Nondeterminisitic Finite Automata

Lin Chen

Email: Lin.Chen@ttu.edu

Grader: zulfi.khan@ttu.edu

TEXAS TECH
UNIVERSITY

Deterministic Finite Automata

- Deterministic finite automata are

  - Deterministic: given the current state and next input symbol, it moves deterministically to a next state.

  - Finite: consists of finite number of states

  - Automata: machine

Nondeterministic Finite Automata

- Nondeterministic finite automata are

  - Nondeterministic: given the current state and next input symbol, it moves to no or one of several legal states.

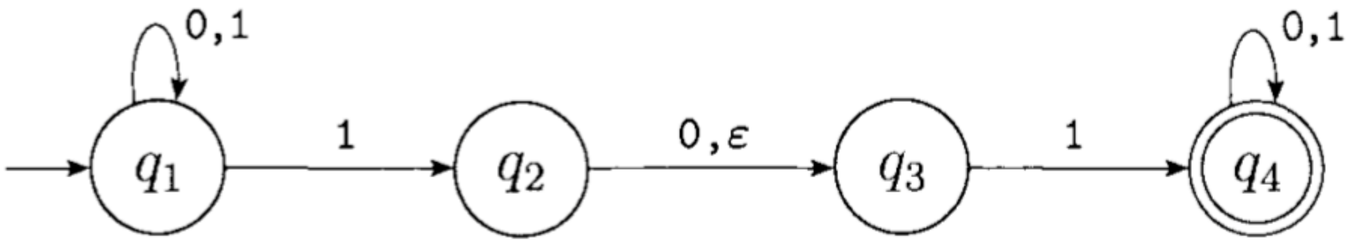
  - A Nondeterministic Finite Automata can have 0, 1 or more transitions for a single state/symbol pair

Nondeterministic Finite Automata

- Nondeterministic finite automata are

  - Nondeterministic: given the current state and next input symbol, it moves to no or one of several legal states.

  - A Nondeterministic Finite Automata can have 0, 1 or more transitions for a single state/symbol pair

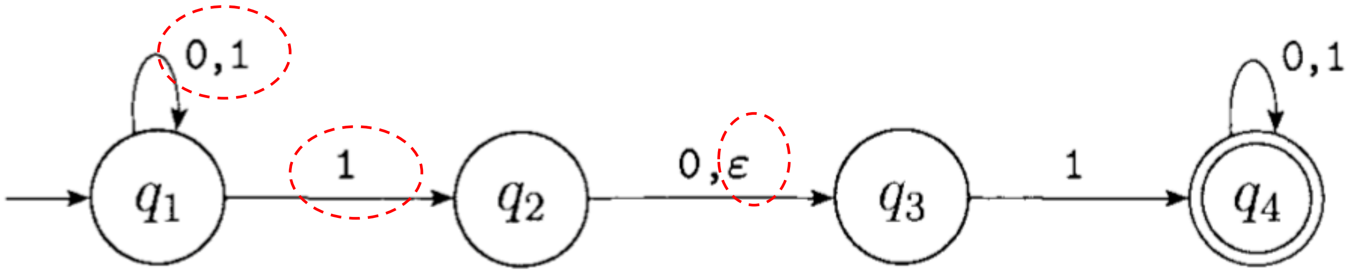


Nondeterministic Finite Automata

- Nondeterministic finite automata are

  - Nondeterministic: given the current state and next input symbol, it moves to no or one of several legal states.

  - A Nondeterministic Finite Automata can have 0, 1 or more transitions for a single state/symbol pair



