CS 1511 Homework 12

Mathew Varughese, Justin Kramer, Zach Smith

Wednesday, Feb 27

21. b)

EXACT INDSET = $\{\langle G, k\rangle$ | the largest independent set in G has size exactly k$\}$.

INDSET = $\{\langle G, k\rangle$ | there exists an independent set in G with size k$\}$.

LE INDSET = $\{\langle G, k\rangle$ | all sets in G have size $\leq$ k $\}$.

EXACT INDSET = INDSET $\cap$ LE INDSET

INDSET $\in$ NP. This is shown in a previous theorem. The polynomial verifier would just check the given set and check if it is independent, if it is in G, and if it is of length k.

LE INDSET is in CoNP. This is because it is a "for all" (for all set in G ...). More formally, the Turing Machine verifier would be the same as that above.

EXACT INDSET is the union of these. This is by definition. Every element in EXACT INDSET will be in INDSET. If the largest independent set in G has size exactly k, then there definitely exists a set of size k. Every element is also in LE INDSET. If the largest set has size exactly k, then it all sets will have size less than or equal to k.

So, EXACT INDSET is in DP. L_1 = INDSET, $L_1 \in$ NP, L_2 = LE INDSET . $L_2 \in$ CoNP, L = $L_1 \cap L_2$.

21 c)

We need to show $\forall L \in DP \leq$ EXACT INDSET in poly time

In order to show this, we will need two TM's T and W.f

First off, any language in DP consists of the intersection of two languages, L1 and L2. L1 $\in$ NP and L2 $\in$ co-NP. In this case, we will show that this reduction will work for an arbitrary L1 and L2 by reducing a NP-Complete problem to EXACT INDSET with TM T and by reducing a co-NP-Complete problem to EXACT INDSET with TM W, both in poly-time. This way, any intersection of L1 and L2 for arbitrary L1's and L2's will work.

Our NP-Complete language will be INDSET, which is proven to be NP-Complete in the book.

Below is our algorithm.

Begin with T(G, k, n) with a graph G and integers k and n

Loop this Turing Machine, beginning with k = 0 and going until k = number of nodes in the graph.

This can be done in poly-time.

If this TM doesn't accept when k = n, reject.

If this TM accepts when k > n, reject.

Run EXACT INDSET(G, n)

Our co-NP-Complete language with be coINDSET, the complement of INDSET that is by definition co-NP-Complete. coINDSET is not the same as LE INDSET. coINDSET is the complement, (sets that do not equal size k) and LE INDSET has sets that are less than or equal to k.

Since INDSET is NP Complete, coINDSET is CoNP Complete. This is because every language in CoNP is the complement of a language in NP. So, if any language in NP can be reduced to INDSET, the complement of every language of NP (so every coNP language) can be reduced to coINDSET.

Below is our algorithm

Begin with W(G, k, n) with a graph G and integer k and n

Loop this Turing Machine, beginning with k = 0 and going until k = number of nodes in the graph

This can be run in poly-time

If this TM rejects when k = n, reject.

If this TM rejects when k > n, reject.

Run EXACT INDSET(G, n)

Therefore, if both a NP-Complete language and a co-NP-Complete problem can be reduced to EXACT INDSET in poly-time, every language in DP is poly-time reducible to EXACT INDSET.