NONDETERMINISTIC FINITE AUTOMATA

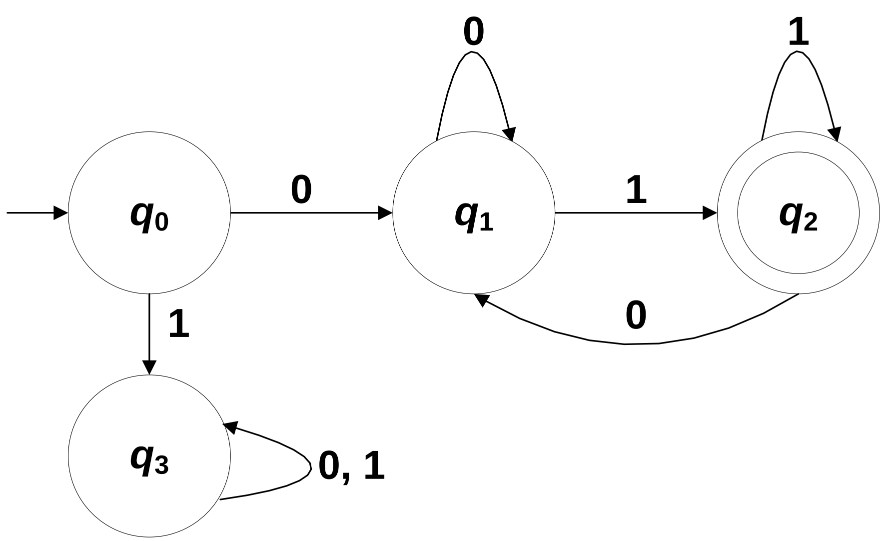
iACADEMY
SCHOOL OF COMPUTING • SCHOOL OF BUSINESS • SCHOOL OF DESIGN

SCHOOL OF COMPUTING
MITCH M. ANDAYA

NONDETERMINISM

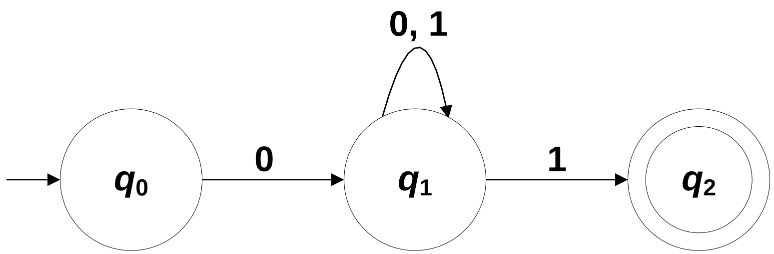
- The main characteristic of a DFA is that for each input symbol, the automaton can move to only one state from any given current state.

- Consider a DFA that accepts strings over the alphabet $\Sigma = \{0, 1\}$ that starts with a 0 and ends with a 1.



NONDETERMINISM

- Consider now the following finite automaton:

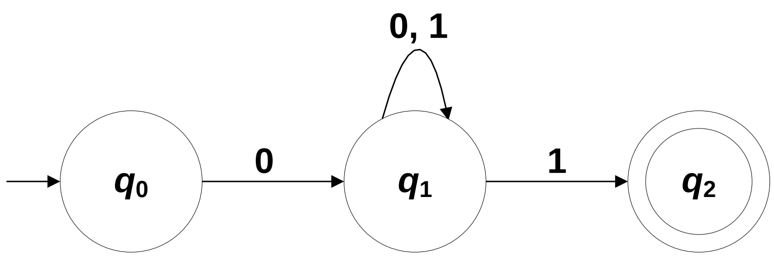


This machine also accepts strings that start with a 0 and end with 1.

Observe that if the machine is at state q_1 and a 1 arrives, the machine goes to state q_2 and stays at q_1 at the same time.

This machine is not a DFA because there is a situation in which an input symbol causes the automaton to move to more than one state.

NONDETERMINISM



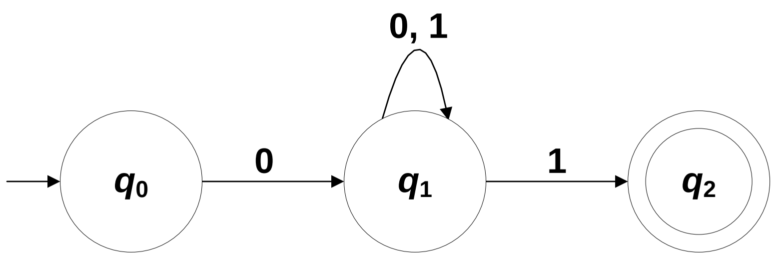
In this case, the NFA will consider both possibilities.

The NFA will make a "copy" of itself. One copy continues the computation at state q_1 while the other continues at state q_2.

Effectively, the NFA is said to be in two states, q_1 and q_2.

This is in contrast with a DFA since DFAs can only be in one state at any given time.

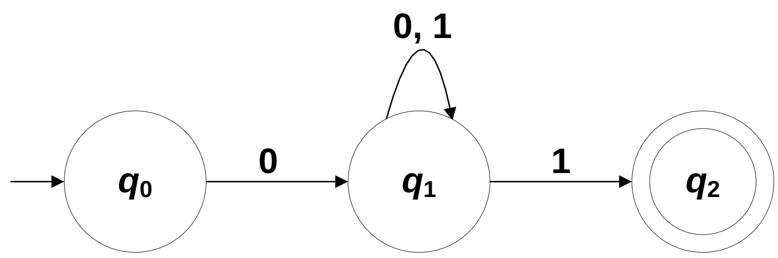
NONDETERMINISM



- The given graph is the state diagram for a ***nondeterministic finite automaton*** (NFA).
- Because an NFA can move to more than one state given a single input symbol, an NFA is said to have "choices."
- Take the given NFA as an example. If the NFA is currently in state q_1 and an input symbol 1 arrives, the NFA has the option of staying in state q_1 or going to state q_2.

NONDETERMINISM

- Notice also that in an NFA, it is possible for a state not to have a transition edge for one or more input symbols.



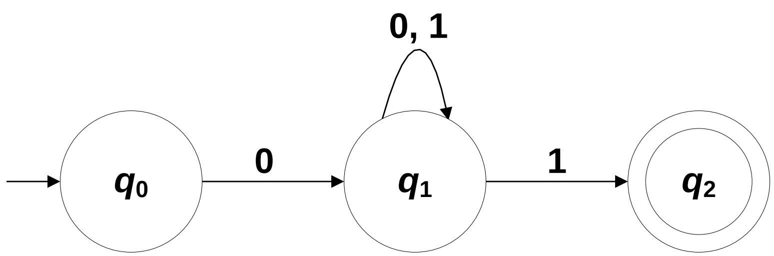
In the given example, take note that for state q_o, there is a transition edge for input symbol 0 but none for 1.

If a copy of the NFA is at state q_o and an input symbol 1 arrives, this copy stops processing because it has nowhere to go. This copy of the NFA simply "dies." Similarly, if a copy of the NFA is at q_2 and a 1 or a 0 arrives, this copy also stops processing and dies.

NONDETERMINISM

- Here is an example to illustrate how an NFA processes strings:

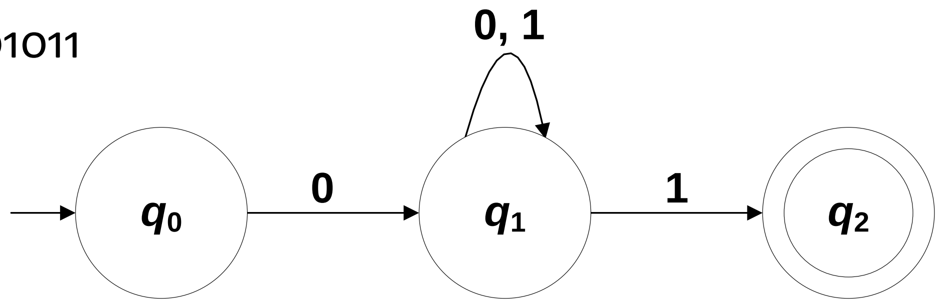
  Given the following NFA:

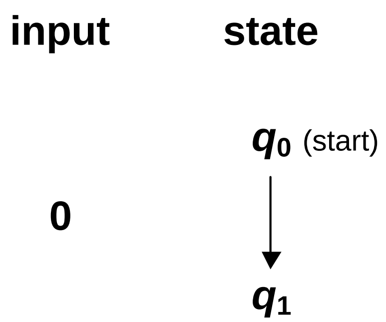


Determine if the string 01011 will be accepted.

