Nondeterministic Finite Automata

Fundamental set operations

Let A and B be two languages. In addition to union and intersection, we consider the following set operations:

- **Concatenation** of A and B, $A \circ B$, is $\{xy \mid x \in A$ and $y \in B\}$,
- **Star** of A, A^*, is $\{x_1 x_2 \cdots x_k \mid k \geq 0$ and $x_1, \ldots, x_k \in A\}$.
- **Complement** of A, $\overline{A}$, is $\Sigma^* - A$, i.e., $\{w \mid w \in \Sigma^* \wedge w \notin A\}$.

Fundamental set operations

Let A and B be two languages. In addition to union and intersection, we consider the following set operations:

- **Concatenation** of A and B, $A \circ B$, is $\{xy \mid x \in A$ and $y \in B\}$,
- **Star** of A, A^*, is $\{x_1 x_2 \cdots x_k \mid k \geq 0$ and $x_1, \ldots, x_k \in A\}$.
- **Complement** of A, $\overline{A}$, is $\Sigma^* - A$, i.e., $\{w \mid w \in \Sigma^* \wedge w \notin A\}$.

We will prove that the regular languages are closed under union, intersection, concatenation, star, and complement;, i.e., that if A and B are regular, then $A \cup B$, $A \cap B$, $A \circ B$, A^*, and $\overline{A}$ are each regular.

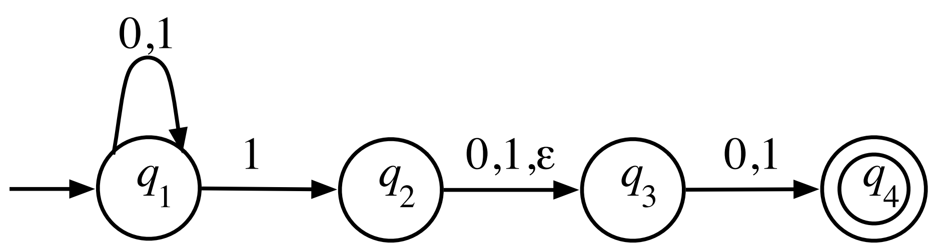
Nondeterministic Finite Automata

The **nondeterministic finite automaton** is a variant of finite automaton with two characteristics:

- **ϵ-transition**: state transition can be made **without reading a symbol**;
- **nondeterminism**: **zero or more than one possible value** may exist for state transition.

An Example Nondeterministic Finite Automaton

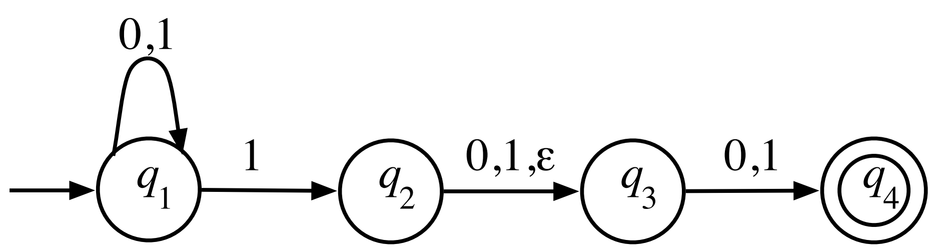
An NFA that accepts all strings over $\{0, 1\}$ that contain a 1 either at the third position from the end or at the second position from the end.



- There are two edges labeled 1 coming out of q_1.
- There are no edges coming out of q_4.
- The edge from q_2 is labeled with ϵ, in addition to 0 and 1.

What the Diagram Says

- If the node you are in has an outgoing edge labeled ϵ, you may choose to follow it.
- After receiving a symbol,
  - if the node you are in has multiple outgoing edges labeled with the symbol, you nondeterministically choose one of them and cross it;
  - if there are no choices, stop there.



Nondeterministic Finite Automata, Formally

A **nondeterministic finite automaton** is a 5-tuple $N = (Q, \Sigma, \delta, q_0, F)$, where δ is a mapping of $Q \times \Sigma_\epsilon$ to 2^Q (alternatively, written $\mathcal{P}(Q)$), and $\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$.

For each $p \in Q$ and each $a \in \Sigma_\epsilon$, $\delta(s, a) = R$ means:

"upon reading an a the automaton N may transition from state s to any state in R."

