Lecture 15: Computational Complexity

Instructor: Prahladh Harsha

Today

- Complexity of Counting
  - ★ Promise Problems
  - ★ Unique SAT (Valiant-Vazirani Thm)
  - ★ Counting Class, #P

---

1. Promise Problems

Qn: Given a graph, check if it is connected.

Underlying language

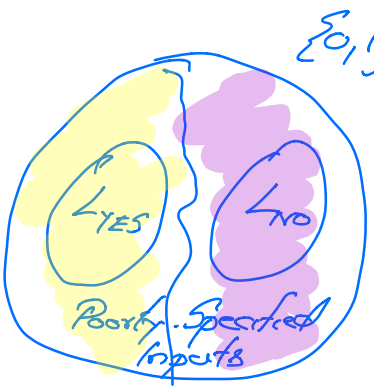
$L = \{\langle G \rangle \mid G \text{ is a graph} \wedge G \text{ is connected}\}$

$\bar{L} = \{\langle G \rangle \mid G \text{ is either not a graph or if a graph, not connected}\}$

Alg: assumes "G is a graph" → promise

$L_{YES} = \{\langle G \rangle \mid G \text{ is a graph} \wedge \text{connected}\}$

$L_{NO} = \{\langle G \rangle \mid G \text{ is a 'disconnected graph'}\}$



Defn: Promise Problem:

$(L_{YES}, L_{NO}) \subseteq \Sigma^* \times \Sigma^*$ s.t $L_{YES} \cap L_{NO} = \phi$

In most cases, promise is easy to detect.

Complete problems for BPP

pr P, pr NP, pr BPP.

Circuit Acceptance Probability Problem (CAP^{ε}) $\varepsilon \in (0, 1/3)$

Promise Problem

$$CAP^{\varepsilon}_{YES} = \{(C,p) \mid C \text{ is a ckt} \; p \in [0,1] \text{ s.t } \Pr_r[C(r)=1] \geq p+\varepsilon\}$$

$$CAP^{\varepsilon}_{NO} = \{(C,p) \mid C \text{ is a ckt} \; p \in [0,1] \text{ s.t } \Pr_r[C(r)=1] \leq p\}$$

Observations:

(1) $CAP^{\varepsilon} \in$ pr.BPP

On input (C,p), choose $r_1 \ldots r_t$ at random & accept if $\#\{i \mid C(r_i)=1\} \geq (p+\frac{\varepsilon}{2})t$ & reject otherwise. (Suffices to choose $t = O(1/\varepsilon^2)$.)

(2). CAP^{ε} is pr.BPP-complete (i.e., every $L \in$ pr.BPP, $L \leq_p CAP^{1/3}$)

$L \in$ prBPP $\Rightarrow \exists$ a prob TM M s.t

$\forall x \in L_{YES} \quad \Pr_r[M(x,r)=1] \geq 2/3$

$\forall x \in L_{NO}, \quad \Pr_r[M(x,r)=1] \leq 1/3$

$x \longmapsto (\langle M(x,\cdot), 1/3 \rangle)$

Qn: Is a planar graph Hamiltonian.

$L_{YES} = \{\langle G \rangle \mid G$ is planar & Hamiltonian$\}$

$L_{NO} = \{\langle G \rangle \mid G$ is planar & non-Hamiltonian$\}$

Possibly easier than checking Hamiltonicity

DNF-counting.

$\#DNF = \{(\varphi, N) \mid \varphi$ is a DNF formula & $\#SAT(\varphi) \geq N\}$

$\#DNF$ is coNP-hard

Promise Version:

(1) $(1+\varepsilon)$-Approx-$\#DNF$ $(\varepsilon \in (0,1))$

$L_{YES} = \{(\varphi, N) \mid \varphi$ is a DNF-formula $\#SAT(\varphi) \geq N(1+\varepsilon)\}$

$L_{NO} = \{(\varphi, N) \mid \varphi$ is a DNF-formula $\#SAT(\varphi) < N\}$

Surprising: $(1+\varepsilon)$-Approx DNF is in prBPP $\forall \varepsilon \in (0,1)$.

Promise is not known to be easy to check.

---

② $(1+\varepsilon)$-Approx Permanent

$L_{YES} = \{(M, \varepsilon) \mid per(M) \geq (1+\varepsilon)^t\}$

$L_{NO} = \{(M, \varepsilon) \mid per(M) \leq t\}$

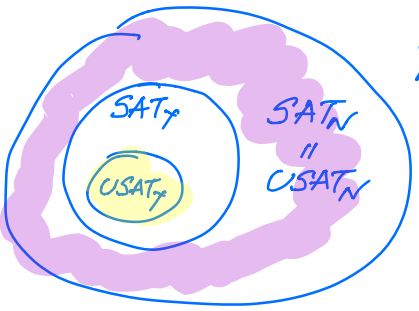
Surprising, $(1+\varepsilon)$-Approx-Per $\in$ pr-BPP $\forall \varepsilon \in (0,1)$

Unique SAT:

USAT: $USAT_Y = \{\varphi \mid \varphi$ is a 3CNF formula φ has a unique sat assign$\}$

$USAT_N = \{\varphi \mid \varphi$ is an unsatisfiable 3CNF formula$\}$

Qn: Is USAT easy?