NONDETERMINISM

01011





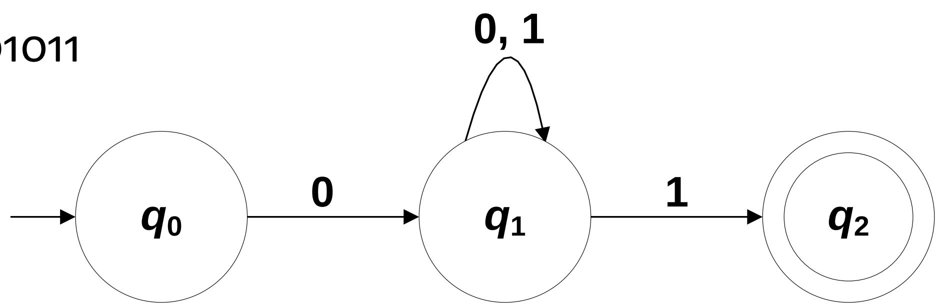
1. Initially, the NFA will be at state q_o.

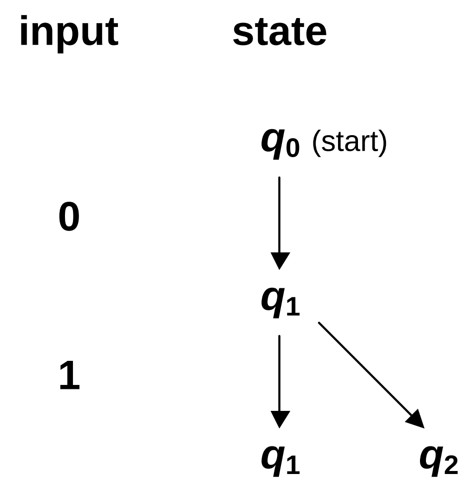
   Assume the first input symbol (0) arrives.

   So the NFA goes to state q_1.

NONDETERMINISM

01011





2. The second input symbol (1) arrives.

   The NFA has two options, either it stays at q_1 or goes to q_2.

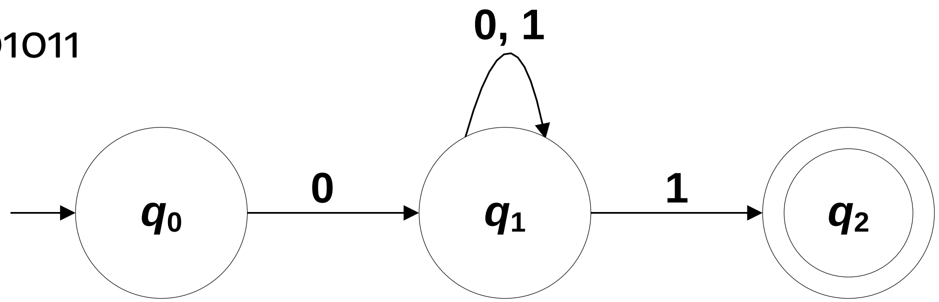
   The machine will try both possibilities at the same time.

The machine made another copy of itself.

One copy to continue the processing at q_1 and the other at q_2.

NONDETERMINISM

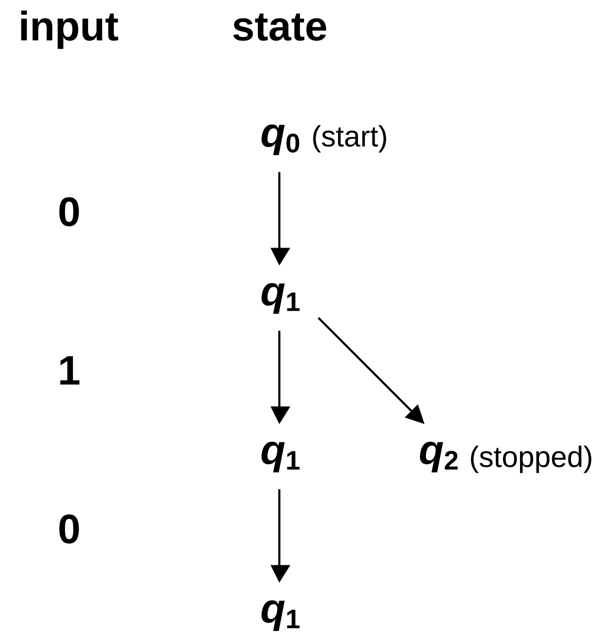
01011



3. Assume now that the third input symbol (0) arrives.

   One copy of the NFA (the one at q_1) stays at q_1.

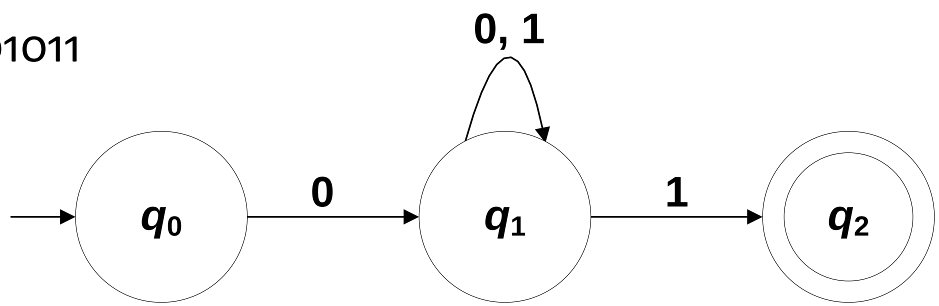
   The other copy (the one at q_2) dies because it has nowhere to go.



At this point, there will only be one remaining copy of the NFA (the one that is still at q_1).

NONDETERMINISM

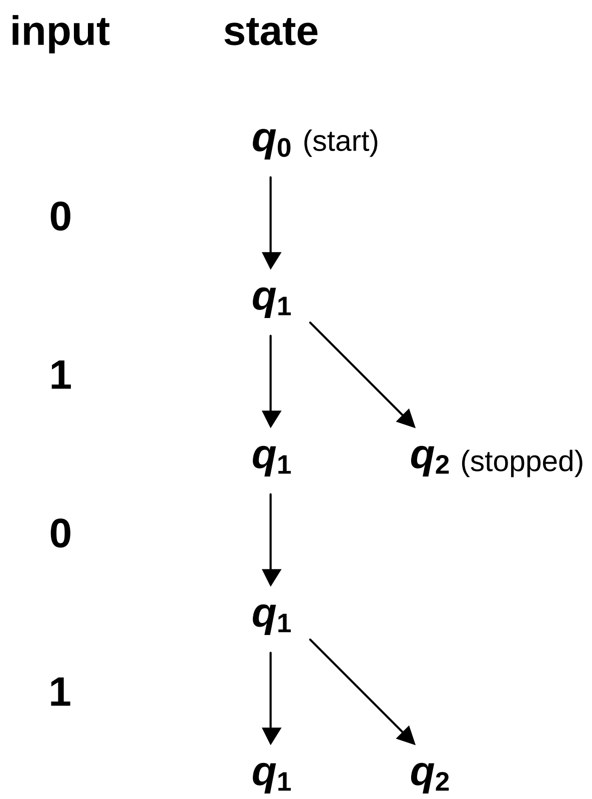
01011



4. Assume now that the fourth input symbol (1) arrives.

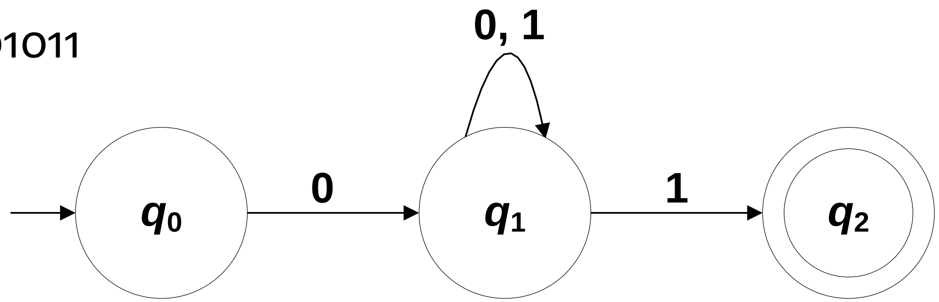
   There are again two options, either stay at q_1 or go to q_2.

   The NFA will again consider both possibilities.