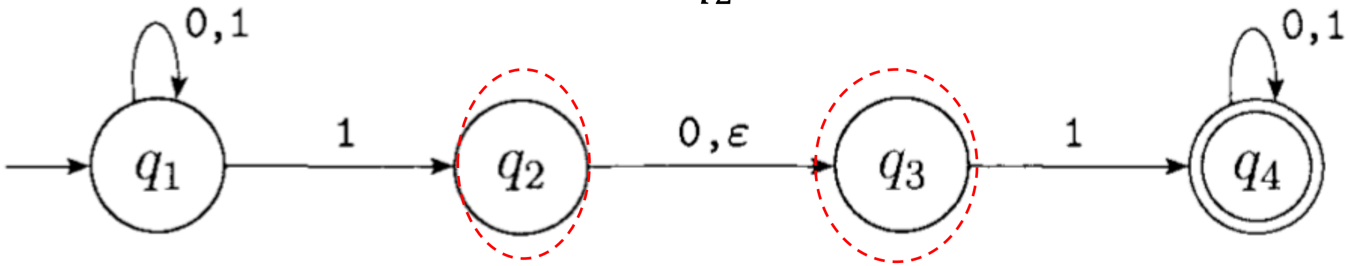
Nondeterministic Finite Automata

- Nondeterministic finite automata are

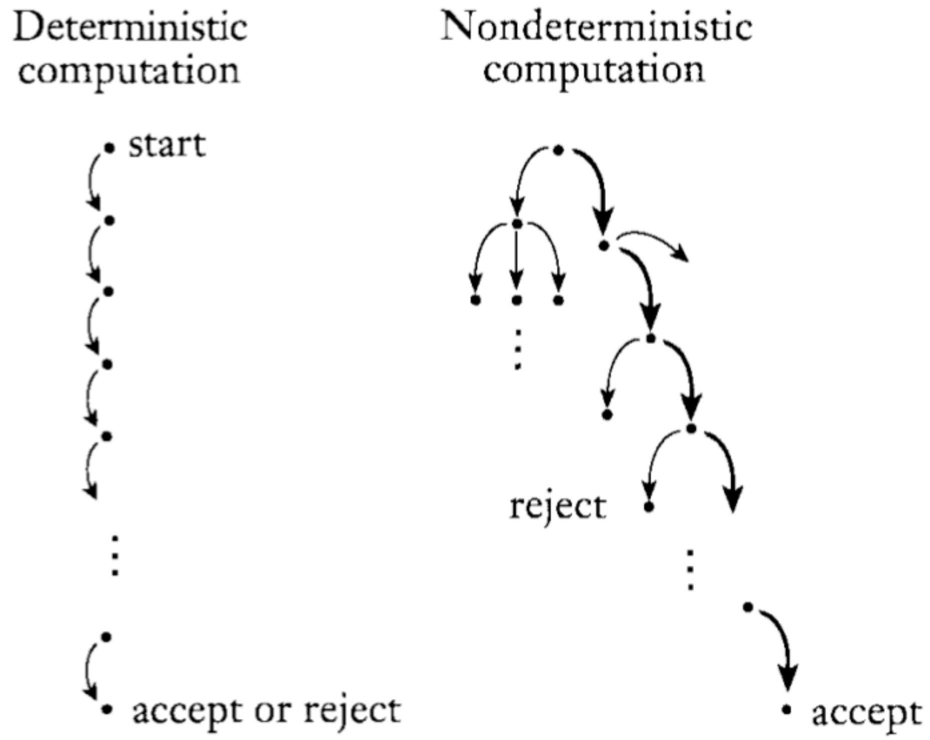
  - Nondeterministic: given the current state and next input symbol, it moves to no or one of several legal states.

  - A Nondeterministic Finite Automata can have 0, 1 or more transitions for a single state/symbol pair

