Problem Set 2

- Due date: **3 Mar, 2023** (released on 15 Feb, 2023)
- The points for each problem is indicated on the side. The total for this set is **65** points.
- The problem set has a fair number of questions so please do not wait until close to the deadline to start on them. Try and do one question every couple of days.
- Turn in your problem sets electronically (PDF; either LaTeXed or scanned etc.) via email.
- Collaboration with other students taking this course is encouraged, but collaboration with others is not allowed. Irrespective of this, all writeups must be done individually and must include names of all collaborators (if any).
- Referring to sources other than the text book and class notes is STRONGLY DISCOURAGED. But if you do use an external source (eg.,other text books, lecture notes, or any material available online), ACKNOWLEDGE all your sources (including collaborators) in your writeup. This will not affect your grades. However, not acknowledging will be treated as a serious case of academic dishonesty.
- Be clear in your writing.

---

1. **[Mahoney's theorem]** (10)

   Show that if there is a sparse language L that is NP-hard, then $\mathsf{P} = \mathsf{NP}$.

   [Hint: Follow the similar sketch as we did in class to *prune* the bag of formulas. You need to find a way to "amplify" the number of satisfiable formulas you have in your bag so that you can force a collision. Think of ways to modify your bag S to a different bag S' of related formulas such that if S had no satisfiable formulas then neither does S' but if S had at least one satisfiable formula, then S' has *many* of them.]

2. **[Baker-Gill-Solovay for NP and coNP]** (15)

   Show that there is a language A such that $\mathsf{NP}^A \neq \mathsf{coNP}^A$.

   [Hint: It might be useful to remember that coNP rejects if *any* of its computational paths reject. Modify the construction seen in class appropriately.]

3. **[Classes and reductions]** (10)

   Show that $\mathsf{NTIME}(n) \neq \mathsf{P}$.

   [Hint: Note that you are being asked to only show that the two classes are different. You may be able to do that without being able to explicitly point out a language that is in one but not in the other. For instance, if you knew for a fact that $\mathsf{NTIME}(n)$ was *not* closed under complementation, that would have been enough for you since you know P is closed under complementation. Alas, we don't know that but try other properties.]

4. **[Unique witness]** (10)

   Define the following language

$$\text{Unique3Col} = \left\{ \langle G \rangle \; : \; \begin{array}{c} \langle G \rangle \text{ encodes an undirected graph} \\ \text{that has a unique 3-colouring,} \\ \text{up to permutation of the three colours.} \end{array} \right\}$$

   Show that Unique3Col $\in \mathsf{P}^{\mathsf{SAT}}$.

5. **[Cook-Levin in the presence of oracles]** (10 + 10)

   A natural question is whether the Cook-Levin reduction continues to hold even in the presence of oracles. Turns out, the answer is 'Yes, and no' — it depends on how precisely the question is posed.

   (a) Let A be an arbitrary language. Define a suitable *relatived* version of CircuitSAT and show that it is NP^A-complete under polynomial time many-one reductions.

   (b) Show that there is a language A and another language L_A such that $L_A \in \mathsf{NP}^A$ but there is no polynomial-time oracle TM M such that M^A is a reduction from L_A to CircuitSAT.

   (That is, giving oracle access to A does not allow us build a reduction from L_A to CircuitSat even though $L_A \in \mathsf{NP}^A$.)

   [Hint: Recall the oracle we used in the Baker-Gill-Solovay theorem. There, we had to diagonalise against polynomial time TMs computing a decision problem. Can you modify it suitably to diagonalise against polynomial time oracle-reductions?]

6. **[A decideable function that is *not* time-constructible]**

   Given an example a decideable function $f : \mathbb{N} \to \mathbb{N}$ that is *not* time-constructible.

   [Hint: You could make use of the time-hierarchy theorem here. Come up with a function f such that computing the value $f(n)$ in $O(f(n))$-time will enable you to solve membership in a given language too quickly and thus contradict the hierarchy theorem.]