TDDD14/TDDD85
Slides for Lecture 3, 2017

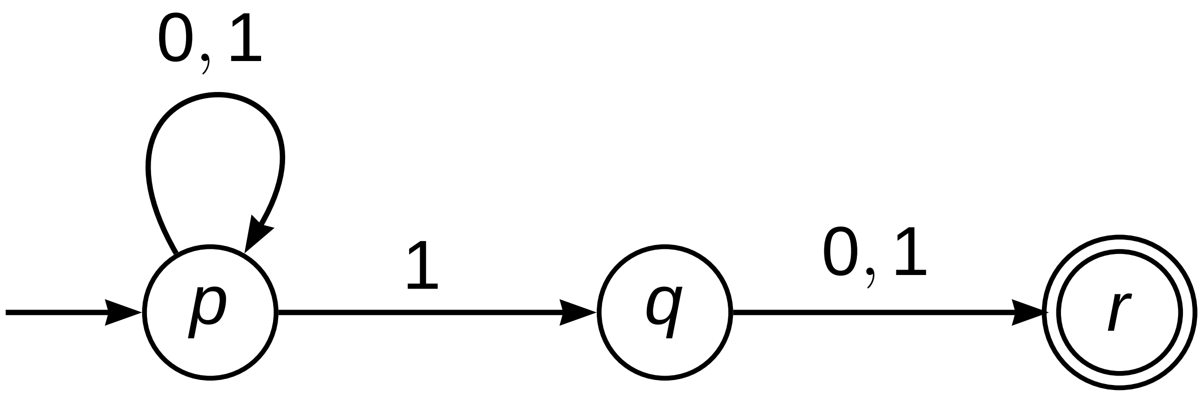
Slides originally for TDDD65 by Gustav Nordh

Some differences to Kozen:
- Kozen allows a set of start states. It may be convenient sometimes, but does not add anything. It is easier to use a single start state and ε-transitions.
- Kozen uses Δ instead of δ for the transition function of an NFA.
- Kozen defines acceptance for NFA by an additional recursive function $\widehat{\Delta}$, as for DFA. However, his definition works only for ordinary NFA, not ε-NFA.
- Kozen defines the subset construction method only for ordinary NFAs, not for ε-NFAs.

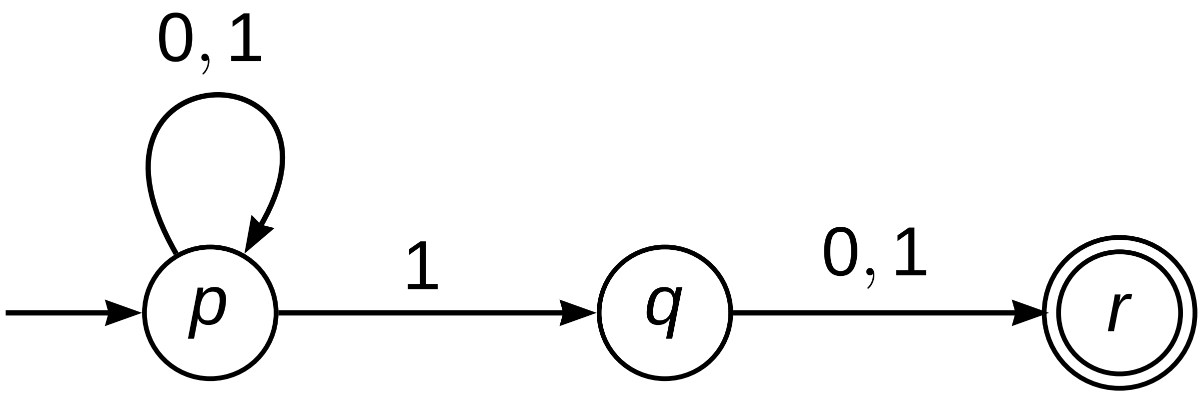
Nondeterminism

- The machines we have seen so far have been deterministic. The next state follows uniquely from the current state and the input symbol
- In a nondeterministic machine several possible next states may follow from the current state and input symbol. These possibilities can be thought of as being explored in parallel
- Understanding the power of nondeterminism is a central topic in the theory of computation (and this course)

Example of a nondeterminstic finite automaton (NFA)

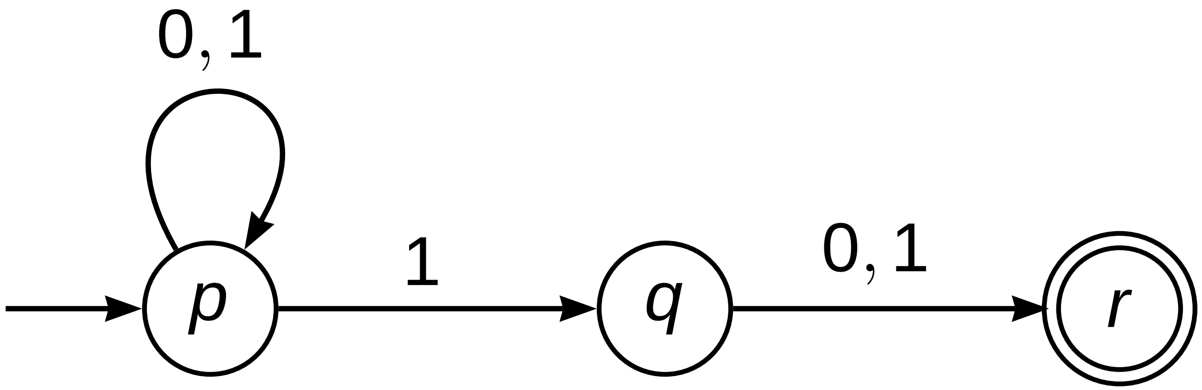


Example of a nondeterminstic finite automaton (NFA)

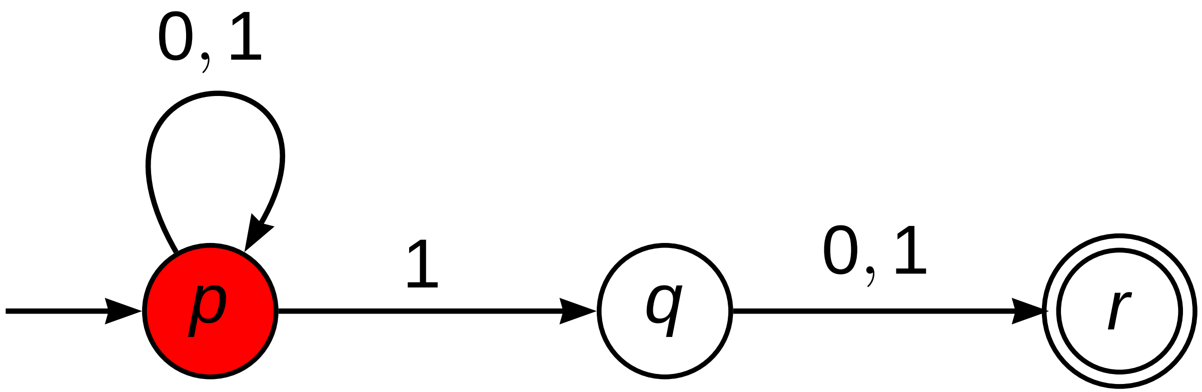


- An NFA accepts a string *s* if, after reading the last symbol of *s*, at least one of its active states is an accept state
- An NFA rejects a string *s* if, after reading the last symbol of *s*, none if its active states is an accept state

Example of a nondeterminstic finite automaton (NFA)

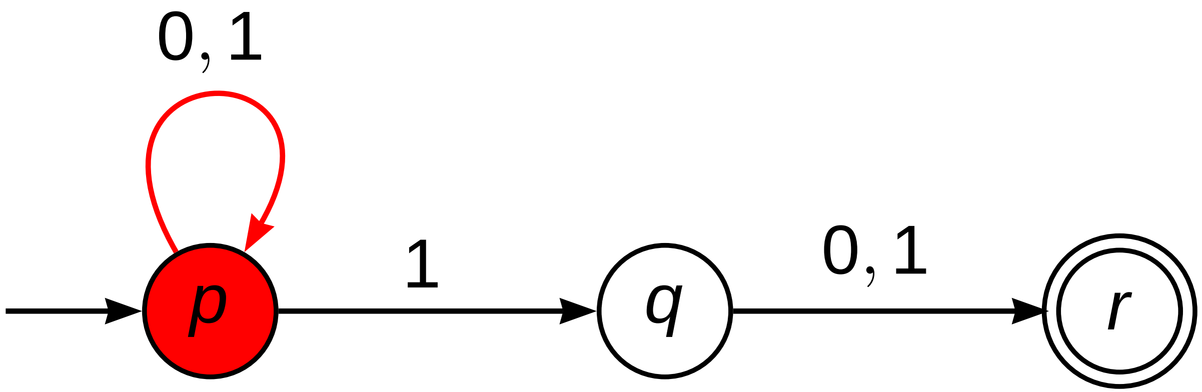


Example of a nondeterminstic finite automaton (NFA)