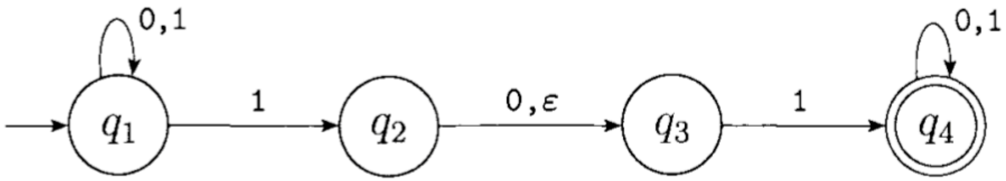
What if I read 1 at q_2



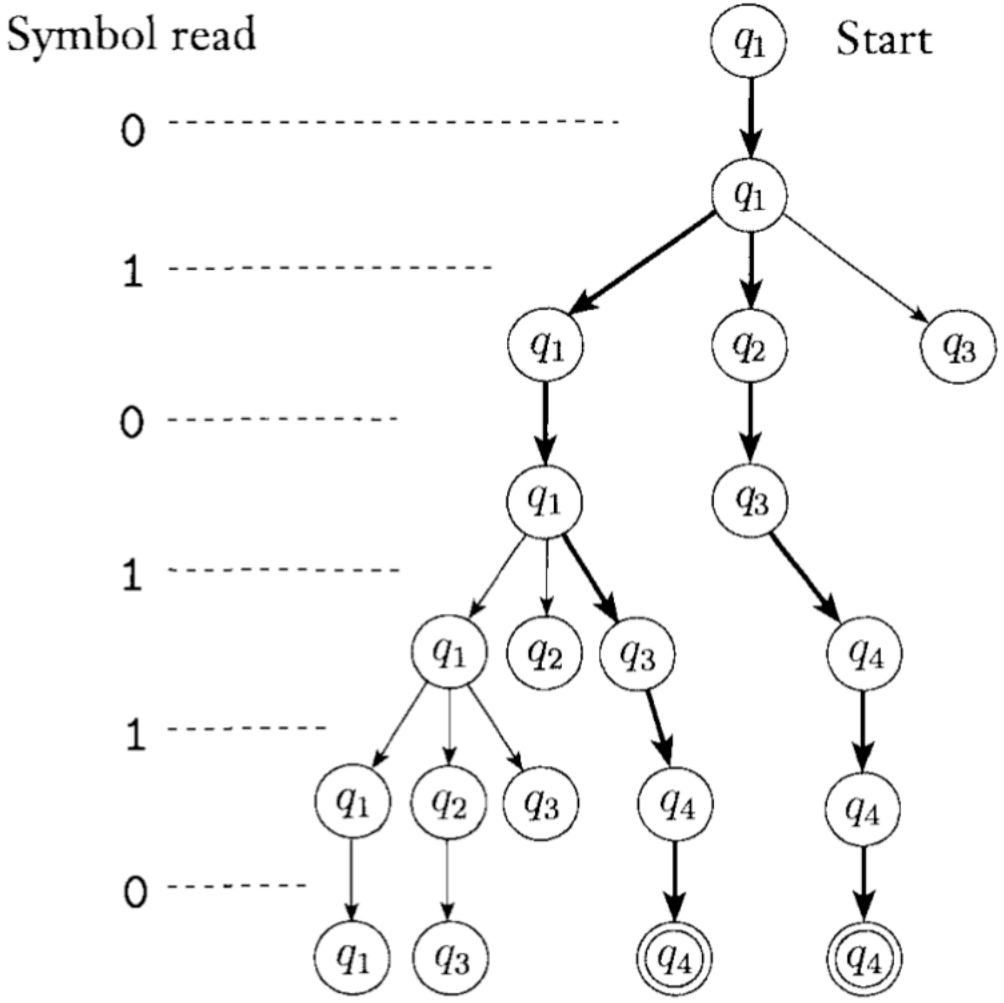
Nondeterministic Finite Automata



Nondeterministic Finite Automata



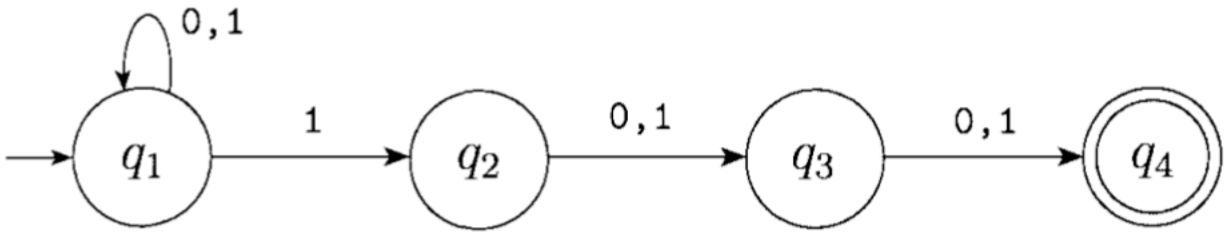
Computation on: 010110



Nondeterministic Finite Automata

- NFA example

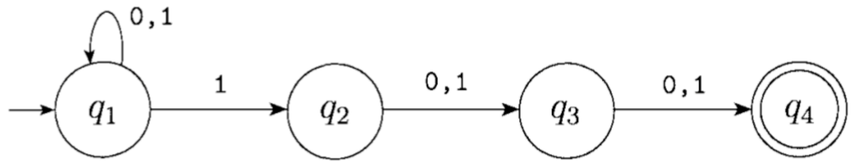
  --All strings over $\{0,1\}$ with 1 at the third position from end (e.g., 000100 is valid, 0011 is not)

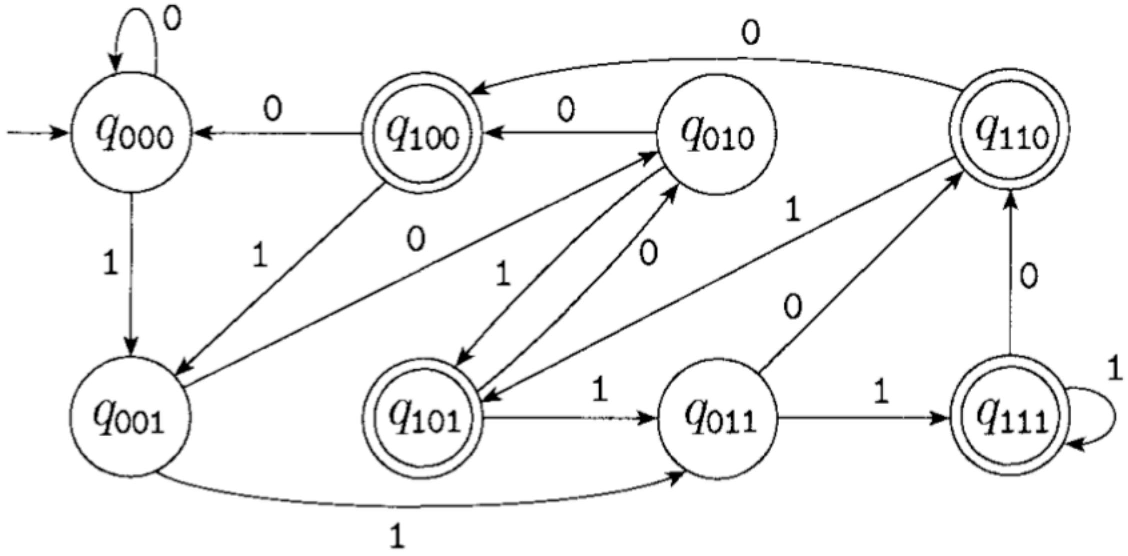


Nondeterministic Finite Automata

- NFA example

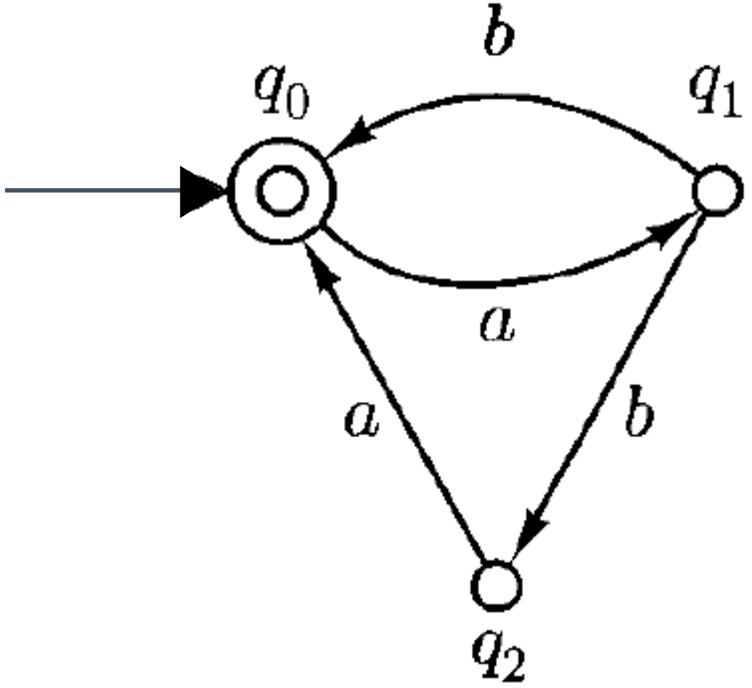
  --All strings over $\{0,1\}$ with 1 at the third position from end (e.g., 000100 is valid, 0011 is not)





