CS221 (Computational Complexity)

Today

1. Administrative Things
2. Outline of Course
3. Review of CS121 (what we need)

Contacts

Lecturer : MADHU SUDAN madhu@cs.harvard.edu

TF : PREETUM NAKKIRAN preetum@cs.harvard.edu

To Do List

- Sign up on Piazza
- Enroll in Course!
- Sign up for scribing
- Look at PS1 (due Friday)

http://madhu.seas.harvard.edu/courses/Spring2018

Grading

1. 3-5 problem sets (40%)
2. SCRIBE WORK (30%)
3. 1 PROJECT (15%)
4. & PARTICIPATION (15%)

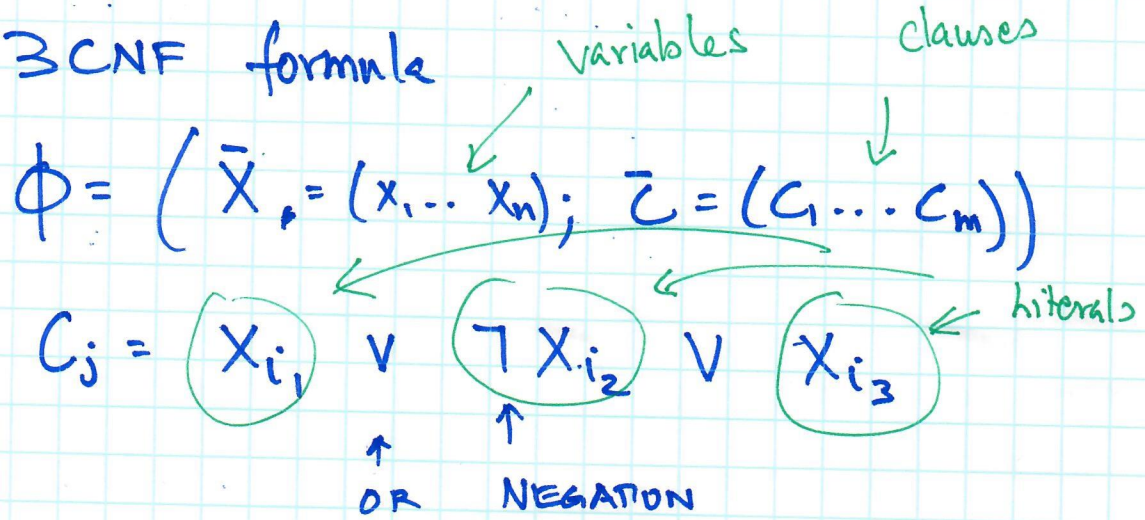
Goals Of Computational Complexity

- Identify important problems (+ related Phenomena / Classes)
- Analyze resources needed & compare tradeoffs.
- Compare with other problems.

What makes a problem interesting?

Example 0: 3SAT

Input: 3CNF formula



Problem: Decide if $\exists$ satisfying Assignment

i.e., Assignment $X_1 = 0,\ X_2 = 0,\ X_3 = 1 \dots$

s.t. $\forall j$ some literal is 1

Example 1: #SAT

Input: 3CNF formula ϕ

Problem: Output # satisfying assignments to ϕ.

Example 2: Permanent

Given: $n \times n$ matrix $A = \left[a_{ij} \right]$

Output:

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

↑ 1-1 function (permutation)

for contrast: $\det(A) = \sum_{\pi: [n] \to [n]} (-1)^{\text{sign}(\pi)} \prod_{i=1}^{n} a_{i\pi(i)}$

CNF Minimization

Given: 3 CNF formula ϕ

Output: Equivalent formula ψ [is $\phi(x) = \psi(x) \ \forall x$]

with min. # clauses.

VC Dimension (Defn. Omitted)

What makes a problem interesting

ⓐ Natural ("Subjective")

And ⓑ Captures other interesting problems ("objective")

