Today

- Complete lower bound for parity.
- Hardness of Uniquely satisfiable instances.

Recall Lemma 1

Lemma 1: If $f : \{0,1\}^n \to \{0,1\}$ is computed by a depth d circuit of size s, then there exists a set $S \subseteq \{0,1\}^n$ of size $|S| \geq 3/42^n$ such that $f : S \to \{0,1\}$ computed by a polynomial over $\mathbb{Z}_3$ of degree $(\log s)^{O(d)}$.

Will summarize theorem and proof later.

But first prove lemma.

Proof of Lemma 1

Main steps:

- Assume w.l.o.g. that circuit has only OR gates and NOT gates (blows up size by constant factor).
- Replace each gate by a polynomial.
- NOT gate maps $x \mapsto 1 - x$: Already a polynomial.
- For OR gates, will pick polynomials of degree $O(\log s)$ probabilistically.
- Will show, that for fixed input, any fixed gate computes output correctly w.p. at least $1 - 1/(4s)$. By union bound, whole circuit computes answer correctly w.p. $3/4$.

- Conclude: Exist polynomials, of degree $\log s$, for each gate that compute output correctly on $3/4$ths of the inputs.
- Degree of output function is then $(\log s)^{O(d)}$.

Prob. polynomial for the OR function

Naive answer: $OR(y_1, \dots, y_k) = 1 - \prod_{i=1}^{k}(1 - y_i)$. Answer is always right. But degree is k - too much.

Step 1: Get the answer right w.p. $1/2$ with polynomials of degree 2.

Basic idea: pick $a_1, \dots, a_k \in \mathbb{Z}_3$ at random. $p_{\mathbf{a}}(\mathbf{y}) = \sum_{i=1} a_i y_i$.

Claim 1: $p_{\mathbf{a}}(\mathbf{0}) = 0$.

Claim 2: $\Pr_{\mathbf{a}}[p_{\mathbf{a}}(\mathbf{y}) = 0] \leq 1/3$.

Proof: Let $Q(\mathbf{z}) = \sum_{i=1}^{k} y_i z_i$. Q is a non-zero polynomial of degree 1 in its argument. Evaluation at random $\mathbf{z} = \mathbf{a}$ leaves it non-zero.

Prob. polynomial for the OR function (contd.)

The polynomial $p_{\mathbf{a}}^2$ is always 0 or 1 and computes the OR function on any fixed input w.p. $2/3$.

Pick $\mathbf{a}_1, \dots, \mathbf{a}_l$, and take the OR of polynomials $p_{\mathbf{a}_i}$.

Gives degree 2ℓ polynomial that is right w.p. $1 - (2/3)^\ell$.

What we gained? Will pick $\ell = \log s$ to make degrees logarithmically smaller than fan-in.

What we lost? Not guaranteed to be right.

Prob. polynomial for circuit

- Replace every gate by degree 2ℓ poly randomly.
- Resulting circuit computes a polynomial of degree $(2\ell)^d$.
- Prob. it gets the output wrong (for fixed input) is at most $s(1/3)^\ell$.
- Lemma follows.

Summarizing proof of parity lower bound

- Small depth circuits compute low degree function of most of the output.
- Parity has small depth circuit implies parity has low-degree polynomial representing it on most inputs.
- Parity has small depth circuit implies $\prod_{i=1}^{n} x_i$ has low-degree polynomial representing it on most inputs.
- $\prod_{i=1}^{n} x_i$ has low degree polynomial, implies all Boolean functions represented by low-degree polynomials on most inputs, and thus are in the linear span of small number ($\sum_{i=0}^{n/2+D}$) of monomial functions.

- But the Boolean functions (and in particular the δ_x functions, given by $\delta_x(y) = 1$ if $x = y$ and 0 o.w.) require large basis on large domains.

New topic: Unique satisfiability

Motivation: Hard functions in cryptography.

Diffie-Hellman motivation for cryptography:

The map $(\phi, \mathbf{a}) \mapsto \phi$, where $\mathbf{a}$ satisfies ϕ is easy to compute but hard to invert.

So maybe similarly the map $(p, q) \mapsto p \cdot q$ is also easy to compute but hard to invert.

Can now start building cryptographic primitives based on this assumption.

Issues

Many leaps of faith:

- Specific problem has changed.
- The inputs have to be generated randomly.
- They have to have known "satisfiability".
- etc. etc.

Initial big worry: The map $(\phi, \mathbf{a}) \mapsto \phi$ loses information, while $(p, q) \mapsto p \cdot q$ does not. And NP-hardness requires "loss of information".

Worry goes away, if we know ϕ has only one satisfying assignment. But then is problem as hard?

Formalizing the problem

Promise Problems: Generalize languages L. $\Pi = (\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}})$, $\Pi_{\mathrm{YES}}, \Pi_{\mathrm{NO}} \subseteq \{0,1\}^*$, $\Pi_{\mathrm{YES}} \cap \Pi_{\mathrm{NO}} = \emptyset$.

Algorithm A solves problem Π, if:
(Completeness): $x \in \Pi_{\mathrm{YES}} \Rightarrow A(x)$ accepts.
(Soundness): $x \in \Pi_{\mathrm{NO}} \Rightarrow A(x)$ rejects.