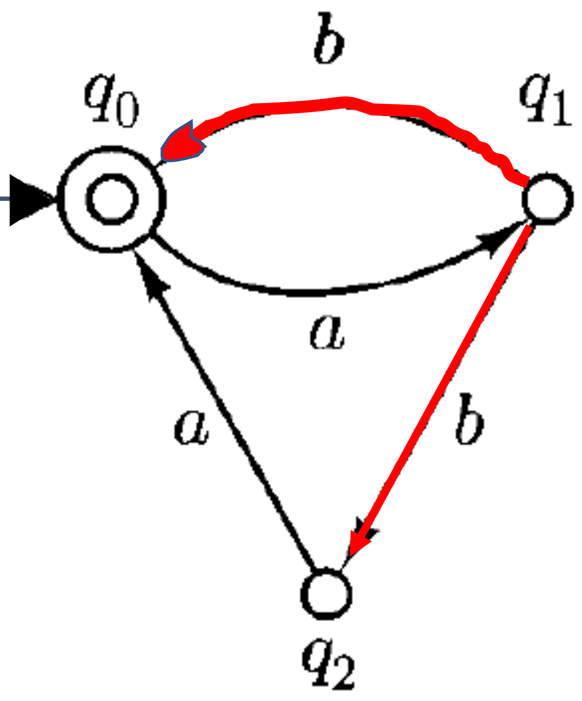
Nondeterministic Finite Automata

- NFA example
  - when NFA is in state q_1 and the input is b ⋯



Nondeterministic Finite Automata

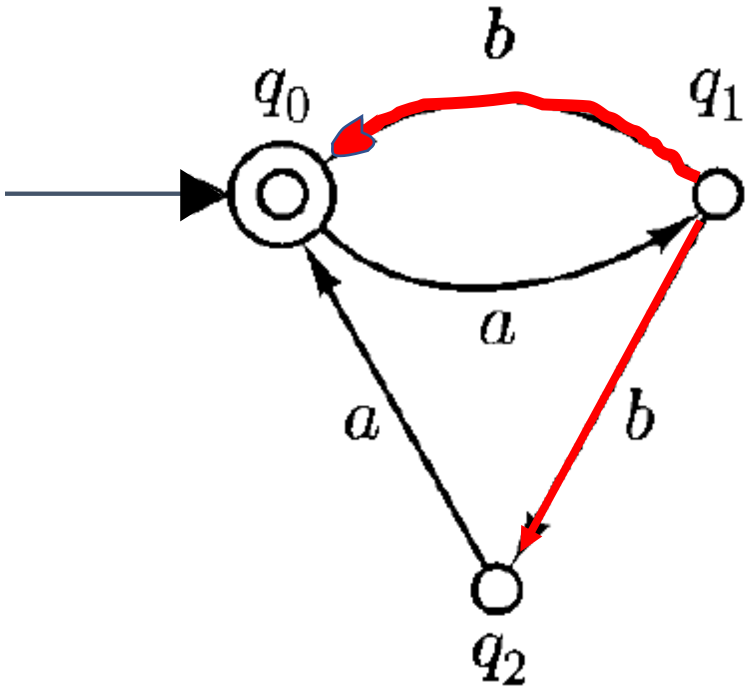
- NFA example
  - when NFA is in state q_1 and the input is b, it can go to two states, both are legal moves
  - A string is accepted if there is some way to get from the initial state (q_0) to a final state (q_0)



Think of the machine split into two copies, one goes to q_0, the other goes to q_2

