Lesson 10: Toda's theorem

Theme: Toda's theorem which states that every language in the polynomial hierarchy can be decided by a polynomial time DTM with oracle access to $\sharp\mathsf{SAT}$, i.e., $\mathbf{PH} \subseteq \mathbf{P}^{\sharp\mathsf{SAT}}$.

Theorem 10.1 (Toda, 1991) $\mathbf{PH} \subseteq \mathbf{P}^{\sharp\mathbf{P}}$.

1 Reduction from $\oplus\mathsf{SAT}$ to $\sharp\mathsf{SAT}$

In the following we will use the notations from Note 11. Recall that $\sharp\varphi$ denote the number of satisfying assignments of a (Boolean) formula φ. For formulas φ and ψ, the formula $\varphi \sqcap \psi$ is a formula such that $\sharp(\varphi \sqcap \psi) = \sharp\varphi \cdot \sharp\psi$.

We define an operation $+$ as follows. Let $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ be the variables in φ and ψ, respectively. Let z be a new variable.

$$\varphi + \psi \quad \stackrel{\mathsf{def}}{=} \quad \Big(\varphi \wedge z \wedge \bigwedge_{i=1}^{m} y_i\Big) \quad \vee \quad \Big(\psi \wedge \neg z \wedge \bigwedge_{i=1}^{n} x_i\Big)$$

Note that $\sharp(\varphi + \psi) = \sharp\varphi + \sharp\psi$.

Lemma 10.2 *There is a deterministic polynomial time algorithm $\mathcal{T}$, that on input formula φ and positive integer m (in unary), outputs a formula ψ such that the following holds.*

- *If $\varphi \in \oplus\mathsf{SAT}$, then $\sharp\psi \equiv -1 \pmod{2^{m+1}}$.*
- *If $\varphi \notin \oplus\mathsf{SAT}$, then $\sharp\psi \equiv 0 \pmod{2^{m+1}}$.*

Proof. We will use the following identity for each $i \geqslant 0$ and n.

(a) If $n \equiv -1 \pmod{2^{2^i}}$, then $4n^3 + 3n^4 \equiv -1 \pmod{2^{2^{i+1}}}$.

(b) If $n \equiv 0 \pmod{2^{2^i}}$, then $4n^3 + 3n^4 \equiv 0 \pmod{2^{2^{i+1}}}$.

On input φ and m, the algorithm $\mathcal{T}$ does the following.

- For each $i = 0, 1, \ldots, \lceil \log(m+1) \rceil$, define a formula ψ_i as follows.

$$\psi_i \stackrel{\mathsf{def}}{=} \begin{cases} \varphi & \text{if } i = 0 \\ 4\psi_{i-1}^3 \ + \ 3\psi_{i-1}^4 & \text{if } i \geqslant 1 \end{cases}$$

  Here $4\psi_{i-1}^3 \ + \ 3\psi_{i-1}^4$ denotes the formula that has $4\sharp(\psi_{i-1})^3 + 3\sharp(\psi_{i-1})^4$ satisfying assignments. Such formula can be constructed using the operators $+$ and $\sqcap$.

- Output the formula $\psi_{\lceil \log(m+1) \rceil}$.

It is not difficult to show that the algorithm $\mathcal{T}$ runs in polynomial time. Its correctness follows directly from the identities (a) and (b). ∎

2 Proof of Theorem 10.1

Let $L \in \mathbf{PH}$. We want to show that $L \in \mathbf{P}^{\sharp\mathsf{SAT}}$. By Theorem 9.6, there is a probabilistic polynomial time algorithm $\mathcal{M}_1$ that on input w, outputs a formula ψ such that the following holds.

- If $w \in L$, then $\mathbf{Pr}[\ \psi \in \oplus\mathsf{SAT}\ ] \geqslant 3/4$.
- If $w \notin L$, then $\mathbf{Pr}[\ \psi \in \oplus\mathsf{SAT}\ ] \leqslant 1/4$.

