Notes on Complexity Theory	Last updated: November, 2011
Lecture 24	
Jonathan Katz	

1 The Complexity of Counting

We explore three results related to hardness of counting. Interestingly, at their core each of these results relies on a simple — yet powerful — technique due to Valiant and Vazirani.

1.1 Hardness of Unique-SAT

Does SAT become any easier if we are guaranteed that the formula we are given has at most *one* solution? Alternately, if we are guaranteed that a given boolean formula has a unique solution does it become any easier to find it? We show here that this is not likely to be the case.

Define the following promise problem:

$$\begin{aligned} \mathsf{USAT} &\stackrel{\text{def}}{=} \{\phi : \phi \text{ has exactly one satisfying assignment}\} \\ \overline{\mathsf{USAT}} &\stackrel{\text{def}}{=} \{\phi : \phi \text{ is unsatisfiable}\}. \end{aligned}$$

Clearly, this problem is in promise-$\mathcal{NP}$. We show that if it is in promise-$\mathcal{P}$, then $\mathcal{NP} = \mathcal{RP}$. We begin with a lemma about pairwise-independent hashing.

Lemma 1 *Let $S \subseteq \{0,1\}^n$ be an arbitrary set with $2^m \leq |S| \leq 2^{m+1}$, and let $H_{n,m+2}$ be a family of pairwise-independent hash functions mapping $\{0,1\}^n$ to $\{0,1\}^{m+2}$. Then*

$$\Pr_{h \in H_{n,m+2}}[\textit{there is a unique } x \in S \textit{ with } h(x) = 0^{m+2}] \geq 1/8.$$

Proof Let $\mathbf{0} \stackrel{\text{def}}{=} 0^{m+2}$, and let $p \stackrel{\text{def}}{=} 2^{-(m+2)}$. Let N be the random variable (over choice of random $h \in H_{n,m+2}$) denoting the number of $x \in S$ for which $h(x) = \mathbf{0}$. Using the inclusion/exclusion principle, we have

$$\begin{aligned} \Pr[N \geq 1] &\geq \sum_{x \in S} \Pr[h(x) = \mathbf{0}] - \frac{1}{2} \cdot \sum_{x \neq x' \in S} \Pr[h(x) = h(x') = \mathbf{0}] \\ &= |S| \cdot p - \binom{|S|}{2} p^2, \end{aligned}$$

while $\Pr[N \geq 2] \leq \sum_{x \neq x' \in S} \Pr[h(x) = h(x') = \mathbf{0}] = \binom{|S|}{2} p^2$. So

$$\Pr[N = 1] = \Pr[N \geq 1] - \Pr[N \geq 2] \geq |S| \cdot p - 2 \cdot \binom{|S|}{2} p^2 \geq |S| p - |S|^2 p^2 \geq 1/8,$$

using the fact that $|S| \cdot p \in [\frac{1}{4}, \frac{1}{2}]$. ■

Theorem 2 (Valiant-Vazirani) *If* $(\mathsf{USAT}, \overline{\mathsf{USAT}})$ *is in promise-*$\mathcal{RP}$*, then* $\mathcal{NP} = \mathcal{RP}$*.*

Proof If $(\mathsf{USAT}, \overline{\mathsf{USAT}})$ is in promise-$\mathcal{RP}$, then there is a probabilistic polynomial-time algorithm A such that

$$\begin{aligned} \phi \in \mathsf{USAT} \quad &\Rightarrow \quad \Pr[A(\phi) = 1] \geq 1/2 \\ \phi \in \overline{\mathsf{USAT}} \quad &\Rightarrow \quad \Pr[A(\phi) = 1] = 0. \end{aligned}$$

We design a probabilistic polynomial-time algorithm B for SAT as follows: on input an n-variable boolean formula ϕ, first choose uniform $m \in \{0, \ldots, n-1\}$. Then choose random $h \leftarrow H_{n,m+2}$. Using the Cook-Levin reduction, rewrite the expression $\psi(x) \stackrel{\text{def}}{=} \big(\phi(x) \wedge \big(h(x) = 0^{m+2}\big)\big)$ as a boolean formula $\phi'(x, z)$, using additional variables z if necessary. (Since h is efficiently computable, the size of ϕ' will be polynomial in the size of ϕ. Furthermore, the number of satisfying assignments to $\phi'(x, z)$ will be the same as the number of satisfying assignments of ψ.) Output $A(\phi')$.

