# CS172 Computability & Complexity (Spring'09)

Instructor: Mihai Pătraşcu        GSI: Omid Etesami

Problem Set 7    *Due: Thursday, April 2*

**1.** Implement a program with the following functionality: it reads a string from standard input, and prints the first position where the string appears in *its own source code*, or "NO" if it never appears. For instance, if your code is in C, and I type `main()`, it should print the position of `main()` in your source code.

Email the source code (in C, Java, or Pyhton) to `mip@alum.mit.edu` as an attachment to an email with the subject `CS172-HW7-P1`. Grading will be done automatically so please follow I/O and email specifications carefully.

*Note 1:* If you submit a file called `abc.c`, you cannot simply open the file `abc.c` and read the source code. I will run your code with the source hidden in a different directory. (For Unix hackers: I use `chroot` for effective hiding.)

*Note 2:* To solve the searching part, you should reuse your KMP implementation from two problem sets ago. If you managed to lose your own KMP code, I can send it to you.

**2.** Rightist Turing Machines are like regular Turing Machines, but do not have the ability to move their head left (they may just read, write, and move towards the right). What is the set of languages decided by Rightist Turing Machines? Prove your answer.

**3.** Which of the following languages are decidable? Prove your answers.

**(a)** $L = \Big\{(M, A) \mid \text{the machine } M \text{ sorts the integer array } A\Big\}$.

**(b)** $L = \Big\{(M, A) \mid$ the machine $M$ sorts the integer array $A$ in time at most $10 \cdot n^3$, where $n$ is the size of $A\Big\}$.

**(c)** $L = \Big\{M \mid$ the Turing Machine $M$ sorts *any* integer array $A$ in time at most $10 \cdot n^3$, where $n$ is the size of $A\Big\}$.

We say a Turing Machine sorts its input if it doesn't loop infinitely, at the end of the computation, the tape contains the input array in sorted order.

**4.** Describe a language $L$ such that $L \subseteq 1^\star$ and $L$ is undecidable.

**5.** Define the "Busy Beaver" function $BB(n)$ = the most number of 1's that a Turing Machine with $n$ states could write on the tape before halting.

[Here is a more formal definition. Consider all Turing machines with $n$ states, running over the binary alphabet $\Sigma = \{0, 1\}$. Imagine running all these machines with an empty initial tape, and ignore the ones that loop infinitely. Among all machines that halt, define the champion beaver as the one who leaves the largest number of 1's on its tape at the end of the computation. Let $BB(n)$ be the number of 1's left on the tape after the champion beaver finishes.]

Prove that $BB(n)$ is an uncomputable functions: there exists no Turing Machine that accepts $n$ as input, and writes $BB(n)$ as its output.

*Hint:* Use a direct proof by contradiction. You will effectively show that $BB(n)$ is larger than any computable function.

**6.** Assuming $BB(n)$ is uncomputable, show that $HALT$ is undecidable by reduction. Remember that we defined $HALT = \{(M, x) \mid \text{the machine } M \text{ halts when run on input } x\}$. (This is the 3rd proof you see for the undecidability of the halting problem.)