(Can extend to probabilistic algorithms naturally.)

Unique SAT: $\mathrm{USAT} = (\mathrm{USAT}_{\mathrm{YES}}, \mathrm{USAT}_{\mathrm{NO}})$:
$\Pi_{\mathrm{YES}} = \{\phi | \phi$ has exactly one sat. assgmnt.$\}$.
$\Pi_{\mathrm{NO}} = \{\phi | \phi$ has no sat. assgmnts.$\}$.

Formal question: Is $\mathrm{USAT} \in P$? (Does there exist a polytime algorithm A solving USAT)?

Valiant-Vazirani theorem

Theorem: $\mathrm{USAT} \in P$ implies NP $= \mathrm{RP}$.

Proved via the following lemma.

Lemma: There exists a randomized reduction from SAT to USAT.

$\phi \mapsto \psi$ such that $\phi \notin \mathrm{SAT}$ implies $\psi \in \mathrm{USAT}_{\mathrm{NO}}$. $\phi \in \mathrm{SAT}$ implies $\psi \in \mathrm{USAT}_{\mathrm{YES}}$ with probability $1/\mathrm{poly}(n)$.

Again: Question stated without randomness, but answer mentions it! (Can also mention answer without randomness: NP $\subseteq$ P$/_{\mathrm{poly}}$ or PH collapses etc.)

Proof Idea

ψ will have as its clauses, all clauses of ϕ and some more. $(\psi(x) = \phi(x) \wedge \rho(x).)$

So hopefully, will reduce # sat. assgnmts to one.

Furthermore, can put any polynomial time decidable constraint $\rho(x)$ (Since every computation can be transformed into SAT. Exercise coming up.)

So what is $\rho(x)$ going to be?

Proof Idea

Suppose we know there exist M sat. assgnmts to ϕ.

Will pick a random function $h : \{0,1\}^n \to \{0,\ldots,M-1\}$.

Hopefully this distinguished satisfying assignments, and we can let $\rho(x)$ be the condition $h(x) = 0$.

Calculations imply this works out with constant probability.

Caveats in the solution

- How to do this reduction in polytime? Not enough time to represent h!
- Don't know M!

Amendments:

- Will pick pairwise independent hash function.
- Will guess M approximately (to within a factor of 2).

Things will work out!

Pairwise independent hash families

Defn: $H \subseteq \{f : \{0,1\}^n \to \{0,1\}^m\}$ is pairwise independent family if for all $\mathbf{a} \neq \mathbf{b} \in \{0,1\}^n$ and $\mathbf{c}, \mathbf{d} \in \{0,1\}^m$

$$\Pr_{h \in H}[h(\mathbf{a}) = \mathbf{c} \text{ AND } h(\mathbf{b}) = \mathbf{d}] = (1/2^m)^2.$$

H is nice if $h \in H$ can be efficiently sampled and efficiently computed.

Example: Pick $A \in \{0,1\}^{m \times n}$ and $b \in \{0,1\}^m$ at random. Let $h_{A,b}(x) = Ax + b$. Then $H = \{h_{A,b}\}_{A,b}$ is a nice, pairwise independent family.

Proof: Exercise.

Randomized reduction from SAT to USAT

Given ϕ:

- Pick $m \in \{2, \ldots, n+1\}$ at random (and hope that $\#$ satisfying assignments is between 2^{m-2} and 2^{m-1}.)
- Pick h at random from nice p.w.i. family H.
- Let $\psi(x) = \phi(x) \wedge (h(x) = 0)$.
- Output ψ.

Analysis

Let $S = \{x | \phi(x)\}$.

Hope: $2^{m-2} \leq |S| \leq 2^{m-1}$.

Claim: $\Pr_m$[Hope is realized] $\geq 1/n$.

Proof: Claim is true for some $m \in \{2, \ldots, n+1\}$. Prob. we pick that m is $1/n$.

Analysis (contd.)

Claim: $\Pr_h[$ Exactly one $x \in S$ maps to 0 — Hope $] \geq 1/8$.

Define G_x: Event that x maps to 0 and no other $y \in S$ maps to 0.

Prob. we wish to lower bound is (conditioned on Hope):

$\Pr_h[\cup_{x \in S} G_x] = \sum_x \Pr_h[G_x]$

(since G_x's are mutually exclusive).

$\Pr_h[h(x) = 0] = 1/2^m$.

$\Pr_h[h(x) = 0 \text{ and } h(y) = 0] = 1/4^m$.

$\Pr_h[h(x) = 0 \text{ and } \exists y \in S - \{x\}, s.t. h(y) = 0] \leq |S|/4^m$.

$\Pr_h[G_x] \geq 1/2^m - |S|/4^m$.

$\Pr_h[\cup_x G_x] \geq |S|/2^m(1 - |S|/2^m) \geq 1/8$.

Concluding the analysis

With probability $1/8n$ reduction produces ψ with exactly one satisfying assignment. If you can decide satisfiability in such cases then can decide satisfiability probabilistically in all cases.