For ϵ in particular, $\delta(s, \epsilon) = R$ means:

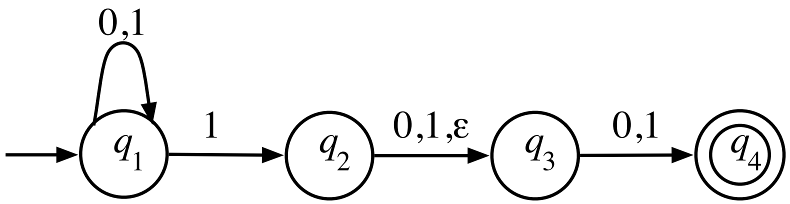
"without reading the symbol the automaton N may transition from state s to any state in R."

Acceptance by Nondeterministic Finite Automata

N accepts a word $w = w_1 \cdots w_n, w_1, \ldots, w_n \in \Sigma$ if: there exist $(p_0, \ldots, p_m), p_0, \ldots, p_m \in Q$ and $y_1, \ldots, y_m \in \Sigma_\epsilon$ such that

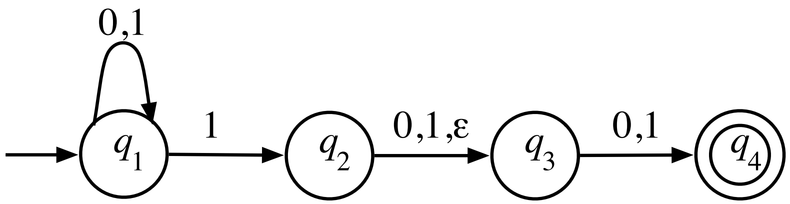
- $w = y_1 \cdots y_m$,
- $p_0 = q_0$,
- $p_m \in F$, and
- for every i, $1 \leq i \leq m$, $p_i \in \delta(p_{i-1}, w_i)$.

Transition Function for the Previous Example



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

Transition Function for the Previous Example



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

This NFA accepts $y = 11$ with respect to state sequence (q_1, q_2, q_3, q_4) and decomposition $y = 1\epsilon 1$.

Nondeterministic Choices

Nondeterministic computation can be thought of as a self-reproducing agent traveling in the state space.

1. At the start of computation the agent is in the initial state.

Nondeterministic Choices

Nondeterministic computation can be thought of as a self-reproducing agent traveling in the state space.

1. At the start of computation the agent is in the initial state.

Nondeterministic Choices

2. Both before and after receiving an input symbol, the agent follows each ϵ-labeled outgoing edge by producing its own clone and sending it along the edge. Thus, the original remains in the current location.

Nondeterministic Choices

2. Both before and after receiving an input symbol, the agent follows each ϵ-labeled outgoing edge by producing its own clone and sending it along the edge. Thus, the original remains in the current location.

Nondeterministic Choices

3. On receiving an input symbol, say a,
   (a) If there is only one a-labeled outgoing edge, the agent follows the edge.
   (b) If there is no a-labeled outgoing edge, the agent evaporates.

