Instructor: Andrej Bogdanov **Notes by:** Bangsheng Tang

One may be familiar with the difference between an NP-problem and a Counting problem. An NP-problem asks "Is there a solution?", while a Counting problem asks "How many solutions are here?". In this lecture, we shall focus on Counting problems.

1 Counting Problems

For an NP-problem R, the *counting version* of R is the function $\#R : \{0,1\}^n \to \mathbb{N}$ given by

$$\#R(x) = |\{y : (x, y) \in R\}|.$$

Recall that when we defined NP-relations we required that the length of y such that $R(x, y)$ holds can be at most polynomial in x, so it must be that $\#R(x) \leq 2^{p(|x|)}$ for some polynomial p.

We define $\#P$ (pronounced as "sharp-P") as the class of all the counting problems of form $\#R$, where R is an NP-relation.

With the counting problems defined, one may wonder how hard are these counting problems. When we proved that BH is complete for NP, namely $\forall L \in \text{NP}$, $L \leq \text{BH}$. This reduction preserves the number of witnesses. This is called a *parsimonious* reduction.

Suppose A and B are two decision problems, if $A \leq B$ under parsimonious reduction, then if we can solve $\#B$, then we can solve $\#A$.

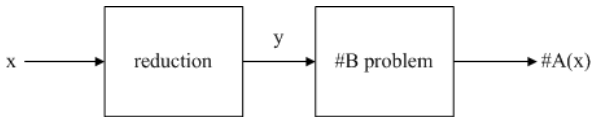

Figure 1: Reduction from #A to #B

$\forall L \in \text{NP}$, $L \leq \text{NP}$ under *parsimonious* says that if we can solve #BHefficiently, then any problem in $\#P$ can be solved efficiently. Moreover, it has been proved that $L \leq \text{BH} \leq \text{CKTSAT} \leq \text{SAT}$, and all these reductions are parsimonious. If #SAT can be solved efficiently, then any problem in $\#P$ can be solved efficiently.

To compare decision and counting, consider the set of problems $\text{P}^{\#R}$, which is decision version of $\#R$. If $\text{P}^{\#\text{SAT}} = \text{P}$, then for every $\#R \in \#\text{P}$, $P^{\#R} = P$, since #SAT is complete for #P.

One should be careful when dealing with parsimonious reductions, since not all NP-reductions are parsimonious. For instance consider the problem of 3-coloring a graph (3COL): Given a given graph G, this problem asks if G is 3-colorable, namely if it is possible to assign colors in the set $\{1, 2, 3\}$ such that neighboring vertices never have the same color.

3COL is NP-complete, but the reduction 3SAT $\leq$ 3COL is not parsimonious. The intuition here is that, for an instance of 3SAT, which is a boolean function ϕ, then ϕ is reduced to a graph G. ϕ is satisfiable if and only if G is 3-colorable. A 3-coloring to G could be converted to an assignment for ϕ. Let V_1, V_2, V_3 be the vertices that are colored 1, 2 or 3 accordingly. Then we totally change all the vertices in V_1 to be colored 2, and all the vertices in V_2 to be colored 1, which produces another valid solution but the corresponding boolean assignment remains the same. There are in fact totally 6 such permutations of colors, therefore an assignment of ϕ would correspond to 6 different colorings.

To resolve that, we introduce another definition.

Definition 1. *We say $\#A$ reduces to $\#B$, if there is a pair of poly-time computable functions (R, Q), such that*

$$\forall x, \#A(x) = Q(\#B(R(x)))$$

According to this definition, #3COL is complete for #P. One should remember that MATCHING $\in$ P, but it can be proved that #MATCHING is complete in $\#P$.

2 The Complexity of Counting Problems

How hard is counting problems? One may observe that $\mathrm{P}^{\#\mathrm{SAT}} \supseteq$ NP, coNP, and BPP. Actually, a much stronger containment holds:

Theorem 2 (Toda)**.** $\forall k,\ \mathrm{P}^{\#\mathrm{SAT}} \supseteq \Sigma_k$

In the other direction, $\mathrm{P}^{\#\mathrm{SAT}} \subseteq$ exp. Since its behaviors could be simulated by exhaustive search.

Then what about approximating counting problems? By approximately counting an NP-relation R, we mean on input x, we want to run in time poly$(|x|, 1/\epsilon)$, output a number N such that $(1-\epsilon)\#R(x) \leq N \leq (1+\epsilon)\#R(x)$.

Notice that approximate counting is at least as hard as NP. If we set $\epsilon = 1/2$, the approximate counting algorithm outputs zero when there are no witnesses for R, and a positive number when there are witnesses for R. However it turns out that approximate counting is not much harder than this: For every NP-relation R, there is a probabilistic algorithm with access to a SAT oracle that runs in expected poly$(|x|, 1/\epsilon)$ and approximately count $\#R$. Namely, approximate counting can be done by a randomized algorithm given access to a SAT oracle.

This indicates that approximate counting is easier than exact counting. If it is not the case, then by Toda's theorem every problem in the polynomial hierarchy can be solved by a randomized algorithm with access to a SAT oracle, which implies that PH $= \Sigma_3$, an unlikely consequence.

3 Counting and Interactive Proofs

This section mainly devoted to the theorem below.

Definition 3. *Denote* IP(poly) *as interactive protocols with unbounded number of rounds.*

Theorem 4. $\mathrm{P}^{\#SAT} \subseteq \mathrm{IP(poly)}$

Proof. It is sufficient to convert an IP for following problems, given ϕ, k: ϕ has exactly k satisfying assignments.

