CS 332: *Theory of Computation* — Prof. Mark Bun
Boston University — December 2, 2021

Homework 10 – Due Thursday, December 9, 2021 at 11:59 PM

Reminder Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to the course staff if asked. You must also identify your collaborators and write "Collaborators: none" if you worked by yourself. Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.

Note You may use various generalizations of the Turing machine model we have seen in class, such as TMs with two-way infinite tapes, stay-put, or multiple tapes. If you choose to use such a generalization, state clearly and precisely what model you are using.

Problems There are 4 required problems.

1. (**Decision vs. search**) You are consulting for a hospital that wants to improve the morale of its surgery teams. A surgical team[1] consists of a surgeon, an anesthesiologist, and a OR technician. The hospital sends you sets X, Y, and Z, with r individuals each, corresponding to the three roles. You also receive a set M of triples, where each triple corresponds to a potential surgery team where all 3 individuals can work together without fighting. An individual can appear in multiple triples. The hospital administration is asking you to find a way to arrange the largest possible number of happy surgical teams. That is, your goal is to select a largest subset of triples from M so that each individual appears in at most one triple. (There may be more than one largest subset, in which case, you are free to pick any largest subset.)

   In this problem, you will prove that if P = NP, you can *find* a largest set of happy surgical teams in polynomial time.

   (a) Consider the following language:
   $TEAM = \{\langle X, Y, Z, M, k\rangle \mid |X| = |Y| = |Z|$, each element of M is a triple (x, y, z) where $x \in X, y \in Y$ and $z \in Z$, and M contains a subset of size k where each element appears in at most one triple$\}$.

   Prove that $TEAM \in$ NP by exhibiting a *nondeterministic polynomial time algorithm.* Make sure to explain why your algorithm is correct and why it runs in (nondeterministic) polynomial time.

   (b) If P = NP, the proof you just gave for part (a) implies that $TEAM \in$ P, that is, in polynomial time you can decide whether it is possible to have k happy surgical teams. Assume you have a subroutine that does it, and use it repeatedly to *find* a largest set of happy surgical teams in polynomial time.

   Hint: Similar to problem 7.40, solved in the book.

2. (**Poly-time Reductions**) Assume $\mathsf{P} \neq \mathsf{NP}$. For each of the following, give a language (if it exists) with the stated property. Explain why your language satisfies the given property, or explain why no such language can exist.

   (a) $A \leq_{\mathrm{p}} SAT$ and A is NP-complete.

[1] omitting several important roles for simplicity

(b) $SAT \leq_{\mathrm{p}} B$ and B is not NP-complete.

(c) $SAT \leq_{\mathrm{p}} C$ and C is not NP-hard.

(d) D is both regular and NP-complete.

3. (**Satisfiability**) Define the language $DSAT = \{\langle \phi_1, \phi_2 \rangle \mid \phi_1, \phi_2$ are Boolean formulas over the same set of variables and there exists x satisfying both formulas simultaneously$\}$. For example, if $\phi_1 = x_1 \vee x_2$ and $\phi_2 = x_1 \wedge \overline{x_2}$, then the assignment $x_1 = 1, x_2 = 0$ satisfies both formulas simultaneously, and hence $\langle \phi_1, \phi_2 \rangle \in DSAT$.

   (a) Prove that $DSAT$ is in NP by describing a deterministic polynomial-time verifier. Make sure to explain why your verifier is correct and why it runs in deterministic polynomial time.

   (b) Show that $SAT \leq_{\mathrm{p}} DSAT$ and use this to conclude that $DSAT$ is NP-complete.

4. (**Systems of linear inequalities**) A *linear inequality* I over variables $x_1, \ldots, x_k$ is an inequality of the form $c_1 x_1 + \ldots c_k x_k \leq b$, where $c_1, \ldots, c_k$ and b are integers. E.g., $5x_1 - 3x_2 + x_3 \leq -1$ is a linear inequality. A *system of linear inequalities* is a set $\{I_1, \ldots, I_m\}$ of inequalities over the same variables. Such a system *has an boolean solution* if one can assign boolean values (either 0 or 1) to all variables in such a way that all inequalities are satisfied.

   Define the language $BI = \{\langle I_1, \ldots, I_m \rangle \mid$ the system $\{I_1, \ldots, I_m\}$ has a boolean solution$\}$.

   (a) Prove that BI is in NP. You may either describe a nondeterministic poly-time TM, or a deterministic poly-time verifier.

   (b) Show that $3SAT \leq_{\mathrm{p}} BI$ and use this to conclude that BI is NP-complete.

5. (**Bonus**) In a directed graph, the ***indegree*** of a node is the number of incoming edges and the ***outdegree*** of a node is the number of outgoing edges. Show that the following problem is NP-complete. Given an undirected graph G and a subset S of the nodes in G, determine whether it possible to convert G to a directed graph by assigning directions to each of its edges so that every node in S has indegree 0 or outdegree 0, and all remaining nodes in G have indegree at least 1.