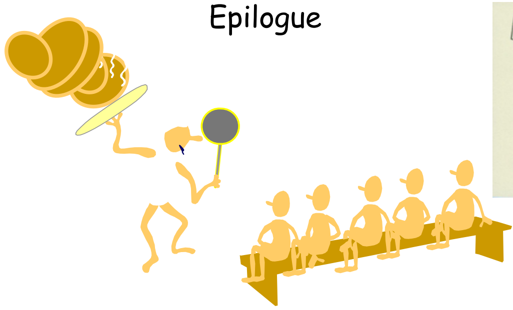
Great Theoretical Ides in Computer Science

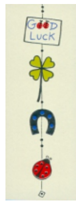
V. Adamchik

CS 15-251 Fall 2014

Carnegie Mellon University

Epilogue





We Had Some Lectures

1. Pancakes
2. Inductive Reasoning
3. Logic — 21-300
4. Proofs — 80-310
5. Counting I
6. Counting II — 21-301
7. Generating Functions
8. Games — 15-859
9. Probability I — 15-359
10. Probability II — 21-325
11. Graphs I
12. Graphs II — 21-484
13. Graphs III
14. Groups
15. Fields, Polynomials — 21-373
17. Error Correction — 21-441
18. Cryptography — 15-503
19. Finite State Automata — 15-452
20. Cantor's Legacy — 15-354
21. Turing's Legacy
22. Gödel's Legacy — 80-311
23. Efficient Reductions — 15-451
24. P vs NP — 15-455
25. Approximation Algorithms
26. Interactive Proofs
27. Epilogue

#1. Pancakes

Sorting by Prefix Reversal

Pancake Number P_n

Breaking-apart $\le P_n \le$ Bring-to-top

Pancake Network

Genome rearrangements

#2. Induction

Standard Form

$[P(0) \wedge \forall k\, (P(k) \Rightarrow P(k+1))] \longrightarrow \forall n\, P(n)$

Strong Form

$[P(0) \wedge (P(0),P(1),\ldots,P(k) \Rightarrow P(k+1))] \longrightarrow \forall n\, P(n)$

Least Element Principal

Structural Induction

#3-4. Axiomatic Systems

Given the axioms and deduction rules.

Goal: to prove other properties, e.g. theorems.

Consistency: we do not want a set of axioms to lead to contradictions.

Soundness: every statement that's proven is true.

Completeness: every true statement can be proven.

Gödel: first and second incompleteness theorems.

#3-4. Propositional Logic

An axiomatic system that is composed of symbols, logical operators ($\wedge,\vee,\neg,\rightarrow$), and parenthesis.

A formula is a propositional formula (aka "well-formed" formula) if and only if it is a 'theorem' in this system.

Propositional Logic is sound and complete

#3-4. First-Order Logic

distinguished from propositional logic by its use of quantifiers ($\forall$, $\exists$) and logical functions.

This logic formalizes
Zermelo-Fraenkel Set theory
and
Peano arithmetic.

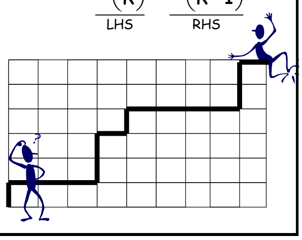
#5-6. Counting

1. The Principle of Inclusion-Exclusion
2. Counting by use of bijection
3. Choice Trees
4. The Pigeonhole Principle
5. The Binomial Theorem $(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$
6. Pirates and Gold
7. Diophantine Equations

#5-6. Counting

1. Pascal's Triangle $\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$
2. Combinatorial Proofs $\underbrace{k\binom{n}{k}}_{\text{LHS}} = \underbrace{n\binom{n-1}{k-1}}_{\text{RHS}}$
3. Manhattan Walk
4. Catalan Numbers

$\frac{1}{n+1}\binom{2n}{n}$



#7. Generating Functions

Given a sequence of numbers

$a_0, a_1, a_2, \dots, a_n, \dots$

The generating function for that sequence is a power series, where x is just a placeholder.

$$\sum_{k=0}^{\infty} a_k x^k$$

closed form

$$1,1,2,3,5,8,\dots \to \sum_{k=0}^{\infty} F_k x^k = \frac{x}{1-x-x^2}$$

#8. Mathematical Games

Two players alternate moves
Rules specify moves from one position to another
A terminal position is one for which there are no moves
No draws, no randomness
The last player to move wins

P-position (win for previous) vs N-position

The game of Nim

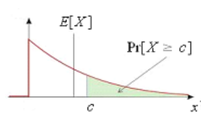
Nim-Sum: addition in base 2 without carry.

