MCS 541 – Computational Complexity
Spring 2023
Problem Set 2*

Lev Reyzin

Due: 2/13/23 at the beginning of class

1. Define a RAM Turing machine to be a Turing machine that has *random access memory* – unlike TMs, this is a model of computation commonly used to analyze algorithms. We formalize this as follows: the machine has an additional work tape called an address tape, two additional symbols in its alphabet that we denote by R (for "read") and W (for "write"), and an additional state we denote by q_{access}. We also assume that the machine has an infinite array A that is initialized to all blanks. Whenever the machine enters q_{access}, if its address tape is of the form $\langle i \rangle R$ (recall that $\langle i \rangle$ denotes the binary encoding of i) then the value $A[i]$ is written in the cell after the R symbol. If its tape takes the form $\langle i \rangle W \sigma$ (where σ is some symbol in the machine's alphabet) then $A[i]$ is set to the value σ. Explain how you could efficiently simulate a RAM Turing machine with a TM.[1]

2. Show that the problem of determining whether a formula in 3-CNF has a satisfying assignment in which *exactly* one literal evaluates to true in each clause is NP-complete. (Recall that a literal is either a variable or its negation.)

3. The "verifier definition" of **coNP** says that a language $L \subseteq \{0,1\}^*$ is in **coNP** if $\exists$ a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time TM M such that for every $x \in \{0,1\}^*$,

$$x \in L \iff \forall u \in \{0,1\}^{p(|x|)} \text{ s.t. } M(x,u) = 1.$$

Prove that this defines the same class as the "standard" definition: $\mathbf{coNP} = \{L : \bar{L} \in \mathbf{NP}\}$.

4. The notion of polynomial-time reducibility in Cook's paper involved efficient Turing reductions, which are now called Cook reductions (and denoted as $\leq_T^P$). A language A is Cook-reducible to a language B if there is a polynomial time TM M that, given an oracle (i.e. a magic black box that can decide a language in a single step) for deciding B, can decide A. The reason that we define **NP** and **coNP** using Karp reductions instead of Cook reductions is that Cook reductions are too powerful to distinguish **NP** from **coNP**. Show that **NP** = **coNP** under Cook reductions.

5. Define the language FACT $= \{\langle n,k \rangle |\ n$ has a prime factor that is smaller than $k\}$. FACT is believed to be neither in **P** nor in **NPC**. Show that FACT $\in$ **NP** $\cap$ **coNP**. You may use the result that PRIMES $\in$ **P** (Agrawal, Kayal, and Saxena 2004), which gives an efficient test for primality.

*Some of these problems are modifications of exercises that appear in Arora-Barak.

[1] In fact, it is known that if a Boolean function f is computable in time $T(n)$ (for some time-constructible T) by a RAM Turing machine, then it is in $\mathbf{DTIME}(T(n)^2)$.