NONDETERMINISM

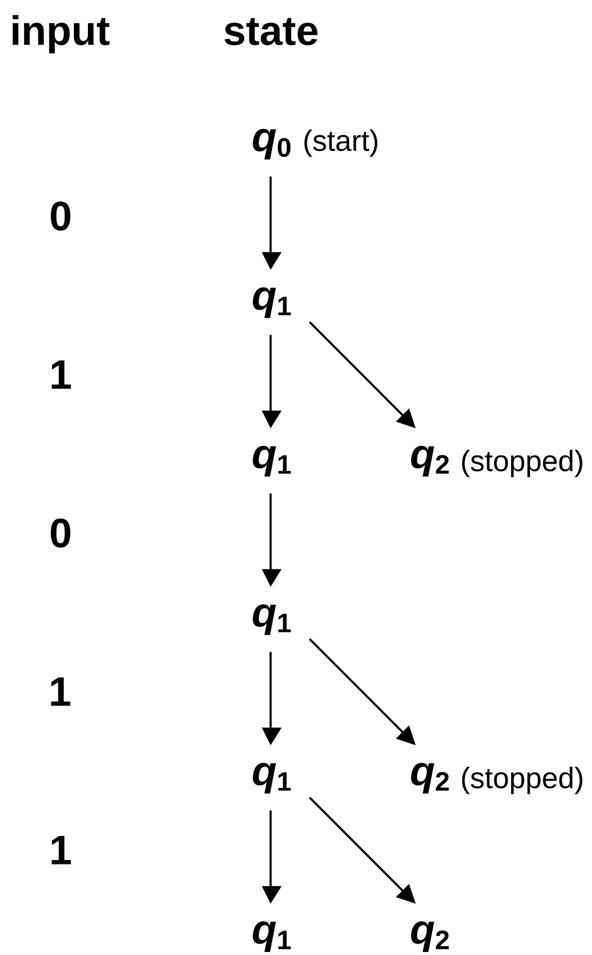
01011



5. Assume now that the fifth and last input symbol (1) arrives.

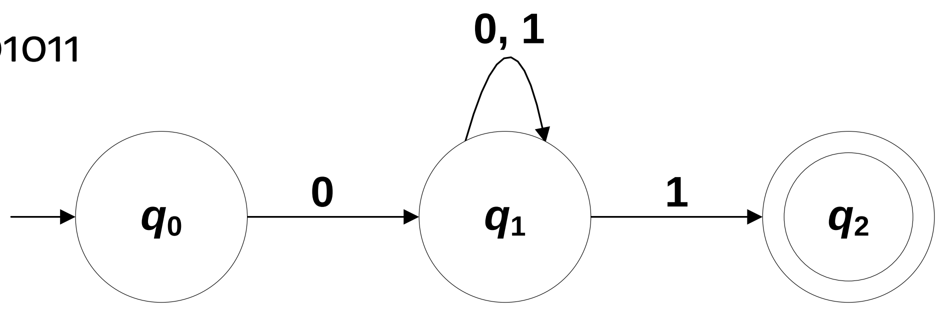
   One copy of the NFA (the one at q_1) has the option of staying at q_1 or going to q_2. As usual, the NFA will consider both.

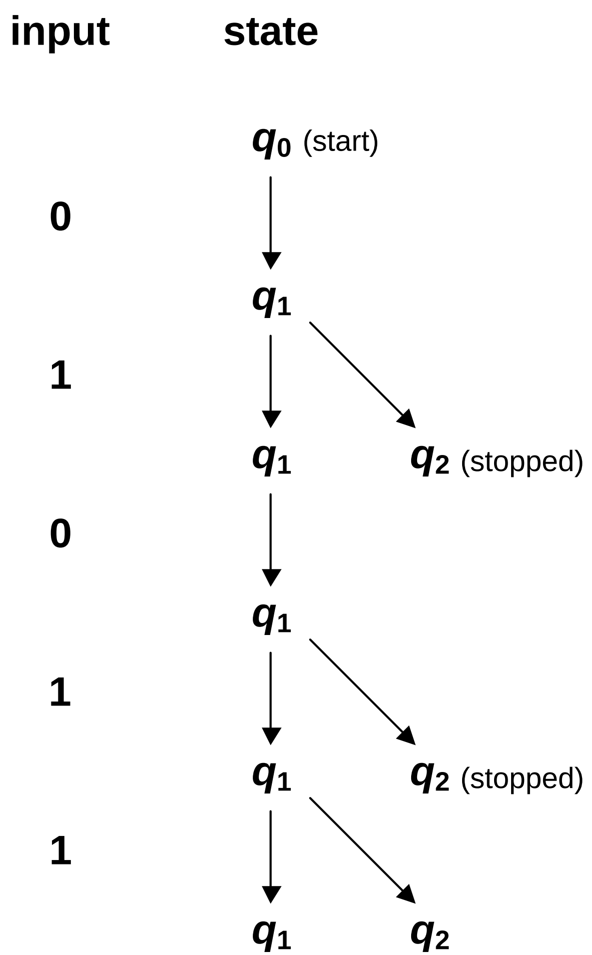
   The other copy (the one at q_2) stops processing.



NONDETERMINISM

01011





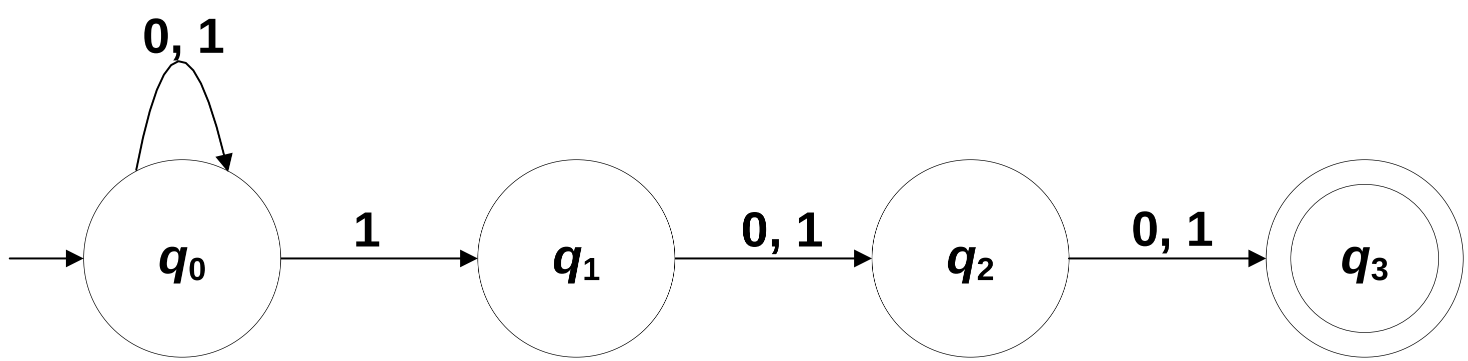
There are two copies of the NFA.

One copy is currently at state q_1 and the other at state q_2. Since one of the copies is at a final state, the NFA accepts the string 01011.

If there is no copy of the NFA that is in a final state, then the string will be rejected.

- Another example:

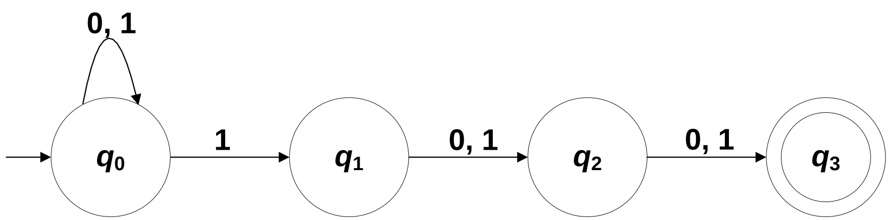
  Given the following NFA:



Determine if the string 110110 will be accepted.

NONDETERMINISM

110110



1. Initially, the NFA will be at state q_o.

   Assume that the first input symbol (1) arrives.

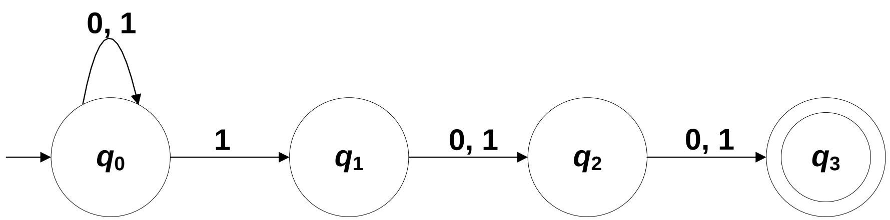
   The NFA has two options, either to stay at q_o or go to q_1. The machine will try both possibilities.

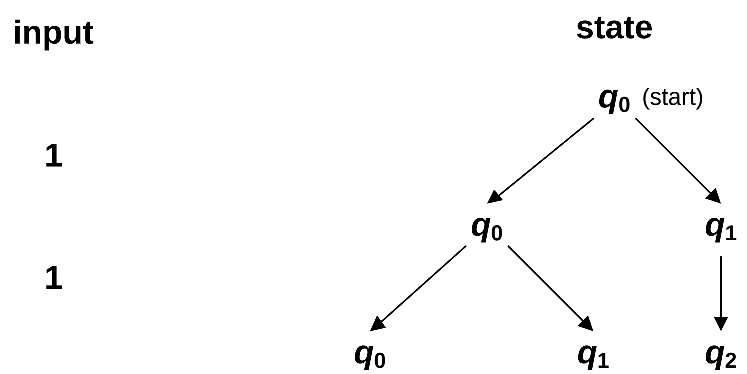
There will now be two copies of the NFA.

