MCS 441 – Theory of Computation I

Spring 2013

Problem Set 7

Lev Reyzin

Due: 3/13/13 at the beginning of class

Related reading: Chapters 3-4.
Instructions: Atop your problem set, write your name, clearly list your collaborators[1] (see syllabus for the collaboration policy), and indicate whether you are an undergraduate or graduate student.
Important note: Problems labeled "U" and "G" are assigned to undergraduate and graduate students, respectively. Undergraduate students can get a small bonus for solving the graduate problems. Graduate students are encouraged to solve the undergraduate problems for practice.

1. [8 pts] A simple observation from number theory is that a decimal number is divisible by 3 if and only if the sum of its digits is divisible by 3.

a. [2 pts] Use this observation to devise an algorithm for testing divisibility by 3 for arbitrarily large inputs. Namely, give a *high level description* of an algorithm that decides

$$L_1 = \{w \mid w \in \{0, 1, \ldots, 9\}^* \text{ and } w \text{ is divisible by } 3\}.$$

b. [6 pts] Give an *implementation description* (not a formal description) of a Turing machine that tests divisibility by 3 using your algorithm from part a.

2. [10 pts] Read the article "*Who Can Name the Bigger Number?*" by Scott Aaronson (now a professor at MIT): `http://homepages.math.uic.edu/~lreyzin/s13_mcs441/Aaronson99.pdf`. This article may include concepts we have not yet covered, but the introduction to them should be sufficiently self-contained.

a. [5 pts] Going back to 10,000 B.C., name some advances in mathematics over time that have effectively allowed people to express bigger numbers than before the advent of those advances. You may do outside research to answer this question.[2] Include at least one advance from Aaronson's article.

b. [5 pts] Would Aaronson say Turing machines are important even if they didn't have the immense real-world impact that they currently have? Why or why not?

[1] If you did not have any collaborators, please say so.
[2] A few examples will suffice; do not go overboard and write a research paper.

3. [12 pts] The goal of this problem is to find an uncomputable (or "incalculable") function. This function, $S(n)$, will grow so fast that even Turing machines cannot keep up with it. Let

$$\varepsilon\text{-HALT}_{\text{TM}} = \{\langle M \rangle \mid M \text{ is a TM and } M \text{ halts on the input } \varepsilon \text{ (i.e. no input)}\}.$$

You can assume the input alphabet to be $\Sigma = \{0, 1\}$, which is sufficient to encode M.

a. [2 pts] Is ε-HALT_{TM} recognizable? Why or why not?

b. [5 pts] Prove that ε-HALT_{TM} is not decidable.
Hint: you can use the fact that HALT_{TM} is not decidable and use that to show that ε-HALT_{TM} being decidable would lead to a contradiction.

c. [5 pts] Let $\mathcal{H}_n$ be the set of n-state TMs that eventually halt when run on the empty input. Let $S(n)$ be the maximum number steps a TM in $\mathcal{H}_n$ can take.

U. Let

$$L_{3.c.U} = \{m \mid m \in \{0,1\}^* \text{ s.t. } m = S(n) \text{ for some } n \geq 1\}.^3$$

Prove that $L_{3.c.U}$ is not decidable.

G. Let $S_{-1}(n)$ be the maximum number of steps that a TM in $\mathcal{H}_n$ can take *that is not equal to* $S(n)$. In essence, we are considering the second longest running machine that halts. Let

$$L_{3.c.G} = \{m \mid m \in \{0,1\}^* \text{ s.t. } m = S_{-1}(n) \text{ for some } n \geq 1\}.^3$$

Prove that $L_{3.c.G}$ is not decidable.

4. [6 pts] Consider the language L_4, which contains only one string, defined as follows:

$$L_4 = \begin{cases} \{1\} & \text{if } L_{3.c.U} \text{ contains infinitely many primes} \\ \{0\} & \text{otherwise.} \end{cases}$$

Is L_4 decidable? Why or why not?
Hint: Think carefully about the definition of decidable languages.

[3] Interpret m as a binary number.