Penn State University
CSE 564 Complexity of Combinatorial Problems

Instructor: Pei Wu

Module
2

Alternations and Polynomial Hierarchy

A list of contents mentioned in the second week. All proofs are sketching high-level ideas. Details can be found in AB's book (Ch. 5)

2.1 NP and coNP (verifier view)

DEFINITION 2.1 (NP). A language L is in NP iff there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a deterministic polynomial-time Turing machine M such that for every input x:

$$x \in L \iff \exists \pi \in \{0,1\}^{p(|x|)} \ \ M(x, \pi) \text{ accepts},$$

equivalently,

$$x \notin L \iff \forall \pi \in \{0,1\}^{p(|x|)} \ \ M(x, \pi) \text{ rejects}.$$

DEFINITION 2.2 (coNP). A language L is in coNP iff $\overline{L} \in$ NP, i.e., iff there exists a polynomial p and a deterministic polynomial-time machine M such that

$$x \in L \iff \forall \pi \in \{0,1\}^{p(|x|)} \ \ M(x, \pi) \text{ accepts},$$

equivalently,

$$x \notin L \iff \exists \pi \in \{0,1\}^{p(|x|)} \ \ M(x, \pi) \text{ rejects}.$$

Example. UNSAT $:= \{\varphi : \varphi \text{ is an unsatisfiable Boolean formula}\} \in$ coNP.

2.2 Generalizing NP and coNP: Σ_2 and Π_2

DEFINITION 2.3 (Σ_k and Π_k via alternating quantifiers). For $k \geq 1$:

$L \in \Sigma_k \iff \exists$ poly p $\exists$ poly-time M s.t.

$$x \in L \iff \exists \pi_1 \in \{0,1\}^{p(|x|)} \forall \pi_2 \in \{0,1\}^{p(|x|)} \cdots Q_k \pi_k \in \{0,1\}^{p(|x|)} \ \ M(x, \pi_1, \ldots, \pi_k) \text{ accepts},$$

$L \in \Pi_k \iff \exists$ poly p $\exists$ poly-time M s.t.

$$x \in L \iff \forall \pi_1 \in \{0,1\}^{p(|x|)} \exists \pi_2 \in \{0,1\}^{p(|x|)} \cdots Q_k \pi_k \in \{0,1\}^{p(|x|)} \ \ M(x, \pi_1, \ldots, \pi_k) \text{ accepts},$$

where the quantifiers alternate and Q_k is $\exists$ if k is odd and $\forall$ if k is even (so Σ_k starts with $\exists$ and Π_k starts with $\forall$).

Padding observation. You can always add "dummy" quantifiers that do not affect the predicate. In particular,

$$\text{NP} = \Sigma_1 \subseteq \Sigma_2 \qquad \text{and} \qquad \text{coNP} = \Pi_1 \subseteq \Pi_2,$$

and also $\text{NP} \subseteq \Pi_2$ and $\text{coNP} \subseteq \Sigma_2$ by adding a leading dummy quantifier.

Some natural problems in low levels of PH

1) MIN-CNF

Define

$$\text{MIN-CNF} := \{\ \varphi\ :\ \text{there is no CNF formula } \psi \text{ with } |\psi| < |\varphi| \text{ such that } \psi \equiv \varphi\ \}.$$

Then MIN-CNF $\in \Pi_2$, because

$$\varphi \in \text{MIN-CNF} \iff \forall \psi\ \big(|\psi| < |\varphi| \Rightarrow \exists x\ (\psi(x) \neq \varphi(x))\big).$$

2) MAX-INDSET

Let $\alpha(G)$ be the size of a maximum independent set in $G = (V, E)$, and define

$$\text{MAX-INDSET} := \{(G, k)\ :\ \alpha(G) = k\}.$$

Then MAX-INDSET $\in \Sigma_2$, since

$$(G,k) \in \text{MAX-INDSET} \iff \exists S \subseteq V\ \forall T \subseteq V,$$
$$\underbrace{S \text{ is an independent set and } |S| = k}_{\text{witness } S}\ \wedge\ (\underbrace{(T \text{ is not an independent set})\ \vee\ (|T| < k)}_{\text{no larger ind. set}}).$$

2.3 The polynomial hierarchy

Definition 2.4 (Polynomial hierarchy).

$$\text{PH} := \bigcup_{k \geq 1} \Sigma_k = \bigcup_{k \geq 1} \Pi_k.$$

(Also $\text{P} \subseteq \Sigma_1 = \text{NP}$ and $\text{P} \subseteq \Pi_1 = \text{coNP}$.)

