CAS CS 535: Complexity Theory

Lecturer: Mark Bun Fall 2026

Lecture Notes 17:

Finish $\mathbf{BPP} \subseteq \mathbf{PH}$, Promise Problems, Randomized Reductions, Valiant-Vazirani Theorem

Reading.

- Arora-Barak § 7.4-7.5, 17.4.1

Last time: Error Reduction, $\mathbf{BPP} \subseteq \mathbf{P}_{/\mathbf{poly}}$, $\mathbf{BPP} \subseteq \mathbf{PH}$

1 BPP and the Polynomial Hierarchy

Last time, we gave evidence that randomness cannot be much more powerful than nondeterminism through the following result.

Theorem 1 (Sipser-Gács-Lautemann)**.** $\mathbf{BPP} \subseteq \mathbf{\Sigma}_2^p \cap \mathbf{\Pi}_2^p$.

Last time, we showed that for every language $L \in \mathbf{BPP}$, we can characterize membership in L using the following Σ_2-type check:

$$x \in L \iff \exists u_1, \dots, u_k \in \{0,1\}^m \ \forall r \in \{0,1\}^m \ (\exists i \in [k] \ M(x, r \oplus u_i) = 1).$$

Here, M is the error-reduced $\mathbf{BPP}$ algorithm for L satisfying

$$\Pr_{r \in \{0,1\}^m}[M(x,r) = L(x)] \geq 1 - 2^{-n}$$

for some $m = \mathrm{poly}(n)$ and $k = 2m/n = \mathrm{poly}(n)$.

Correctness of this simulation rests on the following two lemmas.

Lemma 2. *If $|S_x|/2^m \geq 1 - 2^{-n}$, then there exist shifts $u_1, \dots, u_k \in \{0,1\}^m$ such that $\bigcup_{i=1}^k (S_x \oplus u_i) = \{0,1\}^m$.*

Lemma 3. *If $|S_x|/2^m \leq 2^{-n}$, then for all shifts $u_1, \dots, u_k \in \{0,1\}^m$, we have $\bigcup_{i=1}^k (S_x \oplus u_i) \subsetneq \{0,1\}^m$.*

Here, $S_x = \{r \in \{0,1\}^m \mid M(x,r) = 1\}$ is the set of coin toss strings that cause $M(x,r)$ to accept.

Proof of Lemma 2. We want to show that if $|S|/2^m \geq 1 - 2^{-n}$, there exist $u_1, \dots, u_k$ such that $\bigcup_{i=1}^k (S \oplus u_i) = \{0,1\}^m$. We'll do this via the probabilistic method, showing that a random choice of the u's works with positive probability.

Let us choose $u_1, \dots, u_k \in \{0,1\}^m$ uniformly at random. It now suffices to show that

$$\Pr_u \left[\bigcup_{i=1}^k (S \oplus u_i) = \{0,1\}^m \right] > 0$$

or equivalently, that

$$\Pr\left[\exists r \in \{0,1\}^m \text{ s.t. } r \notin \bigcup_{i=1}^{k}(S \oplus u_i)\right] < 1.$$

We bound the left hand side using a union bound by

$$\begin{aligned}
&\sum_{r\in\{0,1\}^m} \Pr\left[r \notin \bigcup_{i=1}^{k}(S \oplus u_i)\right] \\
&\quad= \sum_{r\in\{0,1\}^m} \Pr[r \notin (S \oplus u_1)] \cdot \Pr[r \notin (S \oplus u_2)] \cdot \ldots \cdot \Pr[r \notin (S \oplus u_k)] \quad \text{by independence} \\
&\quad= \sum_{r\in\{0,1\}^m} \prod_{i=1}^{k} \Pr[r \notin (S \oplus u_i)] \\
&\quad\leq \sum_{r\in\{0,1\}^m} (2^{-n})^k \\
&\quad\leq 2^m \cdot 2^{-n\cdot 2m/n} \\
&\quad= 2^{-m} < 1.
\end{aligned}$$