Nondeterministic Choices

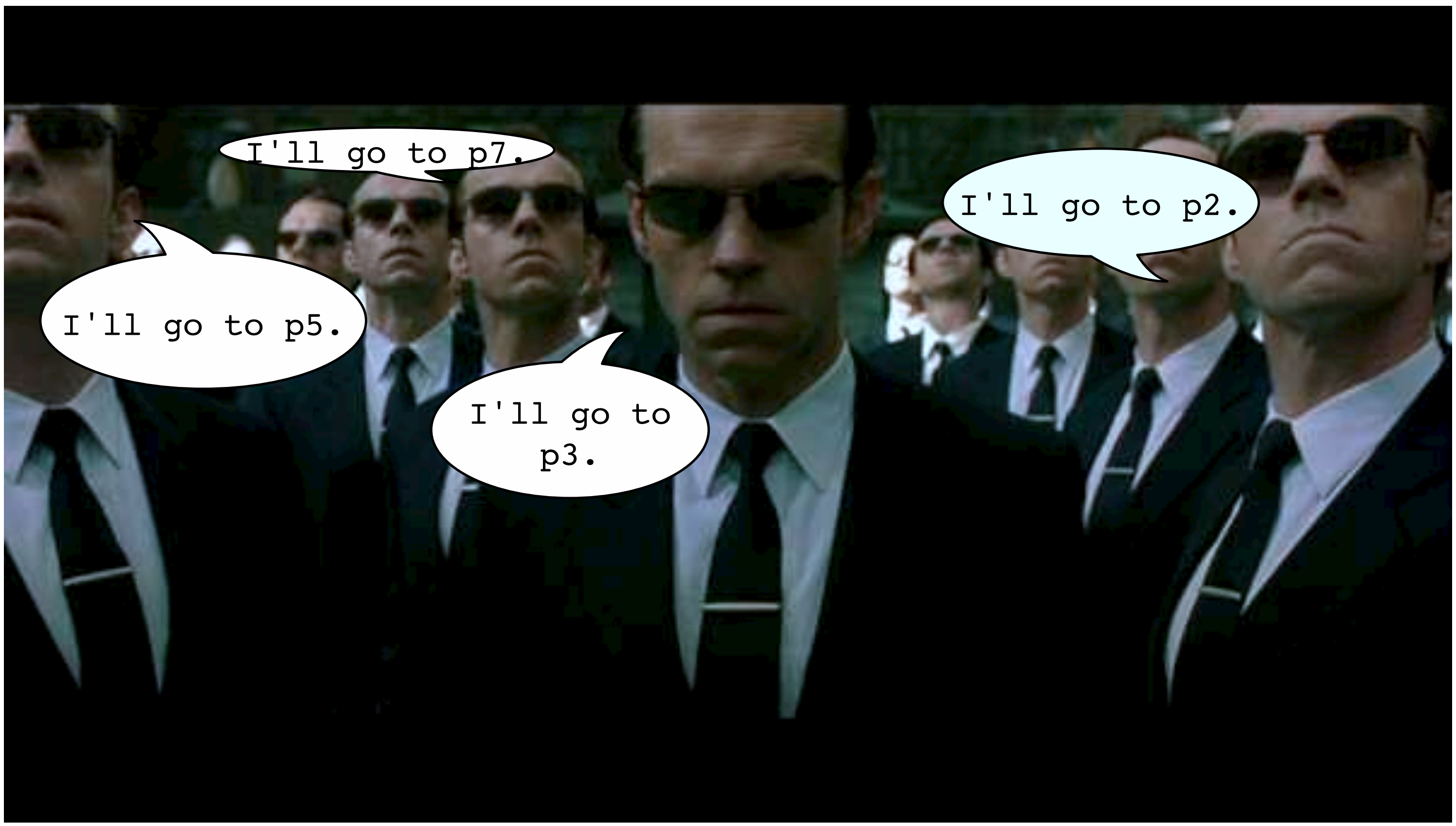
3. On receiving an input symbol, say a,
   (a) If there is only one a-labeled outgoing edge, the agent follows the edge.
   (b) If there is no a-labeled outgoing edge, the agent evaporates.

Nondeterministic Choices

3. On receiving an input symbol, say a,
   (c) If there are $k \geq 2$ a-labeled outgoing edges, the agent produces $k - 1$ clones, make them cross $k - 1$ of the edges, and cross the remaining one by himself.

Nondeterministic Choices

3. On receiving an input symbol, say a,
   (c) If there are $k \geq 2$ a-labeled outgoing edges, the agent produces $k-1$ clones, make them cross $k-1$ of the edges, and cross the remaining one by himself.



