Penn State University
CSE 564 Complexity of Combinatorial Problems

Instructor: Pei Wu

Module
5

Counting classes

So far in the course we have primarily studied *decision* complexity: classes such as NP, coNP, and the polynomial hierarchy PH, defined by whether a language has a polynomial-time verifier with certain quantifier patterns. This lecture studies *counting* complexity. Instead of asking whether a witness exists, we ask *how many* witnesses there are, and we consider both exact counting (the class #P) and weaker decision versions that only ask for parity ($\oplus$P) or majority (PP) information.
After introducing #P and basic examples such as #SAT and the permanent, we define the decision classes $\oplus$P and PP, and show that a PP oracle is strong enough to compute any #P function (via binary search). We then record useful closure properties of $\oplus$P (and similarly PP): closure under complement, intersection, and union.
The second half of the lecture connects these counting classes to randomness and the polynomial hierarchy. We sketch the Valiant–Vazirani theorem, which shows that NP reduces to parity computation with randomness: $\mathsf{NP} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$. We then outline *Toda's first theorem*, $\mathsf{PH} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$, obtained by repeatedly applying the Valiant–Vazirani hashing idea to eliminate quantifiers. Finally, we sketch *Toda's theorem* in its more standard oracle form $\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$, using a counting "mod 2^k" encoding trick.

5.1 The counting class #P

Definition 5.1 (#P). A function $f : \{0,1\}^* \to \mathbb{N}$ is in #P if there exist

- a deterministic polynomial-time Turing machine M, and
- a polynomial $p(\cdot)$

such that for every input x,

$$f(x) \;=\; \big|\{\, y \in \{0,1\}^{p(|x|)} : M(x,y) \text{ accepts} \,\}\big|.$$

Equivalently, $f(x)$ counts the number of NP-witnesses y that cause M to accept on input x.

Example 5.2 (#SAT). Given a Boolean formula φ, the function #SAT(φ) is the number of satisfying assignments of φ. This is a canonical #P function: here $M(\varphi, y)$ simply checks in polynomial time that y satisfies φ.

We expect #P to be computationally hard. Indeed, if we could compute #P functions in polynomial time, then we could decide NP and coNP in polynomial time as well: for a language $L \in \mathsf{NP}$ with verifier V, define $f(x) = |\{y : V(x,y) = 1\}|$. Then $x \in L$ iff $f(x) > 0$, and $x \notin L$ iff $f(x) = 0$.

EXAMPLE 5.3 (#CYCLE). As another typical #P problem, consider counting the number of Hamiltonian cycles in a graph. If there were a polynomial-time algorithm for this counting problem, then in particular we could decide whether a Hamiltonian cycle exists (just check if the count is nonzero), implying $\mathsf{P} = \mathsf{NP}$.

EXAMPLE 5.4 (permanent). For a 0/1 matrix $M \in \{0,1\}^{n \times n}$, its *permanent* is

$$\operatorname{perm}(M) \;=\; \sum_{\sigma \in S_n} \prod_{i=1}^{n} M_{i,\sigma(i)}.$$

If we interpret M as the bipartite adjacency matrix of a bipartite graph G (left vertices $[n]$, right vertices $[n]$), then $\operatorname{perm}(M)$ equals the number of perfect matchings in G.

5.2 Decision versions: $\oplus$P and PP

DEFINITION 5.5 ($\oplus$P). A language $L \subseteq \{0,1\}^*$ is in $\oplus\mathsf{P}$ if there exist a deterministic polynomial-time machine M and a polynomial $p(\cdot)$ such that for every input x,

$$x \in L \quad \Longleftrightarrow \quad \left|\{\, \pi \in \{0,1\}^{p(|x|)} : M(x,\pi) = 1 \,\}\right| \text{ is odd.}$$

In words, $x \in L$ iff the number of accepting witnesses is odd.

DEFINITION 5.6 (PP). A language $L \subseteq \{0,1\}^*$ is in PP if there exist a deterministic polynomial-time machine M and a polynomial $m(\cdot)$ such that for every input x,

$$x \in L \quad \Longleftrightarrow \quad \left|\{\, \tau \in \{0,1\}^{m(|x|)} : M(x,\tau) = 1 \,\}\right| \;\geq\; 2^{m(|x|)-1}.$$

Equivalently, $L \in \mathsf{PP}$ if some polynomial-time probabilistic machine accepts with probability at least $1/2$ on yes-instances and at most $1/2$ on no-instances (with no gap requirement).

