Intro to Theory of Computation

LECTURE 2

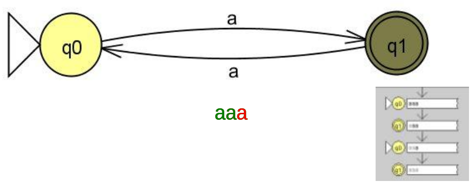
Theory of Computation

- Finite Automata
- Operations on languages
- Nondeterminism

Homework 0 due
Homework 1 handed out

Sofya Raskhodnikova

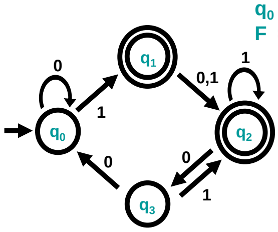
JFLAP http://www.cs.duke.edu/csed/jflap/



Formal definition of FA

$M = (Q, \Sigma, \delta, q_0, F)$ where

$Q = \{q_0, q_1, q_2, q_3\}$
$\Sigma = \{0,1\}$
$\delta : Q \times \Sigma \rightarrow Q$ transition function*
$q_0 \in Q$ is start state
$F = \{q_1, q_2\} \subseteq Q$ accept states



* δ	0	1
q_0		
q_1		
q_2		
q_3		

Language of FA

The **language of M**, **L(M),** is the set of strings that M accepts

A language is regular if it is recognized by a finite automaton

L = { w | w contains 001} is regular

L = { w | w has an even number of 1s} is regular

Regular languages

Many interesting programs accept regular languages

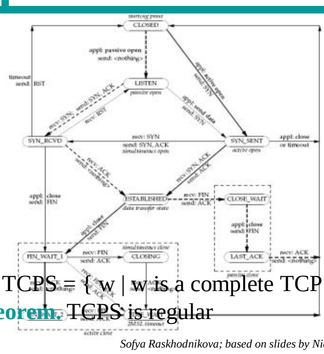
NETWORK PROTOCOLS

COMPILERS

GENETIC TESTING

ARITHMETIC

INTERNET TRANSMISSION CONTROL PROTOCOL



Let TCPS = { w | w is a complete TCP Session}
Theorem. TCPS is regular

CS 464 Compilers

COMMENTS :
Are delimited by /* */
Cannot have NESTED /* */
Must be CLOSED by */
*/ is ILLEGAL outside a comment

COMMENTS = {strings with legal comments}

Theorem. **COMMENTS** is regular.

CS 464 Genetic testing

DNA SEQUENCES are strings over the alphabet {A,C,G,T}

GENES are special substrings

A GENETIC TEST searches a DNA SEQUENCE for a GENE

Theorem. Every GENETIC TEST is regular.

CS 464 Arithmetic

LET $\Sigma_3 = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \begin{bmatrix}1\\0\\0\end{bmatrix}, \begin{bmatrix}1\\0\\1\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}$

- A string over Σ_3 has three ROWS
- Each ROW $b_0b_1b_2\dots b_N$ represents the integer $b_0 + 2b_1 + \dots + 2^N b_N$.
- Let ADD = {S | $\text{ROW}_1 + \text{ROW}_2 = \text{ROW}_3$ }

Theorem. ADD is regular.

CS 464 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: $A^R = \{ w_1 \dots w_k \mid w_k \dots w_1 \in A \}$

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

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

CS 464 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.

CS 464 Closure properties

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

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

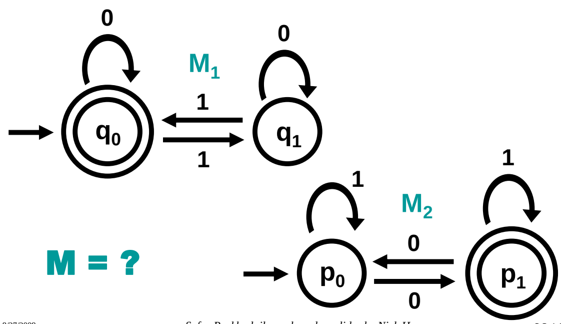
Closure under union

Theorem. The union of two regular languages is also a regular language

Proof: Let
$M_1 = (Q_1, \Sigma, \delta_1, q_0^1, F_1)$ be finite automaton for L_1
and
$M_2 = (Q_2, \Sigma, \delta_2, q_0^2, F_2)$ be finite automaton for L_2

Construct a finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes $L = L_1 \cup L_2$

Example



Proof (continued)

Idea: Run both M_1 and M_2 at the same time!