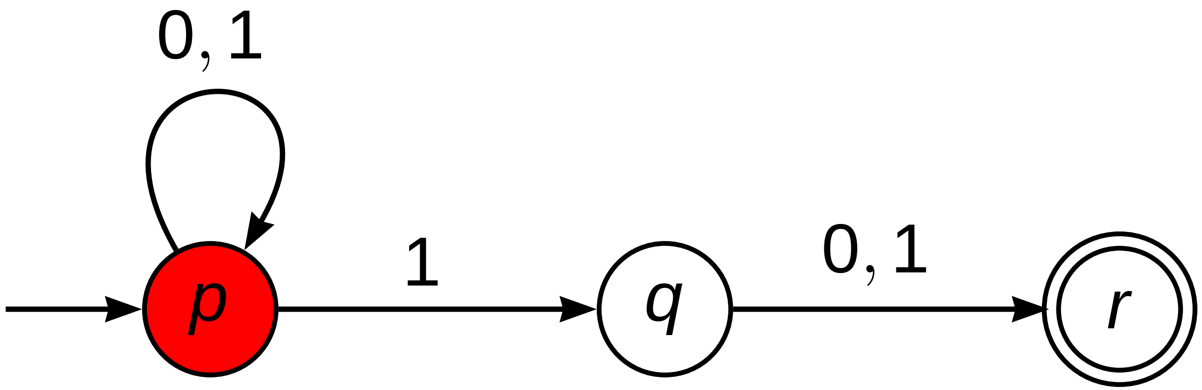
0101

Example of a nondeterminstic finite automaton (NFA)



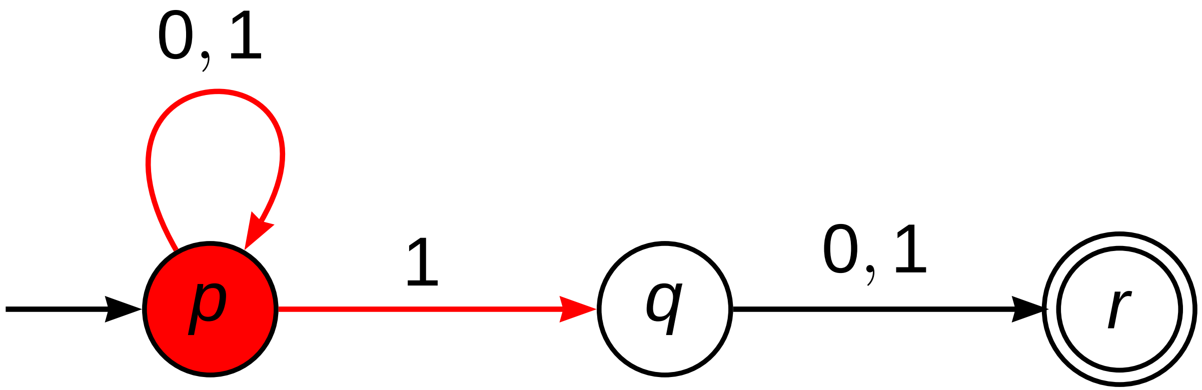
0101

Example of a nondeterminstic finite automaton (NFA)



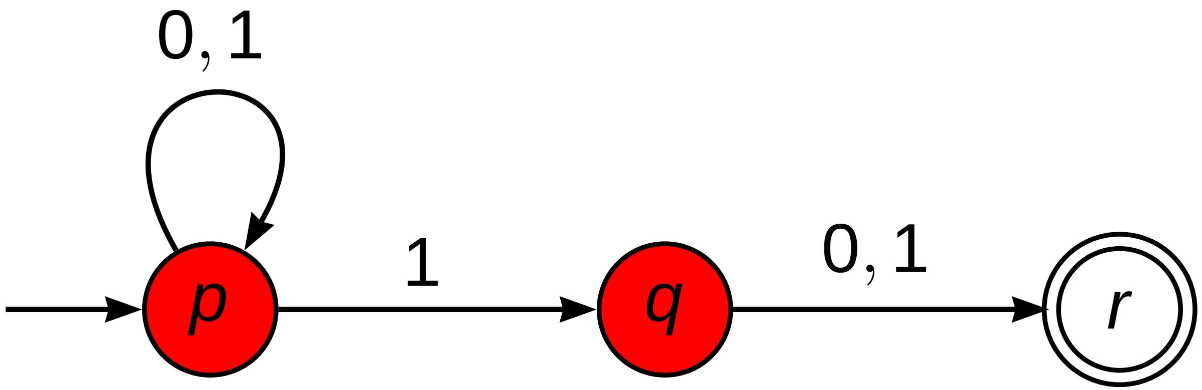
0101

Example of a nondeterminstic finite automaton (NFA)



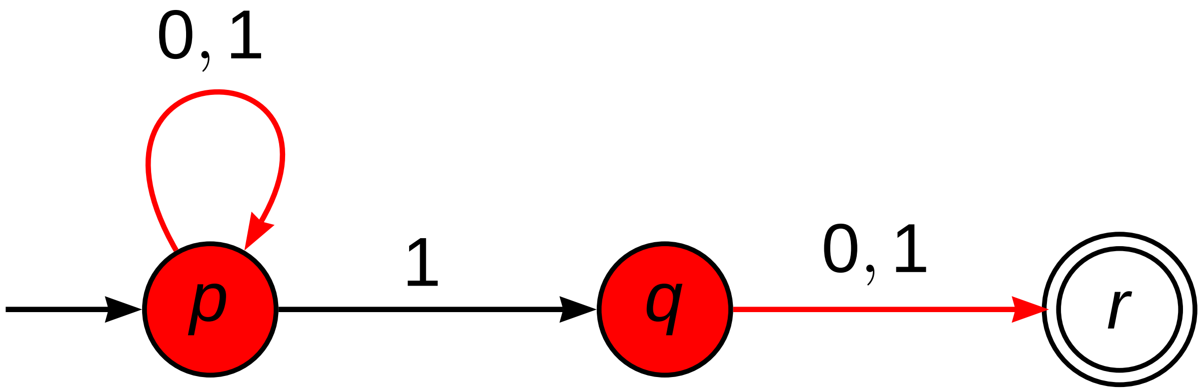
<u>0</u>101

Example of a nondeterminstic finite automaton (NFA)



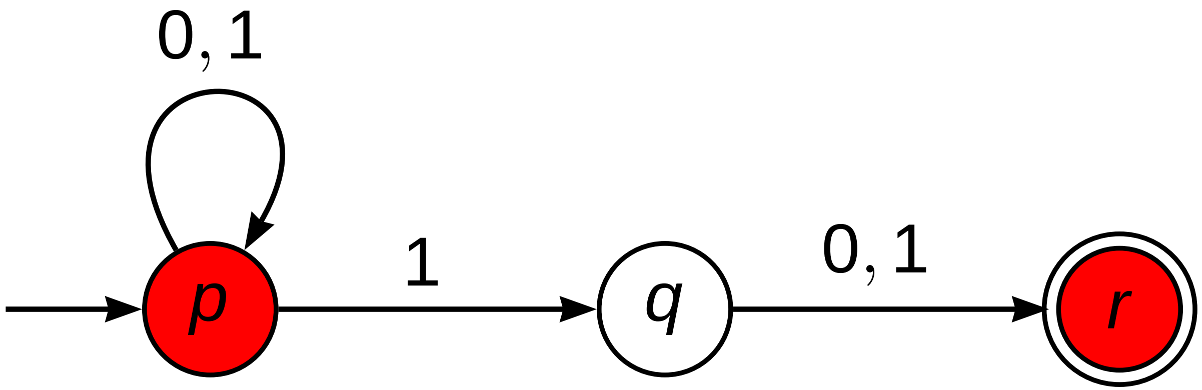
0<u>1</u>01

Example of a nondeterminstic finite automaton (NFA)



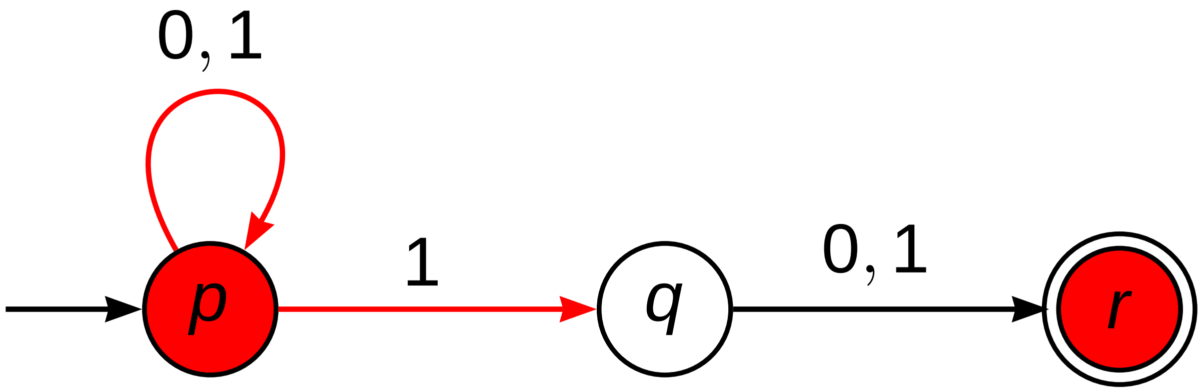
0<u>1</u>01

Example of a nondeterminstic finite automaton (NFA)