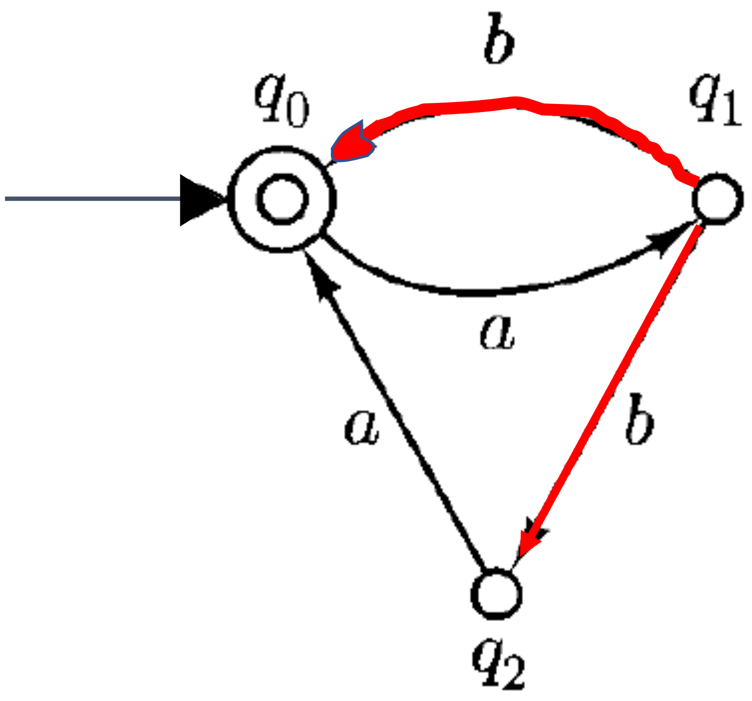
Nondeterministic Finite Automata

- NFA example
  - it does not accept a string starts with b



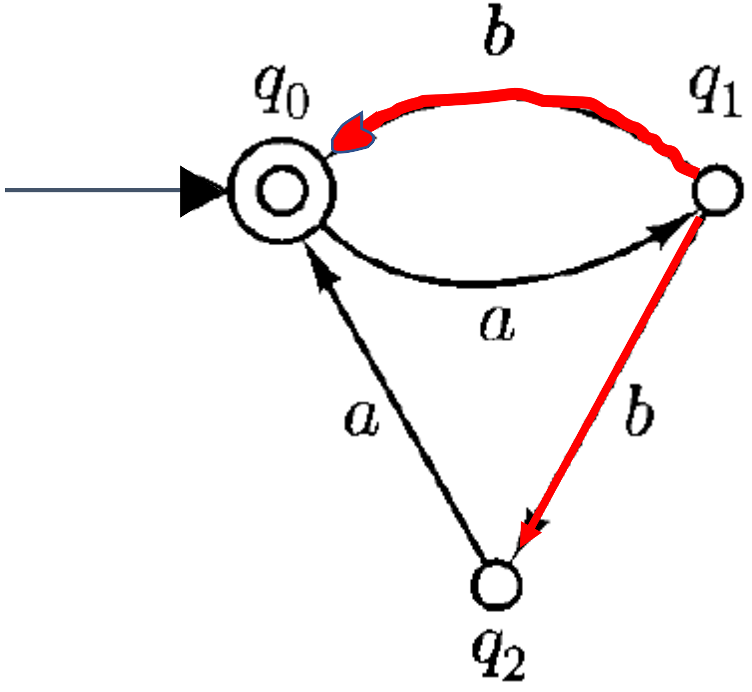
Nondeterministic Finite Automata

- NFA example
  - it does not accept a string starts with b
  - a string starts with a must be followed by b, then it either goes to q_0 or to q_2, which further requires an a to go to q_0



Nondeterministic Finite Automata

- NFA example
  - it does not accept a string starts with b
  - a string starts with a must should be followed by b, then it either goes to q_0 or to q_2, which further requires an a to go to q_0
  - A complete cycle is ab or aba



$$L(M) = \{ab, aba\}^*$$

NFA definition

- A quintuple $M = (Q, \Sigma, \delta, q_0, F)$ where
  - Q is a finite set of states
  - Σ is an alphabet
  - $\mathrm{q}_0 \in Q$ is the initial state
  - $F \subseteq Q$ is the set of final states (can be multiple)
  - δ, the transition function, $Q \times \Sigma_\epsilon \rightarrow 2^Q$

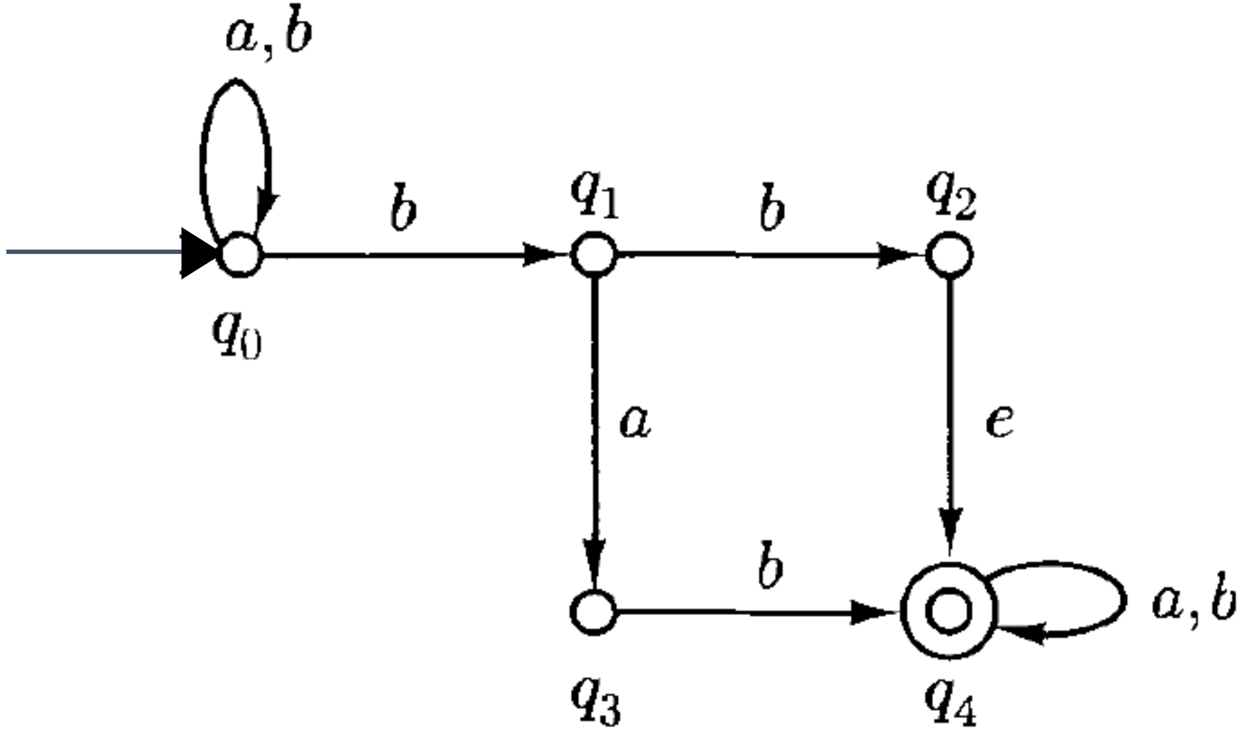
In DFA (- δ, the transition function, a function from $\mathrm{Q} \times \Sigma$ to Q)

NFA definition

- A quintuple $M = (Q, \Sigma, \delta, q_0, F)$ where
  - Q is a finite set of states
  - Σ is an alphabet
  - $\mathrm{q}_0 \in Q$ is the initial state
  - $F \subseteq Q$ is the set of final states (can be multiple)
  - Δ, the transition relation, is a subset of $Q \times (\{\Sigma \cup \epsilon\}) \times Q$

