Review of Complexity Course Material

John Reif

Formal Language Theory:

- **Finite State Automata & Regular Expressions**
- **Non-Determinism in Automata Push Down Automata & Context Free Languages**

Turing Machine Definitions:

- **Deterministic Turing Machines (TMs)**
- **Church-Turing Hypothesis**
- **Turing Machine Variations**
- **Non-Deterministic TMs**
- **Universal Turing Machines**

Countable & Uncountable Sets

- **Countable Sets**
- **Sets shown Uncountable via Diagonalization Proofs**

Decidable & Undecidable Languages

- **Decidable Languages**
- **Languages shown Undecidable via Diagonalization Proofs**

Time and Space Complexity:

- **Complexity Classes**
- **Deterministic Time**
- **Space Complexity**
- **Constant Space Compression**
- **Constant Time Speedup**
- **Time and Space Hierarchy Theorems**

Nondeterminism and NP

- **NonDeterministic TMs**
- **NonDeterministic Time Hierarchy**
- **Ladner's Theorem: If P is not NP then ∃ NP Language neither NP complete nor in P**
- **Sparse Languages and NP**

Language Reductions:

- **Polytime Reductions**
- **Logspace Reductions**

P and NP-Completeness:

- **P- Completeness**
- **NP-Completeness**
- **Cook-Levin Theorem: proof that 3SAT is NP-Complete**
- **Karp's Reduction Proofs of NP Completeness of various Combinatorial Search Problems**

Oracles & Relativized Proof Techniques

- **Oracle Turing Machines**
- **Relativized Proofs**

- An oracle machine is a “black box” machine capable of solving any specified problem (given by a specified language)
- eg P^{SAT} is a polynomial time deterministic Turing machine with access to an oracle for SAT
- What does it mean for a proof to relativize?
- Baker-Bill-Solovoy theorem
- Oracles can be used to define the PH

Relativization

- Many proofs and techniques we have seen **relativize**:
  - they hold after replacing all TMs with oracle TMs that have access to an oracle A
  - e.g. $L^A \subset P^A$ for all oracles A
  - e.g. $P^A \neq EXP^A$ for all oracles A

Polynomial Hierarchy (PH):

- **Definition of PH**
- **Properties of PH**
- **Oracle Characterization of PH**

PSPACE Complete Problems:

- **Quantified Boolean Logic**
- **Complexity of 2 Player Games**

Nondeterministic Space Complexity:

- PSPACE Completeness
- Savitch's proof PSPACE=NPSPACE
- Proof NSPACE = coNSPACE

Nondeterministic LogSpace Complexity

- L = Deterministic Logspace
- NL = Nondeterministic Logspace
- IS Theorem: NL=coNL

Kolmogorov Complexity (KC)

- **Definition of Kolmogorov Complexity**
- **Lower Bounds by Incompressibility Method**

- How to reason about the randomness and information density of a string
- Invariance theorem,
- incompressibility method using Kolmogorov Complexity

Kolmogorov Theory continued…

- Intuitively: $K(x)$= length of shortest description of x
- Define conditional Kolmogorov complexity similarly,
- $K(x|y)$=length of shortest description of x given y.
- Properties of $K(x)$ and $K(x|y)$:
  - $K(xx) = K(x) + O(1)$ since just need TM that generates x
  - $K(xy) \leq K(x) + K(y) + O(\log(\min\{K(x), K(y)\}))$
  - $K(1^n) \leq O(\log n)$ since can use binary encoding of n
  - $K(\pi_{1:n}) \leq O(\log n)$ since can use binary encoding of n
  - For all x, $K(x) \leq |x|+O(1)$ since can encode x in TM
  - $K(x|x) = O(1)$ since just need TM that generates x
  - $K(x|\varepsilon) = K(x)$ since empty string ε provides no additional info on x

13

Parallel Computational Complexity:

- **Circuit Definitions**
- **Circuit Complexity Measures: size & depth**
- **NC Complexity Class**

Uniform & Nonuniform Circuit Families:

- **Definitions of circuit uniformity and advice to construct circuit**
- **Logspace uniform classes of circuits**
- **Poly time uniform classes of circuits**
- **P/poly:polynomial size circuits constructable by polynomial size advise**

  Recall: "**NP** does not have polynomial-size circuits" (**NP** $\not\subset$ **P/poly**) implies **P** $\neq$ **NP**

Basic Circuit Results:

Circuit Lower Bounds:

- **Shannon's Counting Argument,**
- **Formula lower bound on Andreev function**

Karp-Lipton Theorem: if SAT can be solved by polynomial size circuits then the Polynomial Hierarchy PH collapses to its second level.

