Solutions to Homework II
CS6520
Computational Complexity

March 2, 2005

Problem: Show that $\Sigma_k = NP^{CIRCUIT-SAT_{k-1}}$.
Solution: We define $L \in \Sigma_k$ iff there is a deterministic verifier V such that

$$x \in L \iff \exists y_1 \ \forall y_2 \cdots \ Q_k y_k \ \ V(x, y_1, \cdots, y_k) = 1$$

The computation of the verifier can be performed by a circuit as in Cook's theorem. In other words,

$$x \in L \iff \exists y_1 \text{ such that } \ \forall y_2 \cdots \ Q_k y_k \ \ C_{x,y_1}(y_2, \cdots, y_k) = 1$$

The NTM guesses y_1. The problem of deciding whether

$$\forall y_2 \cdots \ Q_k y_k \ \ C_{x,y_1}(y_2, \cdots, y_k) = 1$$

can be done by an oracle for $CIRCUIT-SAT_{k-1}$. So $\Sigma_k \subseteq NP^{CIRCUIT-SAT_{k-1}}$.

For the other direction, the Σ_k machine M^k guesses all the moves of the NTM M (call this string y), all queries made by M, and the answers of the oracle and then simulates M. If M rejects, M^k rejects. If M accepts, M^k needs to check thats its guesses for the oracle-answers are correct. If the answer guessed is YES, then we have to check

$$\exists y_1 \forall y_2 \cdots \ Q_{k-1} y_{k-1} \ \ C(y_1, y_2, \cdots, y_{k-1}) = 1$$

This is in Σ_{k-1}, so it can be done by the Σ_k machine itself. For the queries whose answer was guessed to be NO, we need to check for $i = 1, \cdots, \ell = poly(n)$,

$$\forall y_1^i \exists y_2^i \cdots \ Q_{k-1} y_{k-1}^i C^i(y_1^i, y_2^i, \cdots, y_{k-1}^i) = 0$$

These queries can be combined to a single query checkable by a Π_{k-1} machine

$$\forall y_1^1, \cdots y_1^\ell, \exists y_2^1, \cdots y_2^\ell \; Q_{k-1} y_{k-1}^1, \cdots, y_{k-1}^\ell \vee_{i=1}^\ell C^i(y_1^i, y_2^i, \cdots, y_{k-1}^i) = 0$$

Thus the language can be decided with k alternating quantifiers. $\square$

Problem: Show that $P^{PSPACE} = NP^{PSPACE} = PSPACE$
Solution: We will show that $NP^{PSPACE} \subseteq PSPACE$. Then we have

$$PSPACE \subseteq P^{PSPACE} \subseteq NP^{PSPACE} \subseteq PSPACE$$

and hence all above classes equal $PSPACE$.

The claim that $NP^{PSPACE} \subseteq PSPACE$ is fairly obvious. A PSPACE machine M can go over all possible choices of the NP machine and in addition, whenever the NP machine asks a PSPACE query, the machine M can compute the answer to this query by itself. $\square$

Problem: Can PH have a complete problem ? Show that if $PH = PSPACE$, then PH collapses.
Solution: If PH has a complete problem at level k, then PH will collapse to level k. If PH = PSPACE, then it will have a complete problem, namely Quantified CIRCUIT-SAT. $\square$

Problem: The problem SAT-UNSAT is defined as follows: Given Boolean formulae ϕ, ψ, decide if ϕ is satisfiable And ψ is unsatisfiable. Show this problem is DP-complete.
Solution: Define

$$B = \{< \phi, \psi > | \phi \text{ is satisfiable}\}$$

$$C = \{< \phi, \psi > | \psi \text{ is unsatisfiable}\}$$

Then $B \in NP$, $C \in co - NP$ and $L = B \cap C$.

Given $L = B \cap C$, the problem "Is $x \in B$?" is reducible to "Is ϕ satisfiable?". The problem "Is $x \in C$" is reducible to "Is ψ unsatisfiable". Hence deciding $x \in L$ reduces to deciding $< \phi, \psi > \in SAT - UNSAT$. $\square$

Problem: A graph G is in HC-CRITICAL is G is not Hamiltonian but adding any edge to G will make it Hamiltonian. Show that HC-CRITICAL

is in DP.
Solution: Define

$$NOT - HAM = \{G | G \text{ is not Hamiltonian}\}$$

$$HC = \{G | G + e \text{ is Hamiltonian for any } e \notin G\}$$

It is clear that $NOT - HAM \in co - NP$. Also $HC \in NP$ since we can guess for every $e \notin G$ the Hamiltonian cycle in $G+e$ (and there are at most $\binom{n}{2}$ such edges). $\square$

Problem: Show that $NP^{BPP} \subseteq BPP^{NP}$.
Solution: If $L \in BPP$, then the following language is in BPP.

$$L' = \{< (x_1, a_1), (x_2, a_2), ...(x_n, a_n) > | x_i \in \{0,1\}^*, a_i \in \{YES, NO\}\}$$

where $a_i = YES/NO$ depending on whether or not $x_i \in L$. To show this, simulate BPP machine for L for every i and make sure that the error probability is at most $\frac{1}{n^2}$ so that w.h.p. the machine will be correct for all i.

