Computational Complexity - Lecture 17.

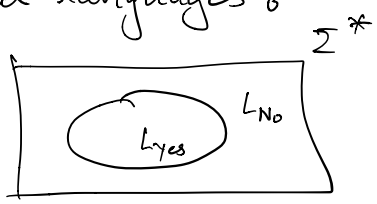
Agenda:
- Promise problems
- Unique SAT.
- Valiant-Vazirani Lemma

} towards understanding the complexity of counting.

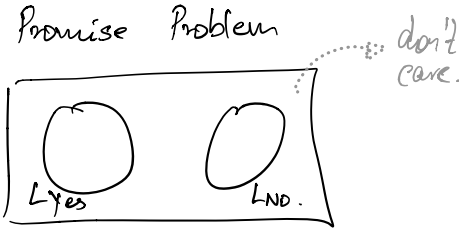
Recap:
- RP, coRP, ZPP, BPP
- Semantic definitions of classes.

What are complete problems?

Typical languages:



Promise Problem



Defn: A promise language is a pair (L_{yes}, L_{NO}) s.t $L_{yes}, L_{NO} \subseteq \Sigma^*$ and $L_{yes} \cap L_{NO} = \emptyset$.

A machine M decides a promise language (L_{yes}, L_{NO}) if
$x \in L_{yes} \Rightarrow M$ accepts x
$x \in L_{NO} \Rightarrow M$ rejects.

$prBPP = \{(L_{yes}, L_{NO}) : \exists$ BPP machine M s.t $x \in L_{yes} \Rightarrow M$ acc. w.p $\geq 2/3$, $x \notin L_{NO} \Rightarrow M$ acc w.p $\leq 1/3\}$

VC of G is a set S s.t every edge has ≥ 1 endpt in S.

Examples: Approx-VC $= L_{yes} \times L_{NO}$

where $L_{yes} = \{(G, k) :$ there is a vertex cover of size $\leq k\}$

$$L_{NO} = \{ (G,k) : \text{smallest vertex cover is of size} \geq 2k \}.$$

▷ Fix an $\varepsilon > 0$.

Circuit Acceptance Prob$^{\varepsilon}$ or $CAP^{\varepsilon} = L_{yes} \times L_{NO}$ where

$$L_{yes} = \{ (C,p) : \Pr_r [C(r)=1] \geq p+\varepsilon \}$$

$$L_{NO} = \{ (C,p) : \Pr_r [C(r)=1] \leq p \}.$$

p - given in binary, few bits.

Claims: (1) $CAP^{\varepsilon} \in$ pr-BPP for any constant $\varepsilon > 0$

(2) CAP^{ε} is ~~BPP~~ hard ~~complete~~, for every constant $\varepsilon > 0$.

Pf: (1): Algo on input (C,p)

- ▷ Pick $r_1, \ldots, r_t$ at random.
- ▷ $k = |\{ i : C(r_i) = 1 \}|$
- ▷ Accept if $k \geq (p + \varepsilon/2) \cdot t$.

Chernoff will tell you that this solves CAP^{ε} w.h.p if t is appropriately chosen.

(2) $L \in BPP \Rightarrow \exists$ Machine $M(-,-)$

$$x \in L \Rightarrow \Pr_r [M(x,r)=1] \geq 2/3$$

$$x \notin L \Rightarrow \Pr_r [M(x,r)=1] \leq 1/3$$

$$x \longmapsto (M(x,\cdot), 1/3) \qquad (\varepsilon \leq 1/3)$$

▷ For a formula φ on n-variables, let $\#SAT(\varphi)$ refer to $|\{ x \in \{0,1\}^n : \varphi(x) = 1 \}|$.

$$L := \{(\varphi, N) \mid \varphi \text{ is a DNF and } \#SAT(\varphi) \geq N\}.$$

$x_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee \ldots$

Obs: L is co-NP-hard.

Pf: TAUT $\longrightarrow (\varphi, 2^n)$.

