CS 21 Decidability and Tractability	**Winter 2024**
Final	
Out: *March 6*	*Due:* **March 13, 1pm**

This is the final exam. You may consult only the course notes and text (Sipser). You may not collaborate. There are 5 problems on two pages. Please attempt all problems. **Please turn in your solutions via Gradescope, by 1pm on the due date.**

Good luck!

1. Consider the following 2-person game. The game is played on a directed acyclic graph whose nodes are labeled with integers. There is a specified start-node s.

   Starting from node s, the players take turns selecting an outgoing edge from the current node: player one selects an outgoing edge from nodes s, which takes them to a node v, then player two selects an outgoing edge from node v, which takes them to a node u, then player one selects an outgoing edge from node u, and so on. We keep a running sum of the integers encountered as this path from s in the graph is traversed. The game ends when a sink (a node with no outgoing edges) is reached. At that point player one wins if the running sum equals zero; otherwise player two wins.

   Given a game instance (a directed acyclic graph G labeled with integers, and a start node s) we can ask whether there is a win for player one (i.e., player one can win no matter what player two does). Prove that the language L consisting of those game instances for which there is a win for player one is PSPACE-complete. In other words prove:

   (a) L is in PSPACE, and

   (b) L is PSPACE-hard. Here it may be useful to recall the two-player game interpretation of QSAT from Lecture 24. Hint: in your reduction, it is sufficient for the graph to be a *layered* graph, with at most 3 nodes per layer. A layered directed graph is one in which the nodes can be partitioned into subsets ("layers") $V = L_1 \cup L_2 \cup L_3 \cup \cdots \cup L_k$ and the only directed edges are edges between adjacent layers; i.e., from nodes in layer L_i to nodes in layer L_{i+1}.

2. Let L be the language over the alphabet $\Sigma = \{a, b, c\}$ consisting of exactly those strings with an *unequal* number of a's and b's (and any number of c's). Is L (i) regular, (ii) context-free but not regular, or (iii) not context free? Prove that your classification is correct.

3. For a language $L \subseteq \Sigma^*$ and a string $w \in \Sigma^*$, the language

$$L_{-w} = \{xy : x \in \Sigma^* \text{ and } y \in \Sigma^* \text{ and } xwy \in L\}$$

   consists of all strings in L with the string w deleted from them.

(a) Prove that if L is regular, then L_{-w} is regular. Hint: make $|w|+1$ copies of a DFA recognizing L.

(b) Prove that if L is R.E., then L_{-w} is R.E.

4. Suppose someone says they wish to prove the following reasonable-sounding statement:

   "If some NP-complete language has an $O(n^2)$-time algorithm, then every language in NP has an $O(n^4)$-time algorithm."

   Show that such a proof is not possible. In other words, prove that *even if* some NP-complete language had an $O(n^2)$-time algorithm, the conclusion (that every language in NP has an $O(n^4)$-time algorithm) is false.

5. Each of the following languages is either in P, or it is NP-complete. **Choose 4 out of the 5 problems, and for each one, prove that it is NP-complete, or prove that it is in P. Please indicate clearly which 4 you are choosing, and provide solutions for only those 4.**

   For two of the problems below, you will need to recall that in a graph, the *degree* of a vertex v, denoted $d(v)$, is the number of edges that touch that vertex; the *maximum degree* of a graph is the maximum, over vertices v, of $d(v)$.

   (a) This problem is a variant of INDEPENDENT SET in bounded-degree graphs. The language in question is the set of all pairs (G,k) for which G is a graph with maximum degree at most 4 containing an independent set of size at least k.

   (b) This problem is a variant of UNDIRECTED HAMILTON PATH in bounded-degree graphs. The language in question is the set of all triples (G,s,t) for which G is an undirected graph with maximum degree at most 2 containing a Hamilton path from node s to node t.

   (c) Given a universe U and a collection of subsets $\mathcal{C} = \{S_1, S_2, S_3, \ldots, S_n\}$, with each $S_i \subseteq U$, we say that a subset $H \subseteq U$ is a *hitting set* if each S_i contains at least one element of H. In this problem we are interested in the case in which each S_i has size at most 2. The language in question is

$$\text{HITTING SET-2} = \{(\mathcal{C}, k) \;:\; \text{for all } S_i \in \mathcal{C},\ |S_i| \le 2, \text{ and there is a hitting set } H \subseteq U \text{ with } |H| \le k\}.$$

   (d) The language consisting of 2-CNF formulas ϕ for which there exists an assignment that satisfies at least 3/4 of the first 100 clauses, and all of the other clauses.

   (e) The language consisting of 2-CNF formulas ϕ for which there exists an assignment that satisfies all of the first 100 clauses, and at least 3/4 of the other clauses.