Nondeterministic Choices

4. When two agents collide in a state, they merge themselves into one.

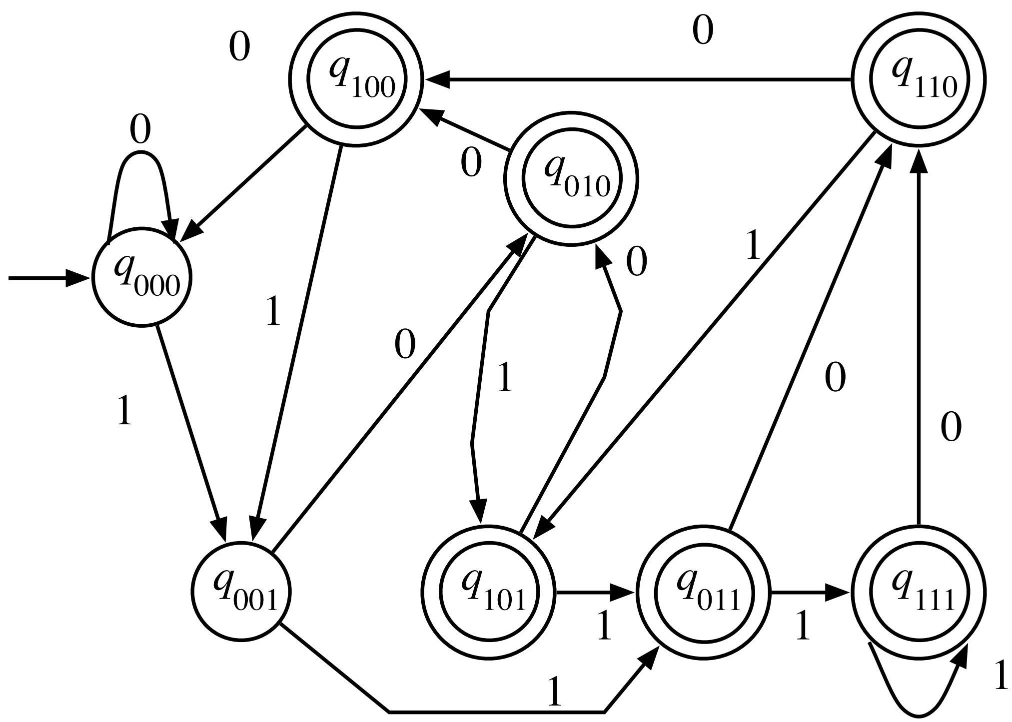
Nondeterministic Choices

4. When two agents collide in a state, they merge themselves into one.

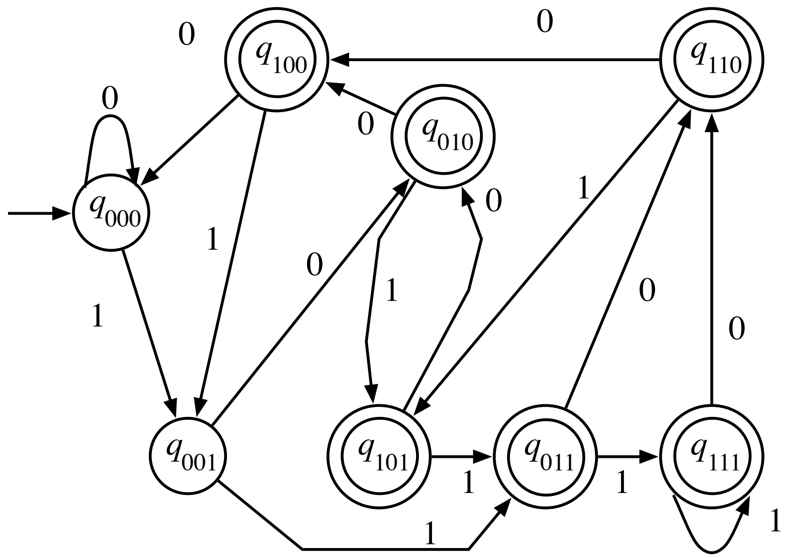
Why Use NFA?

For some languages construction is much easier.

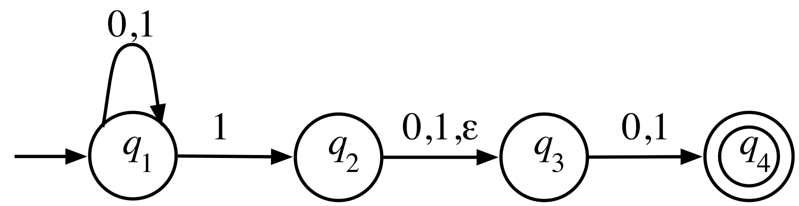
Below is a DFA that accepts the same language by remembering the last three symbols.