Using the alternative definition of PTM, we view $\mathcal{M}_1$ as a DTM with two input (w, r), where r is a random string. Let ℓ be the length of the random string. Let $\mathcal{M}_2$ be the algorithm that on input w and random string r, it outputs the formula:

$$\mathcal{T}(\mathcal{M}_1(w, r), \ell + 2)$$

where $\mathcal{T}$ is the algorithm in Lemma 10.2. That is, it first runs $\mathcal{M}_1(w, r)$ and then runs $\mathcal{T}$ on input $(\mathcal{M}_1(w, r), \ell+2)$ Combining Theorem 9.6 and Lemma 10.2, on input w and random string r, the algorithm $\mathcal{M}_2$ outputs a formula $\psi_{w,r}$ such that the following holds.

- If $w \in L$, then $\mathbf{Pr}_{r\in\{0,1\}^\ell}[\ \sharp\psi_{w,r} \equiv -1 \pmod{2^{\ell+3}}\ ] \geqslant 3/4$.
- If $w \notin L$, then $\mathbf{Pr}_{r\in\{0,1\}^\ell}[\ \sharp\psi_{w,r} \equiv -1 \pmod{2^{\ell+3}}\ ] \leqslant 1/4$.

This is equivalent to the following.

- If $w \in L$, the sum $\sum_{r\in\{0,1\}^\ell} \sharp\psi_{w,r}$ lies in between -2^ℓ and $-\frac{3}{4}2^\ell$ (modulo $2^{\ell+3}$).
- If $w \notin L$, the sum $\sum_{r\in\{0,1\}^\ell} \sharp\psi_{w,r}$ lies in between $-\frac{1}{4}2^\ell$ and 0 (modulo $2^{\ell+3}$).

The sets of values that lie in between -2^ℓ and $-\frac{3}{4}2^\ell$ and in between $-\frac{1}{4}2^\ell$ and 0 (modulo $2^{\ell+3}$) are the following sets P and Q, respectively:

$$P \stackrel{\text{def}}{=} \{28 \cdot 2^{\ell-2}, \ldots, 29 \cdot 2^{\ell-2}\} \qquad \text{and} \qquad Q \stackrel{\text{def}}{=} \{31 \cdot 2^{\ell-2}, \ldots, 2^{\ell+3} - 1\} \cup \{0\}$$

Note that P and Q are disjoint.

The main idea of Theorem 10.1 is that on input word w, the algorithm asks the $\sharp\mathsf{SAT}$ oracle for the value $\sum_{r\in\{0,1\}^\ell} \sharp\psi_{w,r}$ and checks whether the value is in P or Q. To this end, we need to construct a formula whose number of satisfying assignments is exactly $\sum_{r\in\{0,1\}^\ell} \sharp\psi_{w,r}$.

Consider the following NTM $\mathcal{M}'$. On input word w, it does the following.

- Guess a string $r \in \{0,1\}^\ell$.
- Run $\mathcal{M}_2$ on (w, r) to obtain a formula $\psi_{w,r}$.
- Guess a satisfying assignment for $\psi_{w,r}$.
- ACCEPT if and only if the guessed assignment is indeed a satisfying assignment for $\psi_{w,r}$.

Obviously, for every w, the number of accepting runs of $\mathcal{M}'$ on w is precisely $\sum_{r\in\{0,1\}^\ell} \sharp\psi_{w,r}$.

Now, to complete our proof, we present a polynomial time DTM $\mathcal{M}$ decides L (with oracle access to $\sharp\mathsf{SAT}$). On input w, it does the following.

- Construct a formula Ψ_w such that the number of satisfying assignments of Ψ_w is exactly the number of accepting runs of $\mathcal{M}'$ on w.

  Here we use Cook-Levin construction (on w and the transitions in $\mathcal{M}'$). Recall that Cook-Levin reduction is parsimonious.
- Determine the value $\sharp\Psi_w$ (modulo $2^{\ell+3}$) by querying the $\sharp\mathsf{SAT}$ oracle.
- Determine whether $\sharp\Psi_w$ lies in P or Q, the answer of which implies whether $w \in L$.