If ϕ is not satisfiable then ϕ' is not satisfiable, so A (and hence B) always outputs 0. If ϕ is satisfiable, with S denoting the set of satisfying assignments, then with probability $1/n$ the value of m chosen by B is such that $2^m \leq |S| \leq 2^{m+1}$. In that case, Lemma 1 shows that with probability at least $1/8$ the formula ϕ' will have a unique satisfying assignment, in which case A outputs 1 with probability at least $1/2$. We conclude that when ϕ is satisfiable then B outputs 1 with probability at least $1/16n$. ■

1.2 Approximate Counting, and Relating $\#\mathcal{P}$ to $\mathcal{NP}$

$\#\mathcal{P}$ is clearly not weaker than $\mathcal{NP}$, since if we can count solutions then we can certainly tell if any exist. Although $\#\mathcal{P}$ is (in some sense) "harder" than $\mathcal{NP}$, we show that any problem in $\#\mathcal{P}$ can be probabilistically *approximated* in polynomial time using an $\mathcal{NP}$ oracle. (This is reminiscent of the problem of reducing search to decision, except that here we are reducing *counting* the number of witness to the decision problem of whether or not a witness exists. Also, we are only obtaining an approximation, and we use randomization.) We focus on the $\#\mathcal{P}$-complete problem $\#\mathsf{SAT}$. Let $\#\mathsf{SAT}(\phi)$ denote the number of satisfying assignments of a boolean formula ϕ. We show that for any polynomial p there exists a PPT algorithm A such that

$$\Pr\left[\#\mathsf{SAT}(\phi) \cdot \left(1 - \frac{1}{p(|\phi|)}\right) \leq A^{\mathcal{NP}}(\phi) \leq \#\mathsf{SAT}(\phi) \cdot \left(1 + \frac{1}{p(|\phi|)}\right)\right] \geq 1 - 2^{-p(|\phi|)}\,; \tag{1}$$

that is, A approximates $\#\mathsf{SAT}(\phi)$ (the number of satisfying assignments to ϕ) to within a factor $(1 \pm \frac{1}{p(|\phi|)})$ with high probability.

The first observation is that it suffices to obtain a constant-factor approximation. Indeed, say we have an algorithm B such that

$$\frac{1}{64} \cdot \#\mathsf{SAT}(\phi) \leq B^{\mathcal{NP}}(\phi) \leq 64 \cdot \#\mathsf{SAT}(\phi). \tag{2}$$

(For simplicity we assume B always outputs an approximation satisfying the above; any failure probability of B propagates in the obvious way.) We can construct an algorithm A satisfying (1) as follows: on input ϕ, set $q = \log 64 \cdot p(|\phi|)$ and compute $t = B(\phi')$ where

$$\phi' \stackrel{\text{def}}{=} \textstyle\bigwedge_{i=1}^{q} \phi(x_i)\,,$$

and the x_i denote independent sets of variables. A then outputs $t^{1/q}$.

Letting N (resp., N') denote the number of satisfying assignments to ϕ (resp., ϕ'), note that $N' = N^q$. Since t satisfies $\frac{1}{64} \cdot N' \leq t \leq 64 \cdot N'$, the output of A lies in the range

$$\left[2^{-1/p(|\phi|)} \cdot N, \quad 2^{1/p(|\phi|)} \cdot N\right] \subseteq \left[\left(1 - \frac{1}{p(|\phi|)}\right) \cdot N, \quad \left(1 + \frac{1}{p(|\phi|)}\right) \cdot N\right],$$

as desired. In the last step, we use the following inequalities which hold for all $x \geq 1$:

$$\left(\frac{1}{2}\right)^{1/x} \geq \left(1 - \frac{1}{x}\right) \quad \text{and} \quad 2^{1/x} \leq \left(1 + \frac{1}{x}\right).$$

The next observation is that we can obtain a constant-factor approximation by solving the promise problem (Π_Y, Π_N) given by:

$$\begin{aligned} \Pi_Y &\stackrel{\text{def}}{=} \{(\phi, k) \mid \#\mathsf{SAT}(\phi) > 8k\} \\ \Pi_N &\stackrel{\text{def}}{=} \{(\phi, k) \mid \#\mathsf{SAT}(\phi) < k/8\}. \end{aligned}$$

Given an algorithm C solving this promise problem, we can construct an algorithm B satisfying (2) as follows. (Once again, we assume C is deterministic; if C errs with non-zero probability we can handle it in the straightforward way.) On input ϕ do:

- Set $i = 0$.
- While $M((\phi, 8^i)) = 1$, increment i.
- Return $8^{i-\frac{1}{2}}$.

