

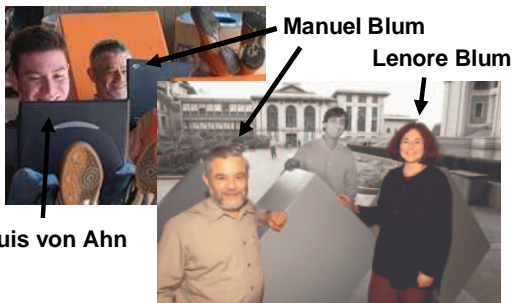
15-453

**FORMAL LANGUAGES,
AUTOMATA AND
COMPUTABILITY**

Suppose $A \subseteq \{1, 2, \dots, 2n\}$ with $|A| = n+1$

TRUE or FALSE: There are always two numbers in A such that one divides the other

INSTRUCTORS



This class is about mathematical models of computation

WHY SHOULD I CARE?
WAYS OF THINKING

WHY SHOULD I CARE?
THEORY CAN DRIVE PRACTICE

Mathematical models of computation
predated computers as we know them

WHY SHOULD I CARE?
THIS STUFF IS USEFUL

PART 1

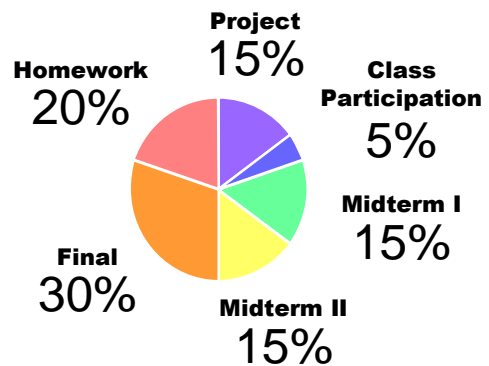
Automata and Languages

PART 2

Computability Theory

PART 3

Complexity and Applications



HOMEWORK

Homework will be assigned every Thursday and will be due one week after at the beginning of class

You must list your collaborators and references in every homework assignment

COURSE PROJECT

Choose a (unique) topic

Learn about your topic

Actually write progress reports

Meet with an instructor once a month

Prepare a 10-minute presentation

WWW.FLAC.WS

Suppose $A \subseteq \{1, 2, \dots, 2n\}$ with $|A| = n+1$

TRUE or FALSE: There are always two numbers in A such that one divides the other

HINT:

HINT 2:

This class will emphasize **PROOFS**

A good proof should be:

Easy to understand

Correct

We expect your proofs to have three levels:

The first level should be a one-word or one-phrase "hint" of the proof

(e.g. "Proof by contradiction,"
"Follows from the pigeonhole principle")

The second level should be a short one-paragraph description or "longer hint"

The third level should be the full proof

DOUBLE STANDARDS?

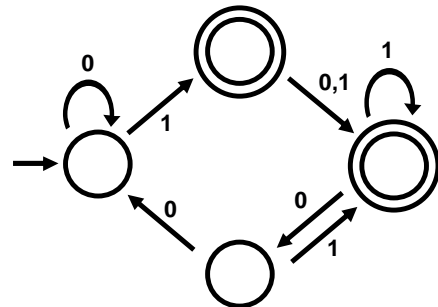
During the lectures, my proofs will usually only contain the first two levels and maybe part of the third

Reasons:

I have to keep you guys awake

I know my proofs are correct

DETERMINISTIC FINITE AUTOMATA



The machine accepts a string if the process ends in a double circle

NOTATION

An alphabet Σ is a finite set (e.g., $\Sigma = \{0,1\}$)

A string over Σ is a finite-length sequence of elements of Σ

For x a string, $|x|$ is the length of x

The unique string of length 0 will be denoted by ϵ and will be called the empty or null string

A language over Σ is a set of strings over Σ

A finite automaton is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$

Q is the set of states

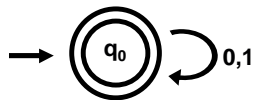
Σ is the alphabet

$\delta : Q \times \Sigma \rightarrow Q$ is the transition function

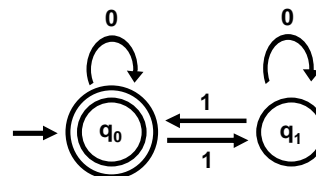
$q_0 \in Q$ is the start state

$F \subseteq Q$ is the set of accept states

$L(M)$ = the language of machine M
= set of all strings machine M accepts



$L(M) =$



$L(M) =$

Build an automaton that accepts all and only those strings that contain 001



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

$L = \{ w \mid w \text{ contains } 001 \}$ is regular

$L = \{ w \mid w \text{ has an even number of } 1\text{s} \}$ is regular

UNION THEOREM

Given two languages, L_1 and L_2 , define the union of L_1 and L_2 as

$$L_1 \cup L_2 = \{ w \mid w \in L_1 \text{ or } w \in L_2 \}$$

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

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

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

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

$$= \{ (q_1, q_2) \mid q_1 \in Q_1 \text{ and } q_2 \in Q_2 \}$$

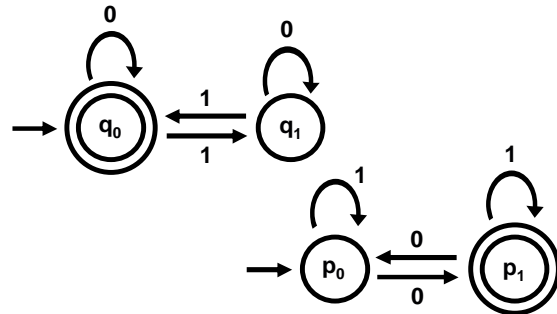
$$= Q_1 \times Q_2$$

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

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

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

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



WWW.FLAC.WS

Read Chapters 0 and 1.1 of the book for next time