Nimber Theorem. N(G)=0 iff P-position

The Game of Dawson's Kayles

#9-10. Discrete Probability

1. Random variable
2. Conditional Probability
3. Law of Total Probability
4. Geometric and Binomial Distributions
5. Expectation $E[X+Y]=E[X]+Y[Y]$
6. Conditional Expectation
7. Tail bounds
8. The Probabilistic Method



#11-13. Graphs

Graph Isomorphism: a vertex bijection that preserves adjacency and non-adjacency structures

Cayley's Formula: the number of labeled trees on n nodes is n^{n-2}. It counts spanning trees in K_n

Prüfer Encoding: a bijection between trees and sequences.

Euler's Formula for Planar Graphs: V – E + F = 2

K_5 and $K_{3,3}$ are not planar

Coloring (planar) graphs.

#11-13. Graphs

Bipartite Matching: Hall's theorem, Hungarian algorithm.

Stable Matching on bipartite graphs $K_{n,n}$.
Basic principle: man proposes, woman disposes

Gale-Shapely Theorem: stable matching is always possible.

Euler and Hamiltonian cycles

#14. Groups

A group G is a pair (S, •), where S is a set and • is a binary operation $S \times S \to S$ such that:

1. (Closure) For all a and b $\in$ S, a • b$\in$S
2. • is associative, (a • b) • c= a • (b • c)
3. (Identity) $\exists$ e $\in$ S s.t. e • a = a • e = a, $\forall$a $\in$ S
4. (Inverses) $\forall$a $\in$ S, $\exists$ b $\in$ S s.t. a • b = b • a = e

$(Z_n, +)$ is a group, $\{Z_n, \times)\backslash\{0\}$ is not a group

$(Z_n^*, \times) = \{x \in Z_n | \gcd(x,n) = 1\}$

Euler Phi function $\phi(n)$ is a size of Z_n^*.

Lagrange's theorem: If G is a finite group, and H is a subgroup then the order of H divides the order of G.

#15. Fields, Polynomials

A field F is a set together with two binary operations + and ×, satisfying

1. (F,+) is a commutative group
2. (F\{0},×) is a commutative group
3. The distributive law (a + b) × c = (a × c) + (b × c)

Infinite fields: $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$. Finite fields: $\mathbb{F}_p$ mod prime p.

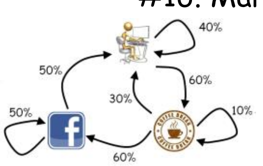
Polynomials with coefficients from a field. Not a field itself.

Over a field, degree-d polynomials have <u>at most</u> d roots.

Lagrange interpolation: restores a unique polynomial of degree at most d from d+1 points.

Vandermonde matrix

#16. Markov Chain



$$M = \begin{array}{c} \\ W \\ C \\ F \end{array} \begin{array}{c} \begin{array}{ccc} W & C & F \end{array} \\ \begin{pmatrix} .4 & .6 & 0 \\ .3 & .1 & .6 \\ .5 & 0 & .5 \end{pmatrix} \end{array}$$

Transition matrix

In this system, the next state depends only on my status at previous state.

Starting at state X at what state will I be after N steps?

Stationary (invariant) distribution: $\pi = \pi M$

Application: random walk on undirected graphs.

#17. Error Correction

Sending a message via noisy channel. The channel may corrupt up to k symbols. How to correct errors?

Reed–Solomon encoding: we represent (d+1) input symbols as the coefficients of a degree d polynomial P(x), and send values P(1), P(2), …, P(d+k+1). If there are up to k erasures, we can restore that polynomial via Lagrange interpolation.

In general case we do not know where the errors are.

We send d+2k+1 values, and find the error locator polynomial by solving a linear system of equations.

#18. Cryptography

Diffie-Hellman Key Exchange, it requires both parties to exchange information to share a secret.

RSA: based on public and private keys.

Pick secret, random large primes: p,q
Multiply n = p*q, "Publish": n
$\phi(n) = (p-1)*(q-1)$
Pick random $e \in Z^*_{\phi(n)}$, "Publish": e
Compute d = inverse of e in $Z^*_{\phi(n)}$
Hence, $e*d = 1$ [mod $\phi(n)$]
"Private Key": d

Encode: $m = input^e (\mathrm{mod}\ n)$

Decode: $m^d (\mathrm{mod}\ n)$

#19. Finite Automata

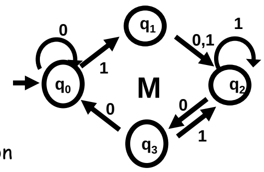
$Q = \{q_0, q_1, q_2, q_3\}$

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

$q_0 \in Q$ is start state

$F = \{q_1, q_2\} \subseteq Q$ accept states

$\delta : Q \times \Sigma \to Q$ transition function



A language $L \subseteq \Sigma$ is regular if there is a DFA which decides it.

