1 Toda's theorem

We will prove Toda's theorem:

Theorem 1 (Toda). $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$.

How do we interpret Toda's theorem? We know that the counting problem #SAT is at least as difficult as the decision problem SAT. Toda's theorem tells us that #SAT is apparently a lot more difficult; every problem into the polynomial hierarchy can be turned into a #SAT question. For instance, the question "Is this the smallest circuit with the given functionality?" can be efficiently turned into a question of the form "How many satisfying assignments does this formula have?"

Of course we do not even know how to rule out the possibility $\mathrm{P}^{\#\mathrm{P}} = \mathrm{P}$. However this would have strange consequences, in particular implying $\mathrm{P} = \mathrm{NP}$ and the collapse of the polynomial hierarchy. Might it still be possible that $\mathrm{P}^{\#\mathrm{P}} = \mathrm{P}^{\mathrm{NP}}$? In other words, does the ability to count solutions give us any more power than the ability to tell if solutions exist? Toda's theorem indicates that in fact counting is more powerful than deciding: If $\mathrm{P}^{\#\mathrm{P}}$ were to equal P^{NP}, then $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{NP}} \subseteq \Sigma_2$ and the polynomial hierarchy would collapse to Σ_2.

2 Proof of Toda's theorem

The starting point for the proof of Toda's theorem is the unique solutions lemma of Valiant and Vazirani: There is a randomized procedure that, given a formula φ on n variables,[1] produces a formula ψ such that if φ is unsatisfiable then so is ψ, but if φ is satisfiable then ψ has a *unique* satisfying assignment with probability at least $1/8n$.

One interpretation of Valiant-Vazirani is that if we had a way to tell whether a formula had one satisfying assignment or zero satisfying assignmnets, then, using randomness, we could solve NP complete problems.

Now suppose that we could tell if a formula had an even number or an odd number of satisfying assignments. In particular we can then tell one from zero satisfying assignmnets, so we can solve everything in NP. In fact we can now solve everything in the polynomial hierarchy.

Lemma 2. *For every k there exists a randomized polynomial-time algorithm R that on input a quantified boolean formula φ with k alternations produces an unquantified boolean formula ψ such*

[1] We won't insist that the formulas be in CNF form.

that

$$\varphi \in \Sigma_k\text{SAT} \longrightarrow \Pr[\#\text{SAT}(\psi) \textit{ is odd}] \geq 2/3$$
$$\varphi \notin \Sigma_k\text{SAT} \longrightarrow \#\text{SAT}(\psi) \textit{ is even.}$$

It looks like we are almost there: To solve Σ_kSAT, we map φ to ψ, use the #P oracle to count the number of satisfying assignments of ψ and return the parity. In fact this is sufficient to show that $\text{PH} \subseteq \text{BPP}^{\#\text{P}}$, but we promised $\text{PH} \subseteq \text{P}^{\#\text{P}}$. To get there we need to do some extra work.

The additional ingredient we need is this:

Lemma 3. *There is a deterministic reduction that runs in time $O(n)$ and on input a formula ψ, produces a formula ψ' such that for every N:*

$$\#\text{SAT}(\psi) = 0 \pmod N \longrightarrow \#\text{SAT}(\psi') = 0 \pmod{N^2}$$
$$\#\text{SAT}(\psi) = -1 \pmod N \longrightarrow \#\text{SAT}(\psi') = -1 \pmod{N^2}$$

Therefore, telling if a formula has an even or odd number of assignments reduces to telling if some other formula has 0 or 3 assignments modulo 4, which in turn reduces to telling if some other formula has 0 or 7 assigments modulo 8, and so on. Each time we apply the lemma the size of the formula increases by a constant factor. If we apply the lemma $\log_2 m$ times to ψ for some m, we obtain a formula ψ' of size $\text{poly}(m)|\varphi|$ and so that if we can tell if ψ' has zero or nonzero assignments modulo 2^m, then we can tell if ψ has an even or odd number of assignments.

Now we put the two lemmas together. First consider the reduction R from Lemma 2. It is randomized, but we think of it as a deterministic procedure that takes φ and a random string r of length $m-1$ and produces a formula ψ_r that depends on r. Applying Lemma 3 to ψ_r $\log_2 m$ times we obtain a formula ψ'_r.