THEOREM 5.7. $\#P \subseteq FP^{PP}$.

Proof sketch (binary search with a PP oracle). Let $f \in \#\mathsf{P}$. Then there exists a polynomial-time machine M and an integer $m = m(|x|)$ such that $f(x) = |\{y \in \{0,1\}^m : M(x,y) = 1\}|$.

We show how, given x and an integer $N \in \{0,1,\ldots,2^m\}$, we can decide whether $f(x) \geq N$ using a single PP query. Define a new polynomial-time machine M' that takes input (x,z,y) where $z \in \{0,1\}$ and $y \in \{0,1\}^m$, and outputs:

$$M'(x,z,y) = \begin{cases} 1 & \text{if } z = 0 \text{ and } y < 2^m - N, \\ 1 & \text{if } z = 1 \text{ and } M(x,y) = 1, \\ 0 & \text{otherwise.} \end{cases}$$

The number of accepting pairs $(z,y) \in \{0,1\} \times \{0,1\}^m$ is $(2^m - N) + f(x)$. Therefore,

$$(2^m - N) + f(x) \;\geq\; 2^m \quad \Longleftrightarrow \quad f(x) \geq N.$$

Since the threshold 2^m is exactly half of the total 2^{m+1} possible choices of (z, y), this comparison is a PP question.

Finally, we recover $f(x)$ by binary searching for the largest N such that $f(x) \geq N$, using $O(m)$ oracle calls. □

Open Problem 5.8. Two natural questions suggested in the notes are:

$$\mathsf{FP}^{\oplus\mathsf{P}} \stackrel{?}{=} \#\mathsf{P}, \qquad \text{and} \qquad \mathsf{P}^{\oplus\mathsf{P}} \stackrel{?}{=} \mathsf{PP}.$$

These relationships remain poorly understood.

5.3 Closure properties of $\oplus\mathsf{P}$

The class $\oplus\mathsf{P}$ behaves nicely under standard Boolean operations. The same statements also hold for PP.

Theorem 5.9 (closure properties of $\oplus\mathsf{P}$). *If $L, L_1, L_2 \in \oplus P$, then*

1. ***Complement:*** $\overline{L} \in \oplus P$.
2. ***Intersection:*** $L_1 \cap L_2 \in \oplus P$.
3. ***Union:*** $L_1 \cup L_2 \in \oplus P$.

Proof sketch. For complement, suppose $x \in L$ iff the number of y with $M(x, y) = 1$ is odd. Add one extra accepting witness to flip parity: define $M'(x, z, y)$ with a fresh bit z by $M'(x, 0, 0) = 1$ and $M'(x, 1, y) = M(x, y)$, and otherwise $M'(x, z, y) = 0$. Then $\#\{(z, y) : M'(x, z, y) = 1\} \equiv \#\{y : M(x, y) = 1\} + 1 \pmod 2$, so $x \notin L$ iff M' has an odd number of accepting witnesses.

For intersection, let M_1, M_2 witness $L_1, L_2 \in \oplus\mathsf{P}$. Define $M'(x, y_1, y_2) = 1$ iff $M_1(x, y_1) = 1$ and $M_2(x, y_2) = 1$. The number of accepting pairs is $\#\mathrm{acc}_{M_1}(x) \cdot \#\mathrm{acc}_{M_2}(x)$, whose parity is 1 iff both factors are odd.

For union, use De Morgan's law: $L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$, together with complement and intersection closure. □

5.4 Valiant–Vazirani: $\mathsf{NP} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$

The key technical tool behind Toda's theorem is the Valiant–Vazirani hashing lemma, which uses randomness to reduce an NP search space to a small region where $\oplus\mathsf{P}$ can distinguish satisfiable from unsatisfiable instances.

Theorem 5.10 (Valiant–Vazirani). $NP \subseteq BPP^{\oplus P}$. *In particular,* SAT $\in BPP^{\oplus P}$.