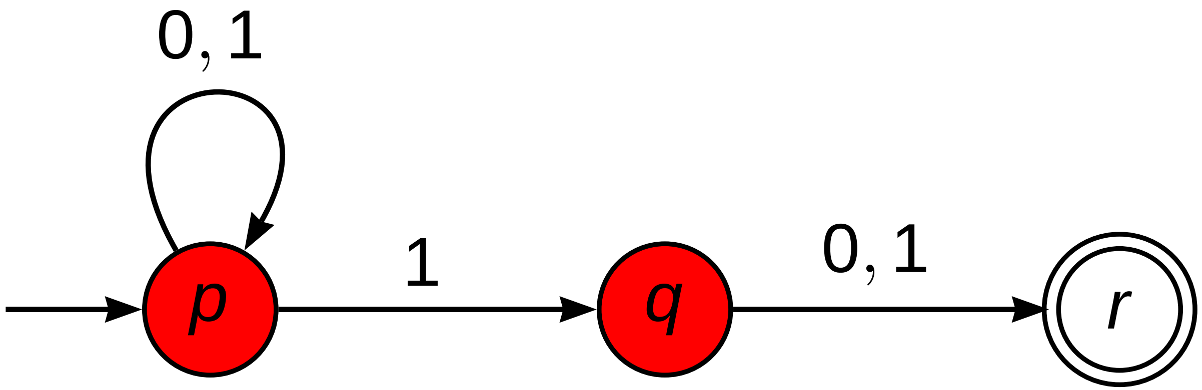
01<u>0</u>1

Example of a nondeterminstic finite automaton (NFA)



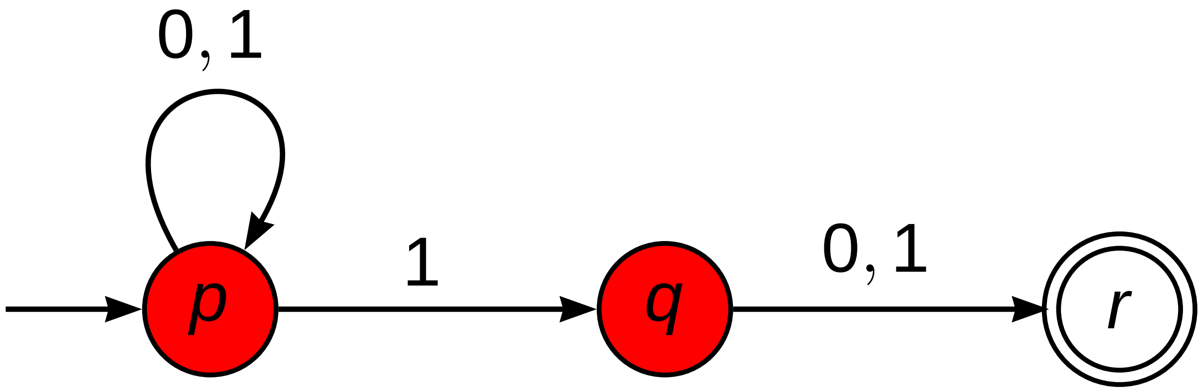
01<u>0</u>1

Example of a nondeterminstic finite automaton (NFA)



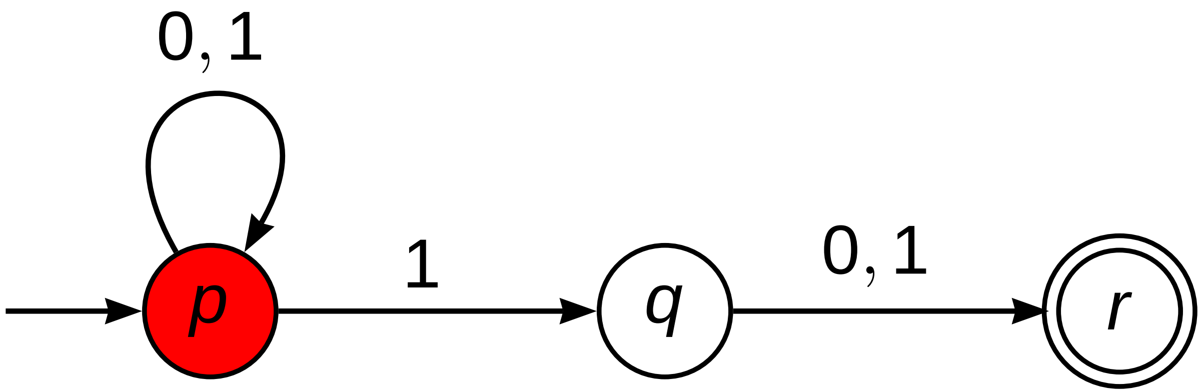
010<u>1</u>

Example of a nondeterminstic finite automaton (NFA)



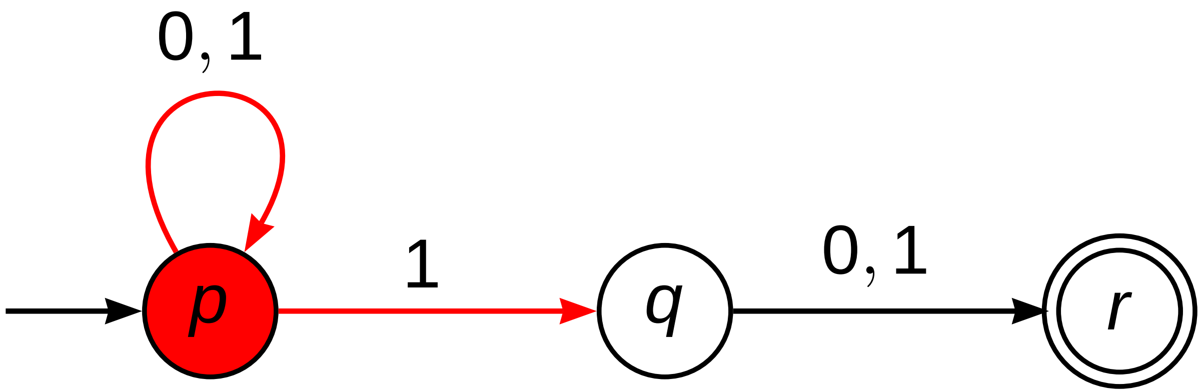
010<u>1</u> REJECT

Example of a nondeterminstic finite automaton (NFA)



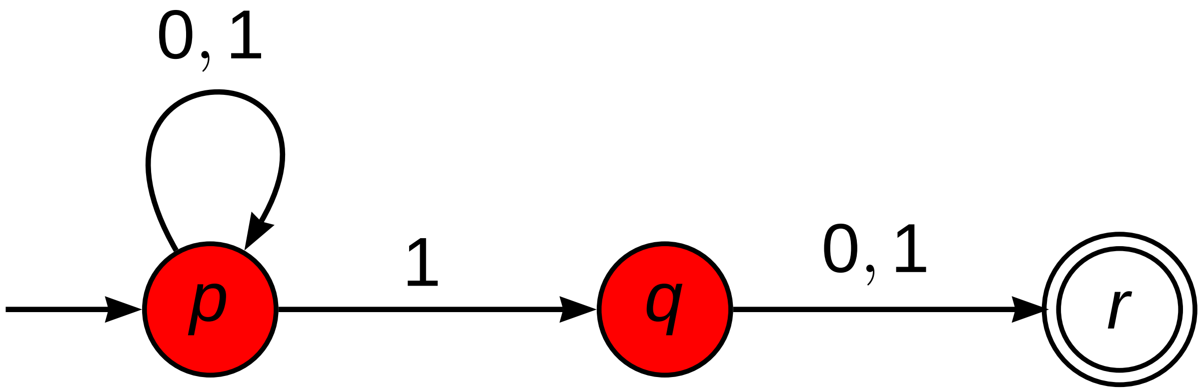
110

Example of a nondeterminstic finite automaton (NFA)



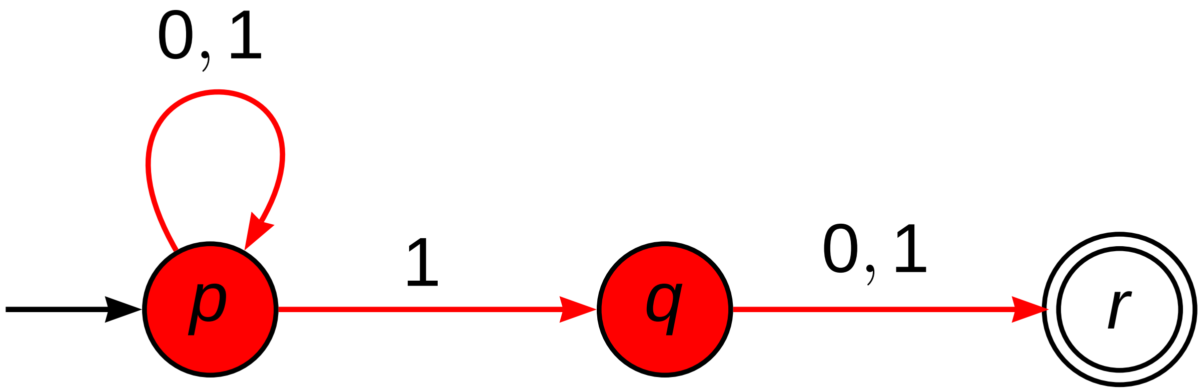
110

Example of a nondeterminstic finite automaton (NFA)



