

6.841/18.405 LECTURE 01

Note Title

2/7/2007

TODAY

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

ADMINISTRIVIA

- LECTURER: MADHU SUDAN machu@mit.edu
- TA: SWASTIK KOPPARTY swastik@mit.edu
- To Do List:
 - ENSURE MEMBER OF MAILING LIST
 - SIGN UP FOR 6.841 today
 - SIGN UP FOR SCRIBING
 - LOOK AT PS 1 (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

$$\Phi = (\bar{X} = (x_1 \dots x_n), \\ \bar{C} = (C_1 \dots C_m)) ;$$

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

Goal: Count # satisfying assignments.

Is this an interesting problem?

EXAMPLE 2: Permanent

Given $n \times n$ matrix

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

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

$\pi: [n] \rightarrow [n]$
permutation

(just like determinant, without the sign)

Is the permanent interesting?

CNF minimization

Given : \exists CNF formula ϕ with m clauses & integer m'

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

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

SAT ?

Permanent ?

CNF minimization ?

} Interesting ?

finally an aesthetic question

... but science/math can evaluate 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

[Joda]

- CNF minimization \leq Permanent

[Lipton]

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

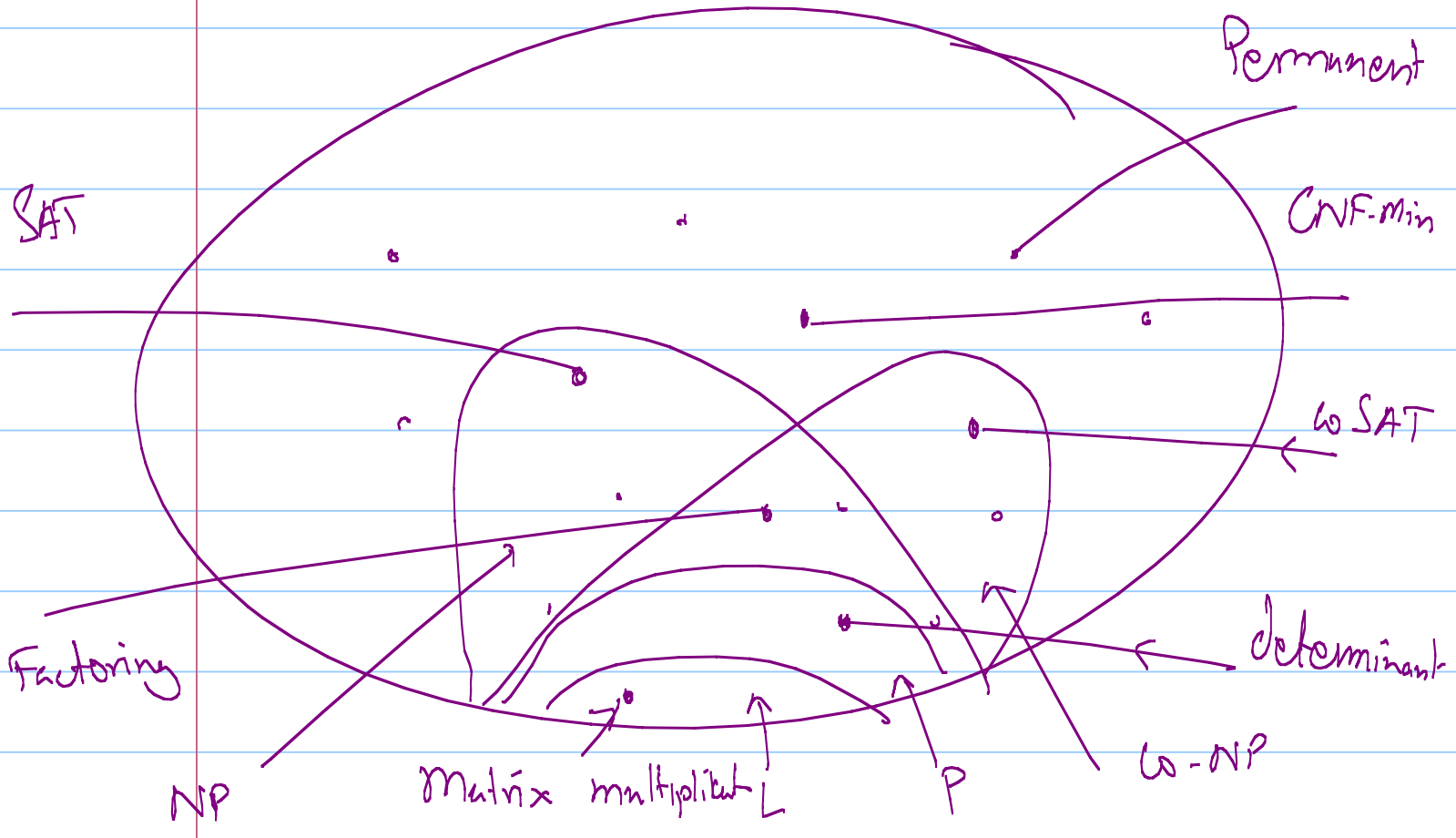
Interesting \Leftarrow Problem occurs in practice, often.

