Intro to Theory of Computation

LECTURE 3

Theory of Computation

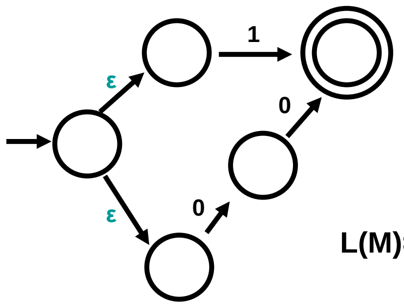
- Equivalence of NFAs and DFAs
- Closure Properties of Regular Languages

Sofya Raskhodnikova

Nondeterminism

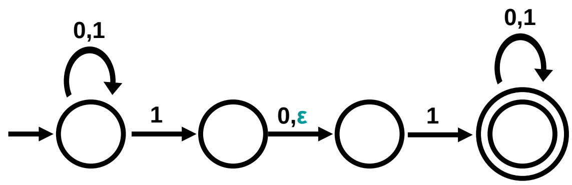
Nondeterministic Finite Automaton (NFA) accepts if there is a way to make it reach an accept state

Example



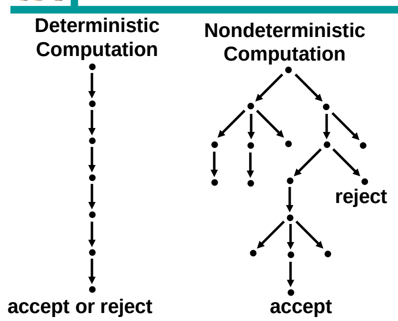
L(M)={1,00}

Example



L(M)={w | w contains 101 or 11}

Nondeterminism



Ways to think about nondeterminism

- **parallel computation**
- **tree of possible computations**
- **guessing the "right" choice**

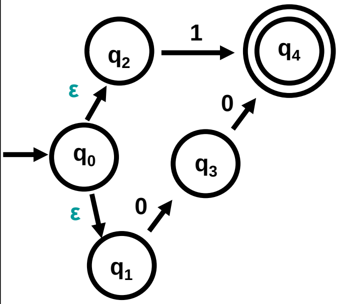
Formal Definition

An ***NFA*** is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$

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

$\mathcal{P}(Q)$ is the set of subsets of Q and $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

Example



$N = (Q, \Sigma, \delta, Q_0, F)$

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

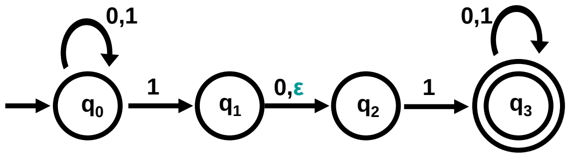
$\Sigma = \{0,1\}$

$F = \{q_4\}$

$\delta(q_2,1) = \{q_4\}$

$\delta(q_3,1) = \varnothing$

Example



$N = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1, q_2, q_3\}$

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

$F = \{q_3\}$

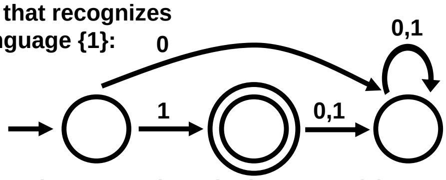
$\delta(q_0,1) = \{q_0, q_1\}$

$\delta(q_1,\varepsilon) = \{q_1,q_2\}$

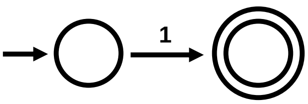
$\delta(q_2,0) = \varnothing$

NFAs ARE SIMPLER THAN DFAs

A DFA that recognizes the language {1}:



An NFA that recognizes the language {1}:



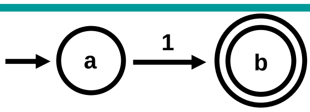
Equivalence of NFAs & DFAs

Theorem. Every NFA has an equivalent DFA

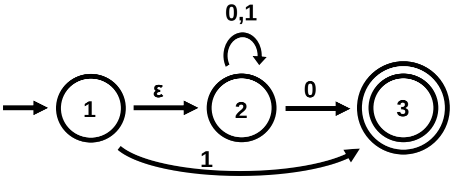
Corollary: A language is regular iff it is recognized by an NFA

Examples: NFA to DFA

1)



2)



Regular Operations on languages

Complement: $\neg A = \{ w \mid w \notin A \}$

Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$

Reverse: $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$

Concatenation: $A \circ B = \{ vw \mid v \in A \text{ and } w \in B \}$

Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$

Closure properties of the class of regular languages

THEOREM. The class of regular languages is **closed** under all 6 operations.

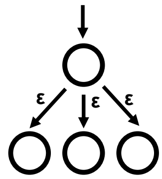
If A and B are regular, applying any of these operation yields a regular language.

Closure under reverse

Theorem. The reverse of a regular language is also regular

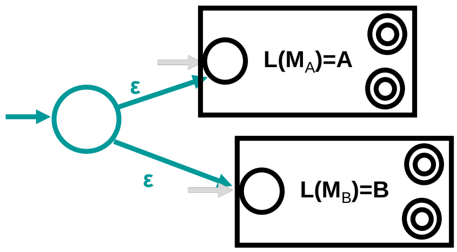
Proof: Let L be a regular language and M be a DFA that recognizes it. Construct a NFA M^R recognizing L^R:

- Define M^R as M with the arrows reversed
- Make the start state of M be the accept state in M^R
- Make a new start state that goes to all accept states of M by ε-transitions.



New construction for A ∪ B

Construct an NFA M:

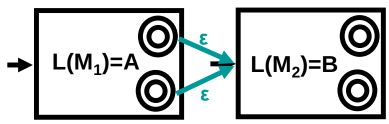


Concatenation operation

Concatenation: $A \circ B = \{ vw \mid v \in A \text{ and } w \in B \}$

Theorem. If A and B are regular, $A \circ B$ is also regular

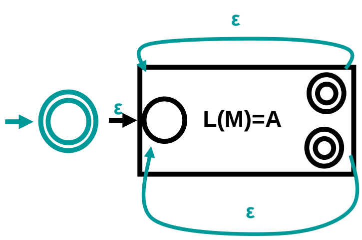
Proof: Given DFAs M_1 and M_2, construct NFA by connecting all accept states in M_1 to start states in M_2



Star operation

Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$

Theorem. If A is regular, A* is also regular.



The class of regular languages is closed under

Regular operations

Union: $A \cup B = \{ w \mid w \in A \text{ or } w \in B \}$

Concatenation: $A \circ B = \{ vw \mid v \in A \text{ and } w \in B \}$

Star: $A^* = \{ w_1 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A \}$

Other operations

Complement: $\neg A = \{ w \mid w \notin A \}$

Intersection: $A \cap B = \{ w \mid w \in A \text{ and } w \in B \}$

Reverse: $A^R = \{ w_1 \ldots w_k \mid w_k \ldots w_1 \in A \}$