0. SAT : Captured many interesting problems ....

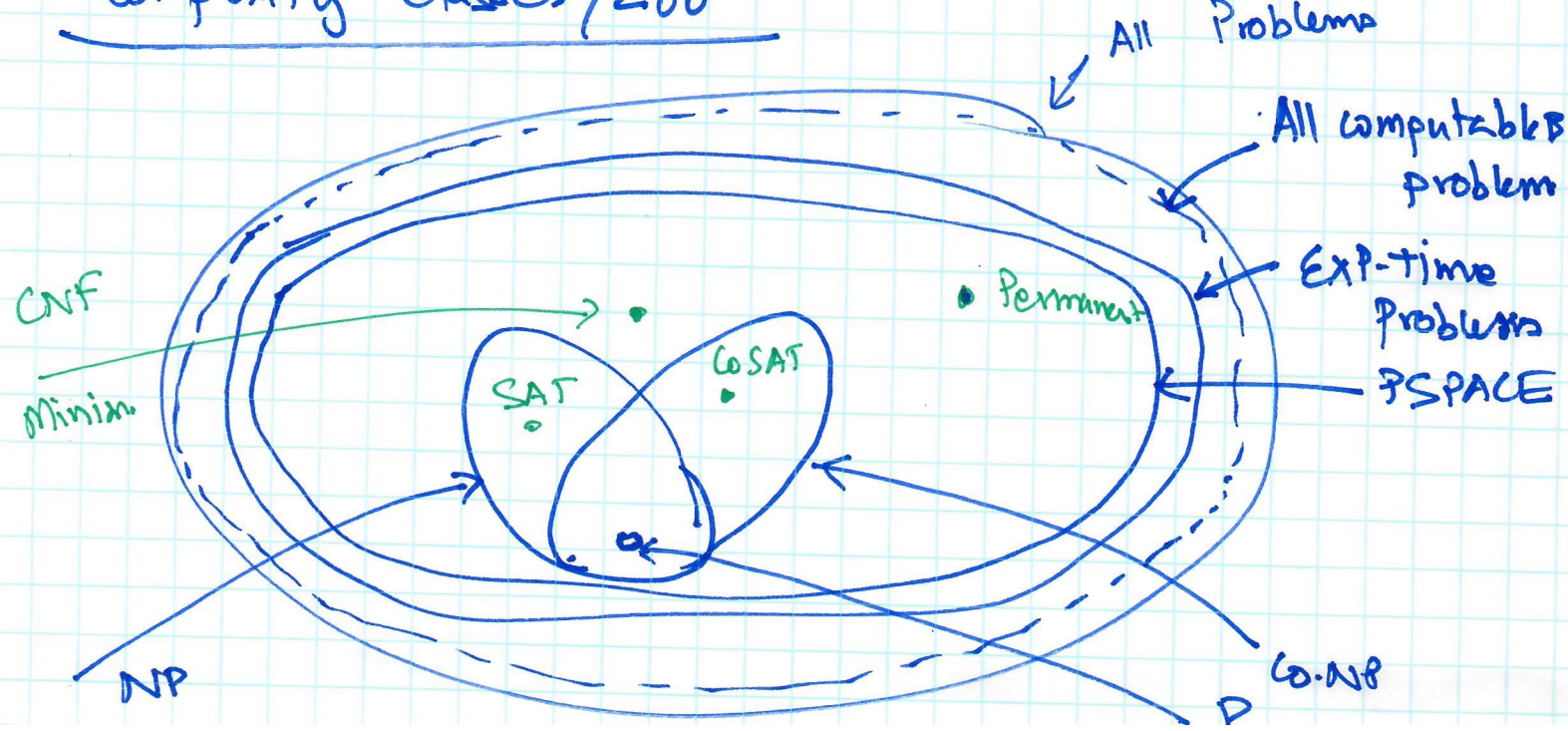
1. #SAT : "Pr[Event]"

2. Permanent : • Till 1900s : Curious relative of determinant.

• Valiant 80 : Permanent ≡ #SAT.

• Toda 86 : CNF Minim. ≤ #SAT.

Complexity Classes / Zoo



- Goal of field: Separate problems based on resources
- State of field:
  - Many conjectures.
  - Hard to prove lower bounds
  - Are able to compare quite well.

$A \longrightarrow B$ $\equiv$ A "no harder than" B

$\Rightarrow$ $A \leq B$

$A \rightleftharpoons B \Rightarrow$ A equivalent to B.

When we find many equivalent problems: time to define a complexity class!

$A_1 \equiv A_2 \equiv A_3 \equiv A_4 \equiv A_5$

Basic Tools for comparisons

① **Languages** ≡ Decision Problems ≡ Boolean functions.

[Convert all problems into Decision Problems].

[YES/NO answer]

② Reduces Problems to one-another.

- (2.1) Many-many reduction / Turing Reduction
- (2.2) Many-one reduction / Karp Reduction

→ $L_1 \leq_m L_2$ if subroutine for L_2 yields algorithm for L_1

→ $L_1 \leq_{Karp} L_2$ if ∃ algorithm A s.t.

$$x \in L_1 \Leftrightarrow A(x) \in L_2.$$

∈

Reduction more restricted ⇒ More refined view of computation.

Example

~~NP = all languages L s.t.~~

SAT : already defined $= \{\phi \mid \exists$ sat. assignment to $\phi\}$

coSAT :

$= \{\phi \mid$ No sat. assignment exists for $\phi\}$.

is SAT $\equiv$ co-SAT ?

Answer 1: YES. $\leftarrow$ Polytime Alg for SAT can be complemented to get Alg. for coSAT

Answer 2: NO : Easy to "prove" $\phi \in$ SAT.

Not known to "prove" $\phi \in$ coSAT.

Course Overview

- Review of Complexity
  - Diagonalization
  - Non-determinism
  - Non-uniformity
  - Space
- Semi-Classical
  - Alternation ("Complexity of Debates")
- Modern
  - Randomness
  - Interaction
    - → AC^0 lower bounds
    - → Toda's Theorem
    - → IP = PSPACE
- Post-Modern
  - → PCPs
  - → ETH
  - → PPAD
  - → TLA ...

Next Lecture

- Diagonalization: "I can't be saying this"
- Time + Space Hierarchies: More ⇒ More.
- Limits of Diagonalization: Relativization.