□

Proof of Lemma 3. It suffices to show that if $|S|/2^m \leq 2^{-n}$, then for all sequences of k bitstrings $u_1, \ldots, u_k \in \{0,1\}^m$, where $k = 2m/n$, we have that

$$\left|\bigcup_{i=1}^{k}(S \oplus u_i)\right| < 2^m.$$

to do this, we use the union bound to upper bound the left hand side via

$$\sum_{i=1}^{k} |S \oplus u_i| = k|S| = \frac{2m}{n} \cdot 2^{-n} \cdot 2^m < 2^m$$

for sufficiently large n, since $m = \text{poly}(n)$. □

2 BPP-Complete Problems?

One notion that's been conspicuously missing from our treatment of randomized computation is *complete problems*. Let's see what happens if we try to identify a $\mathbf{BPP}$-complete problem. Consider the following candidate (inspired, e.g., by our first $\mathbf{NP}$-complete problem).

$$\mathsf{BPA}_{\mathsf{TM}} = \{\langle M, x, 1^t\rangle \mid \text{ PTM } M \text{ accepts } x \text{ within } t \text{ steps w.p. } \geq 2/3\}.$$

One can show that this is $\mathbf{BPP}$-hard under poly-time reductions in exactly the same way we did it for $\mathbf{NP}$. The problem is that it isn't known whether this language is in $\mathbf{BPP}$! To illustrate the difficulty, consider the natural candidate algorithm for solving this problem on a PTM:

On input $\langle M, x, 1^t \rangle$:

Simulate M on x for t steps, and accept iff it accepts.

This works fine as long as, for every input x, the acceptance probability of M after t steps is either at least $2/3$ or at most $1/3$. But what if it's, say $1/2$? Then our algorithm should reject at least $2/3$ of the time, but only rejects $1/2$ of the time. Now you might imagine fixing this with repetitions...but what if M's acceptance probability is, say $2/3 - 2^{-|x|}$?

This issue actually seems fundamental, and in fact, there are no known $\mathbf{BPP}$-complete problems. The workaround to this is to relax our definition of $\mathbf{BPP}$, and indeed of languages themselves, to "promise problems."

Definition 4. A promise problem is a pair of sets (Π_Y, Π_N) such that $\Pi_Y, \Pi_N \subseteq \{0,1\}^*$ and $\Pi_Y \cap \Pi_N = \emptyset$.

The way you should interpret this is that Π_Y is the set of "YES" instances to a problem and Π_N is the set of "NO" instances. Anything outside of $\Pi_Y \cup \Pi_N$ is a "don't care" instance. An equivalent way of thinking about this is that, to solve a promise problem (Π_Y, Π_N), you (an algorithm) are given an input x that is *promised* to be in either Π_Y or Π_N, and your job is to determine which is the case. This notion is most interesting when it's hard (possibly even undecidable) to determine whether the promise holds for an input or not.

Example 5. Consider the unambiguous satisfiability problem USAT defined as follows: Given a Boolean formula φ that is promised to be either unsatisfiable or to have exactly one satisfiying assignment, determine which is the case.

To formalize this as a promise problem, we take

$$\mathsf{USAT}_Y = \{\varphi | \text{ CNF } \varphi \text{ has exactly one satisfying assignment}\}$$
$$\mathsf{USAT}_N = \{\varphi | \text{ CNF } \varphi \text{ is unsatisfiable}\}.$$

A language L is a special case of a promise problem where $\Pi_Y = L, \Pi_N = \overline{L}$, and there are no "don't care"s.

Definition 6. A promise problem (Π_Y, Π_N) is in $\mathbf{PromiseBPP}$ if there exists a polynomial-time PTM M such that

$$x \in \Pi_Y \implies \Pr[M(x) = 1] \geq 2/3$$
$$x \in \Pi_N \implies \Pr[M(x) = 1] \leq 1/3.$$

With this definition in hand, we can now successfully describe $\mathbf{PromiseBPP}$-complete problems. One such problem is obtained by modifying $\mathsf{BPA}_{\mathsf{TM}}$ to include the promise that the TM's acceptance probability is bounded away from $1/2$. A similar problem that's usually easier to work with is the "Circuit Acceptance Probability Problem", which roughly captures the following question: Given a circuit C, what is the probability that C accepts a random input x?