NFA definition

- Example

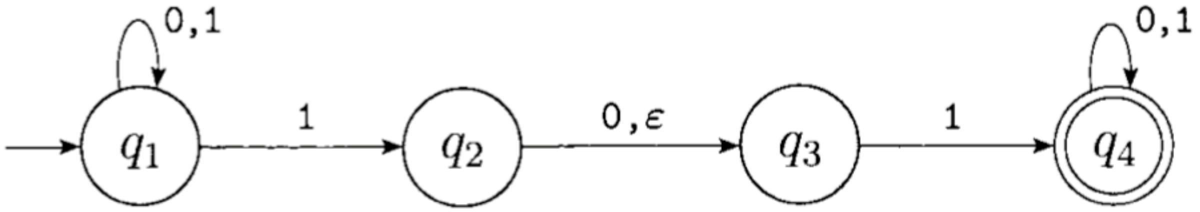


$$Q = \{q_0, q_1, q_2, q_3, q_4\},$$
$$\Sigma = \{a, b\},$$
$$s = q_0,$$
$$F = \{q_4\},$$

$$\Delta = \{(q_0, a, q_0), (q_0, b, q_0), (q_0, b, q_1), (q_1, b, q_2), (q_1, a, q_3), (q_2, e, q_4), (q_3, b, q_4), (q_4, a, q_4), (q_4, b, q_4)\}.$$

NFA definition

- Example



The formal description of N_1 is $(Q, \Sigma, \delta, q_1, F)$, where

1. $Q = \{q_1, q_2, q_3, q_4\}$,
2. $\Sigma = \{0,1\}$,
3. δ is given as

	0	1	ε
q_1	$\{q_1\}$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_3\}$	$\emptyset$	$\{q_3\}$
q_3	$\emptyset$	$\{q_4\}$	$\emptyset$
q_4	$\{q_4\}$	$\{q_4\}$	$\emptyset$

,

4. q_1 is the start state, and
5. $F = \{q_4\}$.

NFA Configurations

- We have learned two ways of describing a NFA
  - A quintuple $M = (Q, \Sigma, \Delta, q_0, F)$
  - A state diagram
- How do we characterize the computation of a NFA?

NFA Configurations

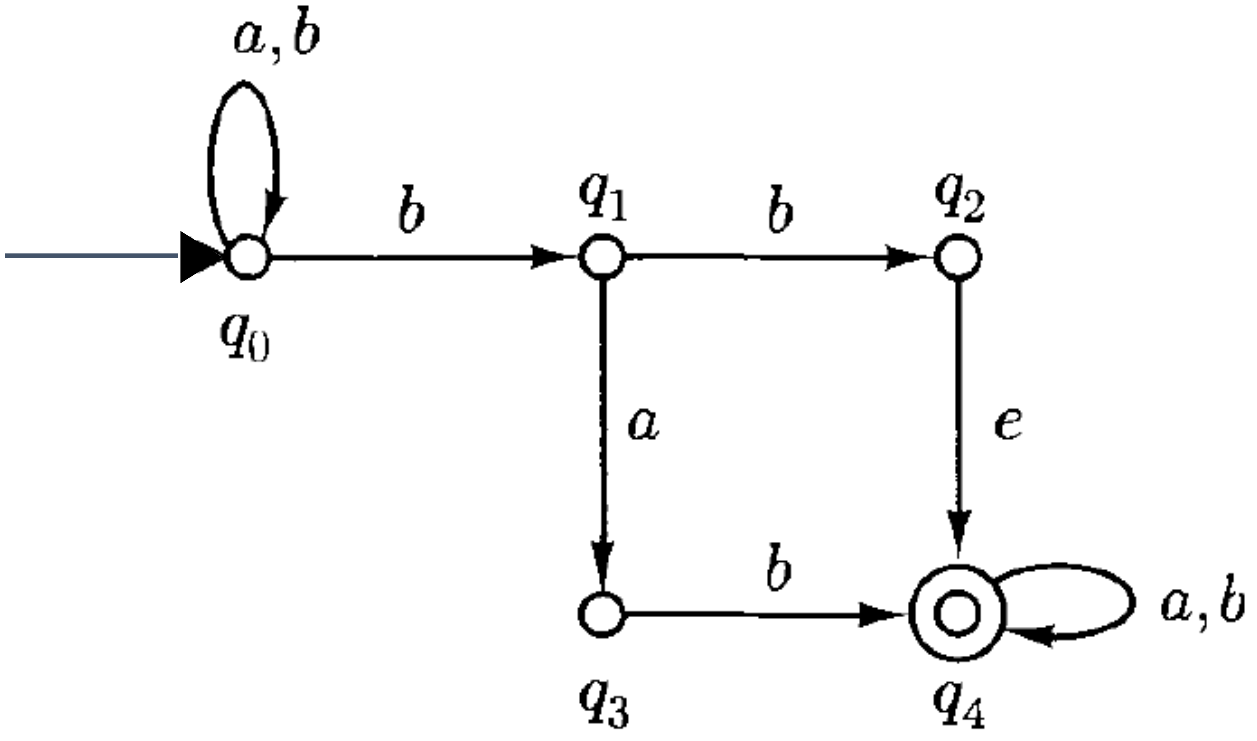
- We have learned two ways of describing a NFA
  - A quintuple $M = (Q, \Sigma, \Delta, q_0, F)$
  - A state diagram
- How do we characterize the computation of a NFA?
  - the computation of a NFA has to be defined on a specific input
  - use a sequence of configurations to represent the computation

NFA Configurations

- Configuration for a DFA $M = (Q, \Sigma, \Delta, q_0, F)$
  - any element of $\mathrm{Q} \times \Sigma^*$
  - the state the NFA currently in
  - the remaining part of the string to be processed