Let i^* be the value of i at the end of the algorithm, and set $\alpha = \log_8 \#\mathsf{SAT}(\phi)$. In the second step, we know that $M((\phi, 8^i))$ outputs 1 as long as $\#\mathsf{SAT}(\phi) > 8^{i+1}$ or, equivalently, $\alpha > i+1$. So we end up with an i^* satisfying $i^* \geq \alpha - 1$. We also know that $M((\phi, 8^i))$ will output 0 whenever $i > \alpha + 1$ and so the algorithm above must stop at the first (integer) i to satisfy this. Thus, $i^* \leq \alpha + 2$. Putting this together, we see that our output value satisfies:

$$\#\mathsf{SAT}(\phi)/64 < 8^{i^* - \frac{1}{2}} < 64 \cdot \#\mathsf{SAT}(\phi),$$

as desired. (Note that we assume nothing about the behavior of M when $(\phi, 8^i) \notin \Pi_Y \cup \Pi_N$.)

Finally, we show that we can probabilistically solve (Π_Y, Π_N) using an $\mathcal{NP}$ oracle. This just uses another application of the Valiant-Vazirani technique. Here we rely on the following lemma:

Lemma 3 *Let $H_{n,m}$ be a family of pairwise-independent hash functions mapping $\{0,1\}^n$ to $\{0,1\}^m$, and let $\varepsilon > 0$. Let $S \subseteq \{0,1\}^n$ be arbitrary with $|S| \geq \varepsilon^{-3} \cdot 2^m$. Then:*

$$\Pr_{h \in H_{n,m}}\left[(1-\varepsilon) \cdot \frac{|S|}{2^m} \leq |\{x \in S \mid h(x) = 0^m\}| \leq (1+\varepsilon) \cdot \frac{|S|}{2^m}\right] > 1 - \varepsilon.$$

Proof Define for each $x \in S$ an indicator random variable δ_x such that $\delta_x = 1$ iff $h(x) = 0^m$ (and 0 otherwise). Note that the δ_x are pairwise independent random variables with expectation 2^{-m} and variance $2^{-m} \cdot (1 - 2^{-m})$. Let $Y \stackrel{\text{def}}{=} \sum_{x \in S} \delta_x = |\{x \in S \mid h(x) = 0^m\}|$. The expectation of Y is $|S|/2^m$, and its variance is $\frac{|S|}{2^m} \cdot (1 - 2^{-m})$ (using pairwise independent of the δ_x). Using Chebychev's inequality, we obtain:

$$\begin{aligned}
\Pr\left[(1-\varepsilon) \cdot \mathbf{Exp}[Y] \leq Y \leq (1+\varepsilon) \cdot \mathbf{Exp}[Y]\right] &= \Pr\left[|Y - \mathbf{Exp}[Y]| \leq \varepsilon \cdot \mathbf{Exp}[Y]\right] \\
&\geq 1 - \frac{\mathbf{Var}[Y]}{(\varepsilon \cdot \mathbf{Exp}[Y])^2} \\
&= 1 - \frac{(1 - 2^{-m}) \cdot 2^m}{\varepsilon^2 \cdot |S|},
\end{aligned}$$

which is greater than $1 - \varepsilon$ for $|S|$ as stated in the proposition. ■

The algorithm solving (Π_Y, Π_N) is as follows. On input (ϕ, k) with $k > 1$ (note that a solution is trivial for $k = 1$), set $m = \lfloor \log k \rfloor$, choose a random h from $H_{n,m}$, and then query the $\mathcal{NP}$ oracle on the statement $\phi'(x) \stackrel{\text{def}}{=} (\phi(x) \wedge (h(x) = 0^m))$ and output the result. An analysis follows.