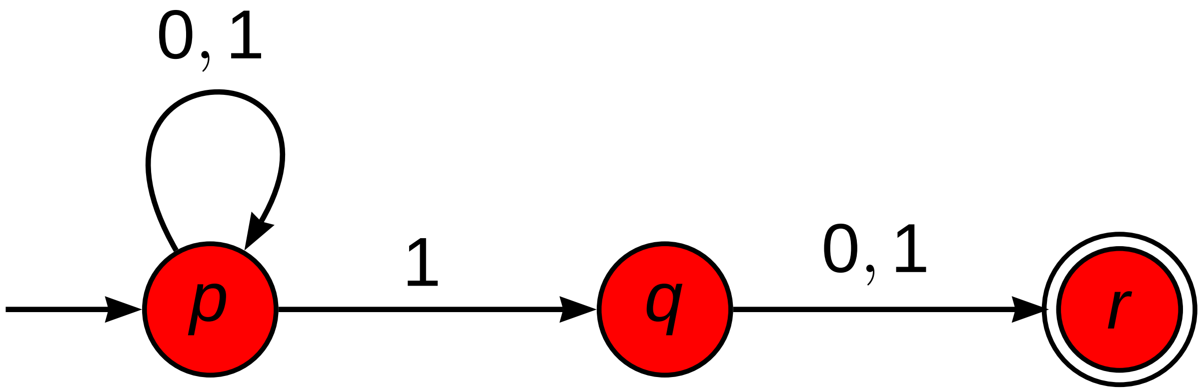
<u>1</u>10

Example of a nondeterminstic finite automaton (NFA)



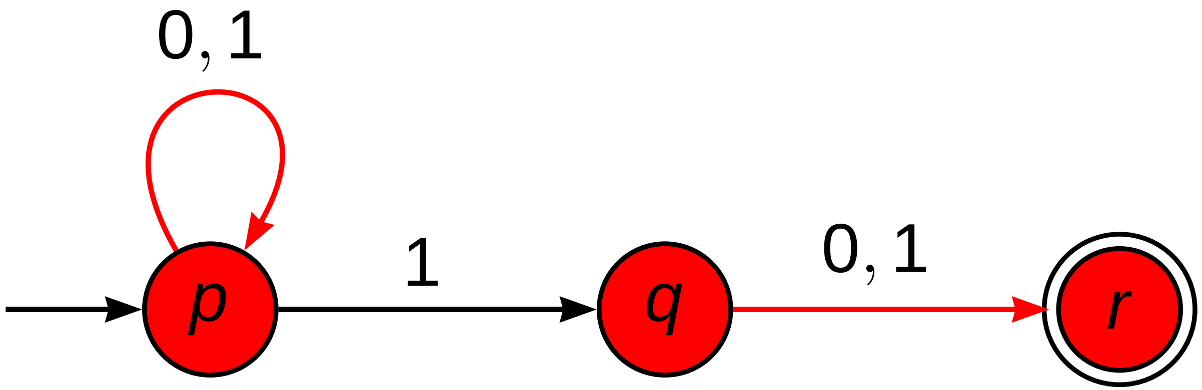
<u>1</u>10

Example of a nondeterminstic finite automaton (NFA)



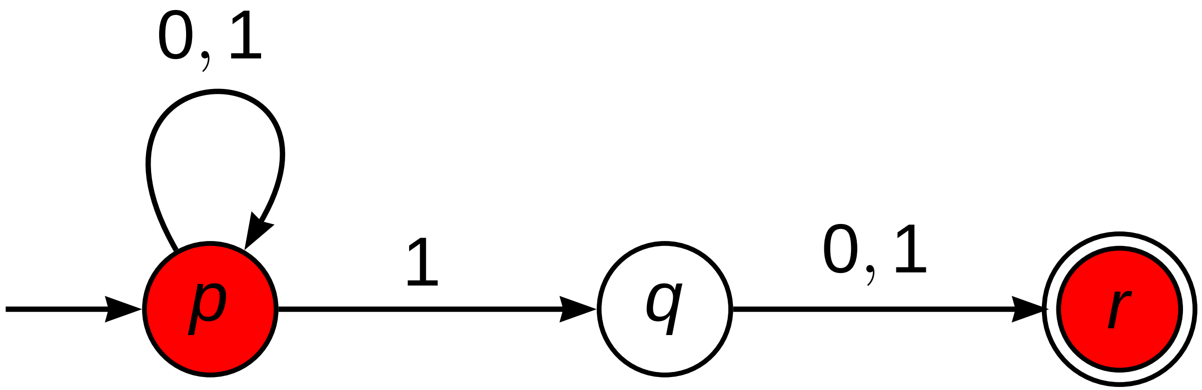
1<u>1</u>0

Example of a nondeterminstic finite automaton (NFA)



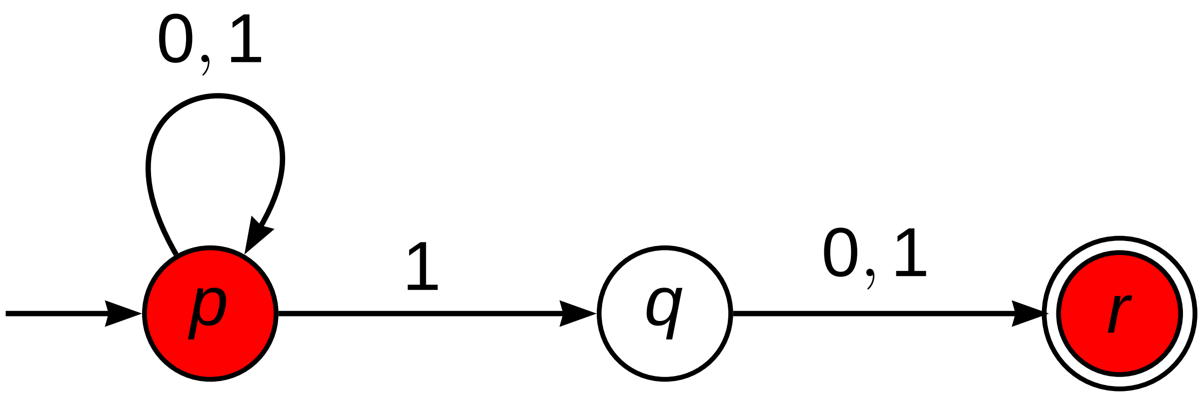
1<u>1</u>0

Example of a nondeterminstic finite automaton (NFA)



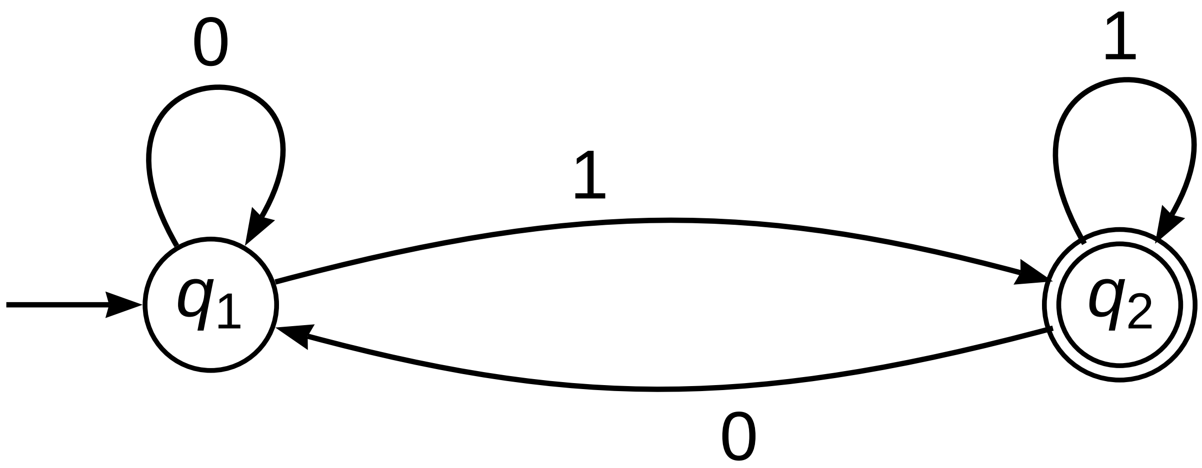
11<u>0</u>

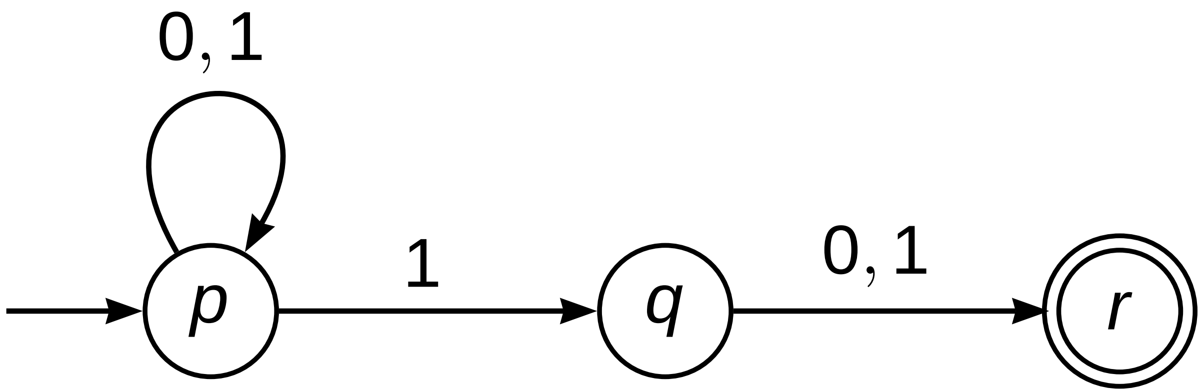
Example of a nondeterminstic finite automaton (NFA)



