CSE431: Complexity Theory November 29, 2018

Homework 4

Anup Rao Due: December 7, 2018

Read the fine print[1]. Each problem is worth 10 points:

1. Prove that **NP** $\neq$ co-**NP** implies that **BPP** $\neq$ **NP**.

2. Consider the following game between two players: Given a directed graph $G = (V, E)$, and a start vertex s, the players (starting with Player 1) alternately choose an outgoing edge incident to the current vertex to reach a vertex that was not previously visited. If one of the players cannot choose a next vertex, he loses. Let **GAME**(G) be the function that is 1 if and only if Player 1 has a strategy that ensures that she always wins no matter what Player 2 does.

   Show that **GAME** is in **PSPACE**.

3. An arithmetic circuit is the same as a boolean circuit, except that every gate either computes the product of the two inputs, or the sum of the two inputs. One can also have constants that feed into the circuit. The circuit maps inputs in $\mathbb{R}^n$ to a real number $\mathbb{R}$. The circuit can also be thought off as encoding a polynomial.

   (a) Suppose you are given two arithmetic circuits such that every gate of each circuit computes a polynomial of degree at most n, and the coefficients of the polynomial is promised to have magnitude at most 2^n. Use the Schwartz-Zippel lemma to give a randomized algorithm in **RP** to decide whether the two polynomials are equal or not. (Be careful when analyzing your agorithm: if x, y are numbers, then $x \times y$ can be significantly larger. You need to make sure that the numbers do not become so big that your algorithm is unable to multiply them!).

   (b) Suppose you are given two arithmetic circuits, with no other promises. Give a randomized algorithm in **RP** to decide whether the polynomials are the same or not. To do this:

      i. Prove that the degree of the polynomials computed by the circuits is at most 2^s, where s is the size of the larger circuit.

      ii. Now, the problem is that we cannot evaluate these circuits on large integers in polynomial time, because the size of the integers might become exponentially large. However, if p is a prime of size at most 2^s, then we can evaluate these circuits

[1] In solving the problem sets, you are allowed to collaborate with fellow students taking the class, but **each submission can have at most one author**. If you do collaborate in any way, you must acknowledge, for each problem, the people you worked with on that problem. The problems have been carefully chosen for their pedagogical value, and hence might be similar to those given in past offerings of this course at UW, or similar to other courses at other schools. Using any pre-existing solutions from these sources, for from the web, constitutes a violation of the academic integrity you are expected to exemplify, and is strictly prohibited. Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers. Some other guidelines for writing good solutions are here: `http://www.cs.washington.edu/education/courses/cse421/08wi/guidelines.pdf`.

modulo p in polynomial time, by carrying out all arithmetic modulo p. Use this fact to give a randomized algorithm to decide whether the polynomials are the same or not in **RP**. (It might be helpful to review the proof of the fingerprinting algorithm discussed in class).

(c) Suppose you could give a polynomial time randomized algorithm to tell whether or not two *boolean* circuits compute the same boolean function. What consequence would this have with regards to the relationship between the classes $\mathbf{P}, \mathbf{BPP}, \mathbf{NP}$?