Complexity Homework 2

Released: March 1, 2007

Part I

Due: March 13, 2007

Problem 1:

A language A is *downward self-reducible* if there is a polynomial-time oracle machine M such that:

- $L(M^A) = A$. That is, when given an oracle for A, M decides A (self-reducibility).
- On input x, M only queries the oracle on strings *smaller* than x (downward reducibility).

The second restriction is necessary to make the property interesting – otherwise, on input x, M could just directly ask the oracle if $x \in A$.

(a) Show that SAT and TQBF are downward self-reducible.

(b) Show that if L is downward self-reducible, then $L \in \mathbf{PSPACE}$.

Problem 2:

Show that $\mathbf{NSPACE}(S) \subseteq \mathbf{ATIME}(S^2)$ for $S(n) \geq \log n$ (and S space-constructible).

(Hint: Recall the algorithm in the proof of Savitch's theorem. Use existential quantifiers to guess the "middle node" in paths and universal quantifier to check both halfs of the path exist. To be clear, give the certificate version of the ATM, clearly stating how many certificate tapes are there, how long each one is, and how the deterministic verification works and in what time.)

Conclude that $\mathbf{PSPACE} \subseteq \mathbf{AP}$. (Of course, we have already seen in class that $\mathbf{PSPACE} = \mathbf{AP}$.)

Problem 3:

What can you say about the class $\mathbf{\Sigma}_k^{p\,\mathbf{\Sigma}_\ell^p}$ for different values of k and ℓ? Prove your claim. (Remember that an oracle can be queried multiple times. Consider cases of k, ℓ being odd/even and 0, 1, 2 etc. For partial credit consider one or more of these special cases.)

Problem 4:

An oracle machine is called a *robust oracle machine* if the language accepted by it remains the same no matter which oracle is used (however the running time may vary). Show that a language L is decided by M^K *in polynomial time* where M is a robust oracle machine and K is some oracle, if and only if $L \in \mathbf{NP} \cap \mathbf{co\text{-}NP}$.

Part II

Due: March 16, 2007

Problem 5:

We have seen in class that PATH is $\mathbf{NL}$-complete with respect to log-space reductions. Define $\mathbf{NC}^1$ reduction and show that in fact PATH is $\mathbf{NL}$-complete with respect to $\mathbf{NC}^1$ reductions.

Problem 6:

We often assume that a circuit has NOT gates only at the input level.

(a) Show how to convert a circuit *with fan-out one for all gates* (except possibly for the input gates) into an equivalent circuit of no larger size, but with NOT gates only at the inputs.

(b) Show how to convert a general circuit (with any fan-out for the gates) into an equivalent circuit of at most twice the size, with NOT gates only at the inputs.

Problem 7:

(Exercise 6.14 from Arora-Barak.) Show that the class of languages decided by DC-uniform circuit families of exponential size (and unbounded depth) is exactly $\mathbf{EXP}$.