NFA Configurations

- Example $(bababab)$

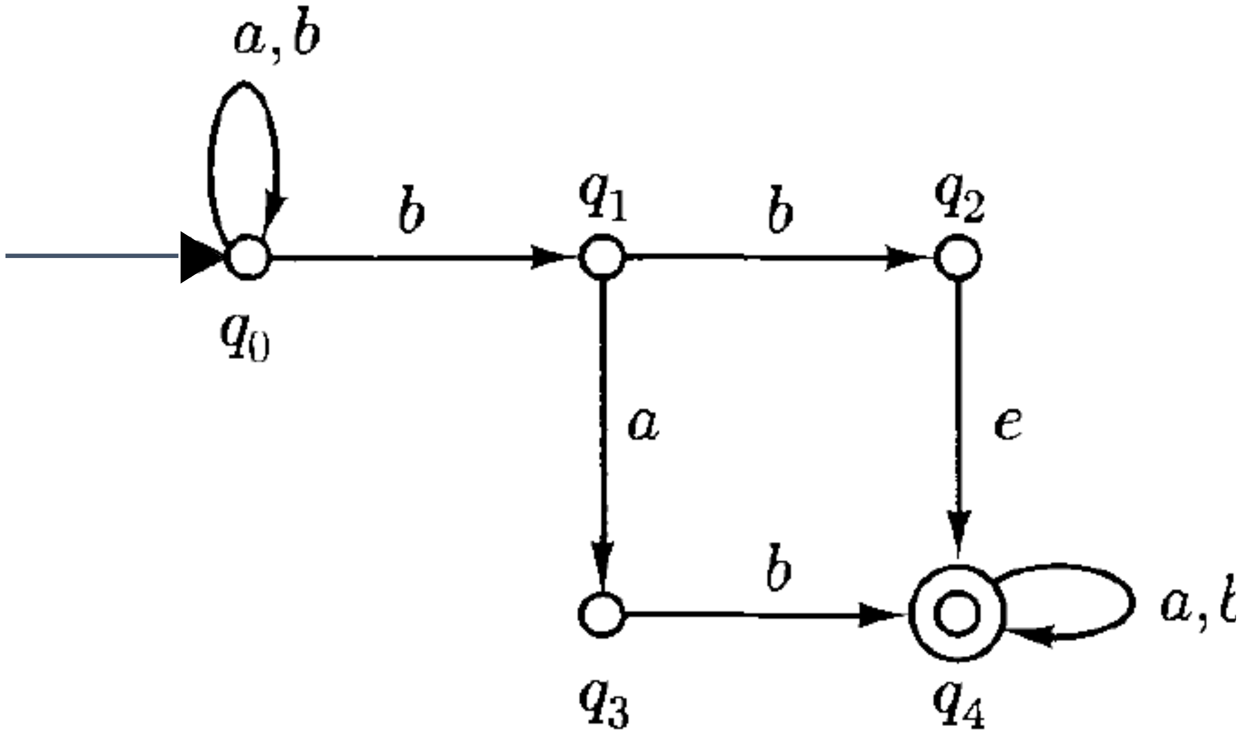


$$
\begin{aligned}
(q_0, bababab) &\vdash_M (q_0, ababab) \\
&\vdash_M (q_0, babab) \\
&\vdash_M (q_0, abab) \\
&\vdots \\
&\vdash_M (q_0, e)
\end{aligned}
$$

$(q_0, bababab) \vdash^* (q_0, e)$

NFA Configurations

- Example $(bababab)$



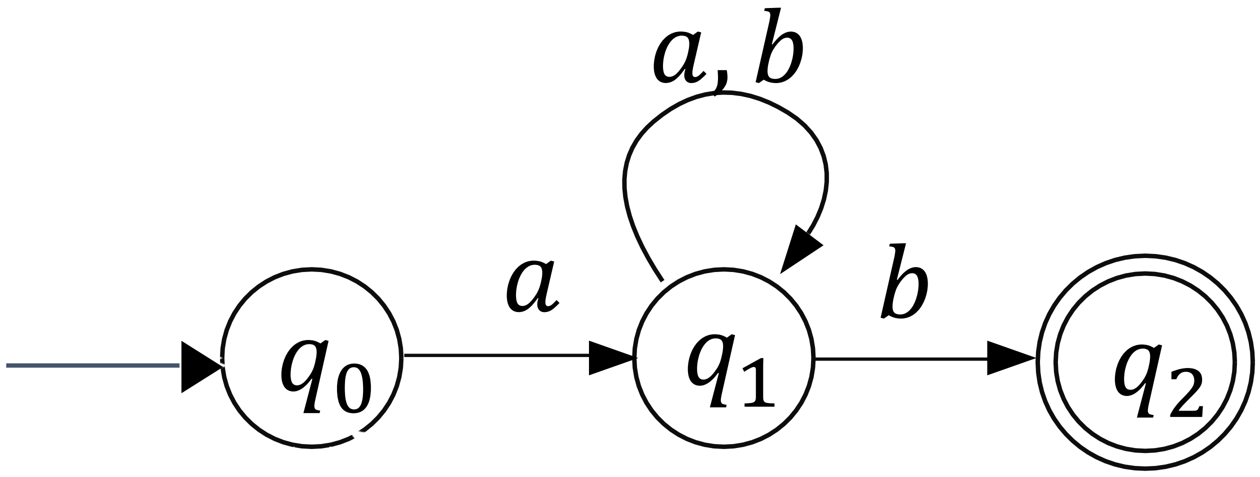
$$
\begin{aligned}
(q_0, bababab) &\vdash_M (q_1, ababab) \\
&\vdash_M (q_3, babab) \\
&\vdash_M (q_4, abab) \\
&\vdash_M (q_4, bab) \\
&\vdash_M (q_4, ab) \\
&\vdash_M (q_4, b) \\
&\vdash_M (q_4, e)
\end{aligned}
$$