Comparison



DFA



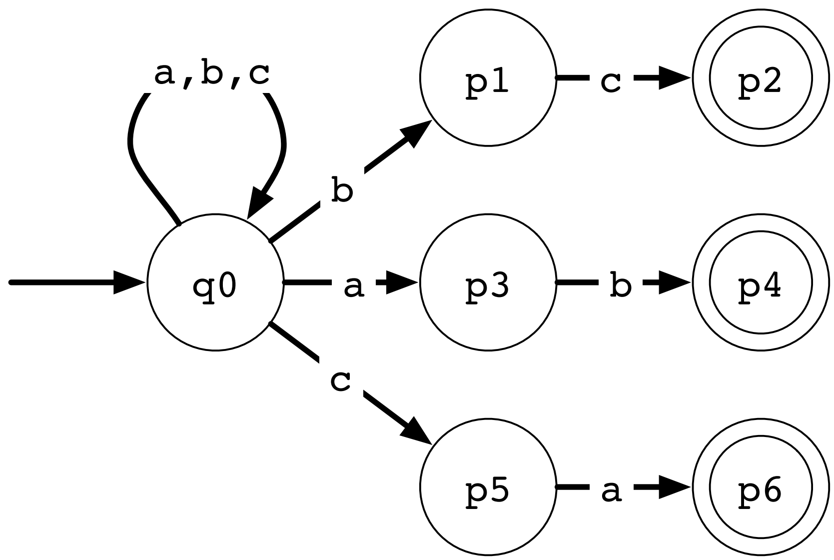
NFA

Example 1

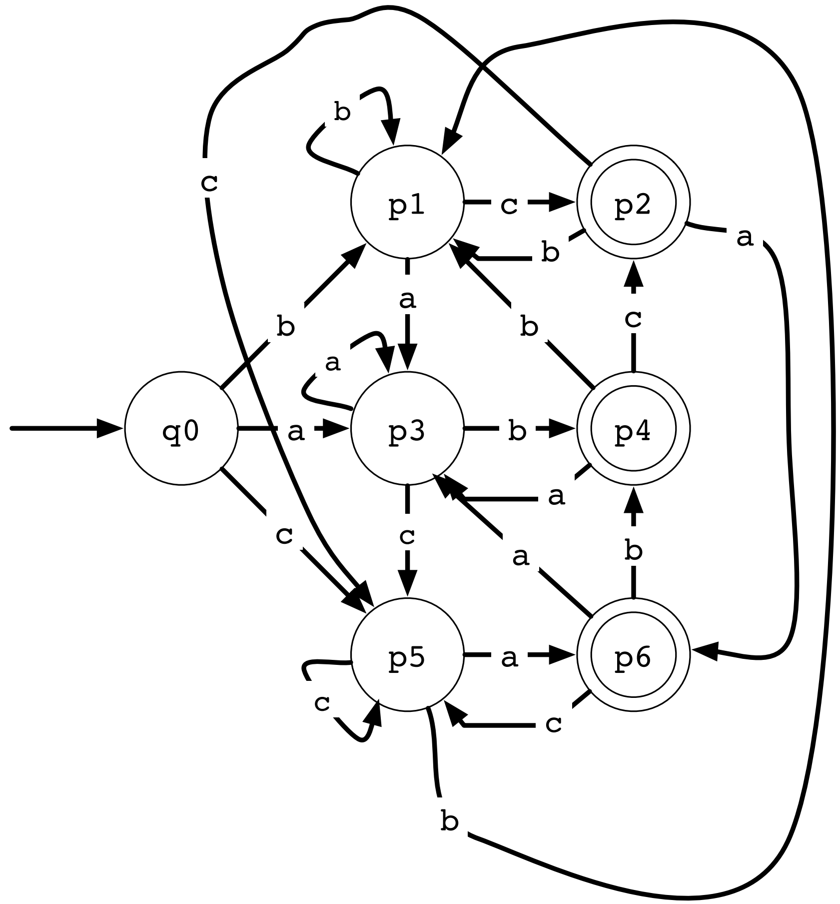
An NFA for the language of all strings over $\{a, b, c\}$ that end with one of ab, bc, and ca.

Example 1

An NFA for the language of all strings over $\{a, b, c\}$ that end with one of ab, bc, and ca.



A DFA Version of the Same Language

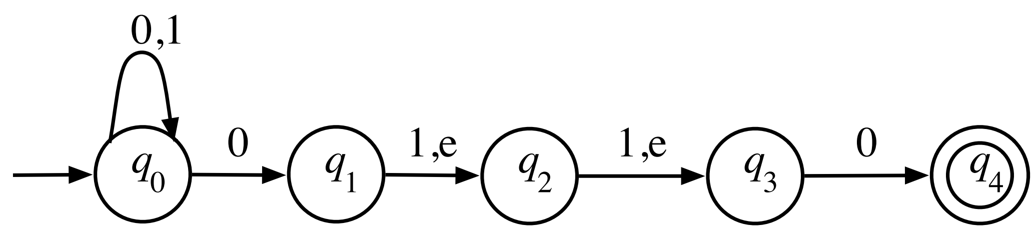


Example 2

An NFA for the language of all strings over $\{0, 1\}$ that end with one of 0110, 010, and 00.

Example 2

An NFA for the language of all strings over $\{0, 1\}$ that end with one of 0110, 010, and 00.

