6.841/18.405 LECTURE 01

Today

1. ADMINISTRIVIA

2. OUTLINE OF 6.841 / COMPLEXITY

3. REVIEW OF 6.840

---

ADMINISTRIVIA

- LECTURER: NAHHU SUDAN nahhu@mit.edu
- TA: BRENDAN JUBA bjuba@mit.edu

- To Do List:
  - ENSURE MEMBER OF MAILING LIST
  - SIGN UP FOR 6.841 today
  - SIGN UP FOR SCRIBING
  - LOOK AT PS4 (due in two weeks)
- Grading: (Not a commitment ...)
  - 3 Problem Sets
  - 1 Scribe
  - 1 Project
  - ≥ Participation) (< will ask for & post emails.

---

Goals of Computational Complexity

- Identify important problems
  (If sufficiently interesting, we get Phenomena / Classes)
- Analyze resources
- Compare with other problems
What makes a problem interesting?

Example 1: \#SAT

Input: 3 CNF formula Boolean

\[ \phi = (\overline{X} = (x_1 \cdots x_n), \overline{C} = (C_1 \cdots C_m)) \]

\[ C_j = 3 \text{ literals } (x_{i_1} \lor \overline{x_{i_2}} \lor x_{i_3}) \]

Goal: count \# satisfying assignments.

Is this an interesting problem?
Example 2: Permanent

Given an $n \times n$ matrix

$$A = \begin{bmatrix} a_{ij} \end{bmatrix}$$

$$\text{perm}(A) = \sum_{\pi: [n] \to [n]} \prod_{i=1}^{n} A_{i, \pi(i)}$$

(permutation)

(just like determinant, without the sign)

Is the permanent interesting?)
CNF minimization

Given: 3 CNF formula $\phi$ with $m$

clauses & integer $m'$

find equivalent formula $\psi$ with

$\leq m'$ clauses

$(\phi = \psi$ if $\forall a \phi(a) = \psi(a)$).

# SAT?
Permanent?
CNF minimization?

finally an aesthetic question....

... but science/math can educate us.
I believe: All 3 are interesting.

CNF minimization: Typical instance of many problems in logic/VLSI etc.

- \(NP = P \implies\) CNF Minimization \(\leq P\)

- Yet seems much harder than SAT...

\#\text{SAT}/\text{Permanent}: Rarely pursued in practice... but equivalent problems arise up...

- Problems are equivalent to each other!!

Valiant
- Permanent \(\rightarrow\) Algebraic analog of \(\text{NP}\)-complete

Ede
- CNF minimization \(\leq\) Permanent

Lipton
- Easy to generate provably hard instances (average case vs. worst-case)
Interesting \leftrightarrow Problem occurs in practice, often.

Interesting \leftrightarrow Problem has many variations with different consequences.

Back to Complexity

Universe of Problems
Would love to know which ones need more resources than others.

Typically hard problem... not much is known; mostly have conjectures.

In the absence of negative results try to accumulate as many positive examples as possible.

$A \leq B = A$ no harder than $B$
Collection of equivalent problems

Arrows going from left to right

⇒ Something interesting is going on.

Perhaps $A$ is a class?

$A \cup B$ is a class?

$B$ is complete?
Aggregation of evidence focuses attention:

& we repeat exercise on central problems.

Classes / Phenomena

\[ \text{NP} \equiv \text{Complexity of Theorem Proving} \]

\[ \text{PSPACE} \equiv \text{Complexity of 2-Player games} \]

\[ \text{IP} \equiv \text{Games against nature / Debates} \]

\[ \text{NP} = \{ L \mid \exists \text{ poly time alg } A \}

\text{ s.t. } L = \{ x \mid \exists y \text{ s.t. } A(x, y) \text{ accepts} \} \]

"y" is a proof that "x \in L"
interesting $L = \llbracket \text{Theorems} \rrbracket$

$= \{ \langle T, x \rangle \mid \exists y \text{ s.t. } |y| \leq n \}$

$\exists \text{ proof of theorem } T \}$

$L \in \text{NP} \land L \text{ is NP-complete}$

$\text{NP} = \text{P} \Rightarrow \text{every theorem is easy to prove (not harder to find proof than to write it down).}$

$\text{NP} = \text{P} \Rightarrow \text{no interesting theorems}$

$\Rightarrow \text{NP} = \text{P} \text{ is Mathematics-complete.}$
Computational Problems

General problem of the type \( R \subseteq \{0,1\}^* \times \{0,1\}^* \)

Given: \( \alpha \) find \( y \) satisfying \( \alpha, y \in R \)

Yet we like

\[ \underline{\text{Languages}} = L \subseteq \{0,1\}^* \]

A the problems: Given \( \alpha \)

Is \( \alpha \in L \)?
Reductions

\[ L_1 \leq L_2 : \text{ } L_2 \text{ can be solved efficiently} \]

\[ \Rightarrow L_1 \]

Two ways:

1. Many-many reduction / Turing reduction
   Subroutine call
   Alg. for \( L_1 \) using \( L_2 \) as subroutine

2. Many-one reduction / Karp reduction
   \[ A : \{0,1\}^* \rightarrow \{0,1\}^* \]
   \[ x \in L_1 \Rightarrow A(x) \in L_2 \]
Why two definitions?

(Karp)
- Restricted reduction easier to find if it exists.
- Tells more about problems.

(Turing)
- Weaker reduction useful when other does not exist.

Example: is SAT = co-SAT?

Answer 1: Yes, since P-time alg. for one implies one for other.

Answer 2: (Probably) No, since can’t "prove" φ ∈ co-SAT easily.

L₁ \leq_{Karp} L₂ \land L₂ \in \text{NP} \Rightarrow L₁ \in \text{NP}.
Agenda for 6.841

- First few weeks
  Whatever little we know about lower bounds.

- Then: Resources
  1. Alternation: "CNF Minimization"
     either SAT & Linear time
     or SAT & Log space
  2. Counting:
     - Is SAT easier if # solution is
       guaranteed to be \( \leq 1 \)?
     - DNF Min. \( \leq \#\text{SAT} \)
3. Proofs, Interaction, Knowledge
4. Distributional Complexity (vs. worst-case)
5. Quantum Computing.

6.840 Review

Time / Space Hierarchy Theorems

- $\text{Time} \left( n^2 \right) \neq \text{Time} \left( n^6 \right)$
- $\text{Space} \left( n^3 \right) \neq \text{SPACE} \left( n^6 \right)$

etc.
Comparing diff resources is hard

\[ \text{Time} (t) \leq \text{SPACE} (t) \leq \text{TIME} (2^t) \]

\( \cup \)

\( \text{NTIME} (t) \)

Space / NSPACE better understood than

Time / NTIME

\[ \text{SPACE} (s) \leq \text{NSPACE} (s) \leq \text{SPACE} (s^2) \]

\[ \text{NSPACE} (s) \leq \text{CONSPACE} (O(s)) \]

\#P \& \text{NP-completeness}

\#P \leq \text{IP} \quad [\text{will cover again}]