# Homework #12

## Problem 1: Matching

In Section 8.6, the book proves that the 3-Dimensional Matching problem is NP-Complete by reducing from 3-SAT. Consider now the problem of 4-Dimensional Matching:

> *Given disjoint sets $W$, $X$, $Y$, and $Z$, each of size $n$, and given a set $T \subseteq W \times X \times Y \times Z$ of ordered 4-tuples, does there exist a set of $n$ 4-tuples in $T$ so that each element of $W \cup X \cup Y \cup Z$ is contained in exactly one of these 4-tuples?*

Prove that the above problem is NP-complete by reduction from 3-Dimensional Matching. Be sure to ensure all of the proof obligations for proving a problem's membership in NP-complete.

**The 3-Dimensional Matching Problem.** *Given disjoint sets X, Y, and Z, each of size $n$, and given a set $T \subseteq X \times Y \times Z$ of ordered triples, does there exist a set of $n$ triples in $T$ so that each element of $X \cup Y \cup Z$ is contained in exactly one of these triples?*

**Solution**

4-Dimensional Matching is in NP, since we can check in $O(n)$ time, using an $n \times 4$ array initialized all to 0, that a given set of $n$ 4-tuples is disjoint.

We now show that *3-Dimensional-Matching* $\leq_p$ *4-Dimensional-Matching*. Consider an instance of *3-Dimensional-Matching* with sets $X$, $Y$, and $Z$ of size $n$ each, and a collection of $C$ ordered triples. We define an instance of *4-Dimensional-Matching* as follows. We have sets $W, X, Y, Z$, each of size $n$, and a collection $C'$ of 4-tuples defined so that for every $(x_j, y_k, z_l) \in C$, and every $i$ between 1 and $n$, there is a 4-tuple $(w_i, x_j, y_k, z_l)$. This instance has a size that is polynomial in the size of the initial *3-Dimensional-Matching* instance.

If $A = (x_j, y_k, z_l)$ is a triple in $C$, define $f(A)$ to be the 4-tuple $(w_j, x_j, y_k, z_l)$; note that $f(A) \in C'$. If $B = (w_i, x_j, y_k, z_l)$ is a 4-tuple in $C'$, define $f'(B)$ to be the triple $(x_j, y_k, z_l)$; note that $f'(B) \in C$. Given a set of $n$ disjoint triples $\{A_i\}$ in $C$, it is easy to show that $\{f(A_i)\}$ is a set of $n$ disjoint 4-tuples in $C'$. Conversely, given a set of $n$ disjoint 4-tuples $\{B_i\}$ in $C'$, it is easy to show that $\{f'(B)\}$ is a set of $n$ disjoint triples in $C$. Thus, by determining whether there is a perfect 4-dimensional matching in the instance we have constructed, we can solve the initial instance of *3-Dimensional-Matching*.

## Problem 2: Magnets

My daughter has recently learned to spell some words. (Seriously, like "cow;" I'm so proud...) To help encourage this, we bought her a colorful set of refrigerator magnets with letters of the alphabet on them (some number of copies of the letter 'A', some number of copies of the letter 'B' and so on), and we use the magnets to try to spell out words that she knows.

Somehow, things always end up getting more elaborate than originally planned, and we started trying to spell out words so as to use up all of the magnets in the full set—that is, picking out words that she knows how to spell, so that once they were all spelled out, each magnet was participating in the spelling of exactly one word. (Multiple copies of the same word are fine; if the set of magnets contains two instances of each 'C', 'A', and 'T', it's ok if we spell 'CAT' twice.)

This has turned out to be pretty difficult. Suppose we consider a general version of the problem *Using Up All the Refrigerator Magnets*, or *Magnets* for short, where we replace the English alphabet with an arbitrary collection of symbols, and we model my daughter's vocabulary as an arbitrary set of strings over this collection of symbols. Prove that the problem of *Magnets* is NP-Complete.

**Solution**

*Magnets* is in NP. A certificate for the problem can be a multiple set of words. We can count the number of types of each kind of magnets used in these words and verify whether that is equal to the given number of that kind of magnets.

Now we will show that *Magnets* is NP-Complete by reducing from *3D Matching*. We have an instance of *3D Matching*, which consists of three sets $X, Y, Z$ such that $|X| = |Y| = |Z| = n$ and set of tuples $M$. We want to find $n$ tuples from $M$ such that each element is covered by exactly one tuple. Create an instance of *Magnets* as follows: each element in $X, Y$, or $Z$ becomes a magnet with a unique letter, (so our alphabet will have $3n$ letters), and every tuple $(x_i, y_j, z_k)$ becomes a word that madison knows. Solving this instance of *Magnets* will sove the instance of *3D Matching*.

We must now show that there is a perfect matching in *3D Matching* if and only if all the magnets can be used up in *Magnets*. If all of the magnets are used up, we must have gotten exactly $n$ words, since each word has 3 letters, and there are a total of $3n$ letters, and therefore we have the desired $n$ tuples. If there is a perfect matching in *3D Matching*, then those words that correspond to the tuples in the matching will exactly use up all of the magnets without overlapping.