| | |
|---|---|
| **CS 21 Decidability and Tractability** | **Winter 2014** |

# Final

*Out: March 12* — *Due:* **March 19, noon**

**This is a final.** You may consult only the course notes and the text (Sipser). *You may not collaborate.* The full honor code guidelines can be found in the course syllabus.

There are 5 problems on 2 pages. Please attempt all problems. **To facilitate grading, please turn in each problem on a separate sheet of paper and put your name on each sheet. Do not staple the separate sheets.** Good luck!

**Instructions for turning in the exam:** Please turn in your exams to Diane Goodfellow in Annenberg 246 before noon on Friday March 19.

1. Consider the following 2-player game. The game is specified by an undirected graph $G = (V, E)$, an integer $k$, and a sequence of pairs of subsets of the vertex set $V$,

$$(S_1, T_1), (S_2, T_2), \ldots, (S_n, T_n).$$

   It is played as follows: player 1 selects one of $S_1$ or $T_1$ and deletes that subset of vertices and their incident edges from $G$; player 2 selects one of $S_2$ or $T_2$ and deletes that subset of vertices and their incident edges from $G$; player 1 selects one of $S_3$ or $T_3$, and so on. In general, in odd-numbered turns $i$, player 1 is selecting one of $S_i$ or $T_i$ (and deleting the specified vertices from $G$), and in even-numbered turns $i$, player 2 is selecting one of $S_i$ or $T_i$ (and deleting the specified vertices from $G$). The game ends after the $n$-th turn. Player 1 wins if the graph that remains contains a $k$-clique; otherwise player 2 wins.

   Given an input $G, k, (S_1, T_1), \ldots (S_n, T_n)$, we can ask whether there is a win for player 1 (i.e., player 1 can win no matter what player 2 does). Prove that the language $L$ consisting of inputs for which there is a win for player 1 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. Your reduction from QSAT will produce a graph with a triple of nodes for each clause, and all possible edges between different triples.

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. Is the following language $L$ (i) decidable, (ii) R.E. but not decidable, (iii) co-R.E. but not decidable, or (iv) neither R.E. nor co-R.E.? Prove that your classification is correct. Recall that for a Turing Machine $M$, we denote by $L(M)$ the language it recognizes.

$$L = \{\langle M_1, M_2, M_3\rangle : M_1, M_2, M_3 \text{ are TMs and } L(M_1) \subseteq L(M_2) \subseteq L(M_3)\}.$$

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

$$L_{-y} = \{xz : x \in \Sigma^* \text{ and } z \in \Sigma^* \text{ and } xyz \in L\}$$

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

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

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

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 CLIQUE 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 100 containing a clique of size at least $k$.

   (c) Say that a "job" consists of a triple (processing <u>t</u>ime, <u>d</u>eadline, <u>p</u>rofit). Given a list of jobs

$$(t_1, d_1, p_1), (t_2, d_2, p_2), \ldots, (t_n, d_n, p_n)$$

   a *schedule* $s_1, s_2, \ldots, s_n$ is a list of starting times for the $n$ jobs. Jobs cannot overlap, so for all $i \neq j$, the intervals $(s_i, s_i + t_i)$ and $(s_j, s_j + t_j)$ must be disjoint. The profit achieved by a schedule is the sum of the profits $p_i$ for jobs that complete before their deadline; i.e., $\sum_{i:s_i+t_i \leq d_i} p_i$.

   A list of $n$ jobs paired with a target profit $P$ (all numbers are represented in binary) is a positive instance of the language if there is schedule for those jobs that achieves at least profit $P$.

   (d) The language consisting of 2-CNF formulas $\phi$ for which there exists an assignment that satisfies at least 3/4 of the first 1000 clauses, and all of the other clauses.

   (e) The language consisting of 2-CNF formulas $\phi$ for which there exists an assignment that satisfies all of the first 1000 clauses, and at least 3/4 of the other clauses.