The main trick here is, instead of reasoning about *formulas*, we want to reason about *polynomials*. Here we convert a 3SAT formula $\phi(x_1, \ldots, x_n)$ with m clauses n variables to a polynomial $q_\phi(x_1, \ldots, x_n)$ according to the following rules:

1. $x_i \to x_i$
2. $\overline{x_i} \to 1 - x_i$
3. $y \wedge z \to y \cdot z$
4. $y \vee z = \overline{\overline{y} \wedge \overline{z}} \to 1 - (1 - y) \cdot (1 - z)$

For q_ϕ we have:

- $q_\phi(x_1, \ldots, x_n) = \phi(x_1, \ldots, x_n)$
- $\deg q_\phi \leq 3m$
- $\#\mathrm{SAT}(\phi) = \sum_{x \in \{0,1\}^n} q_\phi(x) \leq 2^n$

Now we turn to prove that, given: $\sum_{x \in \{0,1\}^n} q_\phi(x)$, $\deg q_\phi \leq 3m$

- If $\sum_{x \in \{0,1\}^n} q_\phi(x) = k$, $\exists P$, such that, $\Pr[(P, V) \text{ accepts}] = 1$.
- If $\sum_{x \in \{0,1\}^n} q_\phi(x) \neq k$, $\forall P^*$, $\Pr[(P^*, V) \text{ rejects}] \geq 2/3$.

We will think of q_ϕ as a polynomial over some finite field $\mathbb{F}$, where $\mathbb{F}$ is a prime field of size $> 2^n$. $\sum_{x \in \{0,1\}^n} q_\phi(x) = \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} q_\phi(x_1, x_2, \ldots, x_n)$.

The idea is, P helps V to strip the Σ's one by one.

Let $q(z) = \sum_{x_2, \ldots, x_n} q_\phi(z, x_2, \ldots, x_n)$. $q(z)$ is polynomial of degree $\leq 3m$, and it can be described by $3m + 1$ coefficients like $q(z) = a_0 + a_1 z + \cdots + a_{3m} z^{3m}$. Note that V cannot calculate these coefficients using only polynomial time, while P knows.

Now at the first round, P sends a description of a polynomial $q'(z)$ of degree at most $3m$ to V. When V receives the description, it checks that whether $q'(0) + q'(1) = k$. If not, it rejects. Otherwise, it picks a random number $r \in \mathbb{F}$, and asks P to prove that $\sum_{x_2, \ldots, x_n} q_\phi(r, x_2, \ldots, x_n) = q'(r)$. And this case turns out to be an instance of the same problem with a polynomial of $n - 1$ variables and $k = q'(r)$. This procedure could be done recursively.

There are two claims to confirm the correctness of the protocol.

Claim 5. *If $q(0)+q(1)=k$ and $q'=q$, then for all r, $\sum_{x_2,\ldots,x_n} q_\phi(r,x_2,\ldots,x_n)=q'(r)$.*

Claim 6. *If $q(0)+q(1)\neq k$ then for every q', $\Pr_r\left[\sum_{x_2,\ldots,x_n} q_\phi(r,x_2,\ldots,x_n)=q'(r)\right]\leq 3m/|\mathbb{F}|$.*

Proof. Since $q'(0)+q'(1)=k\neq q(0)+q(1)$, q and q' must be distinct polynomials, that is $q-q'\not\equiv 0$. By the Schwarz-Zippel lemma we have that $\Pr_r[q(r)-q'(r)=0]\leq 3m/|\mathbb{F}|$. Therefore, $\Pr_r[q(r)=q'(r)]\leq 3m/|\mathbb{F}|$. □

Now comes the full protocol.

At round 0, P sends some prime number p between 2^n and 2^{n+1} to V. V checks that p is prime and sets $\mathbb{F}$ to be the prime field $\mathbb{F}_p$.

Then at round 1 and round 2, P tries to convince that $\sum_{x_1,\ldots,x_n} q_\phi(x_1,\ldots,x_n)=k$ as described above. If V choose to continue, it asks P to prove $\sum_{x_2,\ldots,x_n} q_\phi(r_1,x_2,\ldots,x_n)=k_1$ in the next round, where r is picked randomly by V and k_1 is the expected value of when z is substituted by r.

At round $2i-1$, P tries to prove the problem given by V at the previous round, and at round $2i$, V checks the proof given by P and decide whether to continue. If so, V produces a problem with one variable substituted by a random number r_i and ask P to prove it at round $2i+1$.

After n such iterations, V only needs to check whether $q_\phi(r_1,\ldots,r_n)=k_n$ by itself, and accepts only when this is true.

Claim 7. *If $\sum_{x_1,\ldots,x_n} q_\phi(x)=k$, then $\exists P$, $\Pr[(P,V)\ \textit{accepts}]=1$.*

Proof. By what is given above. □

Claim 8. *If $\sum_{x_1,\ldots,x_n} q_\phi(x)\neq k$, then $\forall P^*$ $\Pr[(P^*,V)\ \textit{rejects}]\geq 2/3$.*

Proof. If V accepts, then there exists an i, such that $\sum_{x_i,\ldots,x_n} q_\phi(r_1,\ldots,r_{i-1},x_i,\ldots,x_n)\neq k_i$, $\sum_{x_{i+1},\ldots,x_n} q_\phi(r_1,\ldots,r_i,x_{i+1},\ldots,x_n)=k_{i+1}$. Denote this event as E_i. Then we have

$$\Pr[\exists i: E_i]\leq \sum_{i=1}^{n}\Pr[E_i]\leq \sum_{i=1}^{n} 3m/|\mathbb{F}| = 3mn/|\mathbb{F}|\leq 3mn/2^n\leq 1/3.$$ □

The two claims complete the proof of the theorem. □