One copy is at state q_o while the other is at state q_1.

NONDETERMINISM

110110





2. Assume the second input symbol (1) arrives

   The copy of the NFA which is at state q_0 has two options— stay at q_0 or go to q_1. The machine will try both possibilities.

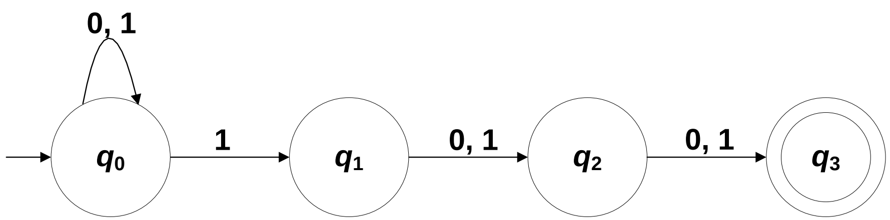
   The copy which is at state q_1 will go to q_2.

There will now be three copies of the NFA.

One is at state q_0, the second one is at state q_1, and the third is at state q_2.

NONDETERMINISM

110110

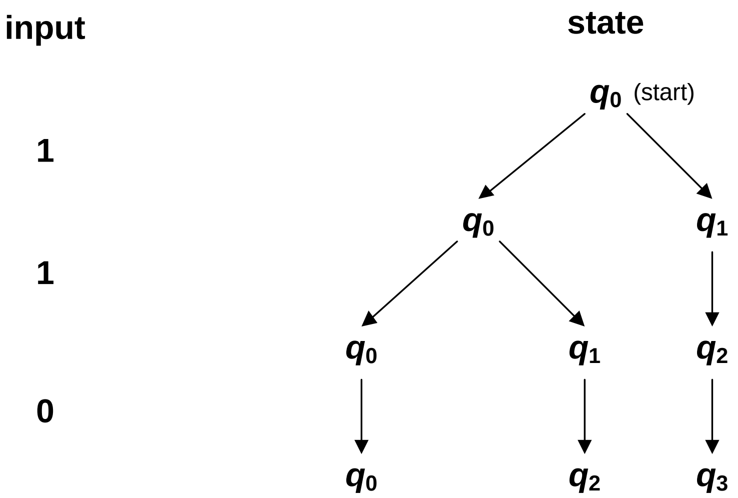


3. Assume the third input symbol (0) arrives.

   The copy of the NFA which is at state q_0 stays at q_0.

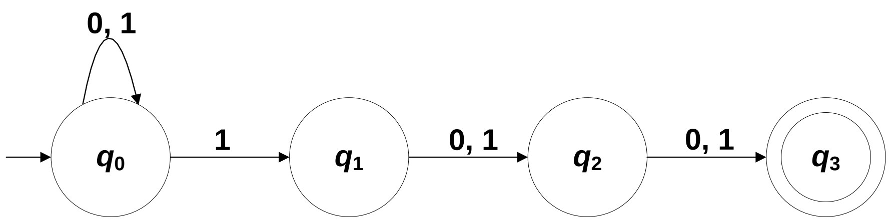
   The copy which is at state q_1 goes to q_2.

   And the copy which is at q_2 goes to state q_3.



NONDETERMINISM

110110

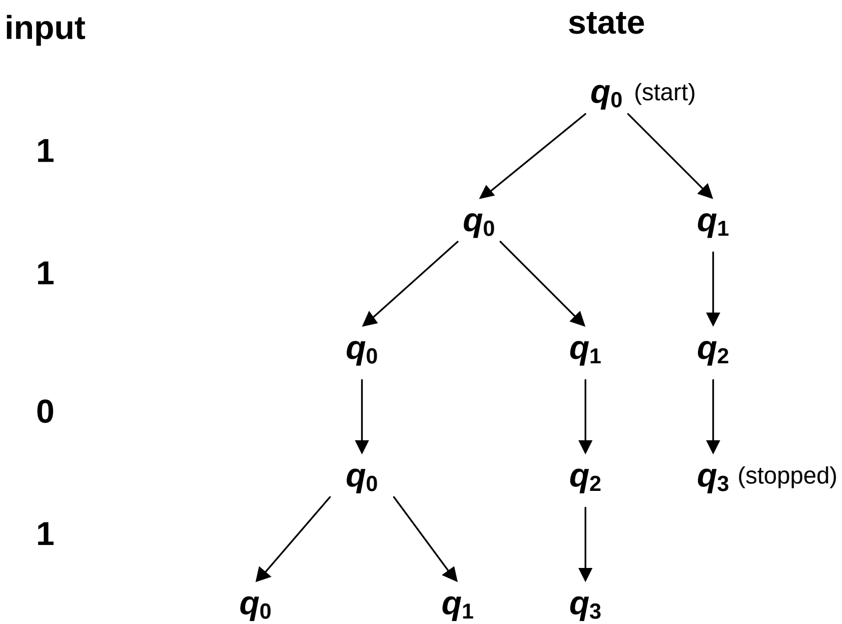


4. Assume the fourth input symbol (1) arrives.

   The copy of the NFA which is at state q_0 has two options. Stay at q_0 or go to q_1. The machine will try both possibilities.

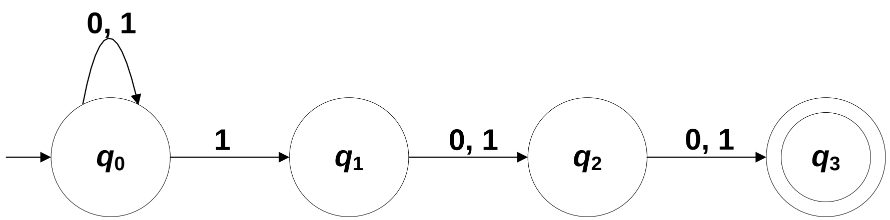
   The copy which is at state q_2 will go to q_3.

   The copy which is at state q_3 stops computing because it has nowhere to go.



NONDETERMINISM

110110

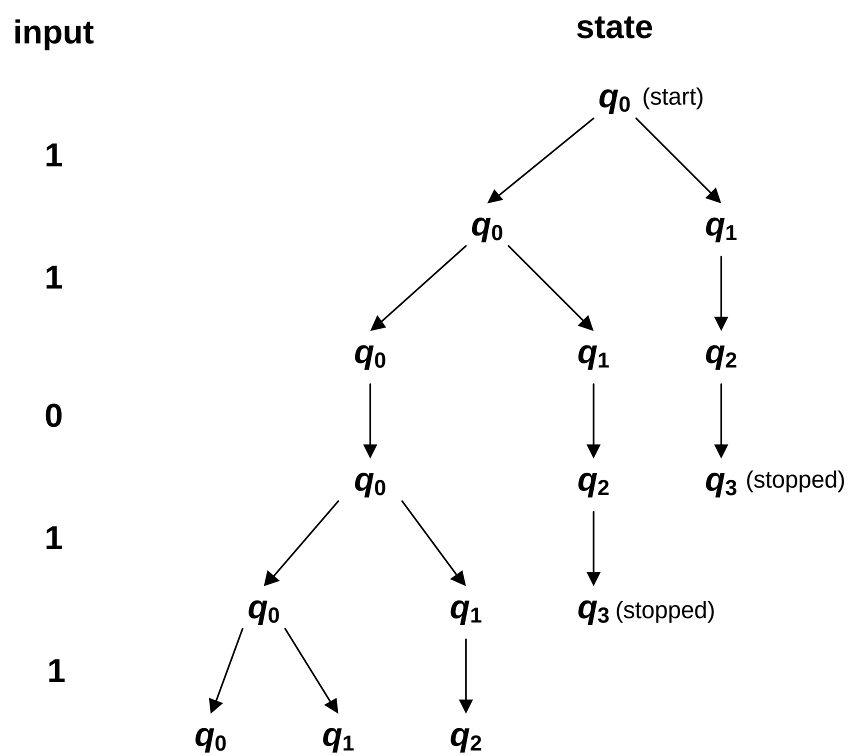


5. Assume the fifth input symbol (1) arrives.

   The copy of the NFA which is at state q_0 has two options—stay at q_0 or go to q_1. The machine will try both possibilities.

