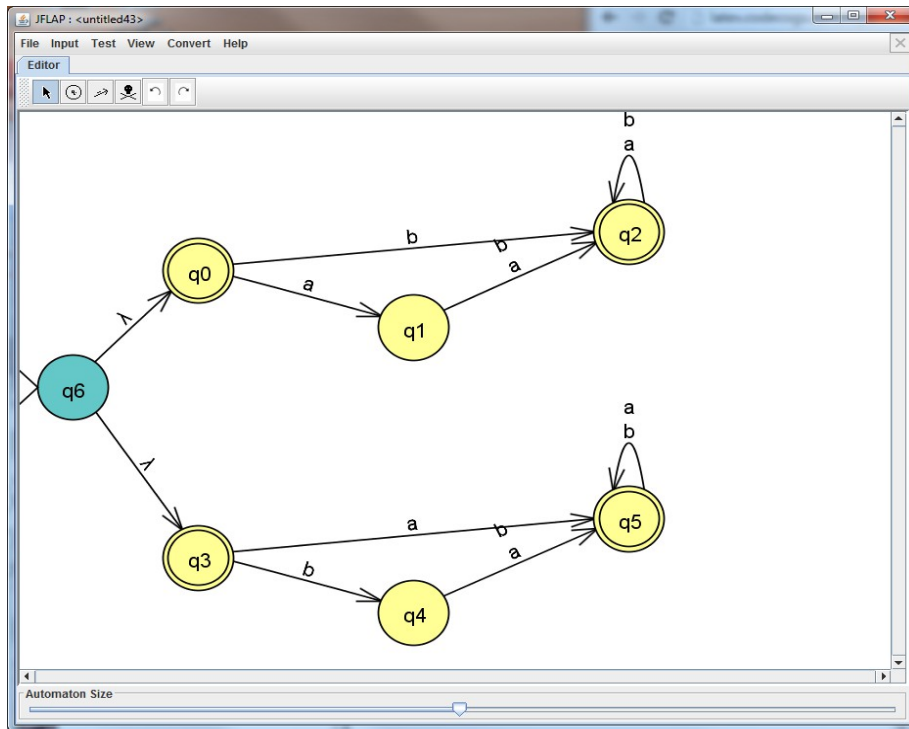Step 4: Take the complement of the final machine.
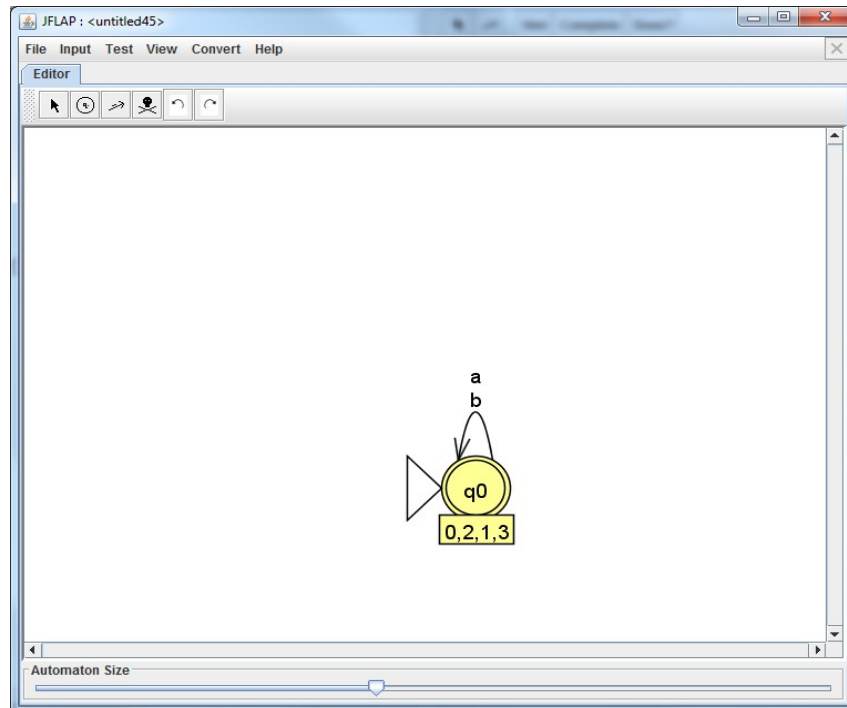
Illustration 5: Result of Step 1 and 2

Illustration 6: Prior to Complement

Questions and Practice

1) After taking the complement of the machine, what is the resulting machine? Is this correct?
   (remember the first machine only accepted *a* and the second *b*).