φ - 3CNF

Valiant-Vazirani: No (Unlikely).

Randomized Red^n

$SAT \leq_{RP} USAT$

Randomized Reductions:

Π, Γ - promise problems

$\Pi \leq_{BPP} \Gamma$ if $\exists$ a randomized alg A s.t

$x \in \Pi_{YES} \Rightarrow Pr[A(x) \in \Gamma_{YES}] \geq \delta(n) + \varepsilon$

$x \in \Pi_{NO} \Rightarrow Pr[A(x) \notin \Gamma_{NO}] \leq \delta(n)$ $\quad (\delta(n) \in (0,1))$

$\varepsilon(n) = 0$; RP-reduction.

Theorem [Valiant-Vazirani] $SAT \leq_{RP} USAT$

More specifically, $\exists$ a rand Alg M s.t

$\varphi \in SAT \Rightarrow Pr[M(\varphi) \in USAT_Y] \geq \frac{1}{8n}$

$\varphi \notin SAT \Rightarrow Pr[M(\varphi) \in USAT_N] = 1$

$(n = \#vars(\varphi))$

Proof: $\varphi \underbrace{\longmapsto}_{\text{randomized}} \psi = \varphi \wedge \underbrace{\varphi'}$

$\varphi' = [f(x) = 1]$

Reduce possibly large # sat assgn to a singleton set.

$n = \#vars(\varphi)$

$2^{m-2} \leq \#sat(\varphi) \leq 2^{m-1}$ for some $m \in \{2, \ldots n+1\}$

(true if $\varphi \in SAT$)

$h: \{0,1\}^n \to \{0,1\}^m$ $\quad \varphi' = [h(x) = \bar{0}]$

$$Pr_h\left[\#\{x \mid h(x) = \bar{0} \wedge \varphi(x) = 1\} = 1\right]$$

$$= \sum_{x \in \varphi^{-1}(1)} Pr_h[x \text{ is the unique string in } \varphi^{-1}(1) \text{ s.t } h(x) = \bar{0}]$$

$$\geq \sum_{x \in \varphi^{-1}(1)} \left(\Pr[h(x) = \bar{0}] - \sum_{y \in \varphi^{-1}(1) - \{x\}} \Pr[h(x) = h(y)] \right)$$

$$= \frac{|\varphi^{-1}(1)|}{2^m} - \frac{|\varphi^{-1}(1)|(|\varphi^{-1}(1)|-1)}{2^{2m}}$$

$$\geq \frac{|\varphi^{-1}(1)|}{2^m} \left(1 - \frac{|\varphi^{-1}(1)|}{2^m}\right) \geq \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$$

$$\left[2^{m-2} \leq |\varphi^{-1}(1)| \leq 2^{m-1} \right.$$

Alg A':

On input φ

1. Pick a random $m \in \{2, \ldots, n+1\}$ $\quad n = \#\text{var}(\varphi)$
2. Pick a $h \in_R \{h : \{0,1\}^n \to \{0,1\}^m\}$
3. Set $\psi = \varphi \wedge [h(x) = \bar{0}]$
4. Output ψ.

Alg A' - (1) satisfies all requirements ✓

(2) Step 2 requires super poly to implement ✗

Qn: Do we need h to be fully random?

Obs: $\mathcal{H}$ = family of fns $h : \{0,1\}^n \to \{0,1\}^m$ st

$\forall\, x \neq y \in \{0,1\}^n$, $a, b \in \{0,1\}^m$: $\Pr[h(x) = a \wedge h(y) = b] = \frac{1}{2^{2m}}$

Defn: $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^m\}$ is a pairwise independent family of hash fns if

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

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

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

$$h_{A,\beta}(x) = Ax + \beta.$$

then $\mathcal{H}_{m,n}$ is pairwise-independent family

Pf:

$$\Pr_{A,\beta}[Ax + \beta = a;\ Ay + \beta = b]$$

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

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

$$= \frac{1}{2^m} \cdot \frac{1}{2^m} = \frac{1}{2^{2m}} \quad \square$$

Alg A: Replace Step 2 w/

(2) Pick $h \leftarrow$ efficiently computable pairwise independent family of hash fns.

Complexity of Counting:

#P : (number-P or sharp-P)

Defn: $f: \{0,1\}^* \to \mathbb{N}$ is in #P if $\exists$ a poly p & a polynomial time alg M s.t

$$f(x) = \#\{y \in \{0,1\}^{p(|x|)} \mid M(x,y)=1\}$$

eg: (1) #SAT = $\varphi \longmapsto$ #SAT(φ)

(2) #MATCHING : $G \longmapsto$ #Matching(G)

Easy: FP - set of f s.t $f: \{0,1\}^* \to \mathbb{N}$ s.t $\exists$ a poly time alg A s.t $\forall x$, $f(x) = A(x)$ (FP - functional P)

Qn #P vs FP

Focus: #P (in next 3 lectures).