College of Computer & Information Science	Spring 2008
Northeastern University	Handout 20
CSG714: Theory of Computation	8 April 2008

Problem Set 6 (due Friday, April 18)

1. (5 + 5 = 10 points) Amplification of soundness error for PCPs

Define $\mathrm{PCP}_\varepsilon(r(n), q(n))$ as the set of languages that have a PCP protocol using $O(r(n))$ random bits and $O(q(n))$ queries to achieve soundness error ε (there is no error with respect to completeness). Thus, $\mathrm{PCP}(r(n), q(n))$ is $\mathrm{PCP}_{1/2}(r(n), q(n))$.

(a) Suppose L is in $\mathrm{PCP}_{1/2}(r(n), q(n))$. Prove that for any $k > 0$, L is in $\mathrm{PCP}_{1/2^k}(kr(n), kq(n))$.

(b) Using the PCP Theorem, show that NP is a superset of $\mathrm{PCP}_{1/n}(\log n, \log n)$.

2. (7 points) ZPP

Problem 10.20 from Sipser's text.

3. (7 points) BPL

Problem 10.22 from Sipser's text.

4. (6 × 6 = 36 points) A random walk through CSG714

(a) Let A be any regular language over the alphabet Σ. Define

$$\textsc{Substring}(A) = \{x \mid \exists w, y : wxy \in A\}$$

Prove or disprove: $\textsc{Substring}(A)$ is a regular language.

(b) Let L be the language

$$\{\langle M \rangle \mid M \text{ is a DFA and accepts all palindromes}\}$$

Prove or disprove: L is decidable.

(c) In our proof of the existence of a language that is not Turing-recognizable, we showed that the set of all Turing machines over a fixed input alphabet Σ is countable, while the set of all languages over Σ is uncountable. In practice, of course, the input alphabets of different Turing machines could be different. Suppose we have an infinite countable set S of symbols $\{a_1, a_2, \ldots, \}$ from which the alphabet for every Turing Machine is chosen. Is it still true that the set of all Turing machines is countable? (Note that even though the set S is infinite, the alphabet for each Turing machine is finite.)

(d) Are all context-free languages in NL?

(e) Consider the following argument. QBF is complete for PSPACE with respect to logspace reductions. QBF is in linear space. Thus, every problem in PSPACE can be reduced to a linear space problem using logspace reductions. So PSPACE is, in fact, equal to linear space.

Is there a fallacy in the above argument?

(f) In our definition of the Interactive Proofs class (IP), we assumed two-sided errors (say, $1/3$ error in both soundness and completeness). Define IP^* as the class of languages that have IP protocols with soundness error at most $1/3$ and no completeness error. Is IP^* identical to IP?