Lecture #14: Introduction to the Polynomial Hierarchy
Exercises and Review

Additional Exercises

1. Suppose you are given a sequence $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ of Boolean formulas, and you wish to know how many of these are satisfiable. A decision problem, concerning this, is as follows.

   How Many are Satisfiable?

   Instance: A sequence
   $$\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$$
   and an integer k such that $0 \le k \le n$.

   Question: Are *exactly* k of the Boolean formulas $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ satisfiable (so that $n-k$ of these Boolean formulas are *unsatisfiable*?

   Let $\widehat{\Sigma}_F = \Sigma_F \cup \{\,(\,,\,)\,,\,\texttt{,}\,\}$ where Σ_F that was introduced in Lecture #11, and used to encode Boolean formulas. Then an instance of the above decision problem, including a sequence of Boolean formulas $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$ and an integer k as above, can be encoded as a string in $\widehat{\Sigma}_F^\star$ consisting of the encodings of each of the formulas $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$, and the unpadded decimal representation of the integer k, separated by commas and encoded by brackets.

   Let $L_{\text{Formulas+Number}} \subseteq \widehat{\Sigma}^\star$ be the language of encodings of instances of this decision problem, and let $L_{\text{HowManySatisfiable}} \subseteq L_{\text{Formulas+Number}}$ be the language of encodings of Yes-instances of this decision problem.

   (a) Sketch a proof that $L_{\text{Formulas+Number}} \in \mathcal{P}$.

   (b) Sketch a proof that $L_{\text{HowManySatisfiable}} \in \Sigma_2\mathcal{P} \cap \Pi_2\mathcal{P}$.

2. Prove that $\Sigma_i\mathcal{P} \cup \Pi_i\mathcal{P} \subseteq \Sigma_{i+1}\mathcal{P} \cap \Pi_{i+1}\mathcal{P}$ for every positive integer i.

Questions for Review

1. What is an ***alternating Turing machine***? How is it similar to a nondeterministic Turing machine — and how is it different?
2. Consider a ***computation tree*** for an alternating Turing machine $\mathcal{M}$ and an input string ω, and a configuration $\mathcal{C}$ in this tree.
   (a) Suppose that $\mathcal{C}$ includes an ***existential*** state. What additional condition(s) must be satisfied in order for $\mathcal{C}$ to be an ***accepting configuration***?
   (b) Suppose, again, that $\mathcal{C}$ includes an ***existential*** state. What additional condition(s) must be satisfied in order for $\mathcal{C}$ to be a ***rejecting configuration***?
   (c) Suppose, once again, that $\mathcal{C}$ includes an ***existential*** state. What additional condition(s) must be satisfied in order for $\mathcal{C}$ to be a ***looping configuration***?
   (d) Suppose, instead, that $\mathcal{C}$ includes a ***universal*** configuration. What additional condition(s) must be satisfied in order for $\mathcal{C}$ to be an ***accepting configuration***?
   (e) Suppose, again, that $\mathcal{C}$ includes a ***universal*** state. What additional condition(s) must be satisfied in order for $\mathcal{C}$ to be a ***rejecting configuration***?
   (f) Suppose, once again, that $\mathcal{C}$ includes a ***universal*** state. What additional condition(s) must be satisfied in order for $\mathcal{C}$ to be a ***looping configuration***?
3. Let $\mathcal{M}$ be an alternating Turing machine with input alphabet Σ.
   (a) What condition(s) must be satisfied in order for $\mathcal{M}$ to ***accept*** a string $\omega \in \Sigma^\star$?
   (b) What condition(s) must be satisfied in order for $\mathcal{M}$ to ***reject*** a string $\omega \in \Sigma^\star$?
   (c) What condition(s) must be satisfied in order for $\mathcal{M}$ to ***loop*** on a string $\omega \in \Sigma^\star$?
4. What does it mean for an alternating Turing machine $\mathcal{M}$, with input alphabet Σ, to ***recognize*** a language $L \subseteq \Sigma^\star$?
5. What does it mean for an alternating Turing machine $\mathcal{M}$, with input alphabet Σ, to ***decide*** a language $L \subseteq \Sigma^\star$?
6. Consider an alternating Turing machine $\mathcal{M}$, with input alphabet Σ, that decides a language $L \subseteq \Sigma^\star$.
   (a) How is the ***time*** used by $\mathcal{M}$ on an input string $\omega \in \Sigma^\star$ defined?
   (b) Let $f : \mathbb{N} \to \mathbb{N}$ be a total function. Give the definition of ATIME$(f(n))$.
   (c) Give the definition of $\mathcal{AP}$. How is this complexity class related to complexity classes that have been considered before this?
7. Define $\Sigma_i\mathcal{P}$ and $\Pi_i\mathcal{P}$ for a positive integer i, as well as the complexity class $\mathcal{PH}$. How are these related? What is conjectured — but not proved — about $\mathcal{PH}$?