Example: Mealy Machine$_{JP}$

Define a Mealy machine that prints the one's complement of an input bit string. That is, the machine transforms every **0** to **1** and every **1** to **0**. For example, the input string **0010** produces output **1101**.

Recall that an Mealy machine is defined as a 5-tuple (Q, Σ, δ, q0, F) where

- Q is a finite set of states
- Σ is a finite alphabet of symbols for forming the input string
- Γ is the finite set of symbols in the output alphabet
- δ is the transition function, $\delta: Q \times \Sigma \rightarrow Q \times \Gamma$
- q0 is the start state (q0 □ Q)

Sample Solution

What are the input and output alphabets?

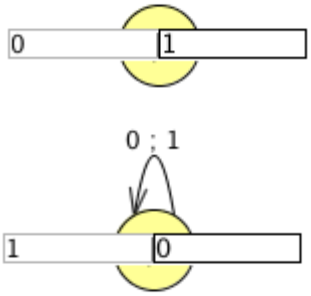
In this case, the input alphabet and output alphabet are identical: $\Sigma = \Gamma = \{0, 1\}$

How is the transition function defined?

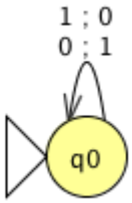
Consider that every transition must output the appropriate symbol for the input symbol just read. In this case, every **0** in the input must result in an output of **1**. Likewise, every **1** in the input must result in an output of **0**.

Note that when you define a transition for a Mealy machine in JFLAP, you must enter both the input symbol (left column) and the output symbol (right column).

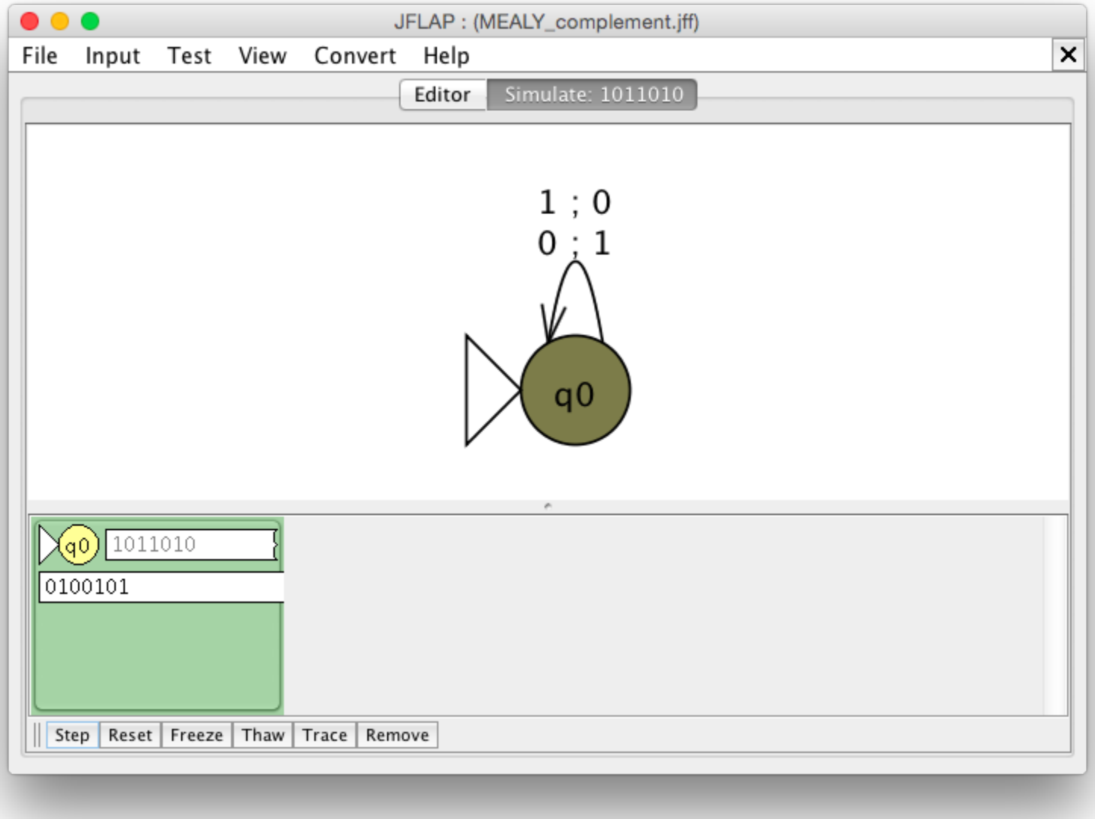
Here is the sequence that demonstrates entering the two appropriate transitions:



Here is the resulting Mealy machine that prints the one's complement of an input bit string (see `MEALY_complement.jff`):



Now step through input strings and observe the output. For example, the input string **0100101** produces output **1011010**.



1011010

0100101