Case 1: $(\phi, k) \in \Pi_Y$, so $\#\mathsf{SAT}(\phi) > 8k$. Let $S_\phi = \{x \mid \phi(x) = 1\}$. Then $|S_\phi| > 8k \geq 8 \cdot 2^m$. So:

$$\begin{aligned}
\Pr\left[\phi' \in \mathsf{SAT}\right] &= \Pr\left[\{x \in S_\phi : h(x) = 0^m\} \neq \emptyset\right] \\
&\geq \Pr\left[\,|\{x \in S_\phi : h(x) = 0^m\}| \geq 4\right] \geq \frac{1}{2},
\end{aligned}$$

which we obtain by applying Lemma 3 with $\varepsilon = \frac{1}{2}$.

Case 2: $(\phi, k) \in \Pi_N$, so $\#\mathsf{SAT}(\phi) < k/8$. Let S_ϕ be as before. Now $|S_\phi| < k/8 \leq 2^m/4$. So:

$$\begin{aligned}
\Pr\left[\phi' \in \mathsf{SAT}\right] &= \Pr\left[\{x \in S_\phi : h(x) = 0^m\} \neq \emptyset\right] \\
&\leq \sum_{x \in S_\phi} \Pr\left[h(x) = 0^m\right] \\
&< \frac{2^m}{4} \cdot 2^{-m} = \frac{1}{4},
\end{aligned}$$

where we have applied a union bound in the second step. We thus have a constant gap in the acceptance probabilities when $\phi \in \Pi_Y$ vs. when $\phi \in \Pi_N$; this gap can be amplified as usual.

1.3 Toda's Theorem

The previous section may suggest that $\#\mathcal{P}$ is not "much stronger" than $\mathcal{NP}$, in the sense that $\#\mathcal{P}$ can be closely approximated given access to an $\mathcal{NP}$ oracle. Here, we examine this more closely, and show the opposite: while *approximating* the number of solutions may be "easy" (given an $\mathcal{NP}$ oracle), determining the *exact* number of solutions appears to be much more difficult.

Toward this, we first introduce the class $\oplus\mathcal{P}$ ("parity $\mathcal{P}$"):

Definition 1 *A function $f : \{0,1\}^* \to \{0,1\}$ is in $\oplus\mathcal{P}$ if there is a Turing machine M running in time polynomial in its first input such that* $f(x) = \#M(x) \bmod 2$.

Note that if $f \in \oplus\mathcal{P}$ then f is just the least-significant bit of some function $\bar{f} \in \#\mathcal{P}$. The class $\oplus\mathcal{P}$ does not represent any "natural" computational problem. Nevertheless, it is natural to study it because (1) it nicely encapsulates the difficulty of computing functions in $\#\mathcal{P}$ *exactly* (i.e., down to the least-significant bit), and (2) it can be seen as a generalization of the unique-SAT example discussed previously (where the difficulty there is determining whether a boolean formula has 0 solutions or 1 solution).

A function $g \in \oplus\mathcal{P}$ is $\oplus\mathcal{P}$-complete (under parsimonious reductions) if for every $f \in \#\mathcal{P}$ there is a polynomial-time computable function ϕ such that $f(x) = g(\phi(x))$ for all x. If $\bar{g} \in \#\mathcal{P}$ is $\#\mathcal{P}$-complete under parsimonious reductions, then the least-significant bit of $\bar{g}$ is $\oplus\mathcal{P}$-complete under parsimonious reductions. For notational purposes it is easier to treat $\oplus\mathcal{P}$ as a language class, in the natural way. (In particular, if $f \in \oplus\mathcal{P}$ as above then we obtain the language $L_f = \{x : f(x) = 1\}$.) In this sense, $\oplus\mathcal{P}$-completeness is just the usual notion of a Karp reduction. Not surprisingly,

$$\oplus\mathsf{SAT} \stackrel{\text{def}}{=} \{\phi : \phi \text{ has an odd number of satisfying assignments}\}$$

is $\oplus\mathcal{P}$-complete. Note that $\phi \in \oplus\mathsf{SAT}$ iff $\sum_x \phi(x) = 1 \bmod 2$ (where we let $\phi(x) = 1$ if x satisfies ϕ, and $\phi(x) = 0$ otherwise).