2) What is the intersection of a machine with itself?

3) What is the intersection of a machine with the machine which accepts everything?

Subtraction of Two Machines.

Now that we have intersection, it is a simple matter to remove a collection of constructs accepted by one machine from another.

$$A \backslash B = A \cap \overline{B} = \overline{\overline{A} \cup B}$$

Where $A \backslash B$ is the set of constructs accepted by A but not by B.

For example, consider the machine (A) which accepts a or b, and remove the states associated with the machine(B) which accepts only b.
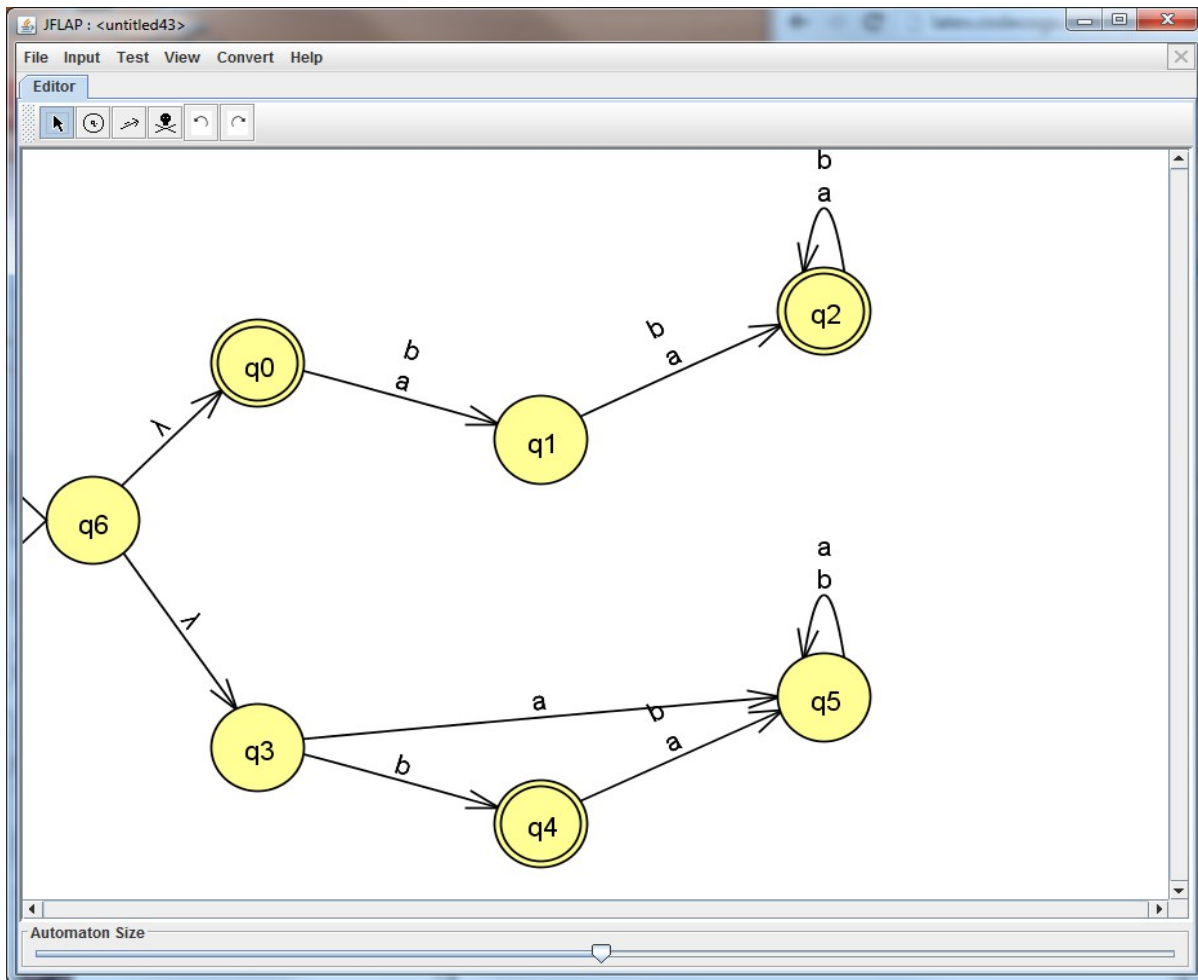

Illustration 7: NFA used to form the union of Complement of A and B

Questions and Practice

1) Identify the complement of the machine which accepts a or b in the NFA above.

