EECS 70 Discrete Mathematics and Probability Theory
Fall 2015 Satish Rao **Discussion 8b**

1. **Countable Set**

   A set S is countable if there is a bijection between S and $\mathbb{N}$ (set of natural numbers) or some subset of $\mathbb{N}$. If $T = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$, is T countable? If yes, can you find two different bijections between T and $\mathbb{N}$.

2. **Count it!**

   For each of the following collections, determine and briefly explain whether it is finite, countably infinite (like the natural numbers), or uncountably infinite (like the reals):

   (a) The integers which divide 8.

   (b) The integers which 8 divides.

   (c) The functions from $\mathbb{N}$ to $\mathbb{N}$.

3. **Computability**

   We say that a computer program P computes the function $f : S \to \{0,1\}$ if, for every $x \in S$, when you run the program P on the input x, the program eventually finishes and returns the value $f(x)$. We say that a function f is *computable* if there exists some computer program that can compute f. (Programs always must be of finite length.)

   Each of the following parts defines a function f. For each part, say whether f is computable or uncomputable.

   1. $f : \mathbb{N} \to \{0,1\}$, defined by
   $$f(a) = \begin{cases} 1 & \text{if } a \text{ is prime,} \\ 0 & \text{otherwise.} \end{cases}$$

   2. $f : \mathbb{N} \to \{0,1\}$, defined by
   $$f(a) = \begin{cases} 1 & \text{if there exist prime numbers } b, c \text{ such that } a = b + c, \\ 0 & \text{otherwise.} \end{cases}$$

   3. $f : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}$, defined by
   $$f(P, I) = \begin{cases} 1 & \text{if program } P \text{ ever divides a number by zero at any point when run on input } I, \\ 0 & \text{otherwise.} \end{cases}$$

   This is a hard question, so we will guide you through the proof:

   Suppose `DivideByZeroChecker` be the program that computes f. I.e. given program `P` and input `I` (represented as bit strings), `DivideByZeroChecker(P, I)` returns "1" if `P` ever divides by 0 when run on input `I`, and returns "0" otherwise.

   Construct another program `Contrarian` as follows:

```
Contrarian(P):
```
1. If `DivByZeroChecker(P,P)` = "yes", then return immediately.
2. Otherwise, divide 1 by 0, and then return.

What happens when `Contrarian` is passed its own source code as its input, i.e., when we run `Contrarian(Contrarian)`. Does it divide by zero?

4. Compute this

(a) Can you write a program that gets n (a natural number) as input and finds the shortest formula that computes n? A formula is a valid sequence consisting of decimal digits, the operators $+$, $\times$,ˆ(raising to the power), and parentheses. The length of a formula is simply the number of characters you need to use to type it (i.e. each operator, decimal digit, or paranthesis counts as one character).

(b) (Optional) Now assume that you want to write a computer program that given the input n (a natural number) finds another computer program (in a specific language, e.g. C or Python) that prints n. The program that is found has to have the minimum length plus execution time amongst all programs that print n, where length is measured by the number of characters in the source code and execution time is measured by a concrete number such as the number of CPU instructions executed. Can this be done?

(c) Consider the set of programs (or functions) that take a single natural number n as input and output a natural number in at most $10^6 + 2^n$ steps (i.e. they always terminate after $10^6 + 2^n$ steps). Let this set be L. A member of L is called **thorough** if every natural number m can be produced as its output (by an appropriate input). Can you write a program that takes a member of L as input and determines whether that member is thorough? The given member of L is guaranteed to be in L, there is no need for your program to verify the membership.

(HINT: If you had such a program, could you somehow use it to solve the halting problem? If so, what would that mean?)