CS310

Nondeterministic Finite Automata

Sections:1.2 page 47

September 11, 2006

Nondeterminism

- What?
  - State diagram can have none, one, or many exiting arrows for each symbol at each state
  - ε-transition automatically taken without any input
- Why?
  - Recognizes same languages as DFA (can convert NFA to DFA)
  - Often easier to build
  - Need machine that will run processes simultaneously

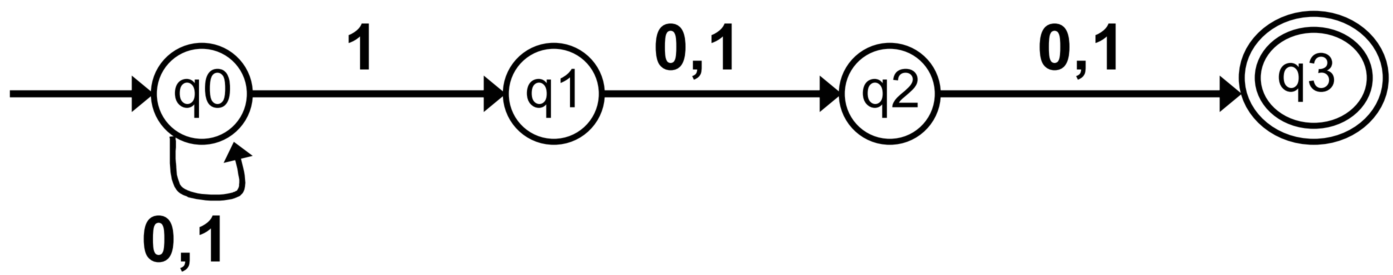
Example (1.30)

- Accept string of at least length three that contains a 1 in the third from end

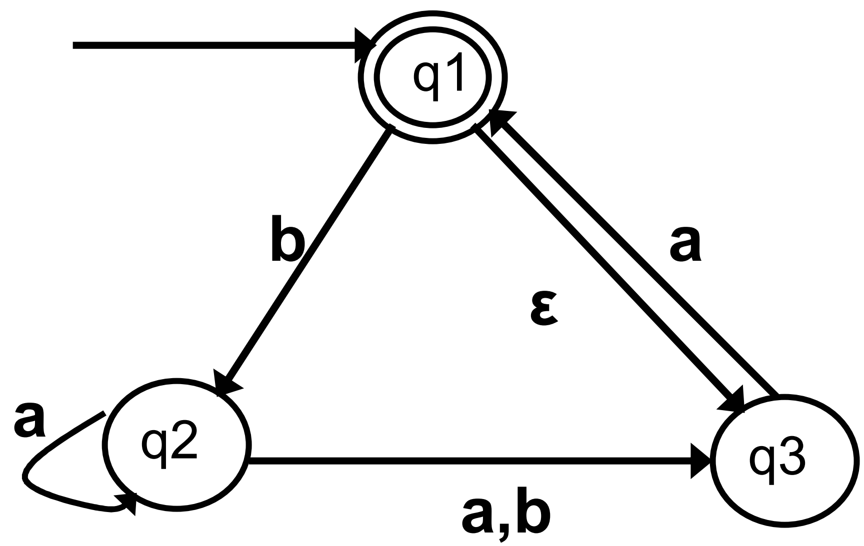
  $\Sigma=\{0,1\}$, $\Sigma^*1(0 \cup 1)(0 \cup 1)$

  What makes this difficult for a DFA?

  Equivalent DFA takes 8 states. Why 8?



Example



Accept: **ε, a, baba, baa**

Reject: b, bb, babba

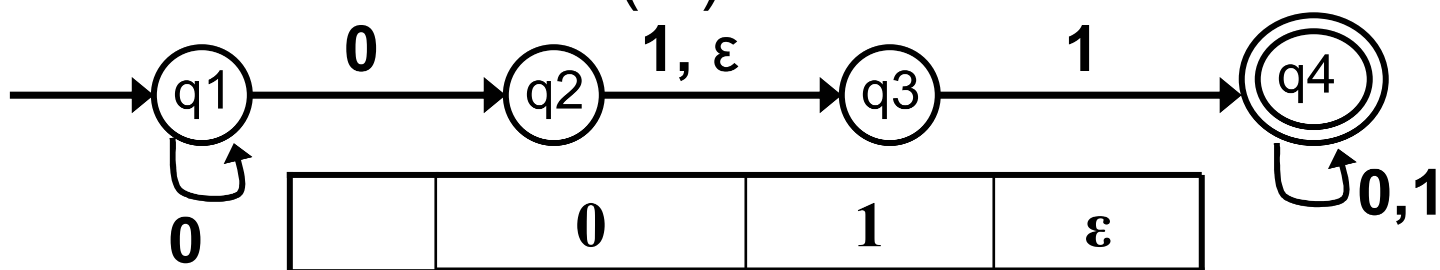
Formal Definition of NFA

- 5 tuple ($Q, \Sigma, \delta, q_0, F$)

  $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

  $\delta : Q \times \Sigma_\varepsilon \rightarrow P(Q)$ (DFA: $\delta : Q \times \Sigma \rightarrow Q$)

  P(Q) is the power set of Q. The collection of all subsets of Q. *P*(Q) has $2^{|Q|}$ members



	0	1	ε
q1	{q1.q2}	ø	ø
q2	ø	{q3}	{q3}
q3	ø	{q4}	ø
q4	{q4}	{q4}	ø

Formal Definition of Computing for NFA

- Given a machine $M= (Q, \sum, \delta, q_0, F)$ and a string $w=w_1w_2\ldots w_n$ over $\sum$, then M ***accepts*** w if there exists a sequence of states $r_0, r_1 \ldots r_n$ in Q such that:
  - $r_0 = q_0$: r_0 is the start state
  - $\delta(r_i, w_{i+1}) = r_{i+1}$, $i=0,\ldots,n-1$: legal transitions
  - $r_n \in F$: stop in an accept state

- Identical to the definition for the DFA!

Practice

- Construct a NFA with three states that recognizes $\{w \mid w \text{ ends with two 0s}\}$

  $\Sigma = \{0,1\}$

Practice

- Construct a NFA with six states

  {w | w even # 0s OR exactly two 1s}

  $\Sigma = \{0,1\}$

Practice

- Construct a NFA with three states

  0*1*0*0

  $\Sigma = \{0,1\}$