Conjecture. Just as we believe $\text{P} \neq \text{NP}$ and $\text{NP} \neq \text{coNP}$, we conjecture that the "hierarchy does not collapse", there are infinitely many different levels:

$$\Sigma_i \neq \Sigma_j, \quad \Pi_i \neq \Pi_j \quad (i \neq j), \qquad \text{and} \qquad \Sigma_i \neq \Pi_i.$$

THEOREM 2.5 (If one level collapses, all higher levels collapse). *If* $\Sigma_i = \Pi_i$ *for some* $i \geq 1$, *then* PH *collapses to the* i*-th level; i.e.,*

$$\forall k \geq i: \quad \Sigma_k = \Sigma_i \quad \textit{and} \quad \Pi_k = \Pi_i.$$

Proof sketch. It suffices to show $\Sigma_{i+1} \subseteq \Sigma_i$. Let $L \in \Sigma_{i+1}$. Then for some poly-time predicate R,

$$x \in L \iff \exists \pi_1 \, \forall \pi_2 \, \exists \pi_3 \, \cdots \, Q_{i+1}\pi_{i+1} \;\; R(x, \pi_1, \ldots, \pi_{i+1}).$$

Fix π_1 and define the derived language

$$L' := \{(x, \pi_1) : \forall \pi_2 \exists \pi_3 \cdots Q_{i+1}\pi_{i+1} \, R(x, \pi_1, \ldots, \pi_{i+1})\}.$$

The predicate defining L' begins with $\forall$ and has i alternations, so $L' \in \Pi_i$. By assumption $\Pi_i = \Sigma_i$, hence $L' \in \Sigma_i$. Therefore

$$x \in L \iff \exists \pi_1 \; (x, \pi_1) \in L' \in \Sigma_i,$$

which gives $L \in \Sigma_{i+1} \subseteq \Sigma_i$ (and similarly for Π_{i+1}). Repeating yields collapse of all higher levels. □

Complete problems

CLAIM 2.6. *For each* $\ell \geq 1$, *the classes* Σ_ℓ *and* Π_ℓ *have complete problems under polynomial-time many-one reductions.*

CLAIM 2.7. *If* PH *had a complete problem (under poly-time many-one reductions), then* PH *would collapse.*

Reason. If L were PH-complete, then $L \in \Sigma_k$ for some k (since $L \in$ PH). Completeness would imply PH $\subseteq \Sigma_k$, hence the hierarchy collapses to level k. □

2.4 Sipser–Gács: BPP $\subseteq \Sigma_2 \cap \Pi_2$

THEOREM 2.8 (Sipser–Gács). BPP $\subseteq \Sigma_2 \cap \Pi_2$.

Proof idea ("shifts" / hitting set). Let $L \in$ BPP. Then there exists a poly-time machine M using $m = \text{poly}(|x|)$ random bits such that

$$x \in L \Rightarrow \Pr_{r \in \{0,1\}^m}[M(x, r) = 1] \geq \tfrac{2}{3}, \qquad x \notin L \Rightarrow \Pr_{r \in \{0,1\}^m}[M(x, r) = 1] \leq \tfrac{1}{3}.$$

Amplify by repetition so that for some $n = \text{poly}(|x|)$,

$$x \in L \Rightarrow \Pr_r[M(x, r) = 1] \geq 1 - 2^{-n}, \qquad x \notin L \Rightarrow \Pr_r[M(x, r) = 1] \leq 2^{-n}.$$

Let $A_x := \{r \in \{0,1\}^m : M(x, r) = 1\}$.

Key combinatorial claim. If $|A_x| \geq (1 - 2^{-n})2^m$, then there exist $p = \text{poly}(m)$ "shifts" $s_1, \ldots, s_p \in \{0,1\}^m$ such that

$$\forall r \in \{0,1\}^m \;\; \exists i \in [p] \;\; r \oplus s_i \in A_x.$$

Equivalently, $\forall r\ \exists i\ M(x, r \oplus s_i) = 1$.

Probabilistic proof of the claim. Fix r. For a uniform random s, we have

$$\Pr_s[M(x, r \oplus s) = 0] = \Pr_s[r \oplus s \notin A_x] = 1 - \frac{|A_x|}{2^m} \leq 2^{-n}.$$

Thus for independent uniform $s_1, \ldots, s_p$,

$$\Pr_{s_1,\ldots,s_p}\Big[\forall i,\ M(x, r \oplus s_i) = 0\Big] \leq 2^{-np}.$$

Taking a union bound over all $r \in \{0,1\}^m$,

$$\Pr_{s_1,\ldots,s_p}\Big[\exists r\ \forall i,\ M(x, r \oplus s_i) = 0\Big] \leq 2^m \cdot 2^{-np}.$$

So if $p \geq m/n + 1$ (still polynomial), the RHS is < 1, meaning there exists a choice of shifts with the desired hitting property.

Putting it into Σ_2 form. For $x \in L$, such shifts exist, hence

$$x \in L \iff \exists s_1, \ldots, s_p\ \ \forall r \in \{0,1\}^m\ \ \exists i \in [p]\ \ M(x, r \oplus s_i) = 1,$$

which is a Σ_2 statement (the inner $\exists i$ is over $O(\log p)$ bits and can be folded into the poly-time predicate).