A useful feature of $\oplus\mathcal{P}$ is that it can be "manipulated" arithmetically in the following sense:

- $(\phi \in \oplus\mathsf{SAT}) \bigwedge (\phi' \in \oplus\mathsf{SAT}) \Leftrightarrow \phi \wedge \phi' \in \oplus\mathsf{SAT}$. This follows because

$$\sum_{x,x'} \phi(x) \wedge \phi'(x') = \sum_{x,x'} \phi(x) \cdot \phi'(x') = \left(\sum_x \phi(x)\right) \cdot \left(\sum_{x'} \phi'(x')\right),$$

  and hence the number of satisfying assignments of $\phi \wedge \phi'$ is the product of the number of satisfying assignments of each of ϕ, ϕ'.

- Let ϕ, ϕ' be formulas, where without loss of generality we assume they both have the same number n of variables (this can always be enforced, without changing the number of satisfying assignments, by "padding" with additional variables that are forced to be 0 in any satisfying assignment). Define the formula $\phi + \phi'$ on $n + 1$ variables as follows:

$$(\phi + \phi')(z, x) = \big((z = 0) \wedge \phi(x)\big) \vee \big((z = 1) \wedge \phi'(x)\big).$$

  Note that the number of satisfying assignments of $\phi+\phi'$ is the sum of the number of satisfying assignments of each of ϕ, ϕ'. In particular, $(\phi + \phi') \in \oplus\mathsf{SAT}$ iff *exactly one* of $\phi, \phi' \in \oplus\mathsf{SAT}$.

- Let '1' stand for some canonical boolean formula that has exactly one satisfying assignment. Then $\phi \notin \oplus\mathsf{SAT} \Leftrightarrow (\phi + 1) \in \oplus\mathsf{SAT}$.

- Finally, $(\phi \in \oplus\mathsf{SAT}) \bigvee (\phi' \in \oplus\mathsf{SAT}) \Leftrightarrow (\phi + 1) \bigwedge (\phi' + 1) + 1 \in \oplus\mathsf{SAT}$.

We use the above tools to prove the following result:

Theorem 4 (Toda's theorem) $\mathsf{PH} \subseteq \mathcal{P}^{\#\mathcal{P}}$.

The proof of Toda's theorem proceeds in two steps, each of which is a theorem in its own right.

Theorem 5 *Fix any $c \in \mathbb{N}$. There is a probabilistic polynomial-time algorithm A such that for any quantified boolean formula ψ with c alternations, the following holds:*

$$\begin{aligned} \psi \textit{ is true} \quad &\Rightarrow \quad \Pr[A(1^m, \psi) \in \oplus\mathsf{SAT}] \geq 1 - 2^{-m} \\ \psi \textit{ is false} \quad &\Rightarrow \quad \Pr[A(1^m, \psi) \in \oplus\mathsf{SAT}] \leq 2^{-m}. \end{aligned}$$

As a corollary, $\mathsf{PH} \subseteq \mathcal{BPP}^{\oplus\mathcal{P}}$.

Proof It suffices to consider quantified boolean formulae beginning with an '$\exists$' quantifier. Indeed, say we have some algorithm A' that works in that case. If ψ begins with a '$\forall$' quantifier then $\neg\psi$ can be written as a quantified boolean formula beginning with an '$\exists$' quantifier; moreover, ψ is true iff $\neg\psi$ is false. Thus, defining $A(1^m, \psi)$ to return $A'(1^m, \neg\psi) + 1$ gives the desired result.

The proof is by induction on c. For $c = 1$ we apply the Valiant-Vazirani result plus amplification. Specifically, let ψ be a statement with only a single $\exists$ quantifier. The Valiant-Vazirani technique gives us a probabilistic polynomial-time algorithm B such that:

$$\begin{aligned} \psi \text{ is true} \quad &\Rightarrow \quad \Pr[B(\psi) \in \oplus\mathsf{SAT}] \geq 1/8n \\ \psi \text{ is false} \quad &\Rightarrow \quad \Pr[B(\psi) \in \oplus\mathsf{SAT}] = 0, \end{aligned}$$

where n is the number of variables in ψ. Algorithm $A(1^m, \psi)$ runs $B(\psi)$ a total of $\ell = O(mn)$ times to obtain formulae $\phi_1, \ldots, \phi_\ell$; it then outputs the formula $\Phi = 1 + \bigwedge_i (\phi_i + 1)$. Note that $\bigvee_i (\phi_i \in \oplus\mathsf{SAT}) \Leftrightarrow \Phi \in \oplus\mathsf{SAT}$; hence

$$\begin{aligned} \psi \text{ is true} \quad &\Rightarrow \quad \Pr[A(1^m, \psi) \in \oplus\mathsf{SAT}] \geq 1 - 2^{-m} \\ \psi \text{ is false} \quad &\Rightarrow \quad \Pr[A(1^m, \psi) \in \oplus\mathsf{SAT}] = 0. \end{aligned}$$

