Homework 13

Joe Baker, Brett Schreiber, Brian Knotten

February 11, 2018

19

a

Problem 5.9 part (a): Show that EXACT INDSET $\in \pi_2^p$.

Proof:
EXACT INDSET can be expressed as the following language in first-order logic:

EXACT INDSET $= \{(G,k) \mid \forall$ Independent Sets $S_1 \in G, \exists$ an Independent Set $S_2 \in G$ such that $|S_2| = k$ and $|S_2| >= |S_1|\}$.

By the definition of π_2^p, EXACT INDSET $\in \pi_2^p$.

b

Problem 5.11: Show that SUCCINCT SET-COVER $\in \Sigma_2^p$.

Proof:
SUCCINCT SET COVER can be expressed as the following language in first-order logic:

SUCCINCT SET COVER $= \Big\{(S,k) \mid \exists\ S' \subseteq S, \forall$ possible assignments of $S',$ then $\big|S'\big| <= k$ and S' true$\Big\}$.

By the definition of Σ_2^p, SUCCINCT SET COVER $\in \Sigma_2^p$.

c

Problem 5.13: Show that VC-DIMENSION $\in \Sigma_3^p$.

Proof:
VC-DIMENSION can be expressed as the following language in first-order logic:

VC-DIMENSION $= \{(C,k) \mid \exists\ X \subseteq U, \forall\ X' \subseteq X, \exists\ i,$ such that $|X| >= k$ and $C_i \cup X = X'\}$.

By the definition of Σ_3^p, VC-DIMENSION $\in \Sigma_3^p$.

d

Problem 5.9 part(b): Show that EXACT INDSET $\in$ DP.

Proof:
Let $L_1 = \{(G,k) \mid \exists$ an Independent Set $S,$ such that $|S| = k\}$.
$L_1 \in$ NP because $L_1 \in \Sigma_1^p$ by definition and $\Sigma_1^p =$ NP.

Let $L_2 = \{(G,k) \mid \forall$ Independent Sets $S,$ such that $|S| <= k\}$.
$L_2 \in$ co-NP because it is in π_1^p by definition and $\pi_1^p =$ co-NP.

Since the intersection of two sets requires both constraints to be met:
Let $L = L_1 \cap L_2 = \{(G, k) \mid \forall \text{ Independent Sets } S_1, \exists \text{ an Independent Set } S_2, \text{ such that } |S_1| <= k \text{ and } |S_2| = k\} =$ EXACT IND SET.

Finally, since $L = L_1 \cap L_2$ and $L_1 \in$ NP and $L_2 \in$ co-NP, then $L \in$ DP.

20

Problem 5.3: Show that if 3SAT is polynomial-time reducible to $\overline{\text{3SAT}}$, then PH = NP.

Proof:
First we will show that if 3SAT is polynomial-time reducible to $\overline{\text{3SAT}}$, then NP = co-NP:
Since 3SAT $\leq_P$ $\overline{\text{3SAT}}$, then there is a bijective function $f : \{0,1\}^* \to \{0,1\}^*$ where $x \in$ 3SAT if and only if $f(x) \in \overline{\text{3SAT}}$. Since 3SAT is NP-complete, for each language $L_{\text{NP}} \in$ NP, there exists some bijective function $f_{L_{\text{NP}}}$ which reduces L_{NP} to 3SAT. Similarly, since $\overline{\text{3SAT}}$ is co-NP-complete, for each language $L_{\text{co-NP}} \in$ co-NP, there exists some bijective function $f_{L_{\text{co-NP}}}$ which reduces $L_{\text{co-NP}}$ to $\overline{\text{3SAT}}$. Thus any language $L_{\text{NP}} \in$ NP can be reduced to some language $L_{\text{co-NP}} \in$ co-NP by applying $f_{\text{NP}}(f(f_{\text{co-NP}}(x)))$ for each $x \in L_{\text{NP}}$ and any language $L_{\text{co-NP}} \in$ co-NP can be reduced to some language $L_{\text{NP}} \in$ NP by applying $f^{-1}_{\text{co-NP}}(f^{-1}(f^{-1}_{\text{NP}}(x)))$ for each $x \in L_{\text{co-NP}}$. Thus 3SAT $\leq_P$ $\overline{\text{3SAT}}$ implies that any language in NP is also in co-NP, and any language in co-NP is also in NP.

Next we will show that if NP = co-NP, then Σ_2^p = NP = π_2^p = co-NP:
Consider an arbitrary language $L \in$ NP. If NP = co-NP, then $L \in$ co-NP.
L being in NP implies that for some string $x \in L$, there exists a certificate c of polynomial size which a poly-time Turing machine can use to verify that $x \in L$.
L being in co-NP implies that for some string $x \in L$, for all arbitrary "counter examples", a poly-time Turing machine can verify these counter examples do not forbid x from being a member of L.

Consider an arbitrary language M such that $M \in \Sigma_2^p$. This means that M can be described as an existential statement followed by a universal. Since NP = co-NP, this latter universal statement can be rewritten as an existential statement, because instead of providing all counter examples, a verifier TM can be given a verifying certificate c. So M can be rewrwitten as two existential statements, which can further be rewritten as a single existential statement requiring a pair of values which would both verify an input string $x \in M$. Since M can be verified by a poly-time verifier and a certificate (containing a tuple), $M \in$ NP. Therefore, for all $M \in \Sigma_2^p \Rightarrow M \in$ NP.

It follows that any language $N \in \Pi_2^p \Rightarrow N \in$ co-NP, because N's universal and existential statement can be rewritten as a universal statement requiring a pair of values. And since $N \in$ co-NP, $N \in$ NP.

This strategy can be applied to all levels of the polynomial hierarchy. Any given class Σ_n^p can have its final existential or universal quantifier be rewritten as a universal or existential quantifier respectively and combined with the previous quantifier to expect a pair of values, which transforms the class into Σ_{n-1}^p, which itself can have its final quantifier be rewritten and be combined with the previous quantifier to expect a 3-tuple of values. This can be repeatedly applied until the class Σ_1^p is reached, which is NP. The same is true for any given class Π_n^p.

Finally, we can repeat the logic used to show that Σ_2^p and π_2^p collapse into NP and co-NP to show that higher levels of the polynomial time hierarchy collapse into NP and co-NP. Thus if 3SAT is polynomial-time reducible to $\overline{\text{3SAT}}$, then PH = NP.