CS 1511 Homework 11

Mathew Varughese, Justin Kramer, Zach Smith

Monday, Feb 25

19. a

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

To show EXACT INDSET $\in \pi_2^p$, we need to demonstrate two things.

$\forall x \forall y \forall z$ T(x, y, z) runs in time $\mid x \mid^k$

$\forall x \exists y$ T(x, y, z) accepts iff x $\in$ EXACT INDSET

Our language will now become $\forall x$, with x being an independent set of G, $\exists w$ where w is the largest independent set of size exactly k.

To solve this in poly-time, we will need 3 read-only tapes and 1 work tape, which is based on a simple scaling from our 2 read-only 1 work tape model with π_1^p.

Basically, it reads on the second tape the set of vertices. It goes through the tape and checks each vertex in the set and marks it on the graph (on the first tape). When it does this, it makes sure adjacent vertices are not touching to ensure it is an independent set.

Then, it checks the third tape, which is another set. It just needs to validate that this set is also a indepdent set and that it is larger or equal to in size to the set on the 2nd tape.

19. b

SUCCINCT SET-COVER = $\{< S, n, k >= :$ There is a subset of the collection of 3-DNF formulas S on n variables called S' where S' evaluates to 1 for every assignment to its k variables$\}$.

To show SUCCINCT SET-COVER $\in \Sigma_2^p$, we need to demonstrate two things.

$\forall x \forall y \forall z$ T(x, y, z) runs in time $\mid x \mid^k$

$\exists x \forall y$ T(x, y, z) accepts iff x $\in$ SUCCINCT SET-COVER

Our language will now become $\exists S'$, with S' being a subset of S, where $\forall w$ combinations of k variable assignments, the formula evaluates to 1.

To solve this in poly-time, we will need 3 read-only tapes and 1 work tape, which is based on a simple scaling from our 2 read-only 1 work tape model with Σ_1^p.

19. c

VC-DIMENSION = $\{< C, k >= :$ C represents a collection S s.t. VC(S) $\geq$ k$\}$.

To show VC-DIMENSION $\in \Sigma_3^p$, we need to demonstrate two things.

$\forall x \forall y \forall z \forall w$ T(x, y, z, w) runs in time $\mid x \mid^k$

$\exists x \forall y \exists z$ T(x, y, z, w) accepts iff x $\in$ VC-DIMENSION

Our language will now become $\exists C$, a Boolean circuit that $\forall X'$, where X' a subset of the size of the largest set, $\exists i$ such that subset Si of S has an intersection with X' that equals k and C(i, x) = 1.

To solve this in poly-time, we will need 4 read-only tapes and 1 work tape, which is based on a simple scaling from our 3 read-only 1 work tape model with Σ_2^p.

20. a

3SAT = $\{< \phi >= : \phi$ is a satisfiable 3CNF formula$\}$.

To reduce 3SAT to its complement, we need to have an algorithm for 3SAT that makes a call to its complement and runs in polynomial time.

Here is the algorithm.

L $\in$ 3SAT

$\phi \in$ L iff $\phi \in$ the compliment of 3SAT

Call to L(ϕ)

$\phi_2 = \neg\phi$

Call to complement 3SAT(ϕ_2)

This would mean that $\Sigma_n^p = \pi_n^p$ with n = 1, and then for any n $\geq$ 1 we could define n TM's, with each having a one-to-one correspondence with their respective oracles of Σ_n^p. Then, we could say a TM M is a concatenation of M1 ... Mn to prove that Σ_n^p answers yes iff M does so.

If this would be possible in polynomial time, then we could take any element in NP, change it to its compliment, run an algorithm in co-NP with that element and show reducability in poly-time.

The same could be done with any element in co-NP, which would therefore make the languages equal.

Also NP $\in \Sigma_1^p$ and co-NP $\in \pi_1^p$, so if the two sets of languages were equal then NP would equal co-NP by definition.

20. b

With the same logic as the previous problem, the hierarchy would eventually collapse so that any Σ_n^p could equal any π_n^p for the n=1 that the hierarchy collapses to. This would be a proof by induction that repeats the previous problem for all complexity classes. In this complete collapse to the bottom of the hierarchy, the union of all class PH would now be able to equal NP.