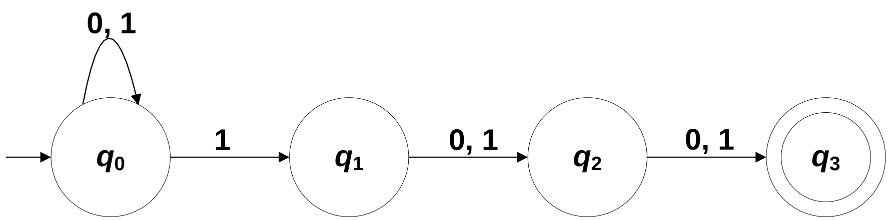
   The copy which is at state q_1 will go to q_2.

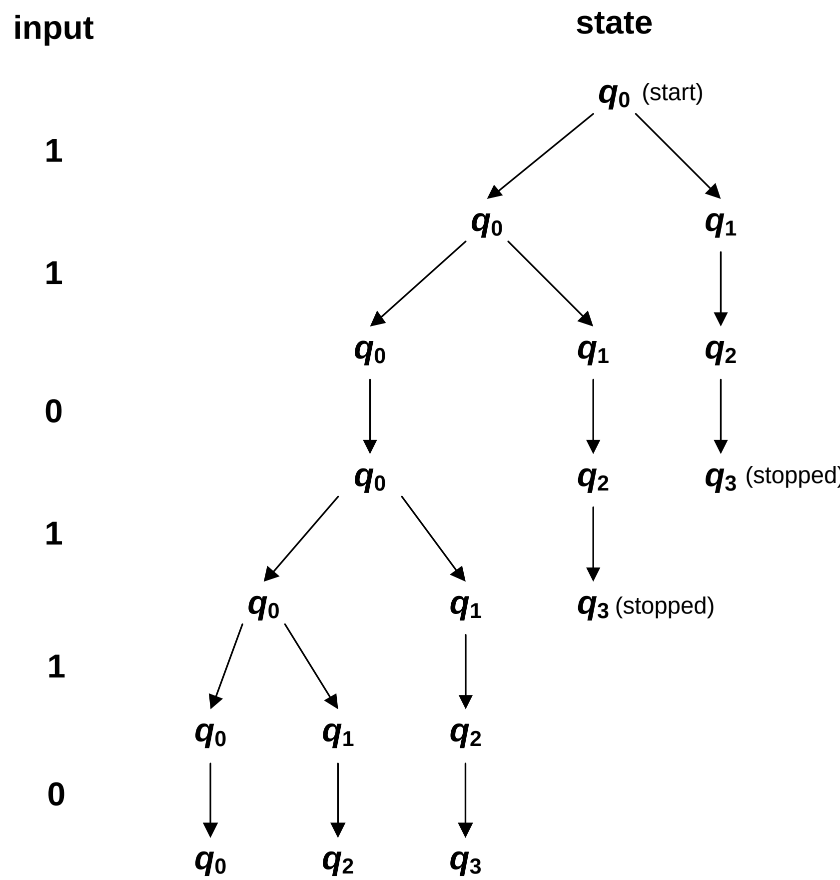
   The copy which is at q_3 dies.



NONDETERMINISM

110110





6. Assume the sixth and final input symbol (0) arrives.

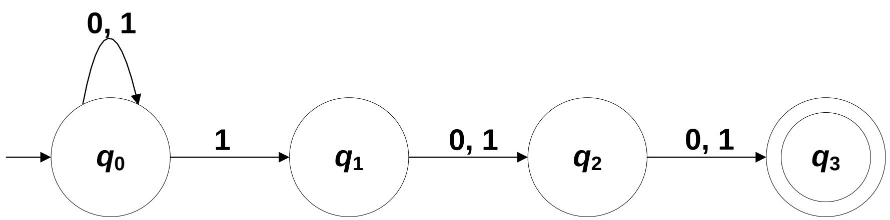
   The copy of the NFA which is at state q_0 stays at q_0.

   The copy at state q_1 goes to q_2.

   The copy at state q_2 goes to q_3.

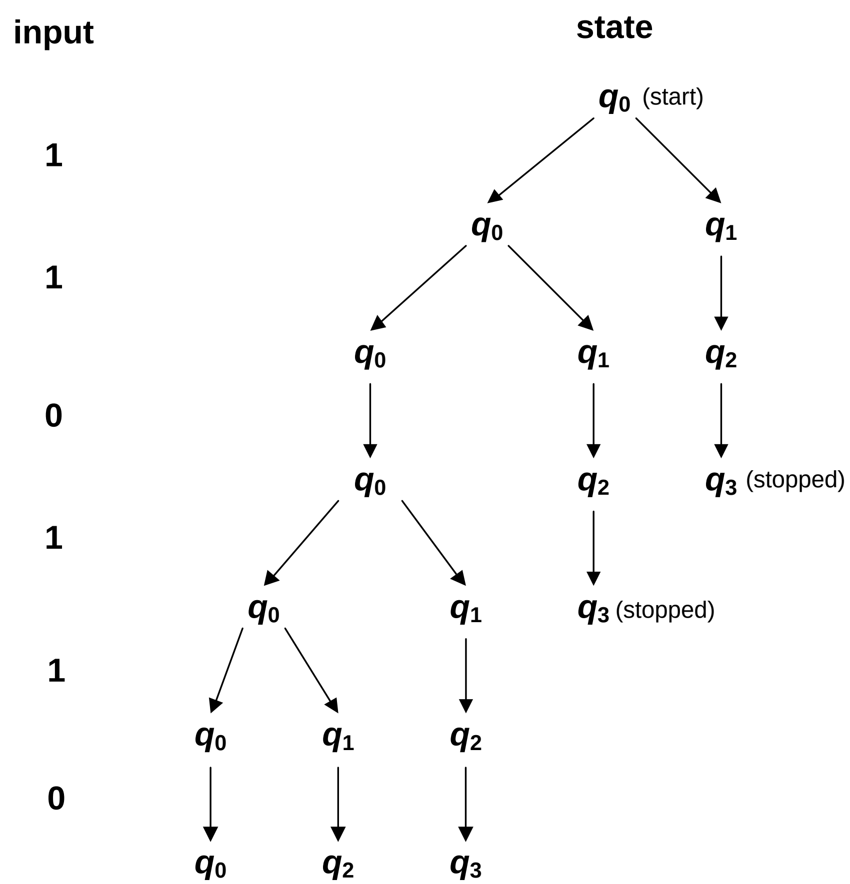
NONDETERMINISM

110110



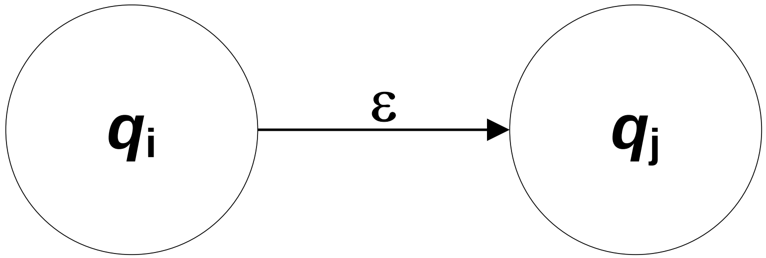
The NFA is now at three states, q_0, q_2, and q_3.

Since one of these states is a final state (q_3), then the NFA accepts the input string 110110.



ε - TRANSITIONS

- One other difference between an NFA and a DFA is that an NFA may have ε-transitions. An ε-transition is shown below:

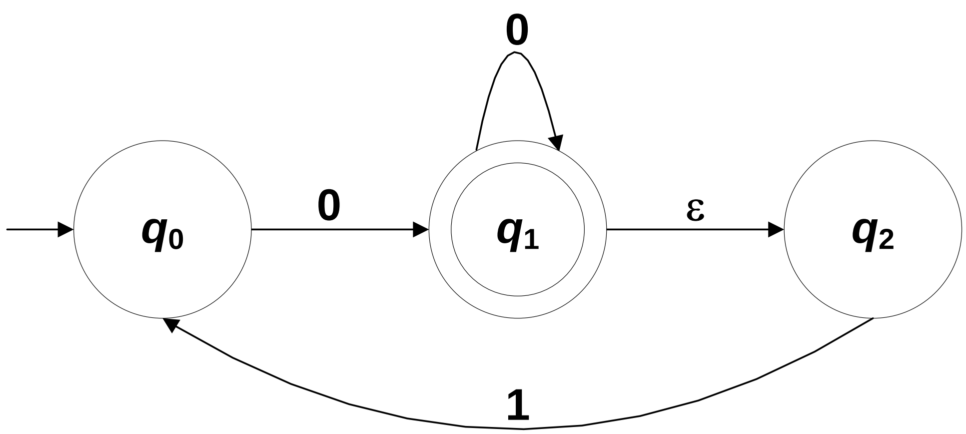


Whenever there is an ε-transition from state q_i to state q_j, this means that once an NFA goes to state q_i, it has the option of staying at q_i or it can go right away to state q_j even without any additional input arriving.