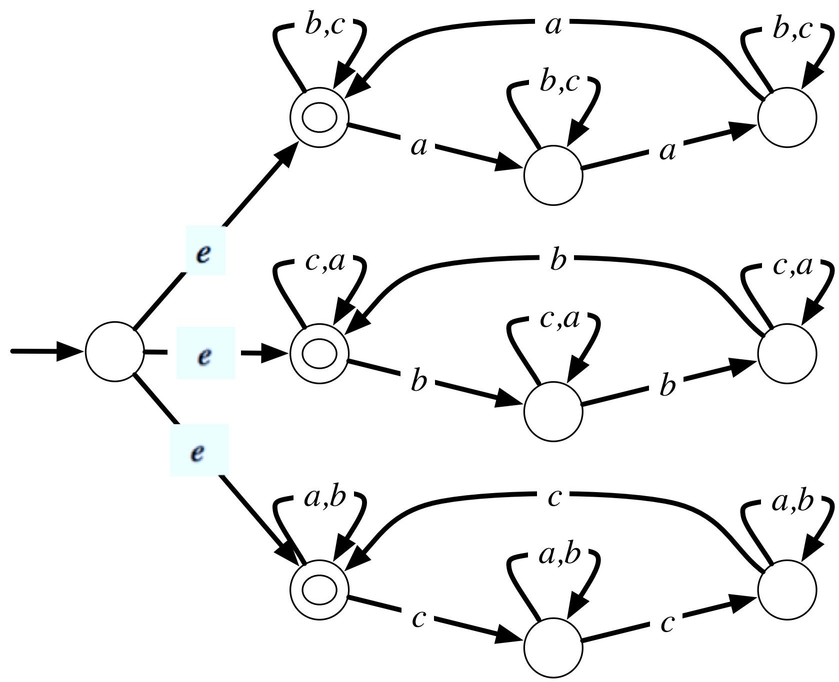


Example 3

An NFA for the language of all strings over $\{a, b, c\}$ for which one of (the number of occurrences of a), (the number of occurrences of b), and (the number of occurrences of c) is a multiple of 3.

Example 3

An NFA for the language of all strings over $\{a, b, c\}$ for which one of (the number of occurrences of a), (the number of occurrences of b), and (the number of occurrences of c) is a multiple of 3.

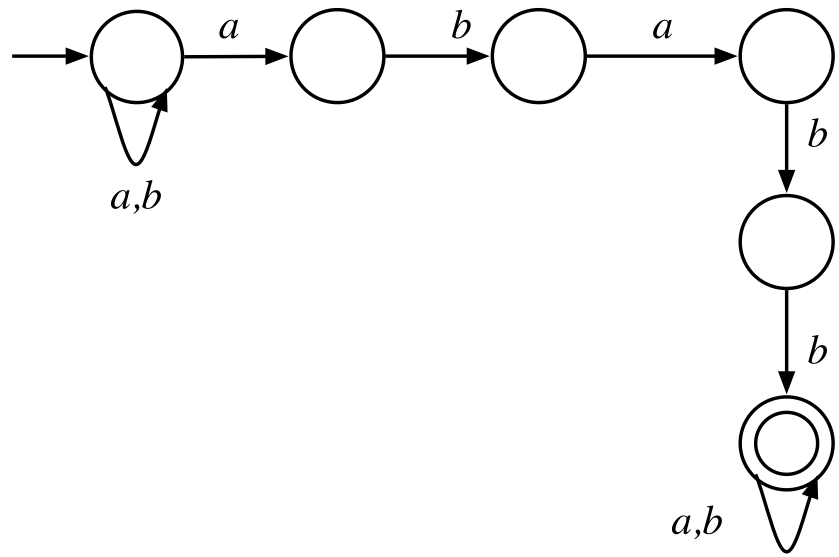


Example 4

An NFA for the language of all strings over $\{a, b\}$ that contain $ababb$.

Example 4

An NFA for the language of all strings over $\{a, b\}$ that contain $ababb$.



The FA Model Versus NFA Model

The NFA simplifies computational design, but the use of nondeterministic selections and ϵ-transitions makes it look very different from FA.

Is that really so?

The FA Model Versus NFA Model

The NFA simplifies computational design, but the use of nondeterministic selections and ϵ-transitions makes it look very different from FA.

Is that really so?

No, the FA model is equivalent to the NFA model. That is, every language accepted by an DFA is accepted by an NFA, vice versa.

Obviously, we have:

Theorem. **Every FA is already an NFA.**

We will show:

Theorem. **Every NFA can be converted to an equivalent FA.**

Proof

The principle is that just before the first symbol is received and after each symbol is read, there will be at most one agent in any state.

Proof

The principle is that just before the first symbol is received and after each symbol is read, there will be at most one agent in any state.

We will thus consider the **"set of all states an agent can be in."**

Proof (cont'd)

Let $N = (Q, \Sigma, \delta, q_0, F)$ be an NFA.