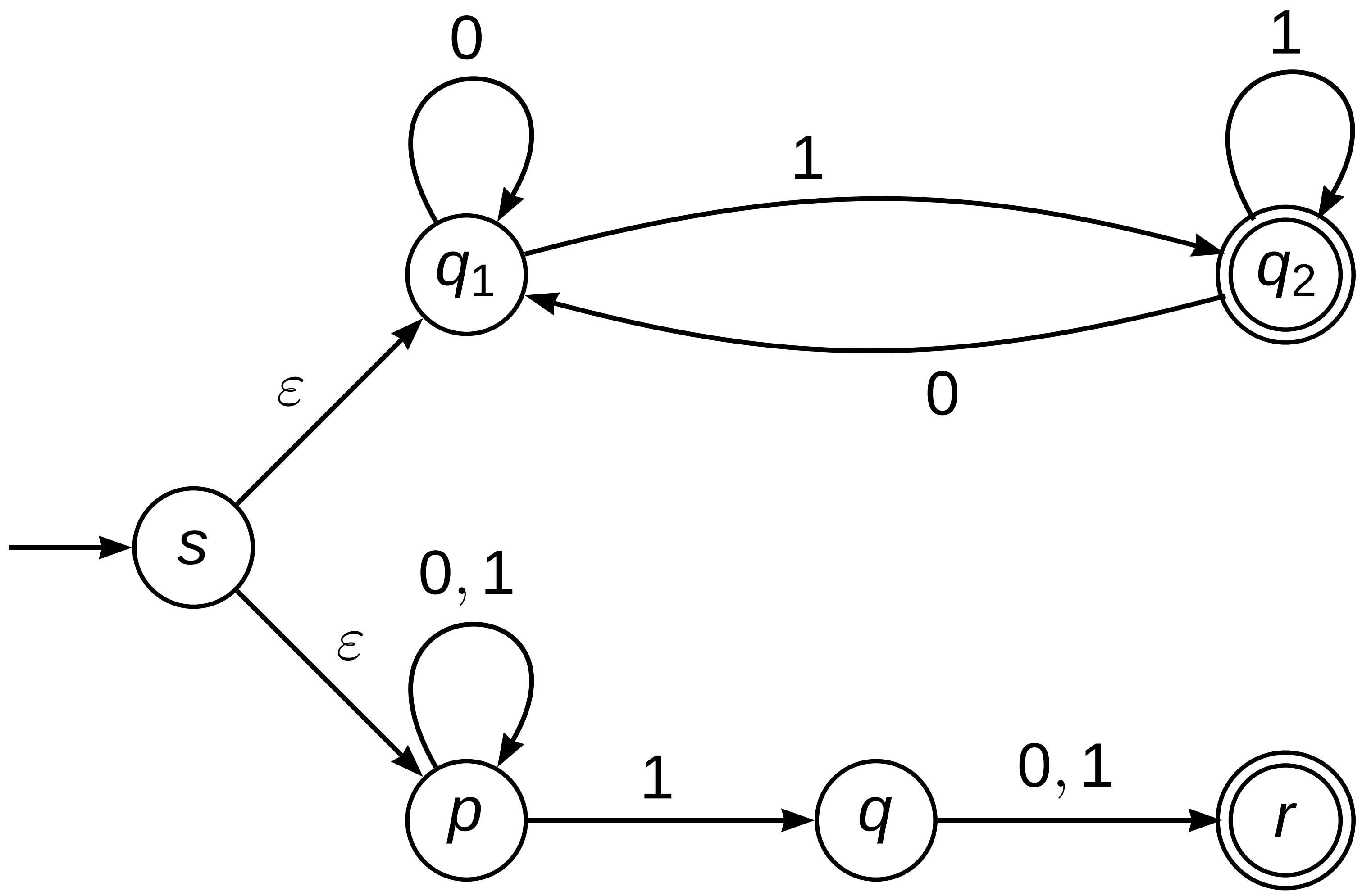
11<u>0</u> ACCEPT

Example of a NFA with ε transitions

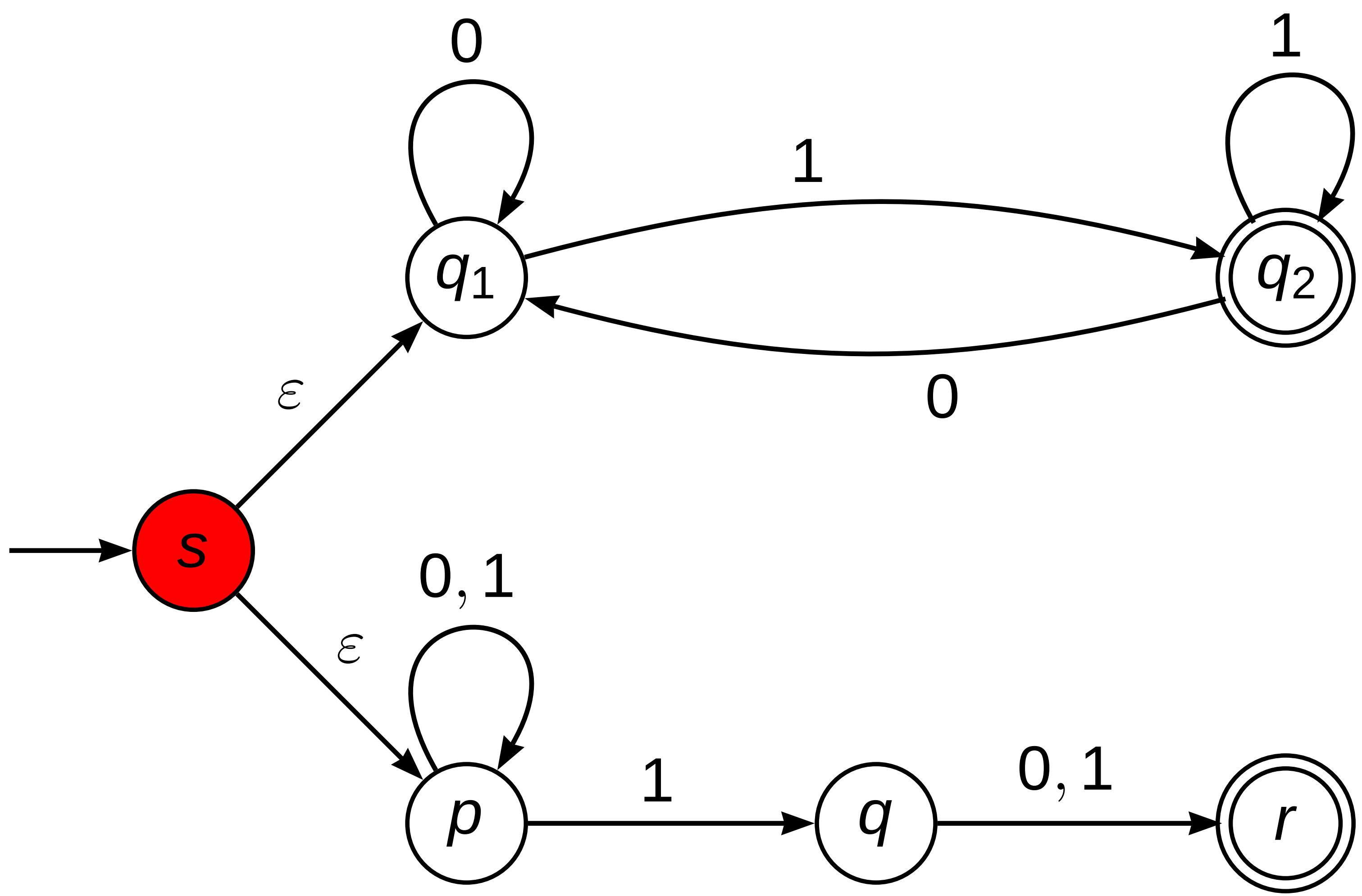




Example of a NFA with ε transitions

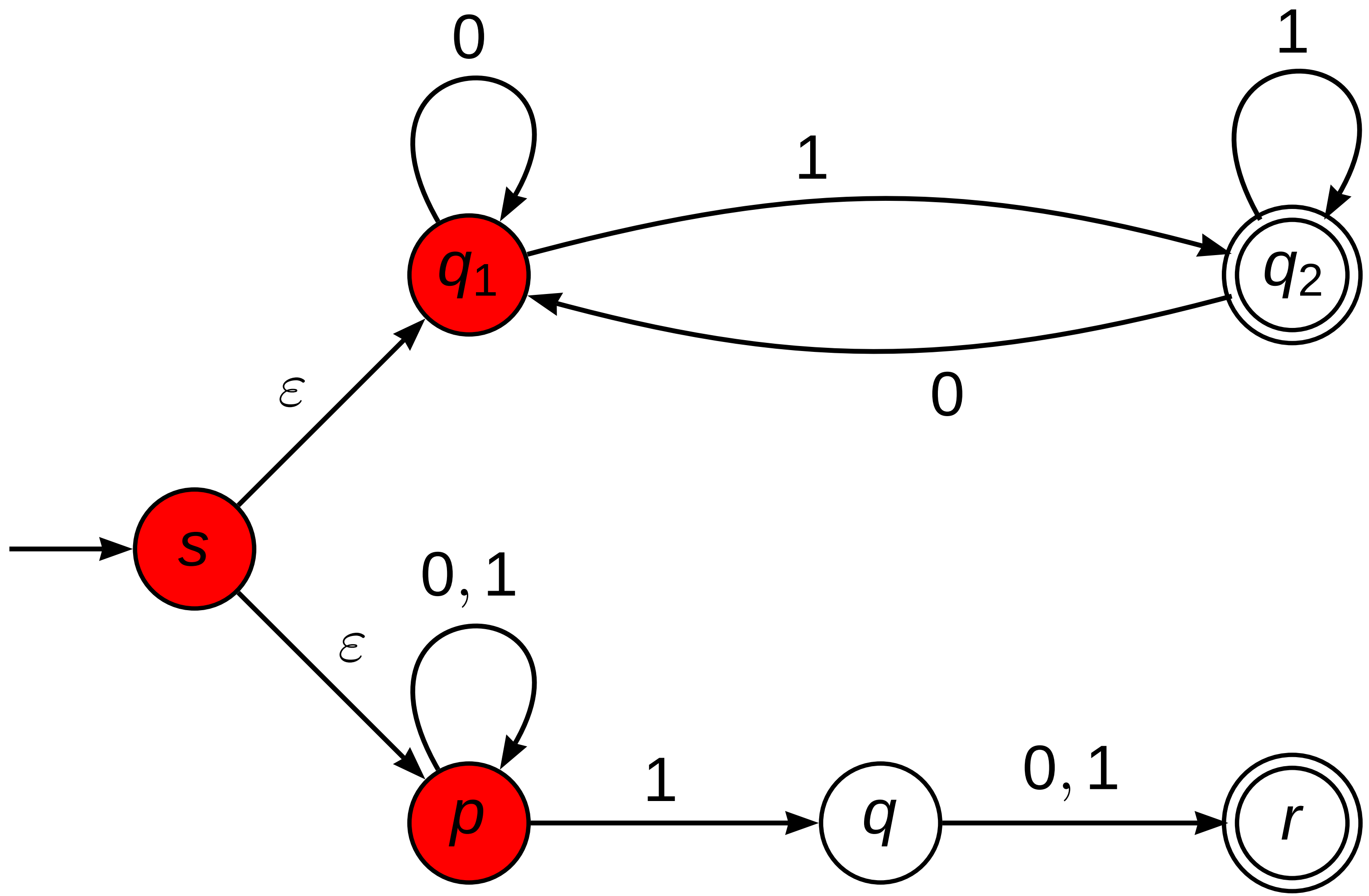


Example of a NFA with ε transitions



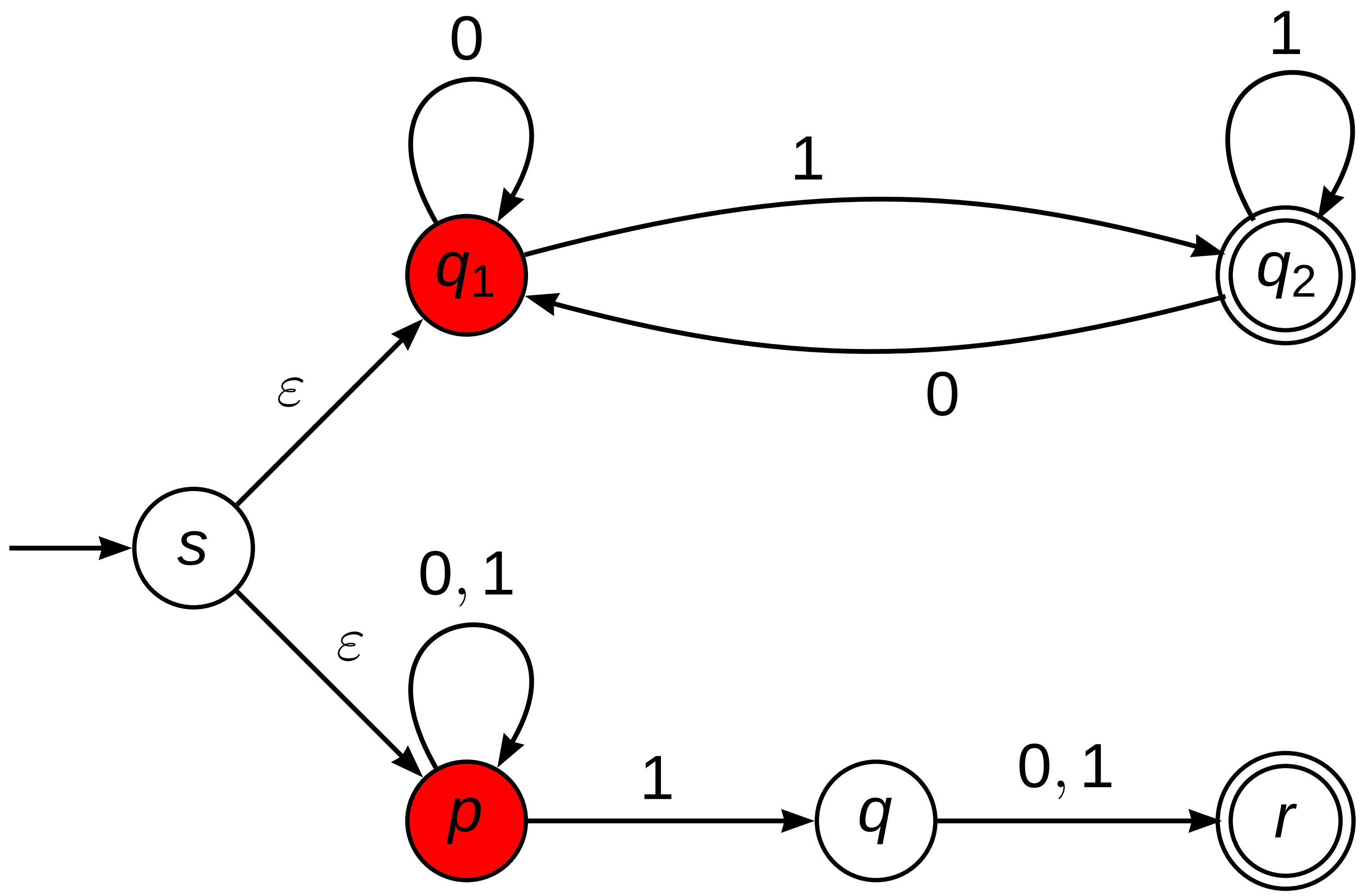
01

Example of a NFA with ε transitions



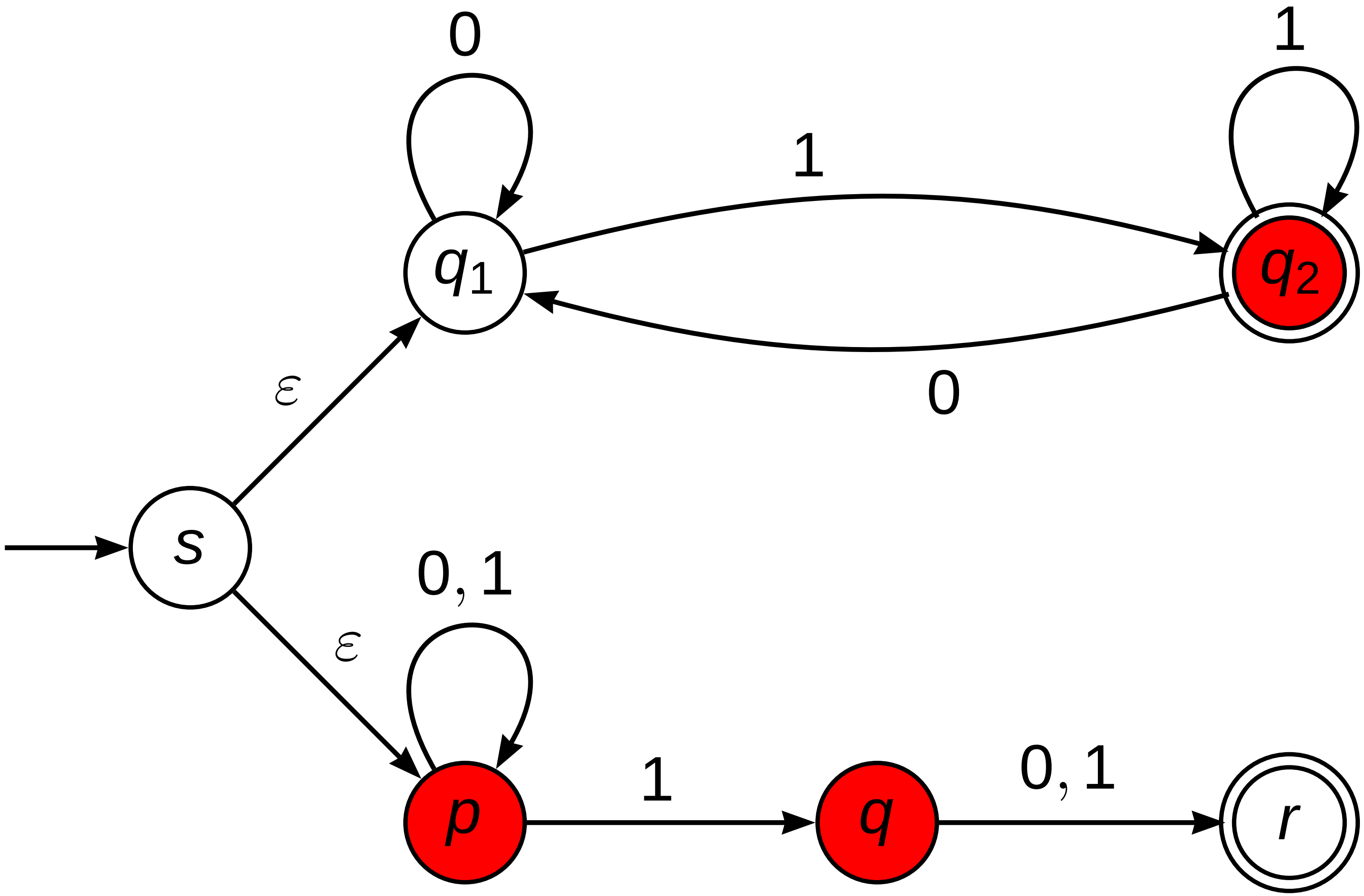
01

Example of a NFA with ε transitions



$\underline{0}1$

Example of a NFA with ε transitions



$0\underline{1}$ ACCEPT

Definition of NFAs

The power set of Q, written $\mathcal{P}(Q)$, is the set of all subsets of Q

Example

If $A = \{1, 2\}$ then $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$

Definition of NFAs

Definition

A nondeterministic finite automaton (NFA) is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$ where

- Q is a finite set called the states
- Σ is a finite set called the alphabet
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ is the transition function
- $q_0 \in Q$ is the start state
- $F \subseteq Q$ is the set of accept states

The language recognized by a NFA

Definition

