6.841/18.405 LECTURE 01

Note Title

TODAY

- ADMINISTRIVIA
- OUTLINE OF 6.841 / COMPLEXITY
- REVIEW OF 6.840

ADMINISTRIVIA

- LECTURER: MADHU SUDAN madhu@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 PS1 (due in two weeks).

- GRADING : (Not a commitment ...)
  - 3 PROBLEM SETS
  - 1 SCRIBE
  - 1 PROJECT
  - ∞ PARTICIPATION) $\leftrightarrow$ 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

$$\phi = \left(\overline{X} = (x_1 \dots x_n), \quad \overline{C} = (C_1 \dots C_m) \right);$$

$C_j =$ 3 literals $(x_{i_1} \vee \overline{x}_{i_2} \vee x_{i_3})$

Goal : Count # satisfying assignments.

Is this an interesting problem?

Example 2 : Permanent

Given $n \times n$ matrix

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

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

(just like determinant, without the sign)

Is the permanent interesting?

CNF minimization

Given: 3 CNF formula ϕ with m clauses & integer m'

find equivalent formula ψ with $\leq m'$ clauses

$$\left(\phi \equiv \psi \text{ if } \forall a \;\; \phi(a) = \psi(a) \right).$$

\# SAT ?

Permanent ?

CNF minimization ?

Interesting ?

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.

- $NP = P \Rightarrow$ CNF minimization $\in P$

- Yet seems much harder than SAT ...

#SAT / Permanent : Rarely pursued in practice ... but equivalent problems come up.

- Problems are equivalent to each other!!

[Valiant] - Permanent ~ Algebraic analog of NP-completeness

[Toda] - CNF minimization $\leq$ Permanent

[Lipton] - Easy to generate provably hard instances (average case vs. worst-case)

Interesting ⇐ Problem occurs in practice, often.

Interesting ⇐ Problem has many variations with different consequences.

Back to Complexity

Universe of Problems

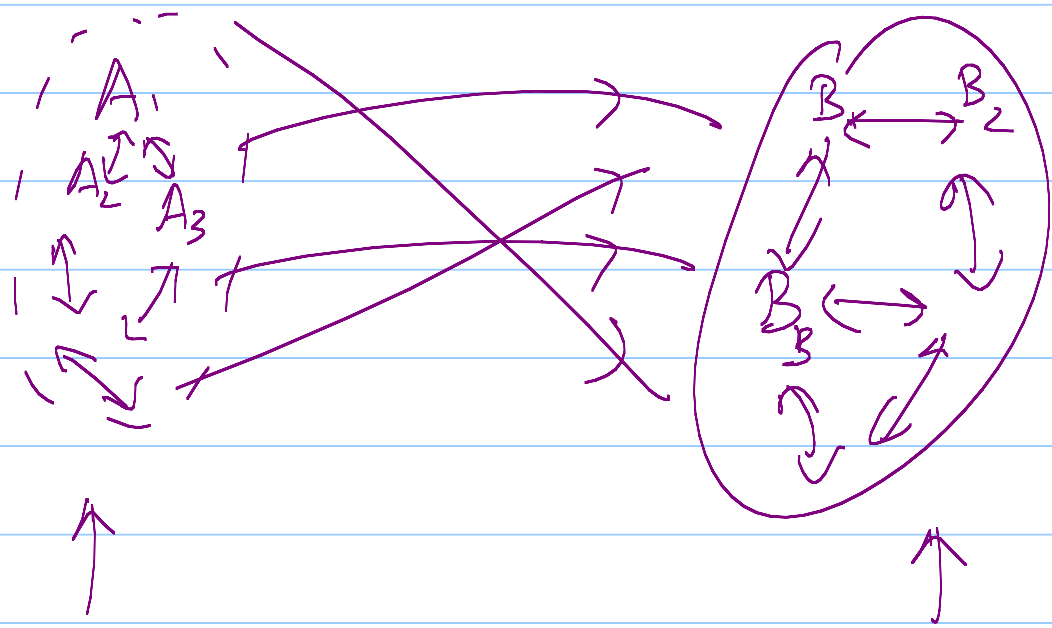
- Would love know which ones <u>need</u> more resources than others

- Typically hard problem ... not much is <u>known</u>; Mostly have conjectures.

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

$$A \longrightarrow B$$

$A \leq B$:= A no harder than B



Collection of equivalent problems

Collection of equivalent problems

Arrows going from left to right

$\Rightarrow$ Something interesting is going on.

Perhaps A is a class?

$A \cup B$ is a class?

B is complete?

Aggregation of evidence focusses attention:

& we repeat exercise on central problems.

Classes / Phenomena

NP $\equiv$ Complexity of Theorem Proving

PSPACE $\equiv$ Complexity of 2-Player games

IP $\equiv$ Games against nature / Debates

$$NP = \{ L \mid \exists \text{ poly time alg } A$$
$$\text{s.t. } L = \{ x \mid \exists y \ A(x, y) \text{ accepts} \} \}$$

"y" is a proof that "$x \in L$"

interesting L = "Theorems"

$$= \{ (T, 1^n) \mid \exists y \ |y| \leq n$$

y proof of theorem

$T \}$

$L \in NP$; L is NP-Complete

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

- $NP \neq P \Rightarrow$ no interesting theorems

$\Rightarrow$ $NP = P$ is Mathematics-complete.

Computational Problems

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

Given : x find y satisfying

$x, y \in R$

Yet we like

Languages = $L \subseteq \{0,1\}^*$

& the problems : Given x

Is $x \in L$?

Reductions

$L_1 \leq L_2$: L_2 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

   $A : \{0,1\}^* \to \{0,1\}^*$

   s.t. $x \in L_1 \iff A(x) \in L_2$

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 $\equiv$ co-SAT ?

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

Answer 2: (Probably) No, since can't "prove" $\phi \in$ co-SAT easily.

$L_1 \leq_{Karp} L_2$ & $L_2 \in NP \Rightarrow L_1 \in NP$.

Agenda for 6.841

- First few weeks

Whatever little we know about lower bounds.

- Then : Resources

① Alternation : "CNF Minimization"

either SAT $\notin$ Linear time

or SAT $\notin$ Logspace

② Counting :

- Is SAT easier if # solutions is guaranteed to be ≤ 1 ?

- DNF Min. $\leq$ #SAT

(3) Proofs, Interaction, Knowledge

(4) Distributional Complexity (vs. worst-case)

(5) Quantum Computing.

6.840 Review

Time / Space Hierarchy theorems

- $\mathrm{Time}(n^2) \subsetneq \mathrm{Time}(n^6)$
- $\mathrm{Space}(n^3) \subsetneq \mathrm{SPACE}(n^6)$

etc.

Comparing diff resources is hard

$$\mathrm{Time}(t) \subseteq \mathrm{SPACE}(t) \subseteq \mathrm{TIME}(2^t)$$

$$\cap\!\!\backslash \quad \mathrm{NTIME}(t) \quad \cup\!\!/$$

Space / NSpace better understood than Time / NTime

$$\mathrm{SPACE}(s) \subseteq \mathrm{NSPACE}(s) \subseteq \mathrm{SPACE}(s^2)$$

$$\mathrm{NSPACE}(s) \subseteq \mathrm{coNSPACE}(O(s)).$$

- NP & NP-Completeness
- $\#P \subseteq IP$ [will cover again]