2) Test and verify the the resulting machine after converting to a DFA and minimizing.
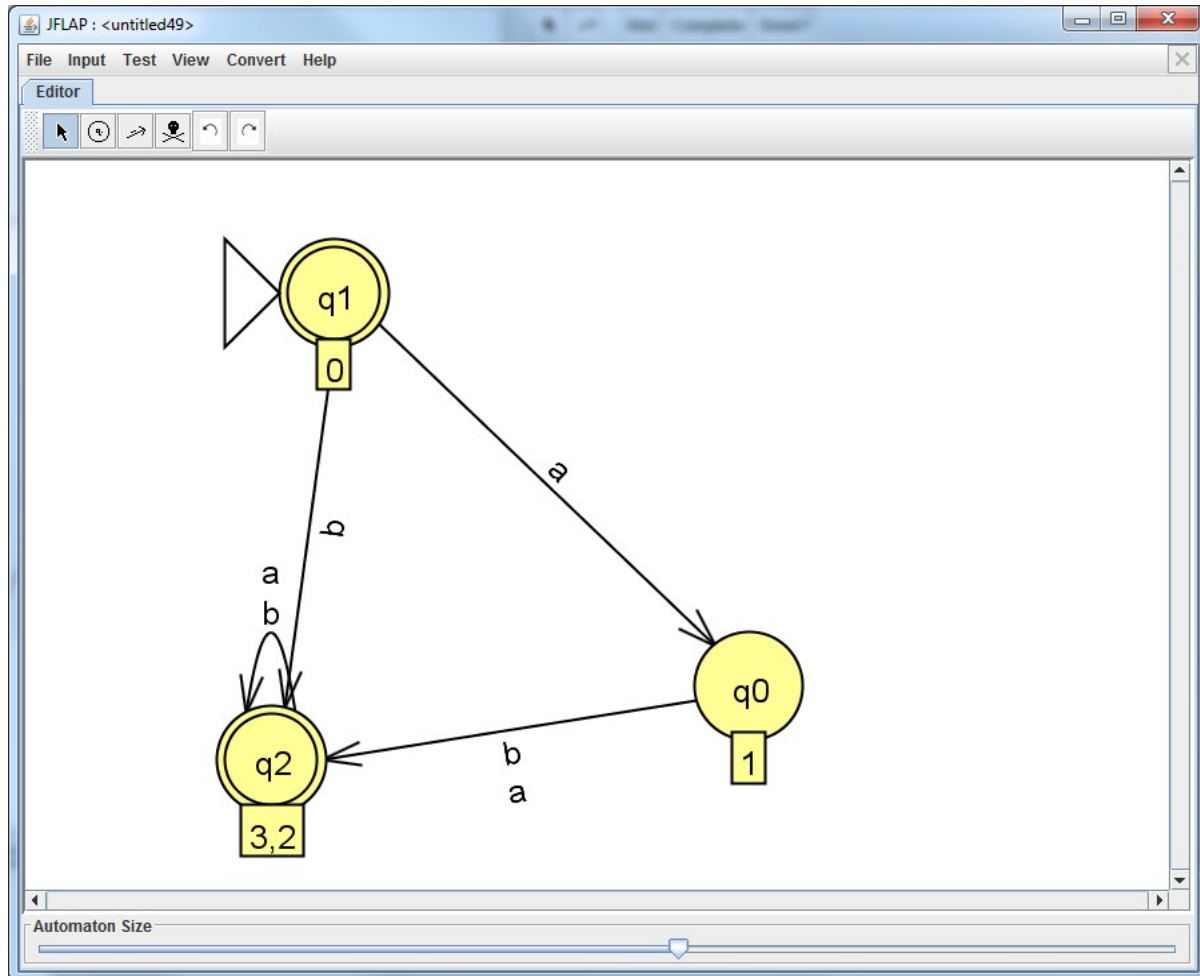

Illustration 8: Result of Removing machine B from A, prior to complement