NP-Completeness

15-750
4/19/10

Basic Topic so far!

Present a prob (eg Max-Flow-Prob) give an algorithm for the prob.

For each prob we gave a bd on time, space, or work (eg $O(n^3)$)

Alice & Bob Prob:

Bob's Job: Write efficient code for 2-coloring a graph!

Alice's Job: " " 3-coloring a graph!

Run-time Guarantees

All have been polynomial time:

$O(n^k)$ for some k

Times like $O(2^n)$ or $O(n^{\log n})$ are not polynomial time guarantees.

Problem Size and Actual size

Prob	Prob Size	Actual Size
$G = (V, E)$	$V + E$	V^2, $(V+E)\log V$
Data Structure Keys	n	$n \log n$
Matrix $n \times n$	n	$n^2 \log n$ $w \log n$

Claim: A polynomial of a polynomial is a polynomial

ie $f, g \in Q[x]$ then $f(g(x)) \in Q[x]$

In particular $\deg(f) = n$ & $\deg(g) = m$

then $\deg(f \circ g) = n \cdot m$

What is a Prob!

Input - Output form

eg

Input: $G = (V, E)$ $a, b \in V$ encoded as a string in binary

Output: Short path from a to b encoded as a string in binary.

Language form: e.g.

Prime $= \{ x \in \{0\,1\}^* \mid x$ is a number n in binary $\wedge$
n is a prime number$\}$

Def $P = \{ L \subseteq \{0,1\}^* \mid \exists$ poly-time alg
for deciding $L \}$

Thm Prime $\in P$ (no proof)

Def A cycle C of G is <u>Hamiltonian</u> if

1) C is simple
2) C contains every vertex of G.

Ham-Cycle $\equiv \{ G \mid G$ has a Ham-cycle$\}$
$\{ \exists$ Ham-cycle of $G \}$

Witness and Certificates

$H = \{ (G, C) \mid C$ is a Ham-cycle of $G \}$

Note: $H \in P$

$$\text{Ham-Cycle} \equiv \{ G \mid \exists \underset{\substack{\uparrow \\ \text{Witness}}}{C} \text{ s.t. } \langle G, C \rangle \in H \}$$

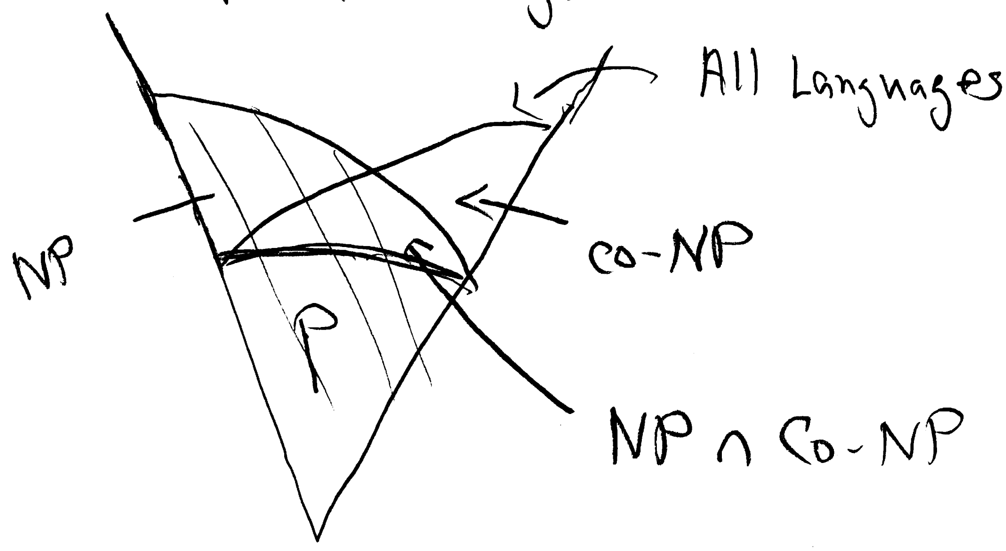
$$NP = \{ L \mid \exists R \text{ (poly-time relation) } \wedge$$
$$x \in L \text{ iff } (\exists y \; R(x,y) \wedge |y| \le |x|^k \text{ some } k) \}$$

Thus Ham-Cycle $\in NP$

$$\overline{L} = \{ x \in \{0,1\}^* \mid x \notin L \}$$

Def $\text{co-}NP = \{ L \subseteq \{0,1\}^* \mid \overline{L} \in NP \}$

Venn Diagram



Reducibility

4/23/08

Many-One: $L_1 \le_p L_2$ if $\exists$ fcn f st

1) f is poly-time computable

2) $x \in L_1$ iff $f(x) \in L_2$

Turing Reduction!

$L_1 \leq_p^T L_2$ iff $\exists$ poly-time alg A st

1) L_1 = Lang accepted by A

2) A may require calls to oracle for L_2.

Claims!

1) $L_1 \leq_p L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$

2) $L_1 \leq_p^T L_2 \leq_p^T L_3 \Rightarrow L_1 \leq_p^T L_3$

3) $L_1 \leq_p L_2 \wedge L_2 \in NP \Rightarrow L_1 \in NP$

<u>Def</u> L is <u>NP-Hard</u> if $\forall L' \in NP \; L' \leq_p L$

<u>Def</u> L is <u>NP-Complete</u> if

1) $L \in NP$
2) L is NP-Hard

Models of computation for which Poly-time are equivalent!

1) RAM

2) Turing Machines

3) Parallel-RAM (poly # of processors)

4) Arrays of finite controls

5) λ-calculus

Not Known

Quantum Computers

The First NP - Complete Prob

Boolean Formula

Variables $x_1, \ldots, x_n$ $\quad \bar{x}_i \equiv$ denotes not x_i

Literals $x_1, \bar{x}_2, \ldots, x_n, \bar{x}_n$ $\quad a_i \equiv$ denotes a literal

$\wedge \equiv$ and $\quad \vee \equiv$ or

Boolean Formula $\equiv$ Formula made from $\wedge, \vee$, Literals

Conjunctive Normal Form $\equiv$ Formula $\wedge\, \vee$(Literals)

eg $\underbrace{(x_1 \vee x_2 \vee \bar{x}_3)}_{\text{Clause}}(x_1 \vee x_2 \vee x_4)(\cdots) \ldots$

CNF $\equiv \{ \varphi$ in cnf $\mid \varphi$ is satisfiable $\}$

ie T: Variables $\longrightarrow \{T, F\}$ st $T(\varphi)$ is true

(Cook's Thm) CNF is NP-Complete

No proof! See Kozen