ε - TRANSITIONS

- Example:

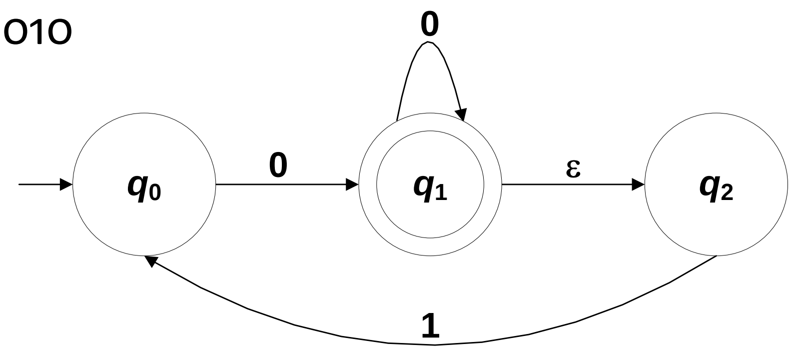
  Given the following NFA:

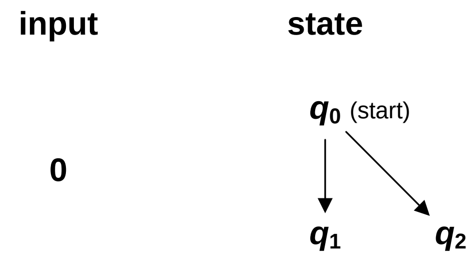


Determine if the string 010010 will be accepted

ε - TRANSITIONS

010010





1. Initially, the NFA will be at state q_0.

   Assume the first input symbol (0) arrives.

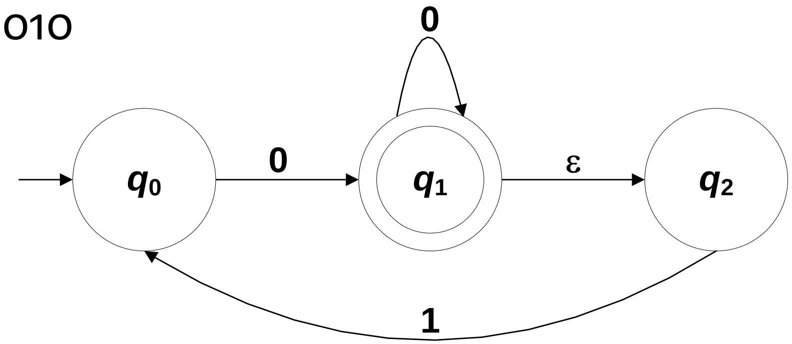
   The NFA goes to q_1.

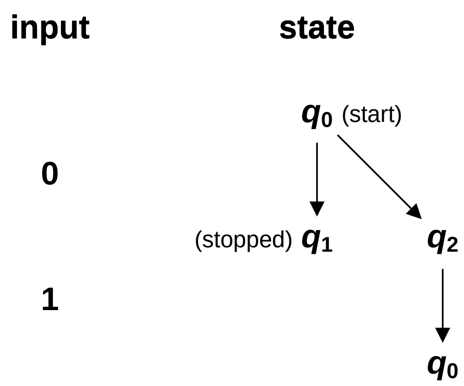
   At q_1, it has the option of immediately going to q_2 without any new input symbol arriving (because of the ε-transition from q_1 to q_2).

   The NFA will consider both and will be in two states, q_1 and q_2.

ε - TRANSITIONS

010010





2. Assume the second input symbol (1) arrives.

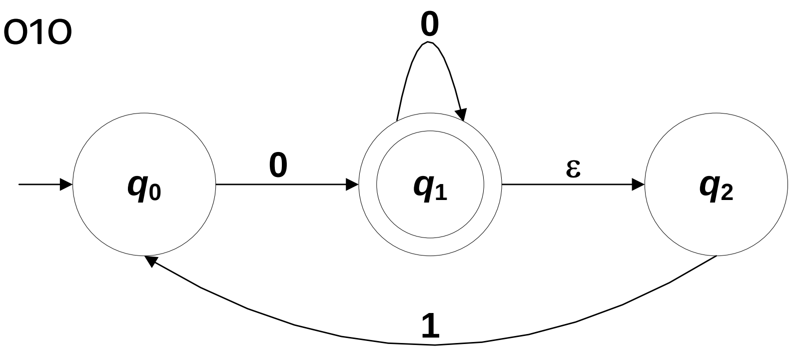
   The copy of the NFA which is at state q_1 stops computing since it has nowhere to go.

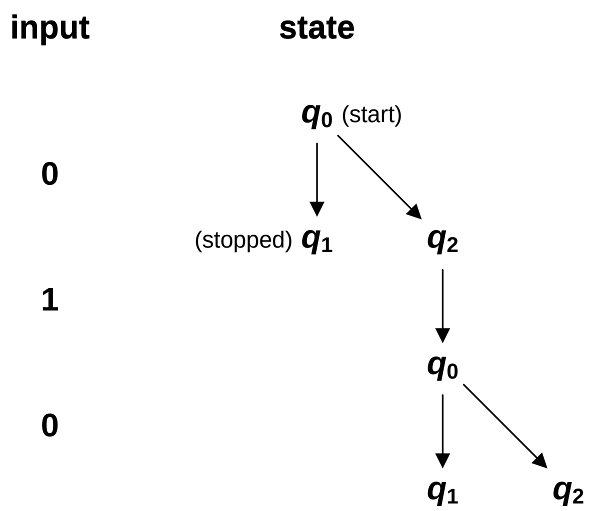
   The copy at q_2 will go to q_0.

   The NFA will only be in one state which is q_0.

ε - TRANSITIONS

010010





3. Assume the third input symbol (0) arrives.

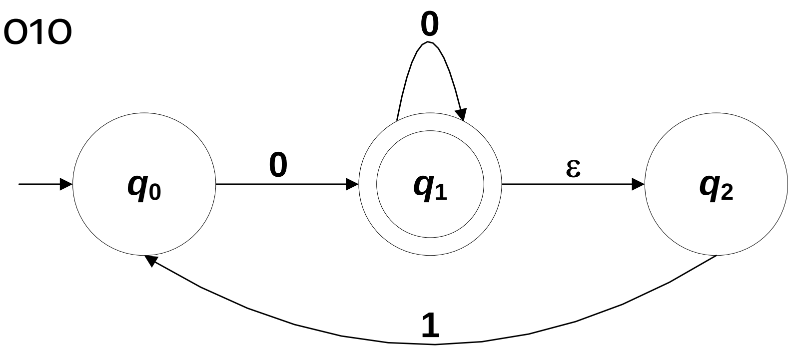
   The NFA goes to q_1.

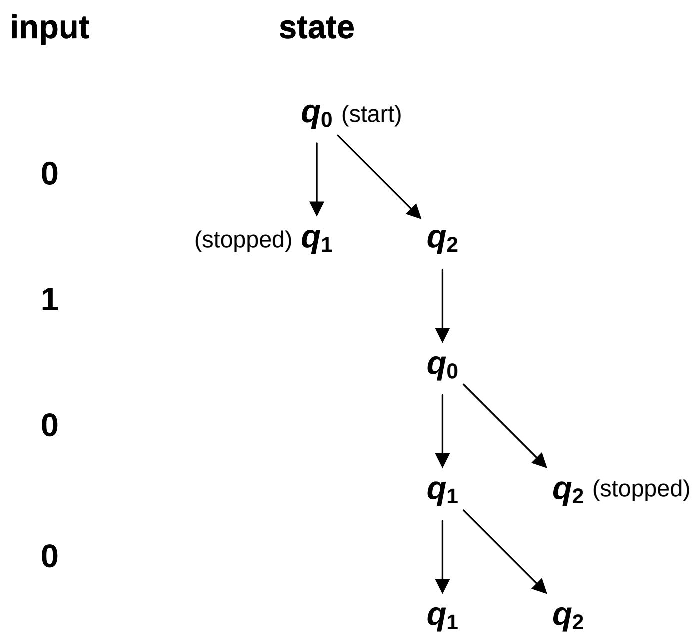
   Once at q_1, it has the option of immediately going to q_2 without any new input symbol arriving.

   The NFA will consider both and will be in two states, q_1 and q_2.