Proof sketch for SAT. Let φ be a CNF formula on n variables, and let $S \subseteq \{0,1\}^n$ be its set of satisfying assignments. If φ is unsatisfiable, then $S = \emptyset$.

Choose a pairwise independent family of hash functions $H_k : \{0,1\}^n \to [2^k]$ (for each k). Pick $k \in \{1, \ldots, n\}$ and choose $h \leftarrow H_k$ and $r \leftarrow [2^k]$ uniformly. Consider the restricted formula

$$\varphi_{h,r}(y) \;:=\; \varphi(y) \;\wedge\; [\, h(y) = r \,].$$

If φ is unsatisfiable, then $\varphi_{h,r}$ is also unsatisfiable, so it has an even number (zero) satisfying assignments.

Now suppose $|S| > 0$. Let i satisfy $2^{i-1} < |S| \le 2^i$ and set $k = i + 1$. A standard inclusion–exclusion argument with pairwise independence shows

$$\Pr_{h \leftarrow H_k,\ r \leftarrow [2^k]} \Big[\big| S \cap h^{-1}(r) \big| = 1 \Big] \ \ge \ \frac{1}{8}.$$

Indeed,

$$\begin{aligned} \Pr\big[\exists!\ y \in S : h(y) = r\big] &\ge \sum_{y \in S} \Pr[h(y) = r] \ - \sum_{\substack{y_1, y_2 \in S \\ y_1 \neq y_2}} \Pr[h(y_1) = r \wedge h(y_2) = r] \\ &= |S| \cdot 2^{-k} \ - \ \frac{|S|(|S|-1)}{2} \cdot 2^{-2k} \\ &\ge \frac{1}{4} - \frac{1}{2} \cdot \frac{1}{4} \ = \ \frac{1}{8}, \end{aligned}$$

where we used $|S| \ge 2^{i-1}$ and $|S| \le 2^i$ with $k = i + 1$.

If $|S \cap h^{-1}(r)| = 1$, then $\varphi_{h,r}$ has exactly one satisfying assignment, hence an *odd* number of satisfying assignments. Thus, by querying the $\oplus\mathsf{P}$ oracle on instances of the form $\varphi_{h,r}$ (and trying all $k \in \{1, \dots, n\}$), we can distinguish satisfiable from unsatisfiable formulas with constant success probability, and standard repetition boosts this to $2/3$. □

A useful way to phrase the above (as in the handwritten notes) is that there is a randomized polynomial-time transformation T such that for every CNF formula φ:

- if $\varphi \in \mathrm{SAT}$, then $\Pr[T(\varphi) \in \oplus\mathsf{P}] \ge 1/8$;
- if $\varphi \notin \mathrm{SAT}$, then $\Pr[T(\varphi) \in \oplus\mathsf{P}] = 0$.

5.5 Toda's theorems

THEOREM 5.11 (Toda's first theorem (sketch)). $\mathsf{PH} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$.

We only sketch the main idea on a representative case. Consider a Π_2 quantified Boolean formula

$$\Psi \ = \ \forall x_1\ \exists x_2\ \varphi(x_1, x_2),$$

where φ is polynomial-time decidable (e.g. a CNF). For a fixed x_1, the subformula $\exists x_2\ \varphi(x_1, x_2)$ is an NP statement about x_2. Applying the Valiant–Vazirani transformation to that existential statement and amplifying its success probability, we can obtain a randomized procedure T such that, for any desired parameter m,

$$\Pr_T\Big[\bigoplus_{x_2} T(\varphi(x_1, x_2)) = 1 \Big] \begin{cases} \ge 1 - 2^{-m} & \text{if } \exists x_2\ \varphi(x_1, x_2) = 1, \\ = 0 & \text{if } \forall x_2\ \varphi(x_1, x_2) = 0. \end{cases}$$

If the Π_2 formula Ψ is true, then every x_1 is a yes-instance of the inner NP problem, so by a union bound over all $2^{|x_1|}$ choices of x_1,

$$\Pr_T\Big[\ \forall x_1,\ \bigoplus_{x_2} T(\varphi(x_1, x_2)) = 1 \Big] \ \ge \ 1 - 2^{-m} \cdot 2^{|x_1|}.$$

Choosing m large enough makes this probability at least $2/3$. This is the "coNP-type" statement indicated in the notes.

