Problem Set 4

This problem set is due on **Monday, October 5, by 10am.**

Use the CS172 drop box.

Write **your name and your student ID number** on your solution. Write legibly. 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.) Be sure to be familiar with the collaboration policy, and read the overview in the class homepage `www.cs.berkeley.edu/~luca/cs172`.

There will be no class on Thursday, September 24. We will have a make-up lecture on Friday, October 2. We'll decide the time of the October 2 lecture next Tuesday, looking for a time that minimizes conflicts. `http://ucfacultywalkout.com/`

In problem 1, describe the Turing machine in some detail, but not to the point of actually specifying the states and drawing a diagram. The level of detail used in the book in Example 3.11 and Example 3.12 is ideal. You are free to use a multi-tape machine or any variant described in Chapter 4 if it makes your solution easier. In problems 2 and 3, describe the algorithms as pseudocode, as you would do in a CS170 assignment; the level of detail in the proof of Theorem 4.2 in the book is a good example.

1. [20 /100] Describe informally a Turing machine that decides the language

$$L := \{p : p \text{ is a prime number } \}$$

2. [20 /100] Show that the Turing-recognizable languages are closed under the star operation. That is, prove that, if L is a recognizable language, then L^* is recognizable.

3. [25/100] Let E be an enumerator that enumerates a language L in lexicographic order.

   Show that L is decidable.

4. [35/100] A *read-only* Turing machine is a Turing machine that is not allowed to write on the tape. The transition function of a read-only Turing machine is of the form

$$\delta : Q \times \Sigma \cup \{\square\} \to Q \times \{L, R\}$$

   and the computation of a read-only Turing machine is defined in the expected way.

   Prove that if M is a read-only Turing machine then the language $L(M)$ recognized by M is regular.

   [Hint: a possible approach is to show that if k is the number of states of M then the number of equivalence classes in $\approx_{L(M)}$ is at most a constant that depends only on k. Be careful that there are k-state read-once Turing machines that recognize languages with $2^{\Omega(k)}$ equivalence classes (can you prove it?), so if your approach leads, for example, to a $O(k^2)$ state upper bound then something is wrong.]