Homework 8: NP-Completeness

CSE 30151 Spring 2020

Due 2020/04/29 at 5pm (no late penalty during grace period of 72 hours)

Instructions

- Create a PDF file (or files) containing your solutions. You can write your solutions by hand, but please scan them in the library or using a smartphone to get them into PDF.
- Please name your PDF file(s) as follows to ensure that the graders give you credit for all of your work:
  - If you're making a complete submission, name it `*netid*-hw8.pdf`, where *netid* is replaced with your NetID.
  - If you're submitting some problems now and want to submit other problems later, name it `*netid*-hw8-123.pdf`, where `123` is replaced with the problem numbers you are submitting at this time.
- Submit your PDF file(s) in Sakai. Don't forget to click the Submit button!

Problems

1. This problem concerns details of the proof of the NP-completeness of *CLIQUE*, which is Theorem 7.32 in the book.
   (a) Convert the formula $\phi = (x \vee x \vee z) \wedge (\bar{x} \vee y \vee y) \wedge (\bar{y} \vee \bar{y} \vee \bar{z})$ into a graph $G = (V, E)$ and integer k, using the construction in the proof of Theorem 7.32 (so that ϕ is satisfiable iff G has a clique of size k).
   (b) For each satisfying truth assignment of ϕ,
   - Please write down the truth assignment.
   - How many subsets of V does it correspond to?
   - Please draw one of them.
   - Is it a clique of G?

   (c) Convert $\phi = (x \vee \bar{y} \vee \bar{y}) \wedge (\bar{x} \vee \bar{x} \vee \bar{x}) \wedge (y \vee y \vee y)$ into a graph G and integer k, and explain why ϕ is not satisfiable and G has no clique of size k.

2. In the *knapsack problem*, you are given a knapsack with maximum weight capacity W kilograms, and a set of k items,

$$S = \{\langle w_1, v_1\rangle, \ldots, \langle w_k, v_k\rangle\}$$

where w_i is the weight (in kilograms) of item i and v_i is the value (in dollars) of item i. The decision version of the problem is: Is there a subset of the items with total weight at most W and total value at least V? More formally,

$$\mathit{KNAPSACK} = \left\{ \langle S, W, V\rangle \mid \exists I \subseteq \{1, \ldots, k\} \text{ s.t. } \sum_{i \in I} w_i \leq W \text{ and } \sum_{i \in I} v_i \geq V \right\}.$$

Show that this problem is NP-complete. You can treat k as the problem size, or you can use $|\langle S, W, V\rangle|$ as the problem size if you want to be more precise.

3. In Theorem 5.15, we considered the modified Post Correspondence Problem (*MPCP*), which is to decide whether a set of dominos

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \ldots \left[\frac{t_k}{b_k}\right] \right\},$$

has a solution, that is, sequence of indices $i_1, \ldots i_l$ such that $i_1 = 1$ and $t_{i_1} \cdots t_{i_l} = b_{i_1} \cdots b_{i_l}$. Define the *string-length* of this solution to be $|t_{i_1} \cdots t_{i_l}|$.

Now let us consider a bounded version of this problem, *BMPCP*, which is to decide, on input $\langle P, \mathtt{1}^n\rangle$, where P is an MPCP instance and $n > 0$, whether P has a solution with string-length n. In this problem, we'll prove that *BMPCP* is NP-complete, by reduction from $\mathit{TIME}_{\mathsf{NTM}}$.

   (a) Prove that *BMPCP* is in NP. You can treat n as the problem size. If you want to be more precise, treat $|\langle P, \mathtt{1}^n\rangle|$ as the problem size.

Next, we need to define a mapping f from triples $\langle N, w, \mathtt{1}^t\rangle$ (where N is an NTM) to pairs $\langle P, \mathtt{1}^n\rangle$, such that N accepts w in at most t steps iff P has a solution with string-length at most n.

The proof of Theorem 5.15 provides such a mapping, with only a small modification. Define $f(N, w, \mathtt{1}^t) = \langle P, n\rangle$, where

- P is the MPCP instance constructed in the proof of Theorem 5.15 for N on w, with the modification that the first domino is

$$\left[\frac{\texttt{\#}}{\texttt{\#}q_0 w \sqcup^{t+1-|w|}\texttt{\#}}\right].$$

  That is, the input string is padded with blanks to be $(t+1)$ symbols long.

- $n = (t+1)(t+3) + \frac{1}{2}(t+2)(t+3) + 1$.

You may assume without proof that neither the nondeterminism of N nor the change to the first domino affects the construction's correctness; that is, N accepts w iff P has a solution.

(b) Prove that if N accepts w in at most t steps, then P has a solution with string-length at most n.

(c) Prove that if P has a solution with string-length at most n, then N accepts w in at most t steps.

(d) Prove that f runs in polynomial time.