QUALIFYING EXAMINATION
THEORETICAL COMPUTER SCIENCE
FRIDAY, OCTOBER 19, 2018
PART II: AUTOMATA AND COMPLEXITY

Instructions:

1. This is a closed book exam.
2. The exam has four problems worth 25 points each. Read all the problems carefully to see the order in which you want to tackle them. You have all day (9am–5pm) to solve the problems.
3. Write clearly and concisely. You may appeal to some standard algorithms/facts from textbooks unless the problem explicitly asks for a proof of that fact or the details of that algorithm.
4. If you cannot solve a problem, to get partial credit write down your main idea/approach in a clear and concise way. For example you can obtain a solution assuming a clearly stated lemma that you believe should be true but cannot prove during the exam. However, please do not write down a laundry list of half-baked ideas just to get partial credit.
5. In problems with multiple parts, if you cannot solve a part, you can still solve subsequent parts (assuming that claims from earlier parts are true) and get partial credit.

May the force be with you.

Problem 1: A string of the form xx for some $x \in \{0,1\}^*$ is called a *square*. Consider the following DFA-Accepts-Some-Square problem: given a DFA M with n states over the alphabet $\{0,1\}$, decide whether $L(M)$ contains a square.

(a) First show that if $L(M)$ contains a square, it contains a square of length at most $2n^2$.

[Hint: Suppose $xx \in L(M)$. Let q_0 be the initial state, and q_m be the state after reading x starting from q_0. For each $i = 1, \ldots, |x|$, consider the pair of states after reading the first i symbols of x starting from q_0 and starting from $q_m \ldots$]

(b) Show that DFA-Accepts-Some-Square is in **P**.

[Hint: Modify your proof of (a) and build a directed graph with $O(n^2)$ vertices.]

Problem 2: In this problem, by "Turing machine", we mean a *deterministic* 2-tape Turing machine with a *read-only* input tape and a *read-write* work tape. In each step such a machine, reads the current cell of the input and work tapes, and based on the current control state, changes its control state, writes a symbol on the work tape, and moves both input and work tape heads, independently, either left or right. A computation of such a machine is said to be *non-erasing* if in each step, if a non-blank symbol is read on the work tape then the same symbol is written back. In other words, the only symbols that are "changed" during the computation are blank symbols on the work tape.

NonErasing is the following problem: Given a Turing machine M and input w, answer "yes" if M's (unique) computation on w is non-erasing. Prove that NonErasing is not recursively enumerable. *Hint:* Can you prove that for any Turing machine M, there is another Turing machine M' that "simulates" M in a non-erasing manner on all inputs?

Problem 3: Consider the following two versions of the "k-SUM" problem:

> VERSION 1. Given a set S of n elements and a "target" number t, where each element is an m-bit number (i.e., an integer in $[0, 2^m)$), do there exist k distinct elements $s_1, \ldots, s_k \in S$ such that $s_1 + \cdots + s_k = t$?
>
> VERSION 2. Given k sets $S_1, \ldots, S_k$ with a total of n elements, and a "target" number t, where each element is an m-bit number, do there exist k (not necessarily distinct) elements $s_1 \in S_1, \ldots, s_k \in S_k$ such that $s_1 + \cdots + s_k = t$?

It is not difficult to see that Version 2 reduces to Version 1 (by adding some leading bits to the numbers). In this problem, you will give a randomized reduction from Version 1 to Version 2.

(a) Given k elements, suppose that we assign each element with a random color independently chosen from $\{1, \ldots, k^2\}$. Show that the probability that all elements receive different colors is at least a positive constant.

(b) Now show that if Version 2 can be solved in $T(k, n, m)$ time, then Version 1 can be solved in $O(T(k^2, n, m + O(\log k)))$ time by a randomized algorithm with correctness probability at least 0.99.

Problem 4: In the *Orthogonal Vectors (OV)* problem, we are given two sets of n d-dimensional 0-1 vectors $A \subseteq \{0,1\}^d$ and $B \subseteq \{0,1\}^d$ with $|A| = |B| = n$, and we want to decide whether there are vectors $\vec{a} \in A$ and $\vec{b} \in B$ that are orthogonal, i.e., such that $\langle \vec{a}, \vec{b} \rangle = \sum_i a_i b_i = 0$. The *Orthogonal Vectors Conjecture (OVC)* of fine-grained complexity states that for every $\delta > 0$ there is a $c \geq 1$ such that OV cannot be solved in $n^{2-\delta}$ time on instances with $d = c \cdot \lg n$. This problem will explore OVC and how it relates to other conjectures in fine-grained complexity.

In particular, one formulation of the *Strong Exponential Time Hypothesis (SETH)* states that for every $\epsilon > 0$ there is a k and a c such that k-SAT for n variables and cn clauses requires time $\Omega((2 - \epsilon)^n)$. (The input to k-SAT is a CNF formula where clauses have length k.)

(a) Show that OV can be solved in $O(n^2\text{poly}(d))$ time.

(b) Show that OV can be solved in $O(n2^{O(d)})$ time.

(c) Show a reduction from k-SAT with n variables and m clauses to OV with $O(2^{n/2})$ vectors in m dimensions, that runs in time $O(2^{n/2}\text{poly}(m))$. [Hint: partition the n variables into two groups of size $\frac{n}{2}$ each.]

(d) Conclude that SETH implies OVC.