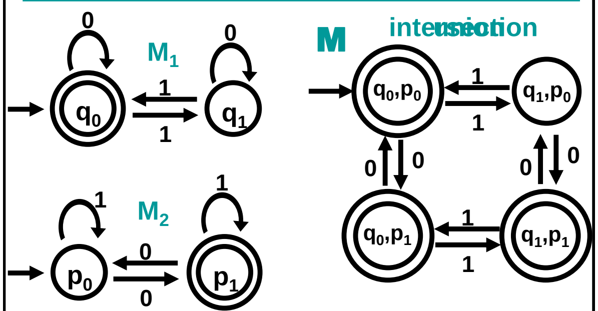
Q = pairs of states, one from M_1 and one from M_2
$= \{ (q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2 \}$
$= Q_1 \times Q_2$

$q_0 = (q_0^1, q_0^2)$

$F = \{ (q_1, q_2) \mid q_1 \in F_1 \text{ or } q_2 \in F_2 \}$

$\delta((q_1,q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$

Example (continued)



Closure properties

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

Closure under reverse

Theorem. The reverse of a regular language is also regular language

Proof: Let L be a regular language and M be a finite automaton that recognizes it.

Construct a finite automaton M^R recognizing L^R

Idea: Define M^R as M with the arrows reversed
Swap start and accept states

CS 464

Closure under reverse

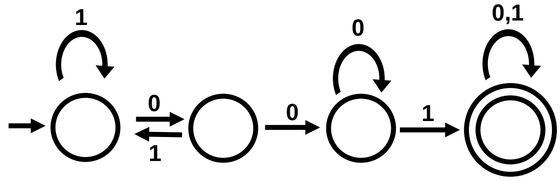
M^R IS NOT ALWAYS A DFA!

It may have many start states

Some states may have too many outgoing edges, or none

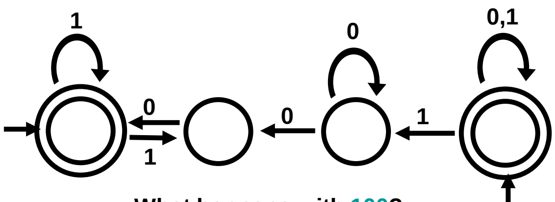
CS 464

Example



CS 464

NONDETERMINISM

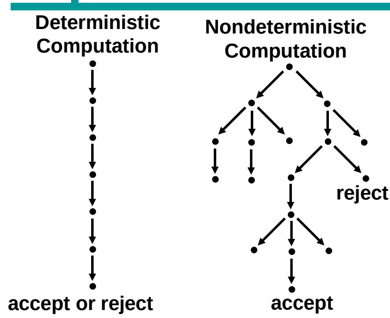


What happens with 100?

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

CS 464

Nondeterminism

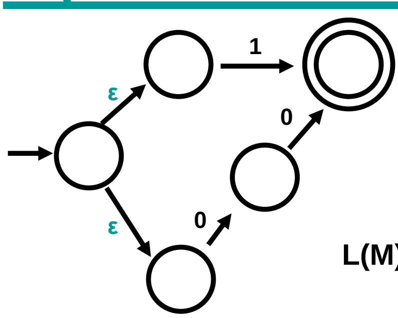


Ways to think about nondeterminism

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

CS 464

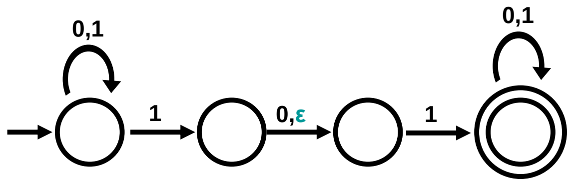
Example



L(M)={1,00}

CS 464

Example



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

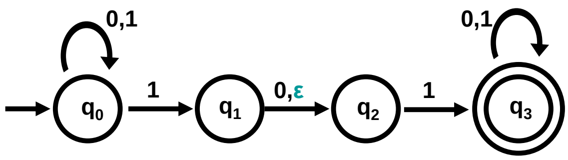
Formal Definition

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



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

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

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

$F = \{q_3\}$

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

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

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