CS154, Lecture 18:

THE END
CS 154 – Final Exam

Thursday December 12, 3:30-6:30 pm
location: TBD

You’re allowed one double-sided sheet of notes
Exam is comprehensive (but will emphasize post-midterm topics)
Practice final and solutions on Piazza
Evaluate CS 154

Your Input Really Matters
Finite Automata (40s-50s):
Very Simple Model (constant memory)
• Characterize what can be computed (through closure properties)
• First encounter: non-determinism (power of verified guessing)
• Argue/characterize what cannot be computed
• Optimization, learning

More modern (algorithmic and complexity-theoretic) perspective:
streaming algorithms, communication complexity
Computability Theory 30’s – 50’s

Very Powerful Models: Turing machines and beyond

(Un)decidability – what cannot be computed at all

• Foot in the door – an unrecognizable language
• Many more problems, through reductions
• Hierarchy of exceedingly harder problems

The foundations of mathematics & computation

Kolmogorov complexity (universal theory of information)

Chapter II
Chapter III

**Complexity Theory: 60’s –**

Time complexity, P vs. NP, NP-completeness
- Non-determinism comes back
- Our foot in the door – SAT, a problem that is likely hard to compute
- Many more problems through (refined) reductions
- An hierarchy of hard problems

Other Resources: space, randomness, communication, power,... Crypto, Game Theory, Computational Lens
Computing: “evolution of an environment via repeated application of simple, local rules” Somebody
Computational Lens
Hitchhiker's Guide to the Galaxy
Computational Game Theory

Markets computing an equilibrium.
Simple dynamics (best response)?

Bounded Rationality:
Prisoners Dilemma
Repeated Games, infinite, finite
Backward Induction
Finite Automata
Always Cooperate, Always Defect, Tit for Tat, Trigger
Limited Resources
Factoring & One-Way Functions

Given two primes P and Q easy to compute N=PQ. For random such N, assume it is hard to find P and Q. Special case of One-Way Functions (the most basic cryptographic primitives).

Random Instances of SAT that are hard
Zero-Knowledge Proofs
Hardness of learning
Pseudorandom Generators
  Deterministically increasing entropy
Randomness is weak
**MAX-SAT**

*Max-SAT* = given a cnf formula how many clauses can be satisfied? A maximization problem: satisfy the most clauses

Can always satisfy a constant fraction of all the clauses. Specifically: When all clauses have at least 3 unique literals, can satisfy at least 7/8 of all clauses (how?) ≥ 7/8 of clauses of clauses in optimal solution (⇒ a 7/8-approximation).

Can we approximate MAX-SAT up to any constant < 1? Can we solve Max-3SAT with (7/8+eps)-approximation?

Not if P≠NP

For other problems no constant-approximation is likely - (clique n^{1-eps})
The PCP Theorem

For some constant $\alpha > 0$ and for every language $L \in \text{NP}$, there exists a polynomial-time computable function $f$ that maps every input $x$ into a 3cnf formula $f(x)$ s.t.

- If $x \in L$ then $f(x) \in \text{SAT}$
- If $x \notin L$ then no assignment satisfies more than $(1 - \alpha)$ fraction of $f(x)$ clauses.

$\Rightarrow$ sufficiently good approximation of MAX-SAT implies $\text{P} = \text{NP}$ (for tight inapproximability need better PCP theorem)
PCPs = Probabilistically Checkable Proofs

Alternative (equivalent) statement of PCP Theorem (informal):

Every statement that has a polynomial-time verifiable proof has such a proof where the verifier only reads $O(1)$ bits of the proof such that

[perfect completeness]: if the statement is correct accept with Probability 1
[soundness]: if the statement is false reject with probability 0.99

Example of the power of randomness (probabilistically checkable)
What can we Prove?

Every problem in NP has a short and easy to verify proof

How about coNP? Can a prover P convince a verifier V that there is no k-clique?

How about PSPACE? Can P convince V that there is a winning strategy for white from a particular position?

Yes!! If we add interaction!
Interactive Proofs

PCPs add randomness to proofs, what if we also add interaction?

$x \in L$?

Interactive Proofs can be used to prove membership in powerful (PSPACE) languages. For example: $V$ knows a winning Chess strategy: $\text{IP}=\text{PSPACE}$
Graph Non-Isomorphism

A graph G and H are isomorphic if can rename vertices of G to get H (the mapping is called isomorphism).

Graph Isomorphism = \{(G,H)| G and H are isomorphic\}
Graph Non-Isomorphism = \{(G,H)| G and H are not isomorphic\}

Graph Isomorphism in NP but can we prove that G and H are not isomorphic?

We will see a simple interactive proof
Interactive Proof for Graph Non-Isomorphism

Prover $P$  

Verifier $V$

Find $c$ such that $H$ and $G_c$ are isomorphic

Select a bit $b$ at random; Set $H$ to be a random isomorphic copy of $G_b$

Accept iff $c = b$

[perfect completeness]: if $G_0, G_1$ are not isomorphic $V$ accept with Probability 1
[soundness]: if $G_0, G_1$ are isomorphic $V$ accept with Probability $\frac{1}{2}$ (no matter what $P$ does)
Zero-Knowledge (Interactive) Proofs

PCPs add randomness to proofs, what if we also add interaction?

$x \in L$?

Prover $P$ Verifier $V$

Zero-Knowledge Proofs – reveal no information apart of $x \in L$

ZK proofs for all of IP (PSPACE)
IP for Non-Isomorphism is ZK (for semi-honest verifier)

Prover $P$

Verifier $V$

Select a bit $b$ at random; Set $H$ to be a random isomorphic copy of $G_b$

Find $c$ such that $H$ and $G_c$ are isomorphic

Accept iff $c = b$

[perfect completeness]: if $G_0, G_1$ are not isomorphic $V$ accept with Probability 1

[soundness]: if $G_0, G_1$ are isomorphic $V$ accept with Probability $\frac{1}{2}$ (no matter what $P$ does)
Where is Waldo?
That's all Folks!