We now consider the quantity $K = \sum_{r \in \{0,1\}^{m-1}} \#\text{SAT}(\psi'_r)$, which counts the number of *pairs* (x, r) such that assignment x satisfies formula ψ'_r.

If $\varphi \notin \Sigma_k$SAT, then regardless of the choice of r, $\#\text{SAT}(\psi'_r)$ must equal 0 modulo 2^m. It follows that $K = 0$ modulo 2^m.

If $\varphi \in \Sigma_k$SAT, let p be the fraction of strings r such that $\#\text{SAT}(\psi'_r)$ is odd. We then have that for a p fraction of strings r, $\#\text{SAT}(\psi'_r) = -1$ modulo 2^m, and for the remaining $1-p$ fraction, $\#\text{SAT}(\psi'_r) = 0$ modulo 2^m. Since $p \in [2/3, 1]$, it follows that K must fall in the range $[-\frac{2}{3}2^{m-1}, -2^{m-1}]$ modulo 2^m, so that $K \neq 0$ modulo 2^m.

We now have our $\text{P}^{\#\text{P}}$ algorithm for Σ_kSAT. On input φ, we run the reductions from Lemmas 2 and 3, and ask the oracle to count the number of pairs (x, r) such that x satisfies ψ'_r. (Observe that this is a #P question.) If the answer divides 2^m, we reject; otherwise we accept.

3 The parity quantifier and proof of Lemma 2

Lemma 2 says the determining if a quantified formula is true can be reduced to computing the *parity* of the number of assignments of some other formula. It will be convenient to view parity

as a *quantifier* (like $\exists$ and $\forall$) over the resulting formula: Given a formula φ, we say the quantified formula "$\oplus x : \varphi(x)$" is true if $\varphi(x)$ is true for an odd number of x, and false otherwise. We define the decision problem

$$\oplus\text{SAT} = \{\varphi : \text{"}\oplus x : \varphi(x)\text{" is true}\}.$$

Just like SAT asks whether a formula is satisfiable, $\oplus$SAT asks if it has an odd number of satisfying assignments. We can do the same for quantified formulas: Define

$$\oplus\Sigma_k\text{SAT} = \{\varphi : \text{"}\oplus x \exists y_1 \forall y_2 \ldots Q y_k : \varphi(x, y_1, \ldots, y_k)\text{" is true}\}$$

and $\oplus\Pi_k$SAT similarly. In particular the language $\oplus\Sigma_k$SAT is hard for Σ_k. So to prove Lemma 2, it is sufficient to design a reduction R such that

$$\varphi \in \oplus\Sigma_k\text{SAT} \longrightarrow \Pr[\psi \in \oplus\text{SAT}] \geq 2/3$$
$$\varphi \notin \oplus\Sigma_k\text{SAT} \longrightarrow \psi \notin \oplus\text{SAT}.$$

We start with the case $k = 1$. We are given $\varphi(x, y)$ and want to determine if "$\oplus x \exists y : \varphi(x, y)$" is true. Let $|x| = |y| = n$. For the moment, let's forget about x and focus on y. What can we do? Using Valiant-Vazirani, we can randomly produce a formula $\varphi'(x, y)$ such that if $\varphi(x, y)$ is satisfiable for some y, then $\varphi'(x, y)$ has a unique satisfying assignment with probability $1/8n$, and otherwise it is not satisfiable.

Let us think wishfully and suppose that instead of working with probability $1/8n$, the Valiant-Vazirani reduction worked with probability one. Then the formula "$\oplus y : \varphi'(x, y)$" would be equivalent to "$\exists y : \varphi(x, y)$", and so the $\oplus$SAT instance "$\oplus x \oplus y : \varphi'(x, y)$" and the $\oplus\Sigma_1$SAT instance "$\oplus x \exists y \varphi(x, y)$" would also be equivalent.