The string is accepted

$(q_0, bababab) \vdash_M^* (q_4, e)$

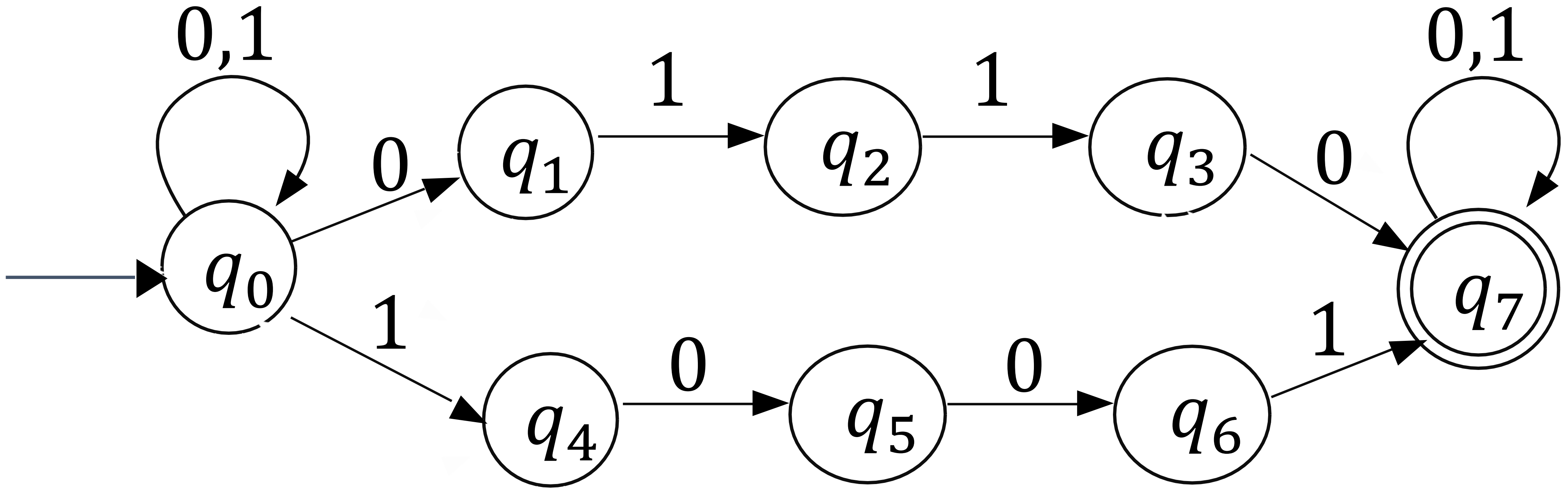
More NFA examples

- All strings starts with a and ends with b



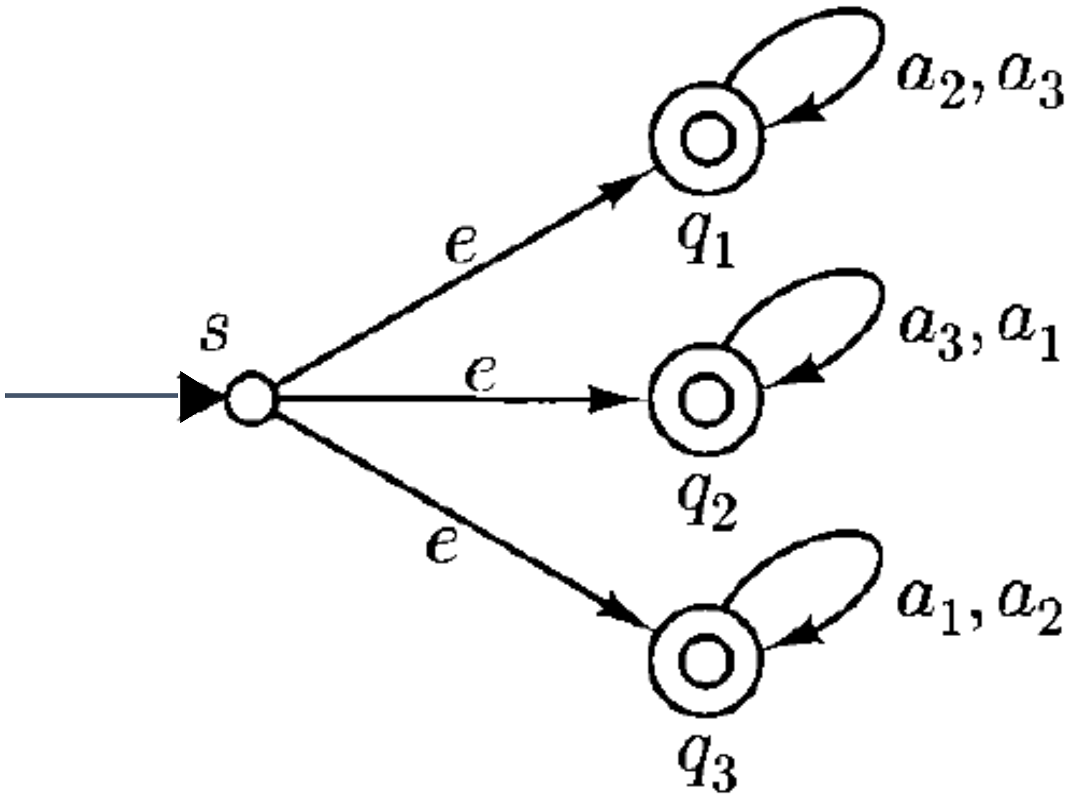
More NFA examples

- All strings that contain 0110 or 1001



More NFA examples

- All strings consisting of at most two symbols out of $\{a_1, a_2, a_3\}$



NFA and languages

- An NFA M accepts a string w if
  - $(q_0, w) \vdash_M^* (q, e)$ where $q \in F$ (ends in a final state)
- The language accepted by M is the set of all strings accepted by M, and denoted as $L(M)$
- NFA languages is the set of all languages accepted by some NFA
  - What is the relationship between NFA and DFA languages?