Using this we can simulate a NP^{BPP} machine for B by a NP^{BPP} machine that makes only one query to the BPP oracle and that too at the end. Thus, if A is the BPP oracle,

$$x \in B \iff \exists\ y_1, y_2, V(x, y_1, y_2) = 1 \ \text{ and } \ y_2 \in A$$

By the definition of BPP,

$$x \in B \implies \exists\ y_1, y_2, V(x, y_1, y_2) = 1 \text{ and } \Pr_{y_3}[V'(y_2, y_3) = 1] \geq 1 - \frac{1}{2^{poly(n)}}$$

$$x \notin B \implies \forall\ y_1, y_2, V(x, y_1, y_2) = 0 \text{ or } \Pr_{y_3}[V'(y_2, y_3) = 1] \leq \frac{1}{2^{poly(n)}}$$

Now we can switch order of (y_1, y_2) and y_3 (verify this ! you need to take union bound over all choices of y_1, y_2). Hence

$$x \in B \implies \Pr_{y_3}[\exists y_1, y_2 \ s.t.\ V(x, y_1, y_2) = 1, V'(y_2, y_3) = 1] \geq 1 - \frac{1}{2^{poly(n)}}$$

$$x \notin B \implies \Pr_{y_3}[\exists y_1, y_2 \ s.t.\ V(x, y_1, y_2) = 1, V'(y_2, y_3) = 1] \leq \frac{1}{2^{poly(n)}}$$

This can be simulated by a BPP^{NP} machine (randomly pick y_3 and the rest is an NP question). $\square$

Problem: If $NP \subseteq BPP$, then PH collapses to BPP.
Solution: If $NP \subseteq BPP$ then

$$\Sigma_2 = NP^{NP} \subseteq NP^{BPP} \subseteq BPP^{NP} \subseteq BPP^{BPP} = BPP$$

(Verify the last step !). Similarly, $\Sigma_i \subseteq BPP \ \forall i$. $\square$

Problem: Implicit Circuit-SAT is NEXP-complete.
Solution: Given a NEXP machine, running the reduction in Cook's theorem gives a circuit C^* of size 2^{n^k}. We claim that this circuit has a small $poly(n)$ size implicit representation C. Note that in Cook's reduction, every cell of the work tape is computed from three other cells using a constant size circuit (which encodes the computation of the NTM). One can use this to get an implicit representation. $\square$

Problem: Given $< M, x, n >$ where n is in binary, deciding if M accepts x in n steps is NEXP-complete.
Solution: Call the above language A. A is in NEXP, since we can simulate M on x for n steps in time $poly(|M| + |x| + n)$ which is at worst exponential in the input size $(|M| + |x| + \log n)$.

Let L in NEXP. Then $x \in L$ iff it is accepted by an NTM M^L running in time $2^{|x|^k}$. This reduces to deciding if $< M^L, x, 2^{|x|^k} > \in A$. $\square$

Problem: Show that $NP^{\Sigma_k \cap \Pi_k} = \Sigma_k$.
Solution: Let $L \in \Sigma_k \cap \Pi_k$. We want to simulate NP^L by a Σ_k machine. Consider a Σ_k machine that guesses the moves of the NP machine and the answers of the oracle for L. We want to verify that the guessed answers are correct. If we guess that $x \in L$, then since $L \in \Sigma_k$, this can be verified by the Σ_k machine. If we guess that $x \notin L$, then $x \in \bar{L}$. But since $L \in \Pi_k$, $\bar{L} \in \Sigma_k$, so the Σ_k machine can also verify that $x \notin L$. $\square$

Problem: Show that if $\Sigma_k = \Pi_k$, then PH collapses to level k.
Solution: Since $\Sigma_k = \Pi_k$ then $\Sigma_k \cap \Pi_k = \Sigma_k$. Then $\Sigma_{k+1} = NP^{\Sigma_k} = NP^{\Sigma_k \cap \Pi_k} = \Sigma_k$ by the previous problem. Now it is easy to see that PH collapses to Σ_k. $\square$

Problem: The problem Graph Consistency (GC) asks, for two given sets A and B of graphs, whether there exists a graph G such that every graph

$g \in A$ is isomorphic to a subgraph of G but each graph $h \in B$ is not isomorphic to any subgraph of G, Show that GC is in Σ_2.

Solution: Note that the input size here is $|A| + |B|$. So first we show that if such a G exists, its size is polynomial in $|A|, |B|$. Indeed assuming G exists, let σ_i denote the isomorphisms from $g_i \in A$ to G. We can take G' to be the subgraph incuded by the images of $\sigma_i(g_i), g_i \in A$. It is easy to show that G' is also a solution to GC and its size is bounded by $\sum_i |g_i| = |A|$.

Now the problem can be stated as

$$GC = \{< A, B > | \exists G, \exists \sigma_i : g_i \to G, \forall \pi_j : h_j \to G, \sigma_i \text{ is an isomorphism but not } \pi_j\}$$

Since the size of G, σ_i, π_j is bounded in $|A| + |B|$, this is in Σ_2. $\square$