We will construct a DFA $M = (S, \Sigma, \gamma, s_0, G)$.

- The state set S is $\mathcal{P}(Q)$.
- The initial state s_0 is the set consisting of q_0 and all the states reachable from q_0 by following only ϵ transitions.
- The final state set G is $\{A \in S \mid A \cap F \neq \emptyset\}$; i.e., the set of all subsets of Q containing an element of F.

Transition Function

The transition function γ is defined as follows:

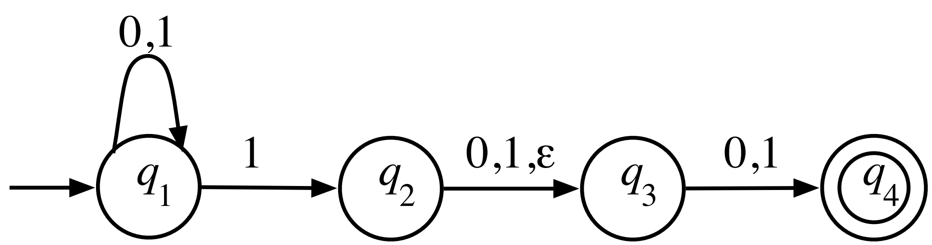
For each $A \in S$ and for each $b \in \Sigma$, $\gamma(A, b) = \bigcup_{p \in A} \delta(p, \epsilon^* b \epsilon^*)$, the collection of all states r that can be reached from a state p in A by following

- any number of ϵ-arrows,
- a b-arrow, and then
- any number of ϵ-arrows.

For all w over Σ, w is accepted by N if and only if the new DFA transitions from s_0 to a state in G on input w. ∎

Example

An NFA that recognizes the language consisting of all strings over $\{0, 1\}$ that contain a 1 at either the third to last position or the second to last position.



Conversion to DFA

The state set consists of: $\emptyset$, $\{q_1\}$, $\{q_2\}$, $\{q_3\}$, $\{q_4\}$, $\{q_1, q_2\}$, $\{q_1, q_3\}$, $\{q_1, q_4\}$, $\{q_2, q_3\}$, $\{q_2, q_4\}$, $\{q_3, q_4\}$, $\{q_1, q_2, q_3\}$, $\{q_1, q_2, q_4\}$, $\{q_1, q_3, q_4\}$, $\{q_2, q_3, q_4\}$, $\{q_1, q_2, q_3, q_4\}$.

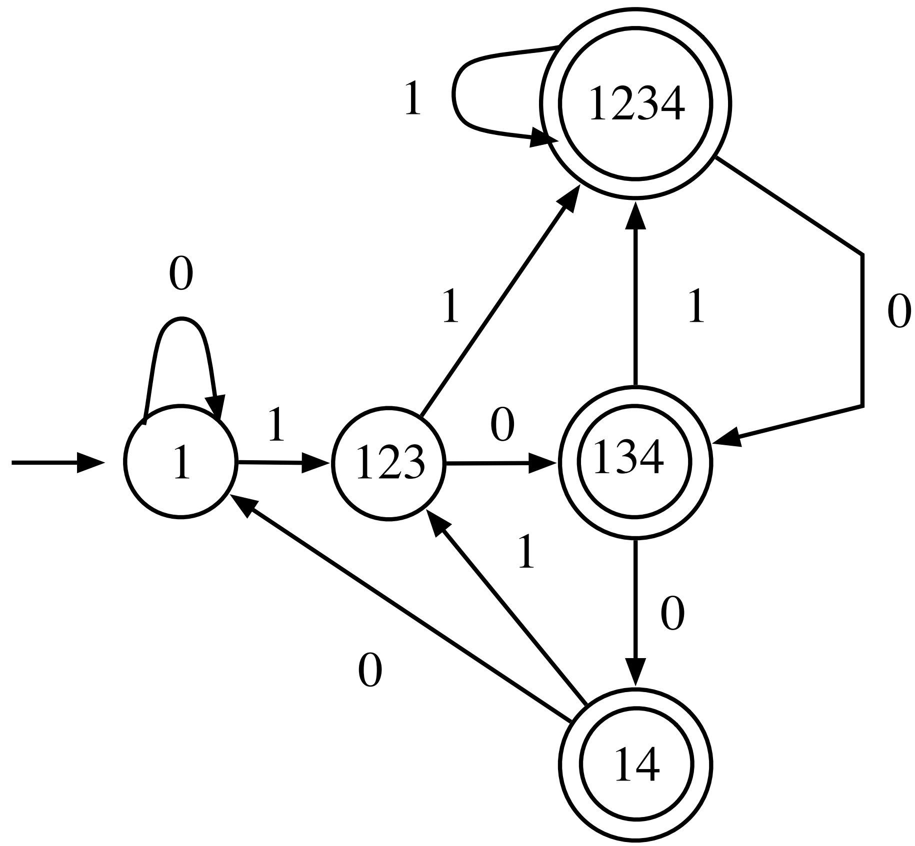
F consists of: $\{q_4\}, \{q_1, q_4\}, \{q_2, q_4\}, \{q_3, q_4\}, \{q_1, q_2, q_4\}, \{q_1, q_3, q_4\}, \{q_2, q_3, q_4\}, \{q_1, q_2, q_3, q_4\}$.