$(1+\varepsilon)$-Approx DNF:

$$L_{YES} = \{(\varphi, N) : \varphi \text{ is a DNF and } \#SAT(\varphi) \geq N(1+\varepsilon)\}$$

$$L_{NO} = \{(\varphi, N) : \varphi \text{ is a DNF and } \#SAT(\varphi) \leq N\}.$$

Surprising fact: $(1+\varepsilon)$-Approx-DNF $\in$ pr BPP for any constant $\varepsilon > 0$.

▷ Unique SAT (USAT)

$$L_{YES} = \{\varphi : \varphi \text{ has a unique satisfying assignment}\}$$

$$L_{NO} = \{\varphi : \varphi \text{ is unsatisfiable}\}.$$

may assume φ is a 3CNF here

How hard is this problem?

[Valiant-Vazirani] - Probably hard.

$$SAT \leq_{RP} \text{Unique SAT}$$

Randomised reductions: Π & Γ are promise problems.

$\Pi \leq_{BPP} \Gamma$ if there is a randomised algo A s.t

$$x \in \Pi_{yes} \Rightarrow Pr[A(x) \in \Gamma_{yes}] \geq 2/3$$

$$x \in \Pi_{NO} \Rightarrow \Pr\left[A(x) \in \Pi_{No}\right] \geq 2/3$$

Thm: [Valiant-Vazirani] $SAT \leq_{RP} USAT$. That is, there is a randomised algo A s.t

$$\varphi \in SAT \Rightarrow \Pr_A\left[A(\varphi) \in USAT_{yes}\right] \geq 1/8n \quad (\#\text{vars in } \varphi)$$

$$\varphi \notin SAT \Rightarrow \Pr_A\left[A(\varphi) \in USAT_{No}\right] = 1.$$

Pf: $\varphi \longmapsto \varphi \wedge \psi$ → is to filter all but one satisfying assignment.

Suppose we knew, such m that $2^{m-2} \leq \#SAT(\varphi) \leq 2^{m-1}$

Idea: $h : \{0,1\}^n \longrightarrow \{0,1\}^m$. random function

Define $\psi(x) = \mathbb{1}(h(x) = \bar{0})$

$$\varphi \notin SAT \Rightarrow A(\varphi) := \varphi \wedge \psi \in USAT_{No}$$

$\varphi \in SAT$ & $2^{m-2} \leq \#SAT(\varphi) \leq 2^{m-1}$

What is the prob that a unique sat. ass. x satisfies $h(x) = \bar{0}$?

$$\Pr_h\left[\#\{x : \varphi(x)=1 \ \& \ h(x)=0\} = 1\right]$$

$$= \sum_{x \in SAT(\varphi)} \Pr\left[x \text{ is the unique elem in } SAT(\varphi) \text{ with } h(x)=0\right]$$

$$= \sum_{x \in SAT(\varphi)} \left[\Pr\left[h(x)=\bar{0} \wedge \left(y \in SAT(\varphi) \ \& \ y \neq x \Rightarrow h(y) \neq \bar{0}\right)\right]\right]$$

$$= \sum_{x \in SAT(\varphi)} \left[\Pr_h[h(x)=\bar{0}] - \sum_{\substack{y \in SAT(\varphi) \\ y \neq x}} \Pr_h[h(x)=h(y)=\bar{0}]\right]$$

$$= \sum_{x \in SAT(\varphi)} \left[\frac{1}{2^m} - \frac{1}{2^m} \cdot \frac{1}{2^m} \left(\#SAT(\varphi) - 1 \right) \right]$$

$$= \frac{1}{2^m} \cdot \#SAT(\varphi) \left[1 - \frac{1}{2^m} \cdot \left(\#SAT(\varphi) - 1 \right) \right]$$

$$\geq \frac{1}{2^m} \cdot 2^{m-2} \cdot \left[1 - \frac{2^{m-1}}{2^m} \right] = \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}.$$

Algorithm A:

- Pick a random $m \in \{2, \ldots, n+1\}$.
- Pick a random $h : \{0,1\}^n \to \{0,1\}^m$
- Return the formula $\varphi'(x) = \varphi(x) \wedge (h(x) = \bar{0})$.