If Ψ is false, then there exists some $x_1^\star$ for which $\forall x_2\ \varphi(x_1^\star, x_2) = 0$, and by the soundness property of the Valiant–Vazirani reduction the parity test fails for that $x_1^\star$ with probability 1, so the above "for all x_1" event has probability 0.

To eliminate the remaining universal quantifier over x_1, one applies the same hashing idea *again* to the set of counterexamples $\{x_1 : \bigoplus_{x_2} T(\varphi(x_1, x_2)) = 0\}$. The $\oplus\mathsf{P}$ closure properties from the previous section ensure that the resulting hashed parity predicates remain in $\oplus\mathsf{P}$, and repetition drives the soundness error down to 2^{-m}. Iterating this elimination of quantifiers yields $\mathsf{PH} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$.

THEOREM 5.12 (Toda's second theorem / Toda's theorem (oracle form)). *$\mathsf{PH} \subseteq \mathsf{P}^{\#\mathsf{P}}$. As a consequence, $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{PP}}$.*

Proof sketch. From Toda's first theorem, any $L \in \mathsf{PH}$ has a $\mathsf{BPP}^{\oplus\mathsf{P}}$ algorithm. Equivalently, there is a polynomial-time randomized reduction $T(x) = T_r(x)$ (using a random string $r \in \{0,1\}^p$) such that

$$\Pr_r\big[T_r(x) \in \oplus\mathsf{P}\big] \begin{cases} \geq 2/3 & \text{if } x \in L, \\ \leq 1/3 & \text{if } x \notin L. \end{cases}$$

The remaining step is to replace the $\oplus\mathsf{P}$ oracle by $\#\mathsf{P}$.

The key ingredient is the following "mod 2^k encoding" lemma (Toda's trick), stated in the handwritten notes.

LEMMA 5.13 (Toda's trick). *Given an instance θ and a unary parameter 1^k, there is a polynomial-time mapping M that outputs a $\#\mathsf{P}$ instance $M(\theta)$ such that, writing $\#M(\theta)$ for the number of accepting witnesses of $M(\theta)$,*

$$\theta \in \oplus P \implies \#M(\theta) \equiv -1 \pmod{2^k}, \qquad \theta \notin \oplus P \implies \#M(\theta) \equiv 0 \pmod{2^k}.$$

Assuming the lemma, consider the multiset of values $\#M(T_r(x)) \bmod 2^k$ over all random strings $r \in \{0,1\}^p$. Each term is either 0 or $-1 \bmod 2^k$, so

$$\sum_{r \in \{0,1\}^p} \#M(T_r(x)) \ \equiv\ -\,|\{r : T_r(x) \in \oplus\mathsf{P}\}| \ =\ -\,2^p \cdot \Pr_r[T_r(x) \in \oplus\mathsf{P}] \pmod{2^k}.$$

If $x \in L$ then this residue lies in the interval $\big[-2^p,\, -\tfrac{2}{3}\cdot 2^p\big]$ modulo 2^k, while if $x \notin L$ it lies in $\big[-\tfrac{1}{3}\cdot 2^p,\, 0\big]$ modulo 2^k. Choosing $k > p$ prevents wrap-around, so these two ranges are disjoint and we can distinguish them in deterministic polynomial time given the residue.

Finally, the quantity $\sum_{r\in\{0,1\}^p} \#M(T_r(x))$ is itself a $\#\mathsf{P}$ function of x: a witness consists of (r, w) where w witnesses that $M(T_r(x))$ accepts. Therefore a $\#\mathsf{P}$ oracle suffices to compute this sum, which completes the simulation of the original $\mathsf{BPP}^{\oplus\mathsf{P}}$ algorithm by $\mathsf{P}^{\#\mathsf{P}}$. Combining this with the earlier inclusion $\#\mathsf{P} \subseteq \mathsf{FP}^{\mathsf{PP}}$ yields $\mathsf{PH} \subseteq \mathsf{P}^{\mathsf{PP}}$. □