Alex Hubers
CSC:341 Spring 2024

HW #11

Due: Apr 23 2024

Instructions. You are allowed to collaborate with others, however you should write up solutions independently. Copying an answer from another source (e.g. the Web) or from another student may yield few or zero points. Write solutions neatly and legibly, or type your solutions in LaTeX. Be sure to number each problem, and indicate a final solution (if relevant). Answers to problems should include justification (show your work).

Acknowledgments. Problems from this homework come from published sources. The specific sources are withheld due to the nature of this assignment.

Academic Honesty. Include the following information at the top of your submission, along with your name.

- Written sources used: (Include textbook(s), complete citations for web or other written sources. Write <u>none</u> if no sources used)
- Help obtained: (Include names of anyone other than the instructor.)

#1 (5 pts)

Prove that polynomial-time reducibility forms a transitive relation. That is, prove the following claim:

Claim 1. *Let A, B, and C be decidable languages such that $A \leq_p B$ and $B \leq_p C$. Then $A \leq_p C$.*

In your answer, let f be a mapping reduction from A to B, decided by M_f, and let g be a mapping reduction from B to C, decided by M_g. Recall that you have three obligations:

1. Describe a mapping reduction h from A to B, decided by TM M_h;
2. prove that your mapping reduction is correct; and
3. prove that M_h runs in polynomial time.

#2

Prove that the following languages are in NP. Be explicit about the certificate and the verification algorithm. Argue that the verification algorithm operates in polynomial time.

#2.1 Graph Coloring (5 pts)

A k-graph coloring of a graph G is an assignment of k colors to nodes such that no two nodes share both the same color and an edge.

$\mathsf{COLOR} = \{\langle G, k\rangle \mid \text{There is a } k\text{-coloring of graph } G\}$.

#2.2 Job Scheduling (5 pts)

Consider a set of J jobs equipped with

1. a partial order $\prec$ on J that determines that for each $J_m, J_n \in J$ whether $J_m \prec J_n$ (understood to mean that J_m must be completed before J_n),
2. a weight determined by the function $W : J \to \mathbb{N}$ that says how long each job takes to complete. There exists k distinct processors on which to run jobs, and each processor may only work on one job at a time.

The goal is to find a scheduling of jobs on these processors that respects the ordering such that all jobs complete in target time t or less.

$\mathsf{SCHED} = \{\langle J, (\prec), W, k, t\rangle \mid \text{There is a schedule of } J \text{ jobs on } k \text{ processors taking time } t \text{ or less}\}$.

#3 (5 pts)

Consider the following combinatorial decision problems:

0-1 Knapsack. Let B be a set of objects equipped with two functions:

- A <u>weight function</u> $w : B \to \mathbb{N}$ assigning weights to objects.
- A <u>value function</u> $v : B \to \mathbb{N}$ assigning values to objects.

Suppose that we have a knapsack that can hold at most W weight. Find a subset $B' \subseteq B$ that can fit in the knapsack and has at least value V.

$$\mathsf{KNAPSACK} = \{\langle B, w, v, W, V\rangle \mid \text{There exists a subset } B' \subseteq B \text{ s.t. } \sum_{b \in B'} w(b) \leq W \text{ and } \sum_{b \in B'} v(b) \geq V\}$$

Subset Sum. Let $S \subseteq \mathbb{N}$ be a (multi-)set[1] of natural numbers. Find a subset $S' \subseteq S$ such that $\sum_{s \in S'} s = t$ for some target number t.

$$\mathsf{subsetsum} = \{\langle S, t\rangle \mid \text{There exists a subset } S' \subseteq S \text{ s.t. } \sum_{s \in S'} s = t\}$$

Assume that $\mathsf{subsetsum}$ is NP-complete. Prove that $\mathsf{knapsack}$ is NP-complete as well.

N.b. Make sure that you explicitly justify the complexity of your verifier and the complexity and correctness of your constructed mapping function in your proof.

[1]A multi-set permits duplicate elements. For example, $\{1, 1, 1\}$ is a valid multi-set, but not a valid set.