ε - TRANSITIONS

010010





4. Assume the fourth input symbol (0) arrives.

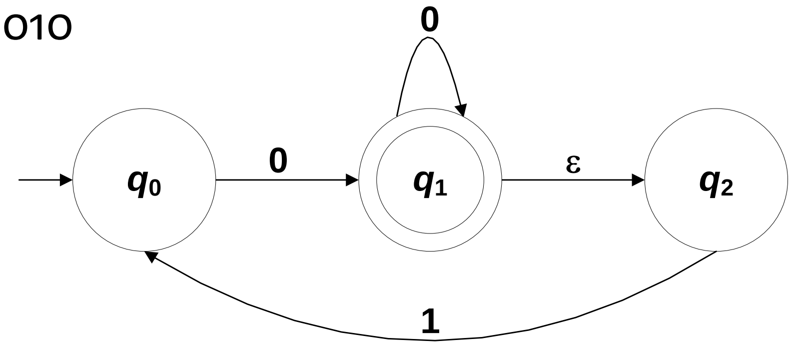
   The copy of the NFA which is at state q_1 stays at q_1.

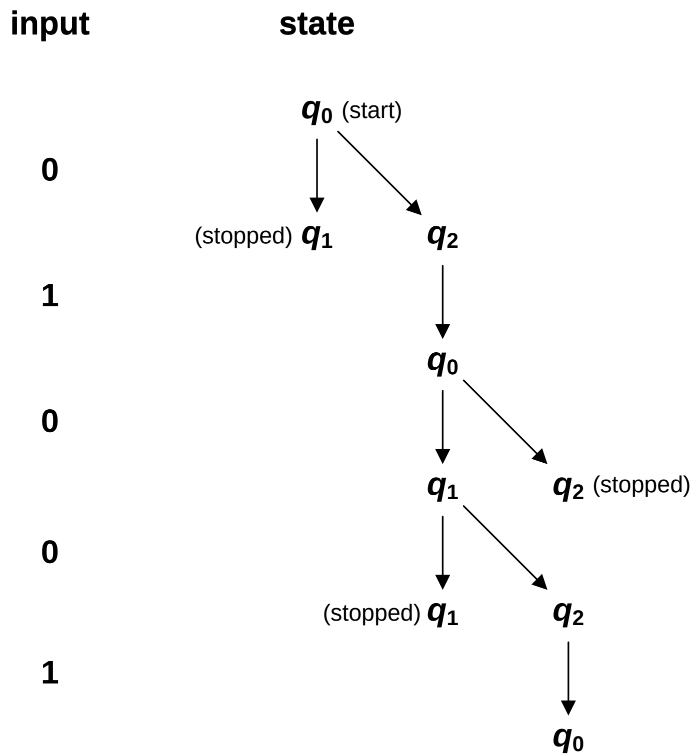
   Since there is an ε-transition from q_1 to q_2, it will also immediately go to q_2.

   The other copy of the NFA (the one at state q_2) stops because it has nowhere to go.

ε - TRANSITIONS

010010





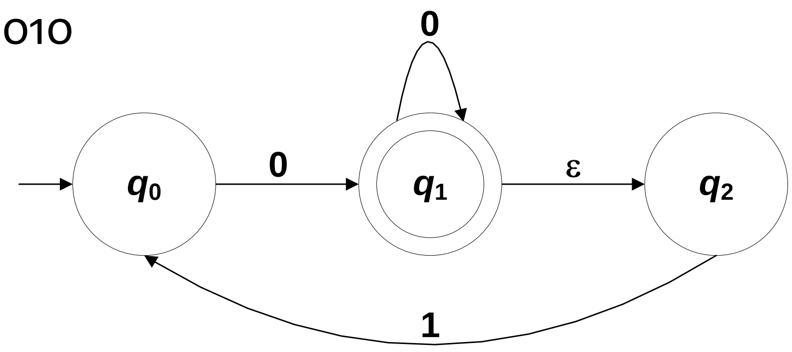
5. Assume the fifth input symbol (1) arrives.

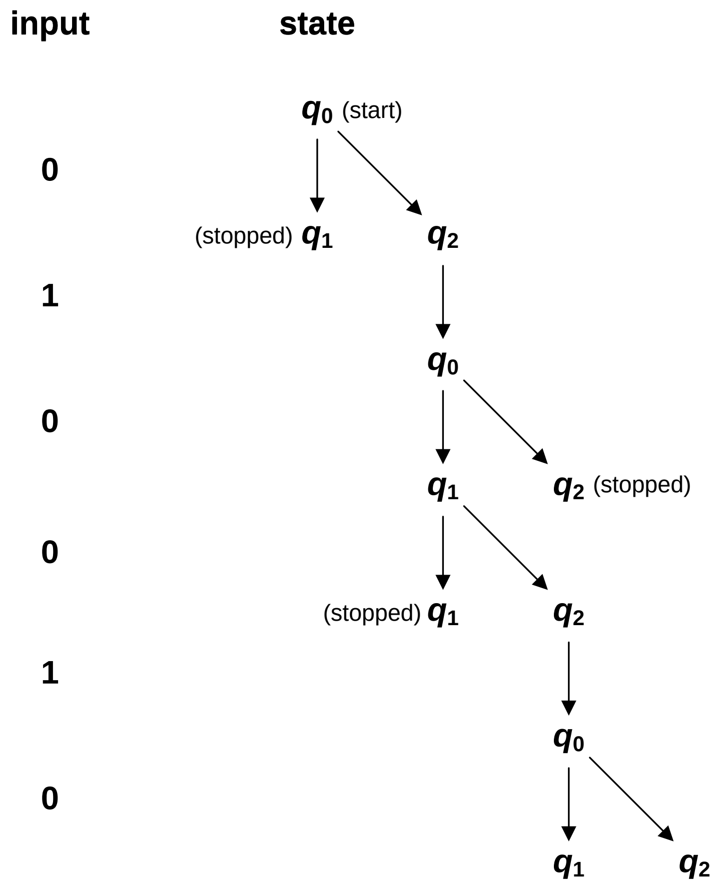
   The copy of the NFA which is at state q_1 stops since it has nowhere to go.

   The remaining copy of the NFA (the one at state q_2) goes to q_0.

ε - TRANSITIONS

010010





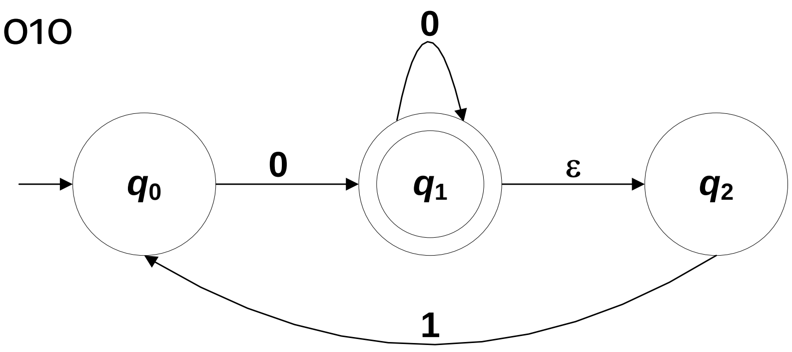
6. Assume the sixth and last input symbol (0) arrives.

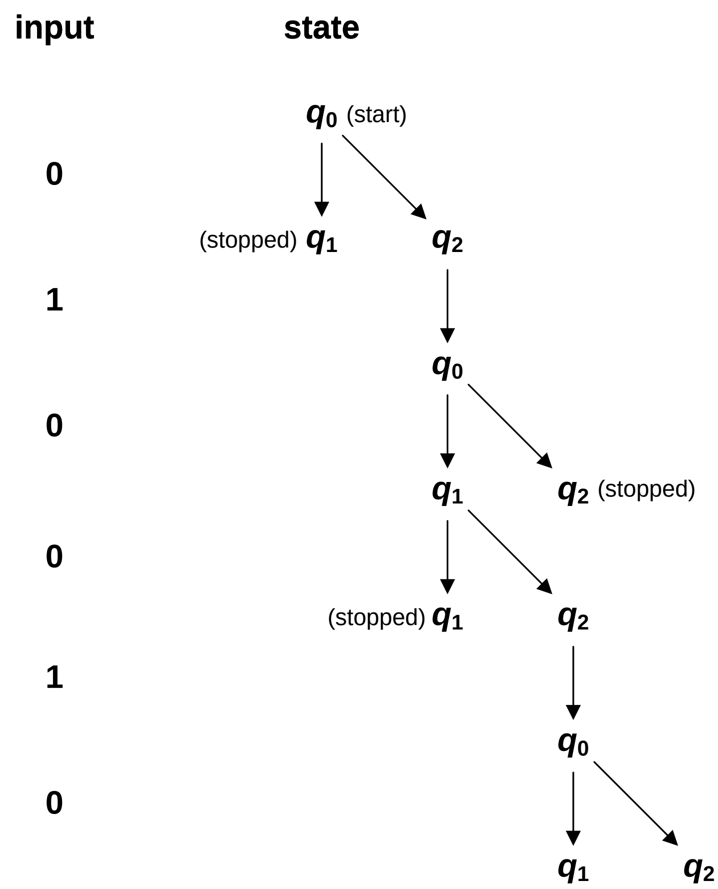
   The NFA goes to q_1.

