CS-E4530 Computational Complexity Theory, Spring 2020
Problem Set 8 **Homework Due Tue 17 Mar, 23:59**

Instructions

- Classroom Problems C8.1–C8.3 will be discussed and solved at the tutorial session on Wed 11 Mar, 14–16, Room T4 (A238). No credit is given for these problems.
- Homework Problems H8.1–H8.3 you should solve on your own, and submit your solutions via the MyCourses interface by the deadline of Tue 17 Mar, 23:59, These problems will be individually graded on a scale of 0–2 points per problem.
- In preparing your solutions to the Homework Problems:
  1. Justify your solutions, be precise, and provide sufficient detail so that it is easy to follow your reasoning.
  2. Submit your solutions as an easily readable, single pdf file, which is either typeset or written in full sentences and clean handwriting.
  3. **[Code of Conduct]** You can discuss the problems with your colleagues and the course's teaching staff, but you must write the presentations of your solutions *independently* and *individually*, without any notes from such discussions.

Classroom Problems

C8.1 By the Prime Number Theorem,[1] about a fraction $1/n$ of all n-bit integers are prime. Based on this fact, show that the expected number of random n-bit numbers one needs to generate and test before finding a prime is $\mathcal{O}(n)$. (*Hint:* Suppose a particular coin has a probability p of coming up heads. How many times must you toss it, before it comes up heads? Method 1: Start by showing that the correct expression is $\sum_{i=1}^{\infty} i(1-p)^{i-1}p$. Method 2: If E is the expected number of coin tosses, show that $E = 1 + (1-p)E$.)

C8.2 Prove that if p and q are distinct primes, then for every integer a and exponent e with $e \not\equiv 0 \pmod{(p-1)(q-1)}$, we have

$$a^e \equiv a^{e \bmod (p-1)(q-1)} \pmod{pq}.$$

C8.3 Explain why the interactive proof system given for the GRAPH NONISOMORPHISM problem at Lecture 14 is in fact a zero-knowledge protocol, i.e. why at the end of the protocol the verifier Bob has learned nothing from Alice that he wouldn't have known also without the interaction, except for the answer to the nonisomorphism question.[2]

[1] `https://en.wikipedia.org/wiki/Prime_number_theorem`

[2] Strictly speaking, this is only true if Bob *follows* the given protocol. Can you see how Bob could cheat Alice to leak information by breaking the protocol?

Homework Problems

H8.1 Design a zero-knowledge interactive proof system for the GRAPH ISOMORPHISM problem. (*Hint:* Compared to the protocol presented at Lecture 14, you might want to let prover Alice make the first move.) Explain why your protocol doesn't give Bob any information which he couldn't have generated by himself, except for the answer to the isomorphism question. (2 points)

H8.2 Show that $\mathbf{P}^{\mathbf{PP}} = \mathbf{P}^{\#\mathbf{P}}$. (Note that in order to define the class $\mathbf{P}^{\#\mathbf{P}}$, one needs to modify the model of oracle Turing machines so that they can also receive output from their queries. Alternatively, one could simply define $\mathbf{P}^{\#\mathbf{P}}$ to equal $\mathbf{P}^{\#\mathrm{SAT(D)}}$.) (2 points)

H8.3 In this problem we continue on the topics of randomised computation and circuit complexity, to derive the fundamental result that $\mathbf{BPP} \subseteq \mathbf{P/poly}$, i.e. all languages with "good" polynomial-time randomised algorithms have also polynomial-size circuits. Referring to Problem H7.3, this result has the notable consequence that if SAT $\in \mathbf{BPP}$, then $\Sigma_2^p = \Pi_2^p$.

We shall use the following characterisation of the class $\mathbf{P/poly}$ from Problem C7.1: For a language L, $L \in \mathbf{P/poly}$ if and only if there exist a polynomially balanced, polynomially decidable binary relation R and a sequence of *advice strings* $w_0, w_1, w_2, \ldots$, such that for each $n \geq 0$:

$$x \in L \;\equiv\; R(x, w_n) \quad \text{holds for all } x \text{ with } |x| \leq n.$$[3]

In the following, we assume for simplicity and w.l.o.g. that all languages are over the binary alphabet, i.e. $L \subseteq \{0,1\}^*$.

(i) Prove that for any language $L \in \mathbf{BPP}$, there is a polynomially balanced, polynomially decidable binary relation R such that for every $n \geq 0$ and x, $|x| = n$:[4]

$$\Pr_w[L(x) \neq R(x,w)] \leq \min\{2^{-2n}, 1/4\}.$$

(*Hint:* Let M be a $\mathbf{BPP}$ machine that decides L with error probability $\leq \frac{1}{4}$. Consider another machine M' that for $n \geq 1$ performs m independent runs of M and then decides according to the majority result. Use the Chernoff bound to estimate how large m needs to be to achieve the indicated error bound.)

[3] In fact, in Problem C7.1 we only proved the "only if" direction of this characterisation, i.e. if L has polynomial-size circuits, then such an "advice-augmented" decision method exists. But the converse is also quite easy to prove, by converting the computations for R on the respective advice strings into circuits.

[4] For simplicity, we denote here $L(x) = 1$ (0) if $x \in L$ (resp. $x \notin L$), and similarly for the relation $R(x, w)$.

(ii) Continuing from above, prove that for every $n \geq 0$ there exists an advice string w such that

$$L(x) = R(x, w), \quad \text{for all } x \text{ with } |x| = n.$$

(*Hint:* For a given $n \geq 0$, estimate the probability

$$\Pr_w[\exists x, |x| = n, \text{ s.th. } L(x) \neq R(x, w)]$$

and show that it is less than 1.) (2 points)