CS 1511 Homework 13

Mathew Varughese, Justin Kramer, Zach Smith

Monday, March 4

24.) According to our textbook, a circuit of size at most S can be represented as a string of 9 * S log S bits. This can create a number of circuits $2^{9S\log S}$. In our example, a circuit of size n^4 would be able to create $2^{9n^4\log n^4}$ circuits and a circuit of size n^2 would be able to create $2^{9n^2\log n^2}$ circuits.

Thus, the larger amounts of circuits buildable with n^4 gates would allow for more functions to be computed than the smaller circuit. There are therefore functions that can be computed with n^4 gates that can not be computed by n^2 gates.

This proof is very similar to showing that when S = $2^n/(10n)$, you can not compute as many functions as there are possible, since there are 2^{2^n} possible functions.

25.) Take a language L in PH.

L = { <x, k>| where $\exists$ a $C_{|x|}$ Boolean circuit with $|\ x\ |^k$ gates, and $\forall$ circuits D with less than $|\ x\ |^k$ gates, the circuit does not compute the same Boolean function as $C_{|x|}$.

This language is clearly in Σ_2^P, so therefore L is a language in PH.

A machine to check if a set of strings x is in L will have a runtime of O(2^{x^k}) when simulated on a Turing Machine.

The Turing Machine will take have to construct every possible circuit with less than $|\ x\ |^k$ gates, which will take O(2^{x^k}).

With this in mind, the amount of space necessary (circuit complexity necessary) will be $\Omega(n^k)$.

This is due to our circuit being able to hardwire in all the possibilities from our Turing Machine in polynomial time.

In exponential time, we can use poly-space to model our Turing Machine.

Thus, for every k > 0 there is a language in PH whose circuit complexity is $\Omega(n^k)$.