In fact, it can be verified that the above holds even if ψ has some free variables x. In more detail, let ψ_x be a statement (with only a single $\exists$ quantifier) that depends on free variables x.[1] The Valiant-Vazirani technique gives us a probabilistic polynomial-time algorithm B outputting a statement ϕ_x (with free variables x) such that, for each x:

$$\begin{aligned} x \text{ is such that } \psi \text{ is true} \quad &\Rightarrow \quad \Pr[\phi_x \in \oplus\mathsf{SAT}] \geq 1/8n \\ x \text{ is such that } \psi \text{ is false} \quad &\Rightarrow \quad \Pr[\phi_x \in \oplus\mathsf{SAT}] = 0. \end{aligned}$$

Repeating this $O(n \cdot (m + |x|))$ times and proceeding as before gives a formula Φ_x where, for *all* x,

$$\begin{aligned} x \text{ is such that } \psi \text{ is true} \quad &\Rightarrow \quad \Pr[\Phi_x \in \oplus\mathsf{SAT}] \geq 1 - 2^{-m} \\ x \text{ is such that } \psi \text{ is false} \quad &\Rightarrow \quad \Pr[\Phi_x \in \oplus\mathsf{SAT}] = 0. \end{aligned}$$

For the inductive step, write $\psi = \exists x : \psi'_x$, where ψ'_x is a quantified boolean formula with $c - 1$ alternations having n free variables x. Applying the inductive hypothesis, we can transform ψ'_x into a boolean formula Φ'_x such that, for all x:

$$x \text{ is such that } \psi'_x \text{ is true} \quad \Rightarrow \quad \Phi'_x \in \oplus\mathsf{SAT} \tag{3}$$

$$x \text{ is such that } \psi'_x \text{ is false} \quad \Rightarrow \quad \Phi'_x \notin \oplus\mathsf{SAT} \tag{4}$$

except with probability at most $2^{-(m+1)}$. We assume the above hold for the rest of the proof.

The key observation is that the Valiant-Vazirani technique applies here as well. We can output, in polynomial time, a boolean formula β such that with probability at least $1/8n$,

$$\begin{aligned} \exists x : \psi'_x \quad &\Rightarrow \quad \exists x : \Phi'_x \in \oplus\mathsf{SAT} \quad \Rightarrow \quad \left|\left\{x : \left(\Phi'_x \in \oplus\mathsf{SAT}\right) \wedge \beta(x)\right\}\right| = 1 \bmod 2 \\ \nexists x : \psi'_x \quad &\Rightarrow \quad \nexists x : \Phi'_x \notin \oplus\mathsf{SAT} \quad \Rightarrow \quad \left|\left\{x : \left(\Phi'_x \in \oplus\mathsf{SAT}\right) \wedge \beta(x)\right\}\right| = 0 \bmod 2. \end{aligned}$$

[1]E.g., ψ_x may be of the form "$\exists z : (z \vee \bar{x}) \wedge x$", in which case ψ_0 is false and ψ_1 is true.

Assume β is such that the above hold. Let $[P]$ evaluate to 1 iff predicate P is true. Then $\exists x : \psi'_x$ implies

$$\begin{aligned} 1 &= \sum_x \left[\left(\Phi'_x \in \oplus\mathsf{SAT}\right) \wedge \beta(x)\right] \bmod 2 \\ &= \sum_x \left[\left(1 = \sum_z \Phi'_x(z) \bmod 2\right) \wedge \beta(x)\right] \bmod 2 \\ &= \sum_x \left[1 = \sum_z \left(\beta(x) \wedge \Phi'_x(z)\right) \bmod 2\right] \bmod 2 \\ &= \sum_{x,z} \left(\beta(x) \wedge \Phi'_x(z)\right) \bmod 2, \end{aligned}$$