Lower Bounds for Poly Size Circuits with Unbounded Fan- in and Constant Depth:

Theorem: Parity is not in AC^0

- **Simple Proof for depth 2 Circuits**

- **Hastad's proof that Parity is not in AC^0:** Hastad's switching lemma starts with constant depth AC circuit, and reduces the depth while preserving size (by randomly assigning some of the inputs).

Decision Tree Complexity:

- **Decision Tree Definitions**

- **Certificate Complexity**

- **Randomized Decision Trees**

- **Distributional Complexity: where the computation is deterministic but the inputs are random**

- **Yao's Lemma on Probabilistic Decision Trees:** Relates Randomized Decision Trees to Distributional Complexity

Randomized Complexity:

- **Randomized Complexity Classes:**
    - **RP,**
    - **BPP,**
    - **ZPP**
- **Two-sided error and one -sided error.**
- **Techniques for Probabilistic TMs Error Reduction**

- **Schwartz-Zippel Lemma for identity testing of multivariate polynomials:** Construction using probabilistic model to solve polynomial identity testing

- **BPP in P/poly (poly size circuits)**

- **Unique Sat:**
    - **The Valiant-Vazirani Theorem:** Randomized technique for reducing the number of satisfying instances of a SAT formula to only a unique solution

Counting Complexity Classes:

- **#P defined**
- **Counting Cycle Covers and Bipartite Matchings are #P Complete**

Parsimonious Reduction: preserves the number of solutions

- many standard **NP**-completeness reductions are parsimonious
- therefore: if #SAT is **#P**-complete we get lots of **#P**-complete problems

Pseudorandom Number Generation & Cryptography:

- **One Way Functions**

- **Pseudorandom Number Generators**

- **Pseudorandom generation for BPP**

- **Goldreich-Levin hard bit**
  Proves the randomness of PRGs

- **Yao's Lemma:** Slight variation also helps to describe PRGs

Communication Complexity

- **Deterministic Communication Complexity**

- **Randomized Communication Complexity**

Interactive Proofs (IP):

- **Interactive Protocols (IP) & Zero Knowledge Proofs**
  - **Interactive proofs of graph non-isomorphism**

2 graphs: $|G_1| = |G_2| = n$
Verifier:

Accept if $G \neq H$, Reject if $G = H$

Protocol: Private-coin Graph Non-isomorphism
V: pick $i \in \{1, 2\}$ uniformly randomly. Randomly permute the vertices of G_i to get a new graph H. Send H to P.
P: identify which of G_1, G_2 was used to produce H. Let G_j be that graph. Send j to V.
V: accept if $i = j$; reject otherwise.

- **Author-Merlin Games**
- **The complexity classes MA and AM**
- **Proof IP = PSPACE**

IP, IP[f(n)]
- Interactive polynomial time, sometimes hard-limited by f(n)
- Private coin protocol (prover cannot see verifier randomness)

AM, AM[f(n)]
- Arthur (honest but limited verifier)-Merlin (untrustworthy but all-powerful prover)
- Public coin protocol (prover can see verifier randomness)
- Should also be noted that counterintuitively, IP[k]$\subseteq$AM[k+2]

Probabilistically Checkable Proofs (PCP):

- **PCP Theorem**
- **Hardness of Approximation:**
  - **Hardness of Approximating MAX-SAT**
- **NP in PCP(poly(n),1)**

Quantum Computation

- **Qubits**

- **Unitary Operations & Projection (Collapse via Observation)**

- **Quantum Circuits**

- **Quantum Algorithms**
  - **Quantum Factoring in Poly time**
  - **Quantum Search**

Relationship with complexity classes

- P/poly
  - Languages accepted by Boolean circuit families of polynomial size
- NP
  - $\mathrm{NP} \not\subset \mathrm{P/poly} \Rightarrow \mathrm{P} \neq \mathrm{NP}$,
  - If any function in NP has exponential-sized circuit (is not “feasibly”
- PSPACE
  - NC (logspace) strictly contained in PSPACE
- LSPACE & NLSPACE ($NC^1 \subseteq L \subseteq NL \subseteq NC^2 \subseteq NC^k \subseteq NC$)
  - Between $\log n$ and $\log^2 n$
- **NC=?P**
  - The most interesting problems about parallelization

Logic & Proof in Complexity Theory:

Why are complexity lower bounds so difficult?

- **Natural Proofs:**
  - **Natural Methods used to attempt to prove complexity bounds such as P vs NP are too powerful,**
  - **implys that one-way functions do not exist**

- **Independence in First Order Logics**
  - **Powerful First order logics such as Peano Arithmetic may not be able to prove complexity bounds such as P vs NP.**