Theorem: $L = \{0^n 1^n : n \in \mathbb{N}\}$ is not regular

An axiomatic system for regular languages, the Kleene theorem

Theorem (Rabin, Scott).

For every NFA there is an equivalent DFA.

#20. Cantor's Legacy

Sets A and B have the same 'cardinality' (size), written $|A| = |B|$, iff there exists a bijection between them.

$\mathbb{N}$ and $\mathbb{Q}$ have the same cardinality, $|\mathbb{N}| = |\mathbb{Q}| = \aleph_0$

$\mathbb{R}$ is uncountable, diagonalization argument.

Continuum Hypothesis:
there is NO a set S with $|\mathbb{N}| < S < |\mathbb{R}|$

Cantor Theorem: There is no a bijection from S onto its power set P(S), and $|S| < |P(S)|$.

The cardinal numbers $\aleph_0 < \aleph_1 < \aleph_2 < \ldots$

Cantor Sets – tiny measure zero sets.

#21. Turing's Legacy

Describes a new model of computation

Turing Machine is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

A language $L \subseteq \Sigma^*$ is decidable if there is a Turing Machine which:

1. Halts on every input $x \in \Sigma^*$.
2. Accepts inputs $x \in L$ and rejects inputs $x \notin L$.

Church-Turing Thesis: Any natural / reasonable notion of computation can be simulated by a TM.

Turing theorem: There is no program to solve the halting problem

#22. Gödel's Legacy

Gödel's Completeness Theorem: there is a (computable) axiomatic system, so a TM can "check" if a proof is a correct

Examples: Peano arithmetic (PA).

Intuition, Turing's halting problem suggests that there are true statements that cannot be proven in PA.

First Incompleteness Theorem: any mathematical proof system which has computable axioms cannot be both complete and sound.

Second Incompleteness Theorem: any mathematical proof system which has computable axioms cannot be both complete and consistent.

#23-24. P vs. NP

P : a set of all languages L s.t. $\exists$ a polynomial time algorithm that decides on L

NP : a set of all languages L s.t. $\exists$ a polynomial time algorithm that verify $x \in L$.

Mapping reduction $A \leq_p B$:

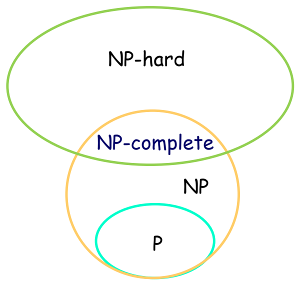
1. f is a polynomial time computable
2. $x \in A$ if and only if $f(x) \in B$.

NP-hard = $\{ L \subseteq \{0,1\}^* \mid \forall X \in NP \text{ and } X \leq_p L \}$

NP-complete iff
1) $L \subseteq NP$
2) $L \subseteq$ NP-hard

Cook-Levin Theorem:
SAT is NP-complete

#23-24. P vs. NP



#23-24. P vs. NP

A recipe for proving any $L \in$ NP-complete:

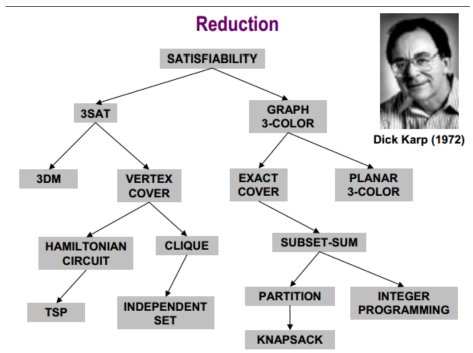
1) Prove $L \in$ NP

2) Choose $A \in$ NPC and reduce it to L

2.1) Describe mapping $f: A \to L$

2.2) Prove $x \in A$ iff $f(x) \in L$

2.3) Prove f is polynomial



All of these problems poly-reduce to one another!

#25. Approximation Algorithms

Suppose we are given an NP-complete problem to solve. Can we develop polynomial-time algorithms that always produce a "good enough" solution?

Let P be a minimization problem, and I be an instance of P. Let ALG(I) be a solution returned by an algorithm, and let OPT(I) be an optimal solution. Then ALG(I) is said to be a c-approximation algorithm, if for $\forall I$, $ALG(I) \leq c \cdot OPT(I)$.

Examples: Vertex Cover, Metric TSP.

Problems can be categorized to the best accuracy achieved by an approximation algorithm.

For some optimization problems, the approximation algorithms are unlikely to be possible. It is NP-hard to approximate them.

Faculty Course Evaluations

http://cmu.onlinecourseevaluations.com

Please fill one in!!

Final Exam

Sun., December 14

1:00pm - 4:00pm

GHC 4401 & GHC 4307

Format:

10 short questions, 4 pts each

6 long questions, 10 pts each

We allow a cheat sheet, both sides, any font size.

Review: Fri, Dec. 12. Time/room will be posted later. Practice Exam will be posted.