Putting it into Π_2 form. For $x \notin L$, $|A_x| \leq 2^{-n}2^m$, which implies that for any fixed shifts $s_1, \ldots, s_p$, a random r is unlikely to land in A_x after any shift; in particular there exists an r that avoids all of them:

$$\forall s_1, \ldots, s_p\ \ \exists r \in \{0,1\}^m\ \ \forall i \in [p]\ \ M(x, r \oplus s_i) = 0.$$

This is a Π_2 statement. Hence $L \in \Sigma_2 \cap \Pi_2$. □

2.5 Upper bound: PH $\subseteq$ PSPACE

THEOREM 2.9. PH $\subseteq$ PSPACE.

Proof idea (recursive evaluation). Let $L \in$ PH. Then for some $k = \text{poly}(|x|)$ there is a poly-time predicate R such that

$$x \in L \iff \exists \pi_1\ \forall \pi_2\ \exists \pi_3\ \cdots\ Q_k \pi_k\ \ R(x, \pi_1, \ldots, \pi_k).$$

A deterministic algorithm can evaluate this by recursion on $\ell = 1, 2, \ldots, k$:

- If $\ell = k + 1$, output $R(x, \pi_1, \ldots, \pi_k)$.
- Otherwise, iterate over all choices of $\pi_\ell \in \{0,1\}^{m_\ell}$, recursively evaluate the suffix, and combine the answers by OR if $Q_\ell = \exists$ (odd ℓ) or by AND if $Q_\ell = \forall$ (even ℓ).

This takes exponential time in general but uses only polynomial space: the recursion depth is $k = O(\text{poly}(|x|))$ and each level stores only polynomially many bits (the current π_ℓ plus the work tape of the poly-time predicate R). □

2.6 Quantified Boolean Formulas and PSPACE-completeness

One can imagine have polynomially many alternations of quantifiers. Then we reach PSPACE.

Definition 2.10 (TQBF). TQBF is the set of true fully-quantified Boolean formulas with alternating quantifiers, e.g.

$$\exists x_1\, \forall x_2\, \exists x_3\, \cdots\, Q_m x_m\ \ \varphi(x_1, \ldots, x_m),$$

where $m = \text{poly}(|\varphi|)$ and φ is a Boolean formula.

Theorem 2.11 (Stockmeyer–Meyer). TQBF *is* PSPACE*-complete.*

Membership TQBF $\in$ PSPACE**.** This is essentially the same recursive-evaluation idea as for PH $\subseteq$ PSPACE.

Hardness idea (configuration reachability; why we need quantifiers). Let M be a polynomial-space TM deciding some $L \in$ PSPACE. A configuration of M on input x has length $s = \text{poly}(|x|)$, so there are at most $2^{O(s)}$ configurations. The computation may take *exponential* time (up to $2^{O(s)}$ steps), so a Cook–Levin style "computation tableau" reduction to SAT would be exponentially large.

Define a predicate $\Psi_t(C', C'')$ meaning:

$$\Psi_t(C', C'') : \quad \text{configuration } C' \text{ can reach } C'' \text{ in at most } 2^t \text{ steps.}$$

A natural recursion is

$$\Psi_t(C', C'') \ \equiv\ \exists D\, \Big(\Psi_{t-1}(C', D)\ \wedge\ \Psi_{t-1}(D, C'')\Big),$$

i.e., split a path of length $\le 2^t$ at a midpoint configuration D.

Size issue and the "one-copy" trick. The displayed recursion duplicates Ψ_{t-1}, which would blow up the formula size. A standard workaround (the one sketched in the notes) is to write an equivalent formula that contains only *one* copy of Ψ_{t-1} by universally quantifying over which subcall we are checking:

$$\Psi_t(C', C'') \ \equiv\ \exists D\ \forall D', D''\ \Big(\big((D', D'') = (C', D) \vee (D', D'') = (D, C'')\big)\ \Rightarrow\ \Psi_{t-1}(D', D'')\Big).$$

Intuitively, for the two specific pairs (C', D) and (D, C'') we force Ψ_{t-1} to hold; for all other pairs, the implication is vacuous.

With a suitable base case $\Psi_0(C', C'')$ encoding "C'' is reachable from C' in one step" (a polynomial-size local check), this yields a quantified Boolean formula of size polynomial in s and t. Choosing $t = O(s)$ makes 2^t large enough to cover any computation (or at least any simple path through the configuration graph), so:

$$x \in L \iff \exists C_{\text{acc}}\, \Big(\Psi_t(C_{\text{start}}, C_{\text{acc}})\ \wedge\ C_{\text{acc}} \text{ is accepting}\Big),$$

giving a polynomial-time reduction from L to TQBF.