Definition 7. The Circuit Acceptance Probability Problem (CAPP) is the promise problem defined as follows.

$$\Pi_Y = \{\langle C, p \rangle \mid \Pr_x[C(x) = 1] \geq p + 1/10\}$$
$$\Pi_N = \{\langle C, p \rangle \mid \Pr_x[C(x) = 1] \leq p - 1/10\}.$$

3 Randomized Reductions and Valiant-Vazirani

(Deterministic) poly-time reductions help us study questions like "$\mathbf{NP} \stackrel{?}{=} \mathbf{P}$" and "$\mathbf{BPP} \stackrel{?}{=} \mathbf{P}$" and their relationship to other problems. But how about questions like "$\mathbf{NP} \stackrel{?}{=} \mathbf{RP}$"?

Recall the Unambiguous Satisfiability problem

$$\begin{aligned}\mathsf{USAT}_Y &= \{\varphi | \text{ CNF } \varphi \text{ has exactly one satisfying assignment}\}\\ \mathsf{USAT}_N &= \{\varphi | \text{ CNF } \varphi \text{ is unsatisfiable}\}.\end{aligned}$$

At first glance, you might imagine that USAT is easier than SAT itself. Imagine you're given a YES (satisfiable) instance φ, and you're trying to find a satisfying assignment. If you're promised that φ has at most one satisfying assignment, you might imagine being able to use this information to guide your search.

The Valiant-Vazirani Theorem shows that this is <u>not</u> the case, and that (at least for randomized algorithms), Unambiguous SAT is at least as hard as SAT itself.

Theorem 8 (Valiant-Vazirani)**.** *There exists a poly-time randomized algorithm A mapping n-variable CNF formulas to n-variable CNF formulas such that*

$$\begin{aligned}\varphi \in \mathsf{SAT} &\implies \Pr[A(\varphi) \in \mathsf{USAT}_Y] \geq \frac{1}{8n}\\ \varphi \notin \mathsf{SAT} &\implies \Pr[A(\varphi) \in \mathsf{USAT}_N] = 1.\end{aligned}$$

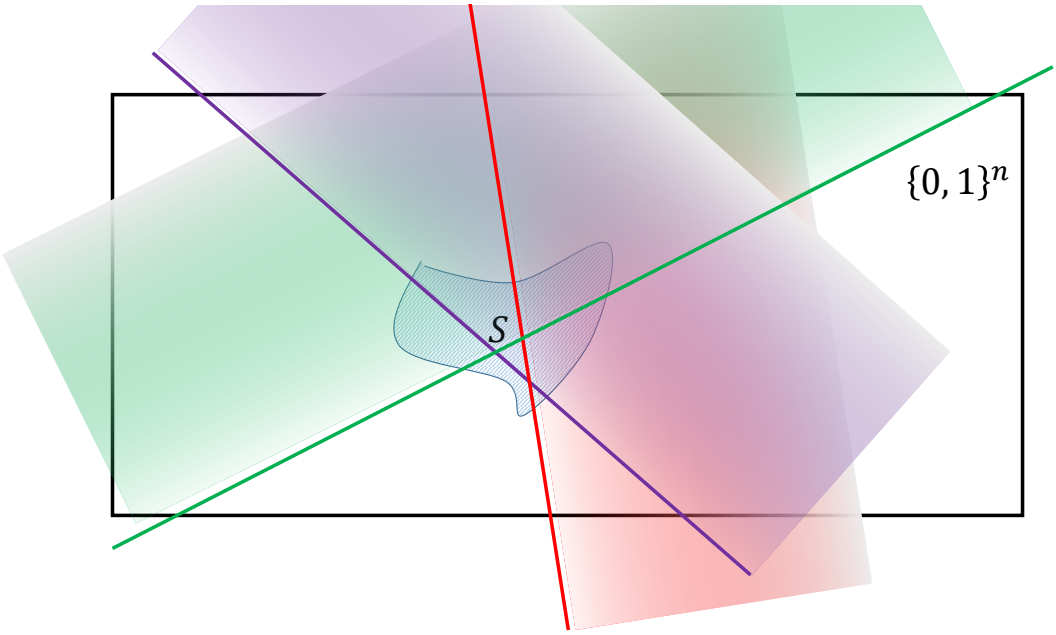
This result has a number of interesting consequences. (Left as exercises.) One is the headline statement about the hardness of USAT:

Corollary 9. $\mathbf{NP} = \mathbf{RP}$ *if and only if* $\mathsf{USAT} \in \mathbf{PrRP}$.