   At q_1, it has the option of immediately going to q_2 without any new input symbol arriving (because of the ε-transition from q_1 to q_2).

ε - TRANSITIONS

010010





The NFA will then be at state q_1 and state q_2.

Since one of these states is a final state (q_1), the string 010010 is accepted.

FORMAL DEFINITION OF NFAs

- The formal definition of a NFA is similar to that of a DFA.
- The main difference lies in the transition function.
- The transition function of an NFA must consider the possibility of state transitions even if there is no input symbol (ε-transitions).
- The transition function must also consider the possibility that the NFA may go to several states given the same input symbol.

DIFFERENCES BETWEEN NFAs AND DFAs

- To summarize the differences between an NFA and a DFA:

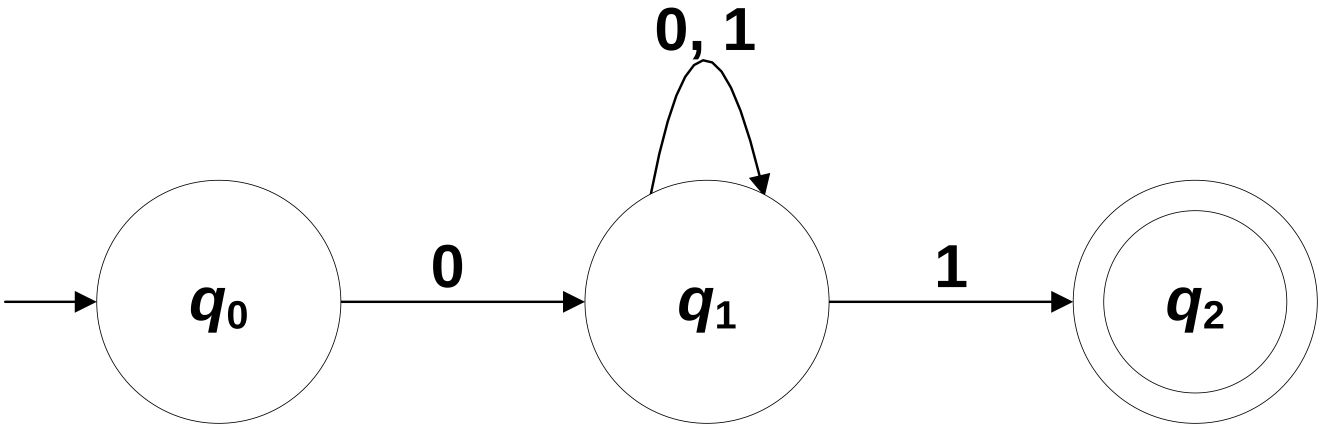
1. NFAs can move to more than one state from any given current state. They can therefore be in several states at the same time.

2. In a given state, NFAs may not necessarily have a transition edge for each symbol in the alphabet.

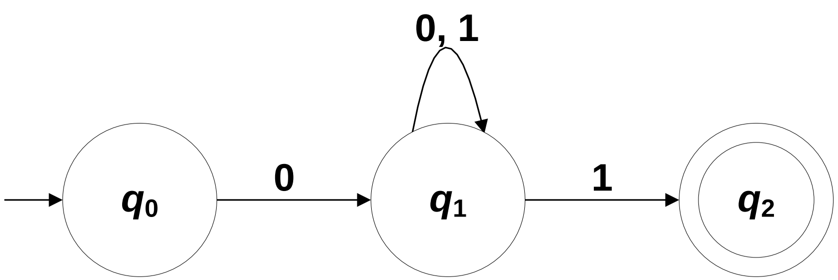
3. NFAs can move to a state even without any input symbol arriving (ε-transitions).

FORMAL DEFINITION OF NFAs

- Example:



FORMAL DEFINITION OF NFAs



- The formal definition of this NFA is a 5-tuple $N_1 = \{Q, \Sigma, \delta, q_0, F\}$ where:

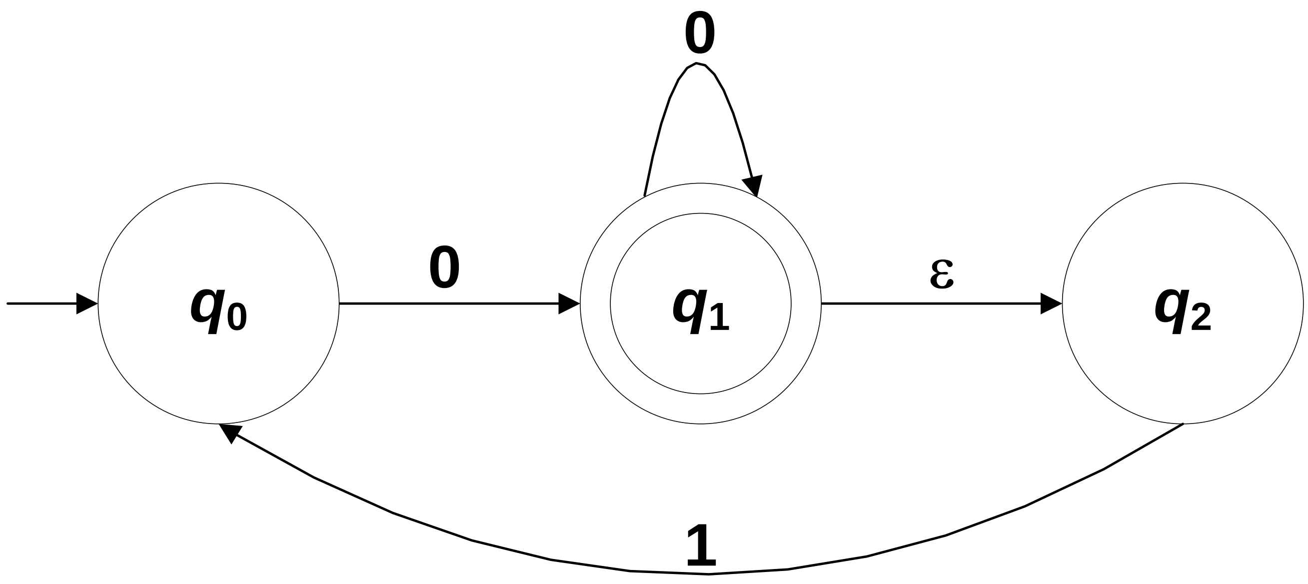
1. $Q = \{q_0, q_1, q_2\}$
2. $\Sigma = \{0, 1\}$
3. δ :

	0	1	ε
q_0	q_1	--	--
q_1	q_1	q_1, q_2	--
q_2	--	--	--

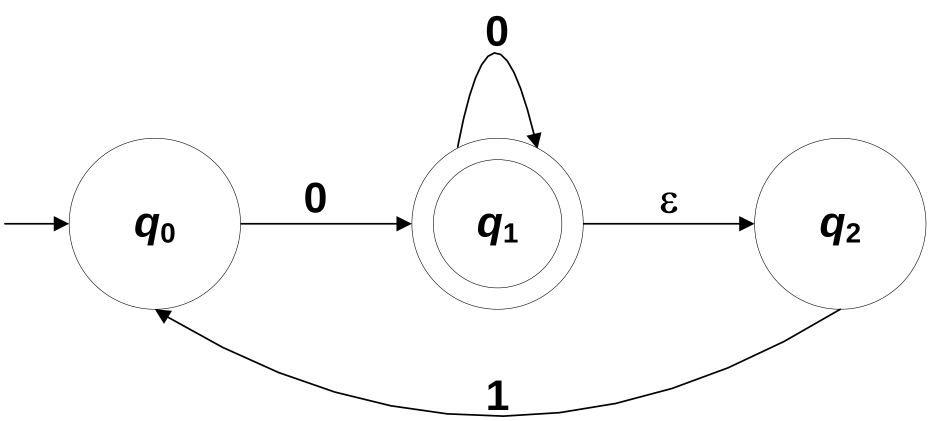
4. Start State = q_0
5. $F = \{q_2\}$

FORMAL DEFINITION OF NFAs

- Another example:



FORMAL DEFINITION OF NFAs



- The formal definition of this NFA is a 5-tuple $N_2 = \{Q, \Sigma, \delta, q_0, F\}$ where:

1. $Q = \{q_0, q_1, q_2\}$
2. $\Sigma = \{0, 1\}$
3. δ:

	0	1	ε
q_0	q_1	--	--
q_1	q_1	--	q_2
q_2	--	q_0	--

4. Start State = q_0
5. $F = \{q_1\}$