Let $N = (Q, \Sigma, \delta, q_0, F)$ be a NFA and $s = s_1 s_2 \cdots s_m$ a string over Σ. N accepts s if there is a sequence of states $r_0, r_1, \ldots r_n$ from Q and s can be written as $s = s_1, s_2 \ldots, s_n$ where each $s_i \in \Sigma \cup \{\varepsilon\}$ such that

- $r_0 = q_0$,
- $r_{i+1} \in \delta(r_i, s_{i+1})$ ($i = 0, \ldots, n - 1$), and
- $r_n \in F$

Equivalence of DFAs and NFAs

Definition

Two machines are equivalent if they recognize the same language

Equivalence of DFAs and NFAs

Theorem

Every NFA has an equivalent DFA

Equivalence of DFAs and NFAs

Theorem

Every NFA has an equivalent DFA

Proof idea.

Given a NFA we need to construct a DFA that simulate the NFA

- The DFA need to keep track of the set of active states of the NFA at each step
- If k is the number of states of the NFA, then the DFA might need up to 2^k states (one for each subset of states of the NFA)
- So, the states of the DFA should be $\mathcal{P}(Q)$ where Q is the states of the NFA

□

Equivalence of DFAs and NFAs

Theorem

Every NFA has an equivalent DFA

Notation: Let $E(R)$ be the set of states that can be reached from R using 0 or more ε transitions.

Equivalence of DFAs and NFAs

Theorem

Every NFA has an equivalent DFA

