CS 351: Problem Set 1
(assigned 10/3/2002; due 10/10/2002)

Despite the length of this document, this is a fairly easy set of problems!

On each problem, you may work with at most one other person, but you must write up the solutions independently. Please turn in each problem on a separate sheet of paper that also contains your name, the name of your collaborator (if any), and other sources (eg., papers, books) that you consulted.

In Problem (1), we explore some simple complexity-theoretic consequences of "padding," a very simple kind of reduction. Problem (2) considers some special cases of Boolean satisfiability, each of which turns out to be solvable quite efficiently. In Problem (3), we begin the development of *communication complexity*, a very important methodology that has touched diverse areas in complexity theory, especially concrete complexity. (See the citation for Andy Yao's Turing award.) In Problem (3), we will use communication complexity to prove an $\Omega(n^2)$ lower bound for the PALINDROMES problem. Throughout this quarter, we will continue to develop (via problem sets) tools and techniques in communication complexity as it pertains to the study of the usual time and space complexity notions.

(1) In parts (a) and (b) of this problem, we explore the idea of *padding* of decision problems, that is, a reduction from problem A to problem B, where B is usually defined as $\{x1^t \mid x \in A\}$ for a suitable choice of t (that may depend on x).

(1a) Show that if P = NP, then EXP = NEXP.

(1b) Show that $\mathrm{E} = \bigcup_{c>0} \mathrm{DTIME}[2^{cn}]$ is not closed under polynomial-time reductions, that is, there are problems A and B such that A is polynomial-time reducible to B, $B \in \mathrm{E}$, but $A \notin \mathrm{E}$. (Hint: Use the time hierarchy theorem.)

(1c) Use (1b) with suitable observations to show that $\mathrm{NP} \neq \mathrm{E}$.

(2) We know that SAT is NP-complete even when each clause is restricted to have no more than 3 literals. Several restrictions of SAT continue to be NP-complete, but some interesting variations turn out to be solvable by efficient algorithms (than are known for 3SAT).

(2a) [Reading assignment] Theorem 9.1 and its Corollary from Papadimitriou's book, where it is shown that 2SAT is in NL (the corollary uses the Immerman–Szelepcsényi Theorem, which we will prove during the week of Oct. 7).

(2b) [Reading assignment] Theorem 4.2 in Papadimitriou's book, where it is shown that HORNSAT is in P; HORNSAT denotes the satisfiability probloem when the clauses are restricted to be Horn clauses (disjunctions of literals where at most one literal is positive, equivalently, clauses of the form $(x_{i_1} \wedge x_{i_2} \wedge \ldots x_{i_k}) \longrightarrow x_{i_\ell}$).

(2c) Problem 9.5.4(a) in the book: Show that the special case of SAT in which each variable can only appear twice is in P.

(3) *Communication complexity* (Yao (1979); the book by Kushilevitz and Nisan (1997) contains an excellent coverage of the subject)

Consider two parties, commonly called Alice and Bob, where Alice has an element $x \in X$ and Bob has $y \in Y$, and they wish to evaluate a function $f : X \times Y \longrightarrow Z$. Alice and Bob communicate with each other by writing messages on a common blackboard. The objective is to study the minimum amount of information (in bits) that must be exchanged between the parties (by writing on the blackboard) to compute function f.

A *protocol* for a function f is a specification of a set of rules that govern the information exchange so that $f(x, y)$ may be computed from the contents of the blackboard, when Alice is given x and Bob is given y. A protocol must specify whose turn it is to write the next message, when the exchange is over, etc. All of this is formalized by the following definition.

Definitions. (adapted from Kushilevitz and Nisan (1997))

A *protocol* $\mathcal{P}$ over domain $X \times Y$ with range Z is a binary tree where each internal node v is labeled either by a function $a_v : X \longrightarrow \{0, 1\}$ or by a function $b_v : Y \longrightarrow \{0, 1\}$, and each leaf is labeled by an element $z \in Z$. (Intuitively, a node labeled by a function a_v indicates that it is Alice's turn to write the next message (one bit), and a node labeled by a function b_v indicates that it is Bob's turn to write the next message. Notice that these functions could be arbitrarily complex—the key point is that functions of the form a_v depend only on x, and functions of the form b_v depend only on y.)

The *value of the protocol* $\mathcal{P}$ *on input* (x, y) is the label of the leaf reached by starting from the root, and walking on the tree. At each internal node v labeled by a_v, the walk goes left if $a_v(x) = 0$ and right if $a_v(x) = 1$, and at each internal node v labeled by b_v, the walk goes left if $b_v(y) = 0$ and right if $b_v(y) = 1$. The *cost of the protocol* $\mathcal{P}$ *on input* (x, y) is the length of the path taken on the input (x, y). The *transcript of the protocol* $\mathcal{P}$ *on input* (x, y) is a binary string, of length equal to the cost of $\mathcal{P}$ on input (x, y), that describes the path taken on the input (x, y). The *cost of the protocol* $\mathcal{P}$ is the height of the tree, that is, the length of the longest path in the tree.