Another is an application to search-to-decision for SAT. Recall that the search-to-decision reduction we saw builds a satisfying assignment to a formula φ by setting one variable at a time, solving the SAT decision problem on an adaptively chosen sequence of inputs. If the satisfying assignment is unique, then these inputs can be chosen *non-adaptively*, all in a single batch; hence, Valiant-Vazirani gives a non-adaptive randomized search-to-decision reduction for SAT.

Proof idea. The reduction A will add randomly chosen additional constraints to φ to eliminate all but 1 satisfying assignment (if one exists).

In pictures. Suppose we have some additional information about our input formula φ: Either it's unsatisfiable, or that the number of satisfying assignments is between 2^{k-1} and 2^k.



Let S be the set of satisfying assignments to φ.

We'll add new constraints $C_1, \ldots, C_k$ which each (almost) independently eliminate a random half of $\{0,1\}^n$ as possible satisfying assignments. The expected size of the portion of S that survives is $2^{-k} \cdot |S| \in [1/2, 1]$. We'll show that $|S| = 1$ with constant probability.

One remaining issue: How do we know how to set k? We don't know! We'll just guess it at random from $1, \ldots, n$, which will be correct with probability $1/n$.

3.1 Some Technical Tools

The details of the proof actually go somewhat differently than the sketch outlined above, in that instead of adding constraints one at a time, we'll "do it all in one go" and sample a single random constraint that eliminates all but a 2^{-k} fraction of satisfying assignments.

The first idea we might try to do this is to let $h : \{0,1\}^n \to \{0,1\}^k$ be a uniformly random "hash" function. Then the constraint "$h(x) = 0^k$" is indeed satisfied by a 2^{-k} fraction of points in S. The problem is that a truly random function is very complex to describe and evaluate; the truth table takes $k \cdot 2^n$ bits to write down, and Shannon's counting argument shows that almost all such functions cannot be described by circuits substantially smaller than this.

The solution is to not choose a h truly uniformly at random, but in a manner that is "just random enough." Specifically, we'll draw h from a *pairwise independent* family of hash functions.

Definition 10. Let $k \leq n$. A set of functions $\mathcal{H} = \{h : \{0,1\}^n \to \{0,1\}^k\}$ is pairwise independent if for all $x \neq x' \in \{0,1\}^n$ and all $y, y' \in \{0,1\}^k$, we have

$$\Pr_{h \leftarrow \mathcal{H}}[h(x) = y \wedge h(x') = y'] = \Pr_{h \leftarrow \mathcal{H}}[h(x) = y] \cdot \Pr_{h \leftarrow \mathcal{H}}[h(x') = y'] = 2^{-k} \cdot 2^{-k} = 2^{-2k}.$$

That is, a function h drawn from a pairwise independent family "looks random" as long as you restrict your attention to a single pair of inputs.

The upshot of pairwise independence is that we have constructions of such families that are easy to sample from and easy to compute.

Example 11. Let $k \leq n$ and define

$$\mathcal{H} = \{h_{A,b} \mid A \in \{0,1\}^{n \times k}, b \in \{0,1\}^k\}$$

where $h_{A,b} : \{0,1\}^n \to \{0,1\}^k$ is defined by $h_{A,b}(x) = Ax + b$, where the arithmetic is done mod 2. This is a pairwise independent hash family. (Proof left as an exercise.)

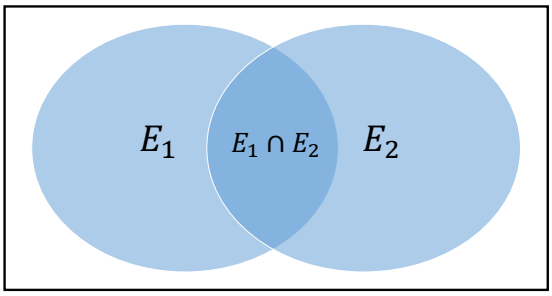
Observe that sampling $h \leftarrow \mathcal{H}$ takes time $\mathrm{poly}(n)$, and each $h \in \mathcal{H}$ can be described by a $\mathrm{poly}(n)$-size circuit (in fact, with each output bit describable by a CNF), and hence can be evaluated in polynomial time.