Notation: Let $E(R)$ be the set of states that can be reached from R using 0 or more ε transitions.

Proof.

The subset construction: Given a NFA $N = (Q, \Sigma, \delta, q_0, F)$ we construct an equivalent DFA $M = (Q', \Sigma, \delta', q_0', F')$ where

- $Q' = \mathcal{P}(Q)$
- $q_0' = E(\{q_0\})$
- $F' = \{R \in Q' \mid R \cap F \neq \emptyset\}$

□

Equivalence of DFAs and NFAs

Theorem

Every NFA has an equivalent DFA

Notation: Let $E(R)$ be the set of states that can be reached from R using 0 or more ε transitions.

Proof.

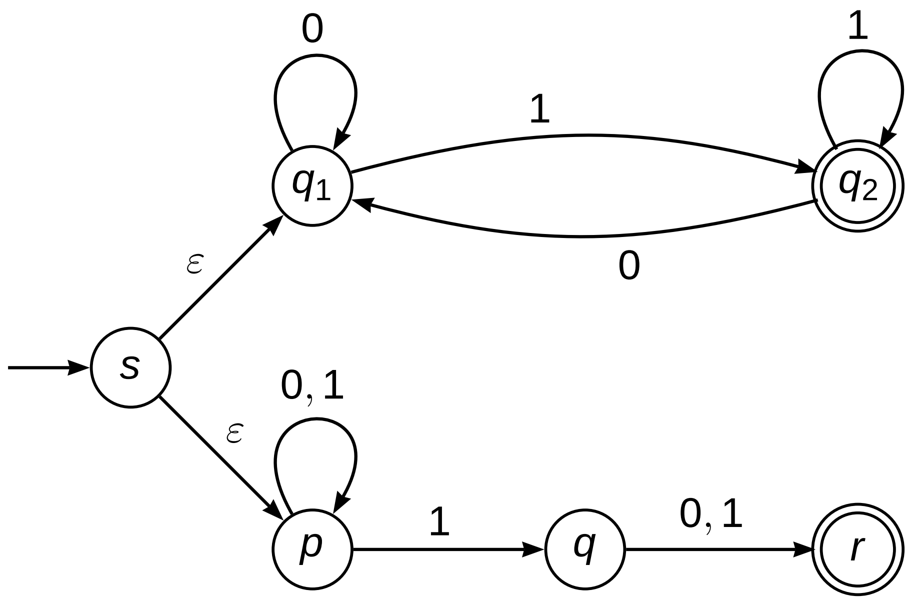
The subset construction: Given a NFA $N = (Q, \Sigma, \delta, q_0, F)$ we construct an equivalent DFA $M = (Q', \Sigma, \delta', q_0', F')$ where

- For $R \in Q'$ and $a \in \Sigma$,
  $\delta'(R, a) = \{q \in Q \mid q \in E(\delta(r, a)) \textit{ for some } r \in R\}$

  $\delta'(R, a)$ is the set of all states that can be reached (in the NFA) by first following a transition labeled a from a state in R and then following 0 or more ε transitions

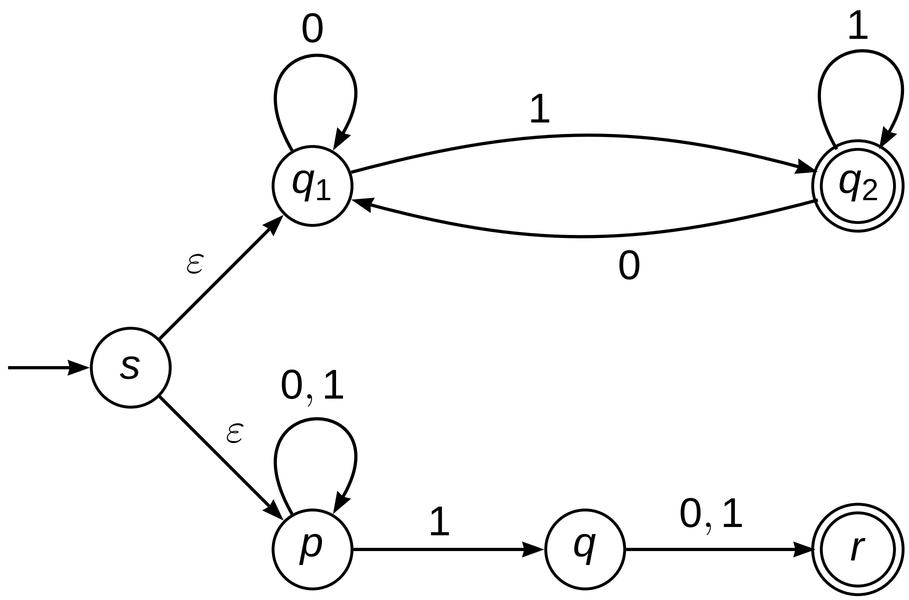
□

Equivalence of DFAs and NFAs: Example



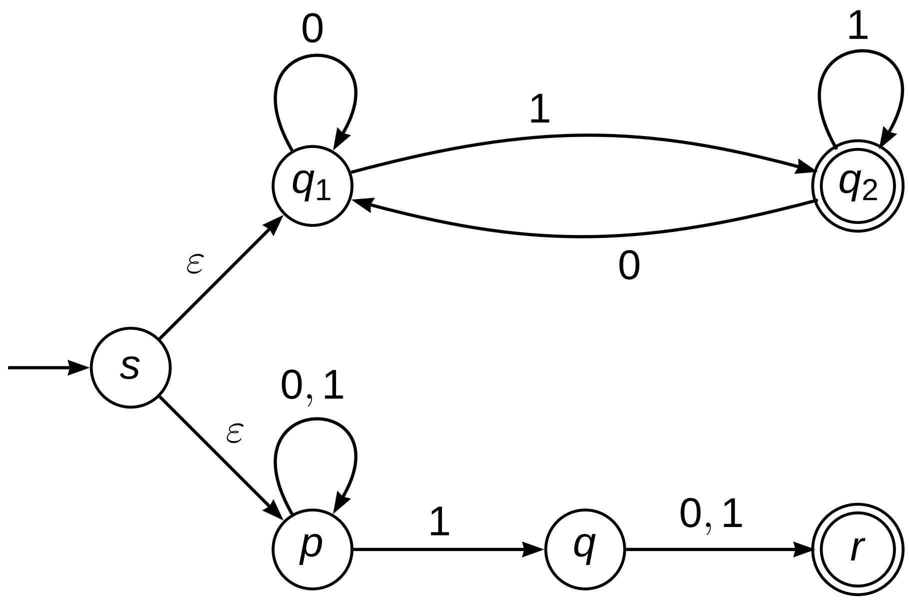
	0	1
$\rightarrow \{s, q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$