A protocol $\mathcal{P}$ over domain $X \times Y$ and range Z naturally defines a function from $X \times Y$ into Z, defined by its value for each $(x, y) \in X \times Y$; this is called the *function computed by the protocol* $\mathcal{P}$. For a function $f : X \times Y \longrightarrow Z$, the *(deterministic) communication complexity of* f is the minimum cost of $\mathcal{P}$, over all protocols $\mathcal{P}$ that compute f. **End of definitions**

Example. Alice and Bob hold subsets $x, y \subseteq \{1, \ldots, n\}$ respectively, and wish to compute $\textsc{max}(x, y)$, the largest number in $x \cup y$. The easy protocol where Alice first writes the largest element of x, then Bob writes the largest number in y, has cost $2 \log n$. Thus the communication complexity of $\textsc{max}$ is at most $2 \log n$.

Exercise. Setting similar to above example, but Alice and Bob wish to compute the median of the elements in the multiset $x \cup y$. Design a protocol of cost $O(\log^2 n)$ for this problem.

(3a) A *combinatorial rectangle* (often, simply a *rectangle*) in $X \times Y$ is a subset $R \subseteq X \times Y$ such that $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$. Prove that $R \subseteq X \times Y$ is a rectangle if and only if for every $x_1, x_2 \in X$ and $y_1, y_2 \in Y$, we have:

$$(x_1, y_1) \in R \text{ and } (x_2, y_2) \in R \implies (x_1, y_2) \in R \text{ and } (x_2, y_1) \in R,$$

that is, R is a rectangle if whenever it contains two points (x_1, y_1) and (x_2, y_2), it also contains the points (x_1, y_2) and (x_2, y_1) obtained by "cut-and-paste" from the first two points.

(3b) Let $\mathcal{P}$ be a protocol and v be a node of the protocol tree. Let R_v denote the set of inputs (x, y) that reach node v. Prove that for every v, R_v is a combinatorial rectangle. (Hint: There are two approaches to this, one following the definition of a rectangle and using induction on the depth of v, and the other using the alternate characterization of rectangles from Problem (3a).)

A particular consequence of (3b) is that the set R_ℓ of inputs that reach a leaf ℓ of the protocol tree is a rectangle. On the other hand, these (leaf-) rectangles form a partition of the input domain $X \times Y$. Thus if $\mathcal{P}$ computes f, then for every $(x_1, y_1), (x_2, y_2) \in X \times Y$, if (x_1, y_1) and (x_2, y_2) both reach the same leaf of $\mathcal{P}$, then $f(x_1, y_1) = f(x_2, y_2) = f(x_1, y_2) = f(x_2, y_1)$.

(3c) Show that any protocol that computes the equality function $\text{EQ} : \{0,1\}^n \times \{0,1\}^n \longrightarrow \{0,1\}$, defined by $\text{EQ}(x, y) = 1$ if and only if $x = y$, must have at least 2^n leaves; since EQ has a trivial protocol of cost $2n$, it follows that the communication complexity of EQ is $\Theta(n)$.

Finally, we turn to an application of communication complexity, specifically the result of Problem (3c), in proving a lower bound on the product of time and space required to solve the PALINDROMES problem on a multitape Turing machine.

(3d) Consider the set $\{x0^n y \mid x, y \in \{0,1\}^*, |x| = |y| = n, y = x^R\}$, a subset of PALINDROMES, where x^R denotes the bitwise reversal of x. Show that if there is a multitape TM that solves PALINDROMES in time $T(n)$ and using space $S(n)$, then there is a communication protocol for the EQ problem of cost at most $O(TS/n)$. (Hint: Assume such a TM exists, and design a protocol for EQ, letting Alice and Bob alternate in simulating it on $x0^n y^R$, depending on which part of the input the TM is reading at any time. Use the blackboard to exchange sufficient information whenever the simulation is to be suspended by one party and resumed by the other.)

Conclude, using (3c), that PALINDROMES $\notin \mathrm{DTISP}[o(n^2)]$, where $\mathrm{DTISP}[f(n)]$ denotes the class of all decision problems that can be solved by a Turing machine such that the product of the time taken and the space used is at most $O(f(n))$.

Contrast this result with the trivial RAM algorithm for PALINDROMES that uses space $O(\log n)$ and runs in time $O(n)$, so $TS = O(n \log n)$. This is an instance where we have a "nontrivial" result for the multitape TM model that simply isn't true for the RAM model. Indeed, a DTISP lower bound of $\Omega(n^2)$ on the RAM model for any natural decision problem would be a breakthrough result in complexity theory. The best known lower bound for the RAM model is a DTISP lower bound of $n^{\sqrt{2}}$ for SAT, and this was proved only a couple of years ago!