The other tool we need is the Inclusion-Exclusion Principle. To motivate this, let's first recall

Union Bound: $\Pr\left[\bigcup_{i=1}^n E_i\right] \leq \sum_{i=1}^n \Pr[E_i]$

Inclusion-Exclusion: $\Pr\left[\bigcup_{i=1}^n E_i\right] \geq \sum_{i=1}^n \Pr[E_i] - \sum_{1 \leq i < j \leq n} \Pr[E_i \cap E_j]$.

Proof by picture: In the case of 2 sets, we actually have equality: $\Pr[E_1 \cup E_2] = \Pr[E_1] + \Pr[E_2] - \Pr[E_1 \cap E_2]$.



The general case follows by induction, with the ordinary union bound facilitating the inductive step.

3.2 Back to Valiant-Vazirani

The proof relies on the following key lemma.

Lemma 12 (Isolation Lemma)**.** *If $\mathcal{H}$ is a pairwise independent family of functions of the form $h : \{0,1\}^n \to \{0,1\}^k$, and $S \subseteq \{0,1\}^n$ satisfies $2^{k-2} \leq |S| \leq 2^{k-1}$, then*

$$\Pr_{h \leftarrow \mathcal{H}}[\textit{There is a unique } x \in S \textit{ s.t. } h(x) = 0^k] \geq \frac{1}{8}.$$

Proof. We write the probability we're interested in as

$$\begin{aligned}
&\sum_{x\in S} \Pr[h(x) = 0^k \wedge (\forall x' \in S \setminus \{x\}, h(x') \neq 0^k)] \\
&\qquad = \sum_{x\in S} \Pr[h(x) = 0^k] - \Pr[h(x) = 0^k \wedge (\exists x' \in S \setminus \{x\}, h(x') = 0^k)] \\
&\qquad \geq \sum_{x\in S} \left(2^{-k} - \sum_{x'\in S\setminus\{x\}} \Pr[h(x) = 0^k \wedge h(x') = 0^k] \right) && \text{(union bound)} \\
&\qquad \geq \sum_{x\in S} \left(2^{-k} - \sum_{x'\in S\setminus\{x\}} 2^{-2k} \right) && \text{(pairwise independence)} \\
&\qquad = |S| \left(2^{-k} - (|S|-1)2^{-2k} \right) \\
&\qquad \geq 2^{k-2}(2^{-k} - 2^{k-1}2^{-2k}) \\
&\qquad = 2^{k-2}2^{-k-1} = 2^{-3} = 1/8.
\end{aligned}$$

□

Proof of Valiant-Vazirani. Let $\mathcal{H}_{n,k}$ denote an efficiently sampleable/computable pairwise independent family of functions $h : \{0,1\}^n \to \{0,1\}^k$. The following randomized algorithm A reduces SAT to USAT.

On input φ:

1. Sample $k \in \{2, \ldots, n+1\}$ uniformly at random
2. Sample $h \leftarrow \mathcal{H}_{n,k}$
3. Output a CNF ψ corresponding to $\varphi(x) \wedge (h(x) = 0^k)$.[1]

By construction, this reduction runs in randomized poly-time. For correctness, we check:

- If $\varphi \notin \mathsf{SAT}$, then ψ is definitely unsatisfiable, so $\psi \in \mathsf{USAT}_N$ with probability 1.
- If $\varphi \in \mathsf{SAT}$, then $\Pr[\psi \in \mathsf{USAT}_Y] \geq \Pr[2^{k-2} \leq |\psi^{-1}(1)| \leq 2^{k-1}] \cdot \frac{1}{8} = \frac{1}{8n}$.

□

[1] The constraint $h(x) = 0^k$ may not literally be expressible using a small CNF. But one can introduce auxiliary variables y to create a formula $\psi(x, y)$ that is uniquely satisfiable if and only if $\varphi(x) \wedge (h(x) = 0^k)$ is, the same way one reduces SAT to $3\mathsf{SAT}$.