Equivalence of DFAs and NFAs: Example



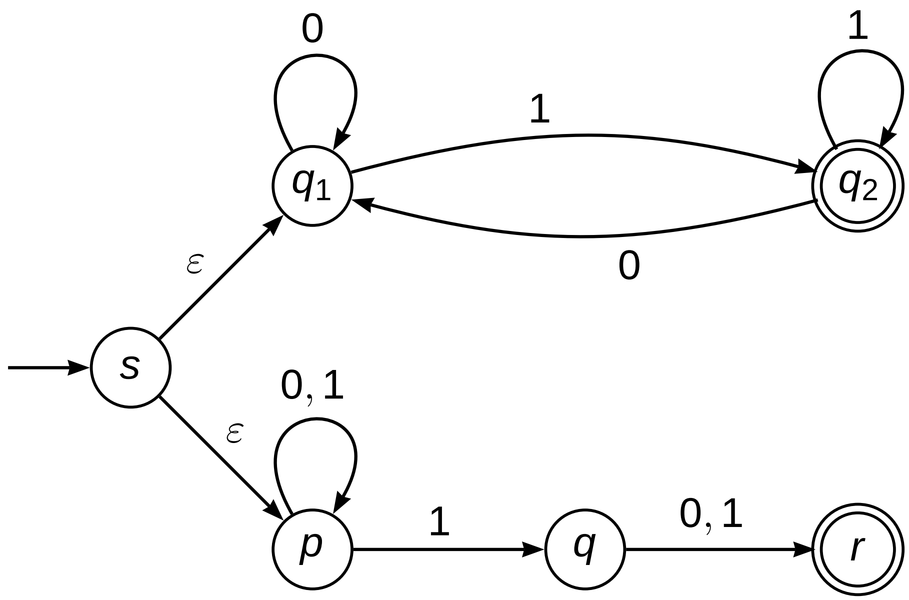
	0	1
$\rightarrow \{s, q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
$\{q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$

Equivalence of DFAs and NFAs: Example



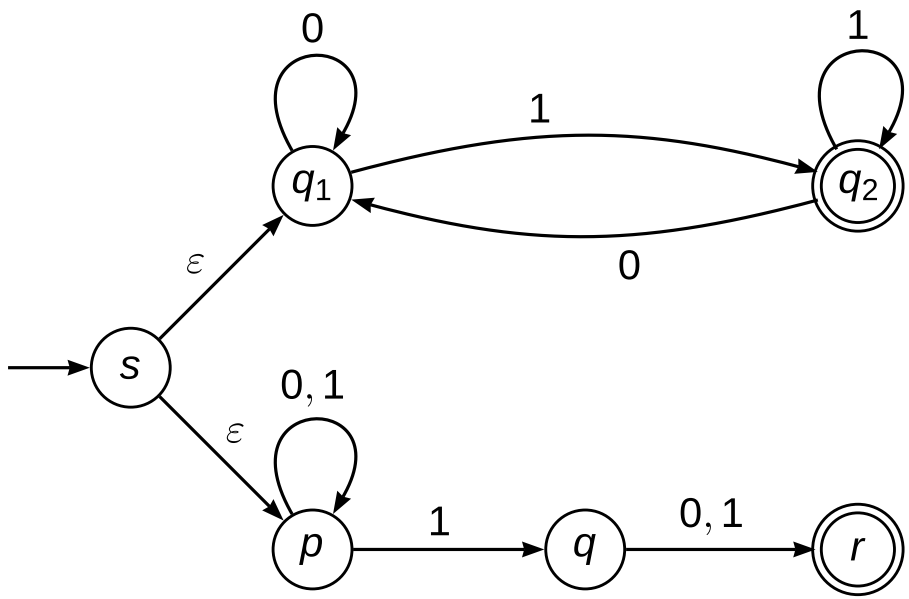
	0	1
$\rightarrow \{s, q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
$\{q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
F $\{q_2, p, q\}$	$\{q_1, p, r\}$	$\{q_2, p, q, r\}$

Equivalence of DFAs and NFAs: Example



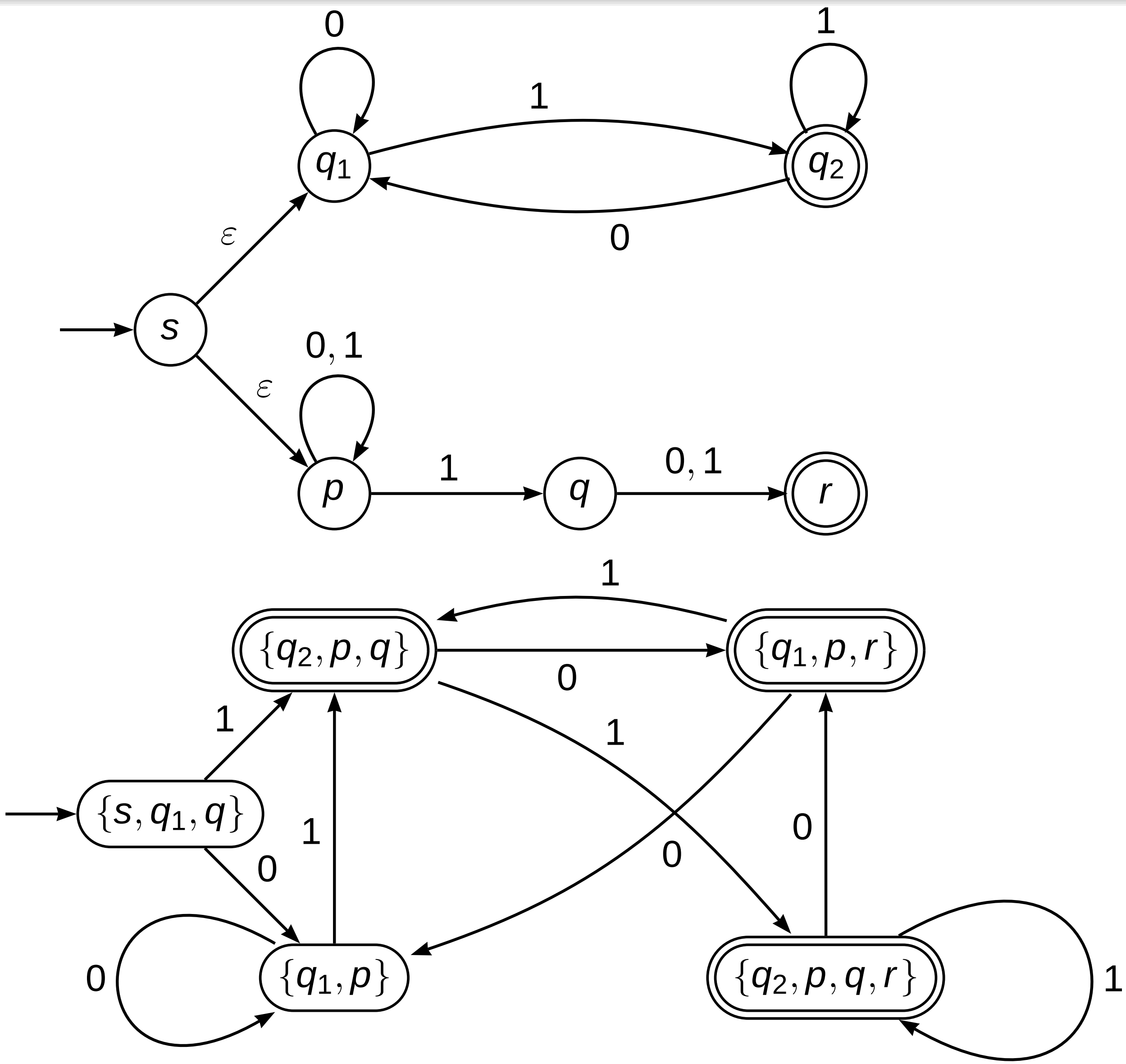
	0	1
$\rightarrow \{s, q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
$\{q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
F $\{q_2, p, q\}$	$\{q_1, p, r\}$	$\{q_2, p, q, r\}$
F $\{q_1, p, r\}$	$\{q_1, p\}$	$\{q_2, p, q\}$

Equivalence of DFAs and NFAs: Example



	0	1
$\rightarrow \{s, q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
$\{q_1, p\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
$F\ \{q_2, p, q\}$	$\{q_1, p, r\}$	$\{q_2, p, q, r\}$
$F\ \{q_1, p, r\}$	$\{q_1, p\}$	$\{q_2, p, q\}$
$F\ \{q_2, p, q, r\}$	$\{q_1, p, r\}$	$\{q_2, p, q, r\}$

Equivalence of DFAs and NFAs: Example



Equivalence of DFAs and NFAs

Theorem

Every NFA has an equivalent DFA

Corollary

Every language recognized by a NFA can be recognized by a DFA