Interesting \Leftarrow Problem has many variations with different consequences.



Back to Complexity

Universe of Problems

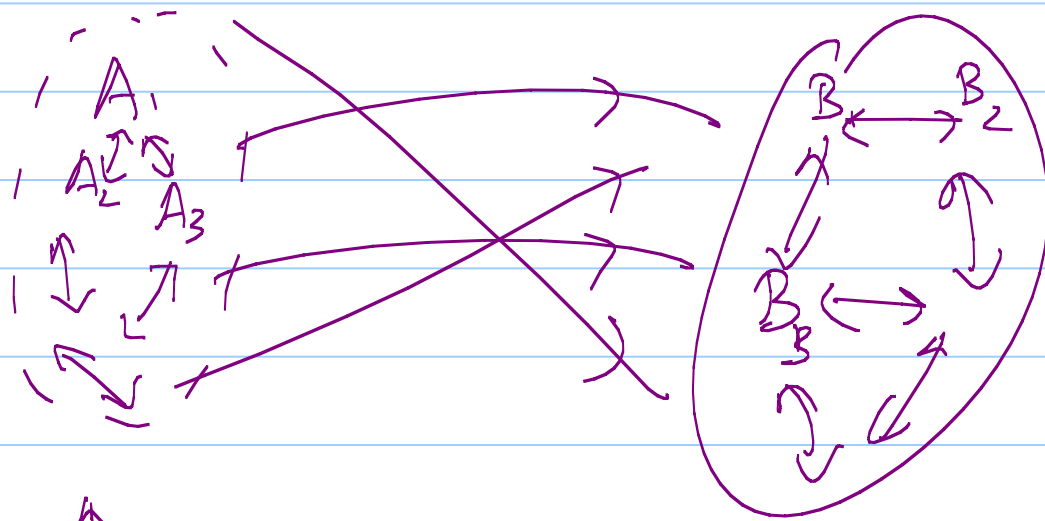


- Would love know which ones need more resources than others
- Typically hard problem ... not much is known; Mostly have conjectures.

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



$A \leq B \neq A$ no harder than B



↑
Collection of equivalent problems

↑
Collection of equivalent problems

Arrows going from left to right

⇒ Something interesting is going on.

Perhaps \textcircled{A} is a class?

$\textcircled{A} \cup \textcircled{B}$ is a class?

\textcircled{B} is complete?

Aggregation of evidence focuses 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 = $\{L \mid \exists \text{ poly time alg } A$

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

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

interesting $L = \text{"Theorems"}$

$$= \{ (T, 1^n) \mid \exists y \ |y| \leq n \\ y \text{ proof of theorem } T \}$$

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

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

• $NP = P \Rightarrow$ 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

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

$$\text{s.t. } x \in L_1 \Leftrightarrow A(x) \in L_2$$

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" $\emptyset \in CoSAT$ easily.

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

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.

————— x —————

6.840 Review

Time / Space Hierarchy Theorems

• Time (n^2) \subsetneq Time (n^6)

• Space (n^3) \subsetneq SPACE (n^6)

etc.

Comparing diff resources is hard

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

∩
NTIME(t)

Space / NSpace better understood than
Time / NTime

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

$$\text{NSPACE}(s) \subseteq \text{CONSPACE}(O(s))$$

• NP & NP-Completeness

• #P \subseteq IP [will cover again]