Unfortunately Valiant-Vazirani sometimes fails; what we will do instead is obtain a random $\varphi'(x, y)$ such that "$\oplus y : \varphi'(x, y)$" and "$\exists y : \varphi(x, y)$" are equivalent with very high probability over the choice of φ'. We will make this probability as high as $1 - \frac{1}{6}2^{-n}$. Then by the union bound we have that

$$\Pr[\text{For all } x\text{: "}\oplus y : \varphi'(x, y)\text{" is equivalent to "}\exists y : \varphi(x, y)\text{"}] \geq 5/6.$$

so in particular

$$\Pr[\text{The formulas "}\oplus x \oplus y : \varphi'(x, y)\text{" and "}\oplus x \exists y : \varphi(x, y)\text{" are equivalent}] \geq 5/6.$$

Let's now see how to construct φ' from φ. We run the Valiant-Vazirani reduction $m = O(n^2)$ times independently to produce formulas $\varphi'_1(x, y)$ up to $\varphi'_m(x, y)$. If $\varphi(x, y)$ is satisfiable (in y), then with probability $1 - \frac{1}{6}2^{-n}$ at least one of these formulas has a unique satisfying assignment, and otherwise none of them has a satisfying assignment.

We are left with the following task: Given formulas $\varphi'_1, \ldots, \varphi'_m$ produce a single formula φ' such that "$\oplus y : \varphi'(x, y)$" is true iff at least one of "$\oplus y : \varphi'_i(x, y)$" is true. This can be done using the following general construction: Given two unquantified formulas $\psi(y), \psi'(y)$ define

- $\psi \cdot \psi'$ as the formula $\psi(y) \wedge \psi'(z)$, where y and z are disjoint sets of variables. It is easy to check that $\#\text{SAT}(\psi \cdot \psi') = \#\text{SAT}(\psi) \cdot \#\text{SAT}(\psi')$.

- $\psi + \psi'$ as the formula $(w \wedge \psi(y)) \vee (\overline{w} \wedge \psi'(y))$, where w is an additional boolean variable. Then $\#\mathrm{SAT}(\psi + \psi') = \#\mathrm{SAT}(\psi) + \#\mathrm{SAT}(\psi')$.
- 1 as an arbitrary formula with exactly one satisfying assignment.

Then we can set

$$\varphi'(x,y) = 1 + (1 + \varphi'_1(x,y)) \cdot (1 + \varphi'_2(x,y)) \ldots (1 + \varphi'_m(x,y))$$

where we think of the formulas as formulas over y, and x is just a free variable that gets copied in the process of constructing φ'. By construction φ' has an odd number of satisfying assignments (in y) iff at least one of the φ'_i does.

This concludes the case $k = 1$. In general, to go from $\oplus\Sigma_k\mathrm{SAT}$ to $\oplus\Sigma_{k-1}\mathrm{SAT}$ we carry out exactly the same argument to eliminate the outermost existential quantifier of φ. We then obtain an instance ψ of $\oplus\Pi_{k-1}\mathrm{SAT}$. Now observe that $\psi \in \oplus\Pi_{k-1}\mathrm{SAT}$ iff $\overline{\psi} \in \oplus\Sigma_{k-1}\mathrm{SAT}$, and the inductive step is done. We can arrange the probabilities so that the reduction from $\oplus\Sigma_k\mathrm{SAT}$ to $\oplus\Sigma_{k-1}\mathrm{SAT}$ succeeds with probability at least $1/6k^2$. Then even after we put everything together the reduction will work with probability $1 - \sum_k (1/6k^2) \geq 2/3$.

4 Proof of Lemma 3

We now prove Lemma 3. First let us construct a polynomial p such that

$$\begin{aligned} s = 0 \pmod{N} &\longrightarrow p(s) = 0 \pmod{N^2} \\ s = -1 \pmod{N} &\longrightarrow p(s) = -1 \pmod{N^2} \end{aligned}$$

It is not difficult to find such a p: If s^2 factors into p the first property is satisfied. Now if $s = -1$ modulo N, then $s^3 = -1$ modulo N, so $s^3(s^3 + 2) = (s^3 + 1)^2 - 1 = -1$ modulo N^2. We can set $p(s) = s^3(s^3 + 2)$.

Now the formula $\psi' = \psi^3 \cdot (\psi^3 + 2)$ proves the lemma.