CSC C63 Final Exam

April 19, 2016

NAME:

Calculators are not permitted (nor would they be useful).

This is a closed book exam.

You can assume that the following problems are NP-complete.

SAT
Input: A boolean formula, F.
Question: Is F satisfiable?

3-SAT
Input: A CNF boolean formula, F, in which every clause has exactly 3 literals.
Question: Is F satisfiable?

IND-SET
Input: A graph G, and an integer K.
Question: Does G have an independent set of size at least K?

CLIQUE
Input: A graph G, and an integer K.
Question: Does G have a clique of size at least K?

VERTEX-COVER
Input: A graph G, and an integer K.
Question: Does G have a vertex cover of size at most K?

PARTITION
Input: A list of positive integers $a_1, ..., a_n$.
Question: Can the list be partitioned into 2 parts A_1, A_2 such that the sum of each part is the same?

KNAPSACK
Input: A list of weights $w_1, ..., w_n$ and values $v_1, ..., v_n$ along with a capacity W and a target T. All numbers are positive integers.
Question: Is there a subset of indices $I \subset \{1, ..., n\}$ such that $\sum_{i \in I} w_i \leq W$ and $\sum_{i \in I} v_i \geq T$?

HAM-PATH
Input: A graph G, with two specified vertices u, v.
Question: Does G have a Hamilton path from u to v?

HAM-CYCLE
Input: A graph G.
Question: Does G have a Hamilton cycle?

1. **(21 pts)** For each of the following statements, say that it is one of:

   A True.

   B False.

   C No one knows, but most researchers think it is True.

   D No one knows, but most researchers think it is False.

   Do not explain your answer.

   (a) $\mathrm{P} = \mathrm{NP}$

   (b) $\mathrm{P} = \mathrm{NP} \cap \text{co-NP}$

   (c) $\mathrm{L} = \mathrm{NL} \cap \text{co-NL}$

   (d) $\mathrm{L} = \mathrm{PSPACE}$

   (e) HAM-PATH $\in$ PSPACE

   (f) There is a 2-approximation algorithm for the non-decision version of CLIQUE

   (g) There is an algorithm to determine whether a given algorithm will output the square of the input.

2. **(12 pts)** Only short answers are required here.

   (a) **(4 pts)** Name a problem in NP that is not NP-complete and is not thought to be in P.

   (b) **(4 pts)** Name a problem that is NL-complete

   (c) **(4 pts)** How is the transition function of a Nondeterministic Turing Machine different than the transition function of the usual kind of Turing Machine?

3. **(17 pts)** Recall that the Halting Problem is:

   Input: $< P >$, the description of an algorithm (i.e. Turing Machine), and X an input for P.
   Question: Does P halt when given input X?

   Prove that there is no algorithm (i.e. Turing machine) that solves the Halting Problem. Do not use a reduction from another undecidable problem.

4. **(20 pts)** Consider the set

$$A = \{(< P >, k) | \text{ for all } x \geq k, P(x+1) \geq P(x) + 2\}$$

P is a Turing machine that has positive integers as input and output, and k is a positive integer.

The condition "$P(x+1) \geq P(x) + 2$" means that $P(x+1), P(x)$ both halt and $P(x+1)$ returns a number that is at least 2 more than what $P(x+2)$ returns.

(a) **(15 pts)** Is A decidable? Prove your answer.

(a) **(5 pts)** Answer Yes or No. You don't need to prove your answer.

(i) Is A recognizable?

(ii) Is $\overline{A}$ recognizable?

5. **(36 pts)** Prove that each of the following problems is NP-complete. You do not have to prove that they are in NP (until later in the exam).

   (a) **(12 pts)**
   4-SAT
   Input: A CNF boolean formula F where every clause has exactly 4 literals.
   Question: Does F have a satisfying assignment?

(b) **(12 pts)**

WEIGHTED-IND-SET

Input: A graph G, with a positive integer weight on each vertex, and an integer W.

Question: Does G have an independent set with total weight at least W?

(c) **(12 pts)**
BIG-CYCLE
Input: A graph G.
Question: Does G have a cycle going through at least $\frac{1}{3}$ of the vertices of G?

6. **(12 pts)** Choose any two problems from the previous question and prove that they are both in NP. For one problem, do this using a verification algorithm. For the other problem, do this using a nondeterministic Turing Machine.

[blank page]

7. **(12 pts)** A satisfying assignment of a boolean formula is *unlocked* if there is a variable which you can change to obtain another satisfying assignment. A satisfying assignment is *locked* if it is not unlocked; i.e. if it is not possible to obtain another satisfying assignment by changing exactly one variable. Eg. for the formula:

$$(x_1 \vee \overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee x_2) \wedge (\overline{x_2} \vee x_3)$$

The satisfying assignment $x_1 = F, x_2 = F, x_3 = F$ is **unlocked** because changing x_3 produces $x_1 = F, x_2 = F, x_3 = T$ which is also satisfying.

The satisfying assignment $x_1 = T, x_2 = T, x_3 = T$ is **locked** because none of the three assignments obtained by changing exactly one of the variables is satisfying.

LOCK-SAT
Input: A boolean formula F.
Question: Does F have a locked satisfying assignment?

Prove that LOCK-SAT $\in$ PSPACE