Obs 1: If $\varphi \in SAT_{NO}$, then $\varphi \in USAT_{NO}$ w.p 1.

Obs 2: If $\varphi \in SAT_{yes}$ then $\varphi \in USAT_{yes}$ w.p $1/8n$.

Pf: Algo is right if we get lucky on m $\to \frac{1}{n}$
& then on h. $\to 1/8$

$\therefore$ Algo succeeds w.p $\geq 1/8n$. $\square$.

Are we done? Picking h requires $2^n \cdot m$ random bits.

$\mathcal{H} = \{ h_a : x \mapsto a \} \to 2^m$.

What did we really need from the way we chose h?

▷ For any $x \in \{0,1\}^n$, $a \in \{0,1\}^m$: $\Pr_h[h(x) = a] = 1/2^m$.

▷ For any $x \neq y \in \{0,1\}^n$, $a, b \in \{0,1\}^m$: $\Pr_h[h(x) = a \ \& \ h(y) = b] = \frac{1}{2^{2m}}$

Defn: $\mathcal{H} \subseteq \{h : \{0,1\}^n \to \{0,1\}^m\}$ is a pairwise indep. hash family if

▷ $\forall x \neq y \in \{0,1\}^n$ & $a, b \in \{0,1\}^m$

$$\Pr_{h \in \mathcal{H}}[h(x) = a \ \& \ h(y) = b] = \frac{1}{2^{2m}}.$$

An example of such a family:

$$\mathcal{H} = \{h_{A,\beta} : A \in \{0,1\}^{m \times n}, \beta \in \{0,1\}^m\}$$

$$h_{A,\beta}: x = Ax + \beta \pmod 2.$$

Lemma: The above is a pairwise indep hash family.

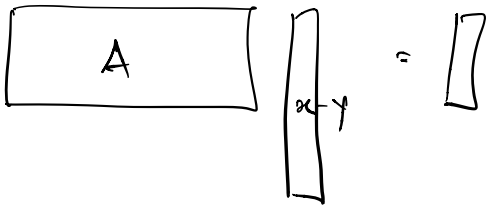
Pf: Fix $x \neq y$ & a, b.

$$\Pr_h[Ax + \beta = a \ \& \ Ay + \beta = b]$$

$$= \Pr_{A,\beta}[A(x-y) = a - b \ \& \ \beta = a - Ax]$$

$$= \frac{1}{2^m} \cdot \frac{1}{2^m}.$$

□.



Imp: If $z_1, \ldots, z_n$ are not all zero,

$$\Pr_r\left[r_1 z_1 + r_2 z_2 + \ldots + r_n z_n = 1 \mod 2\right] = 1/2$$

Algo A':

- Pick $m \in_R \{2, 3, \ldots, n+1\}$.
- Pick $A \in_R \{0,1\}^{m \times n}$ & $\beta \in_R \{0,1\}^m$.
- Return $\psi := \varphi(x) \wedge (Ax \oplus \beta = \bar{0})$.

This is a rand. reduction from SAT $\rightarrow$ Unique SAT.

□.

$\therefore$ USAT $\in$ Pr.P $\Rightarrow$ NP = RP. (unlikely to be true).

Open: Boosting $1/8n$ to even $1/2$. (If PH is infinite then prob $\leq 1/\text{linear}(n)$)

Most of the above problems were about the number of witnesses to something

$$\#P = \left\{ f : \Sigma^* \rightarrow \mathbb{N} : \begin{array}{l}\text{there is non-det poly time machine} \\ M \text{ s.t } f(x) = \#\text{ accepting witnesses} \\ \text{for } M \text{ on } x\end{array} \right\}$$

↓ sharp P, Hash. P, number-P

More in the next few lectures.