Computability and Complexity Theory

Instructor: Alexander Razborov, University of Chicago
razborov@math.uchicago.edu
Course Homepage: www.cs.uchicago.edu/~razborov/teaching/winter17.html

Winter Quarter, 2017

```
   You may work together on solving homework problems, but please
put all the names clearly at the top of the assignment.  Everyone
must turn in their own independently written solutions.  If you
find any of the solutions on-line, it is OK but then you must completely
understand the proof, explain it in your own words and include the
URL.
```

Homework 3, due March 7

1. Let L be a language that is Cook reducible[1] to the language $\left\{ \left(x, 1^{\left(2^{|x|}\right)}\right) \mid x \in L \right\}$. Prove that $L \in \mathsf{P}$.

2. A *perfect matching* in a graph with $2n$ vertices is a set of n edges such that every vertex is incident to exactly one edge in a set. BIPARTITE PERFECT MATCHING is the language that consists of all (encodings of) bipartite graphs that possess at least one perfect matching.

   Show that BIPARTITE PERFECT MATCHING is in $\mathsf{NP} \cap co-\mathsf{NP}$ be exhibiting *explicit*, *direct* and *combinatorial* many-one poly-time reductions to SATISFIABILITY.

The remaining problems were kindly contributed by Prof. Babai.

We study the communication complexity of certain Boolean functions $f(x, y)$. $C(f)$ denotes the deterministic communication complexity, $D_\delta^\mu(f)$ the δ-error distributional complexity under a probability distribution μ on

[1] Cook reducibility is a polynomial analog of the Turing reducibility, see [AroraBarak, Exercise 2.14]

the set of inputs, and $R_\delta(f)$ the δ-error randomized communication complexity with public randomness (Alice and Bob both have access to a shared infinite random binary string $r_1 r_2 \ldots$, where the r_i are independent uniform random bits).

Comparing rates of growth. Recall that for functions $g, h : \mathbb{N} \to \mathbb{R}$ satisfying $g(n), h(n) \geq 0$ we say that $g(n) = O(h(n))$ if there exists a constant c such that $g(n) \leq c \cdot h(n)$ for all sufficiently large n; and $h(n) = \Omega(g(n))$ means $g(n) = O(h(n))$. So "O" indicates an upper bound and "Ω" a lower bound.

3. (Randomized complexity of equality)

   (a) Let $x, y \in \mathbb{F}_2^n$ be column vectors of length n over the field of order 2. Fix $x \neq 0$ and choose y uniformly at random from $\mathbb{F}_2^n$. Prove: $\Pr(x^T y = 0) = 1/2$, where ^{T} denotes "transpose."

   (b) Let EQ denote the "equality" function over the domain $X = \{0,1\}^n$, i.e., for $x, y \in X$ we set $\mathrm{EQ}(x,y) = 1$ if $x = y$ and $\mathrm{EQ}(x,y) = 0$ otherwise. Prove: $R_\delta(\mathrm{EQ}) \leq \lceil \log_2(1/\delta) \rceil$.

4. (Complexity of inequality) Let GE denote the "greater or equal" function on n-bit integers, i.e., for $0 \leq x, y < 2^n$ we set $\mathrm{GE}(x,y) = 1$ if $x \geq y$ and $\mathrm{GE}(x,y) = 0$ otherwise.

   (a) Prove: $C(\mathrm{GE}) = n$.

   (b) Prove: $D_\delta^{\mathrm{unif}}(\mathrm{GE}) = O(\log(1/\delta))$ where "unif" denotes the uniform distribution.

   (c) Prove: $R_\delta(\mathrm{GE}) = O(\log n(\log\log n + \log(1/\delta)))$.

5. (Complexity of forest) Notation: A graph $G = (V, E)$ consists of a set V of vertices and a set E of edges. (E is a set of unordered pairs of vertices.) A graph that has no cycles is called a *forest.*

   Alice and Bob share a set V of n vertices. Alice's input is a graph $X = (V, A)$ and Bob's input is a graph $Y = (V, B)$. Alice and Bob wish to decide whether or not the graph $X \cup Y := (V, A \cup B)$ is a forest. Let us call this problem "FOREST."

   (a) Prove: $C(\mathrm{FOREST}) = O(n \log n)$.

   (b) Prove: $C(\mathrm{FOREST}) = \Omega(n \log n)$.

   (c) Prove: for some constant $\delta > 0$ we have $R_\delta(\mathrm{FOREST}) = \Omega(n)$. (Hint: Use the fact that for some constant $\delta > 0$, the δ-error randomized complexity of "disjointness" of subsets of a set of n elements is $\Omega(n)$.)