The initial state is $\{q_1\}$.

Transition

State	0	1
$\{q_1\}$	$\{q_1\}$	$\{q_1, q_2, q_3\}$
$\{q_1, q_2, q_3\}$	$\{q_1, q_3, q_4\}$	$\{q_1, q_2, q_3, q_4\}$
$\{q_1, q_3, q_4\}$	$\{q_1, q_4\}$	$\{q_1, q_2, q_3, q_4\}$
$\{q_1, q_4\}$	$\{q_1\}$	$\{q_1, q_2, q_3\}$
$\{q_1, q_2, q_3, q_4\}$	$\{q_1, q_3, q_4\}$	$\{q_1, q_2, q_3, q_4\}$

The other states are unreachable from the initial state.


1234
1
0
1
1
0
1
0
123
134
1
1
0
0
14

A Greedy Conversion Algorithm

Step 1 For each state $p \in Q$ and for each symbol $a \in \Sigma$, compute the set $R(p, a)$ of all states that can be reached from state p by:

(a) any number of ϵ transitions,

(b) one transition labeled by a, and then

(c) any number of ϵ transitions.

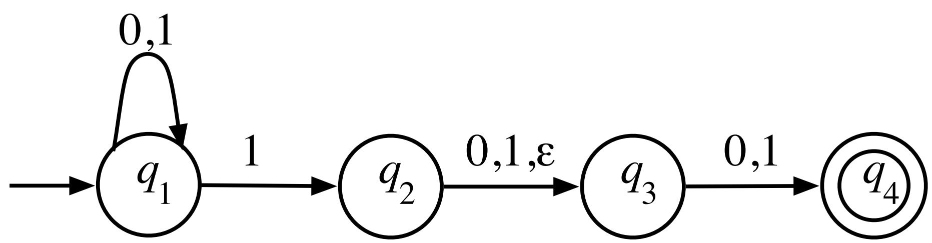
Step 2 Initialize the state collection S as $\{p_0\}$, where p_0 is the set of all states that can be reached from q_0 by following any number of ϵ transitions.

Step 3 While S contains a state with no outgoing edges, select an arbitrary member, say r, of S, and do the following:

- For each symbol a, compute the state r_a as $\cup_{q \in r} R(q, a)$, add r_a to S if r_a is not already in it, and then draw an arc from r to r_a.

Algorithm Execution Example

Use the previous NFA.



Step 1

state	0	1
q_1	q_1	q_1, q_2, q_3
q_2	q_3, q_4	q_3, q_4
q_3	q_4	q_4
q_4	$\emptyset$	$\emptyset$

Step 2

Initially $S = \{\{q_1\}\}$.

Step 3

$\{q_1\}$ on 0 changes the state to $\{q_1\}$
$\{q_1\}$ on 1 changes the state to $\{q_1, q_2, q_3\}$. New!
$\{q_1, q_2, q_3\}$ on 0 changes the state to $\{q_1, q_3, q_4\}$. New!
$\{q_1, q_2, q_3\}$ on 1 changes the state to $\{q_1, q_2, q_3, q_4\}$. New!
$\{q_1, q_3, q_4\}$ on 0 changes the state to $\{q_1, q_4\}$. New!
$\{q_1, q_2, q_3\}$ on 1 changes the state to $\{q_1, q_2, q_3, q_4\}$.
$\{q_1, q_2, q_3, q_4\}$ on 0 changes the state to $\{q_1, q_3, q_4\}$.
$\{q_1, q_2, q_3, q_4\}$ on 1 changes the state to $\{q_1, q_2, q_3, q_4\}$.
$\{q_1, q_4\}$ on 0 changes the state to $\{q_1 3\}$.
$\{q_1, q_4\}$ on 1 changes the state to $\{q_1, q_2, q_3\}$.