and similarly $\nexists x : \psi'_x$ implies

$$0 = \sum_{x,z} \left(\beta(x) \wedge \Phi'_x(z)\right) \bmod 2.$$

Letting $\phi(x,z) \stackrel{\text{def}}{=} \beta(x) \wedge \Phi'_x(z)$ (note ϕ has no free variables), we conclude that

$$\exists x : \psi'_x \Leftrightarrow \phi \in \oplus\mathsf{SAT}.$$

The above all holds with probability at least $1/8n$. But we may amplify as before to obtain Φ such that

$$\begin{aligned} \exists x : \psi'_x &\Rightarrow \Pr[\Phi \in \oplus\mathsf{SAT}] \geq 1 - 2^{-(m+1)} \\ \nexists x : \psi'_x &\Rightarrow \Pr[\Phi \in \oplus\mathsf{SAT}] \leq 2^{-(m+1)}. \end{aligned}$$

Taking into account the error from Equations (3) and (4), we get a total error probability that is bounded by 2^{-m}. ■

The second step of Toda's theorem shows how to derandomize the above reduction, given access to a $\#\mathcal{P}$ oracle.

Theorem 6 $\mathcal{BPP}^{\oplus\mathcal{P}} \subseteq \mathcal{P}^{\#\mathcal{P}}$.

Proof We prove a weaker result, in that we consider only probabilistic Karp reductions to $\oplus\mathcal{P}$. (This suffices to prove Toda's theorem, since the algorithm from the preceding theorem shows that PH can be solved by such a reduction.) For simplicity, we also only consider derandomization of the specific algorithm A from the previous theorem.

The first observation is that there is a (deterministic) polynomial-time computable transformation T such that if $\phi' = T(\phi, 1^\ell)$ then

$$\begin{aligned} \phi \in \oplus\mathsf{SAT} &\Rightarrow \#\mathsf{SAT}(\phi') = -1 \bmod 2^{\ell+1} \\ \phi \notin \oplus\mathsf{SAT} &\Rightarrow \#\mathsf{SAT}(\phi') = 0 \bmod 2^{\ell+1}. \end{aligned}$$

(See [1, Lemma 17.22] for details.)

Let now A be the randomized reduction from the previous theorem (fixing $m = 2$), so that

$$\begin{aligned} \psi \text{ is true} &\Rightarrow \Pr[A(\psi) \in \oplus\mathcal{P}] \geq 3/4 \\ \psi \text{ is false} &\Rightarrow \Pr[A(\psi) \in \oplus\mathcal{P}] \leq 1/4, \end{aligned}$$

where ψ is a quantified boolean formula. Say A uses $t = t(|\psi|)$ random bits. Let $T \circ A$ be the (deterministic) function given by

$$T \circ A(\psi, r) = T(A(\psi; r), 1^t).$$

Finally, consider the polynomial-time predicate R given by

$$R(\psi, (r, x)) = 1 \text{ iff } x \text{ is a satisfying assignment for } T \circ A(\psi, r).$$

Now:

1. If ψ is true then for at least $3/4$ of the values of r the number of satisfying assignments to $T \circ A(\psi, r)$ is equal to -1 modulo 2^{t+1}, and for the remaining values of r the number of satisfying assignments is equal to 0 modulo 2^{t+1}. Thus

$$\left|\{(r, x) \mid R(\psi, (r, x)) = 1\}\right| \in \{-2^t, \ldots, -3 \cdot 2^t/4\} \bmod 2^{t+1}.$$

2. If ψ is false then for at least $3/4$ of the values of r the number of satisfying assignments to $T \circ A(\psi, r)$ is equal to 0 modulo 2^{t+1}, and for the remaining values of r the number of satisfying assignments is equal to -1 modulo 2^{t+1}. Thus

$$\left|\{(r, x) \mid R(\psi, (r, x)) = 1\}\right| \in \{-2^t/4, \ldots, 0\} \bmod 2^{t+1}.$$

We can distinguish the two cases above using a single call to the $\#\mathcal{P}$ oracle (first applying a parsimonious reduction from $R(\psi, \cdot)$ to a boolean formula $\phi(\cdot)$). ■

References

[1] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.