Write every problem on a separate sheet of paper, write *legibly*, and write your name, your student ID, this Homework # and the problem # on top of each page.

The description of your proofs should be as *clear* as possible (which does not mean *long* – in fact, typically, good clear explanations are also short.)

Make sure you are familiar with the collaboration policy, and read the overview in the class homepage theory.cs.stanford.edu/∼trevisan/cs154.

Recall that our late policy is that late homework is not accepted (however the lowest homework grade is dropped).

Refer to the class home page for instructions on how to submit the homework.

Homework 5

Due on Thursday, February 23, 2012, 9:30am

1. Let

$$R := \{x : K(x) \geq |x|\}$$

be the language of binary strings whose Kolmogorov complexity is at least their length. Prove that for every decidable subset $L \subseteq R$ it must be the case that L is finite.

2. (Sipser problem 6.20) Show how to compute the Kolmogorov complexity $K_U(x)$ of a string x with an oracle for A_{TM}.

The definition of an oracle is in Sipser 6.18. An oracle is essentially a subroutine. You could interpret this problem as asking for an algorithm that on input x, computes the descriptive complexity of x, that is, $K_U(x)$, using a subroutine for A_{TM}. On input $\langle M, w\rangle$ the subroutine will return 1 if M accepts w, and 0 otherwise. Whenever you invoke the subroutine on some input $\langle M, w\rangle$, use the terminology "query the A_{TM} oracle on input $\langle M, w\rangle$".

For instance, the machine S in the proof that $HALT_{\mathsf{TM}}$ is undecidable (Sipser 5.1) is an example of a algorithm for $HALT_{\mathsf{TM}}$ using an oracle for A_{TM}. In that example, S only uses the A_{TM} subroutine once. In general (and for this problem), you are allowed to invoke the subroutine any number of times, and the oracle queries may be adaptive (that is, the next query may depend on the answers to the previous ones).

3. (Sipser problem 6.2) We say that a Turing machine M is *minimal* if for all M' such that L(M') = L(M), the number of states of M' is at least the number of states of M. Let $MIN_{TM} = \{M \mid \text{M is a minimal Turing machine}\}$ (Sipser 6.6). Sipser 6.7 shows that MIN_{TM} is not recognizable. Show that for every infinite $L \subseteq MIN_{TM}$, the language L is also not recognizable.
   Hint: Come up with a computable function assuming that L is recognizable, then apply the fixed-point theorem.