CS 70 Fall 2022 — Discrete Mathematics and Probability Theory — Satish Rao and Babak Ayazifar — HW 08

Due: Saturday, 10/22, 4:00 PM
Grace period until Saturday, 10/22, 6:00 PM

Sundry

Before you start writing your final homework submission, state briefly how you worked on it. Who else did you work with? List names and email addresses. (In case of homework party, you can just describe the group.)

1 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 $\mathbb{N}$.

(d) The set of strings over the English alphabet. (Note that the strings may be arbitrarily long, but each string has finite length. Also the strings need not be real English words.)

(e) The set of finite-length strings drawn from a countably infinite alphabet, $\mathscr{C}$.

(f) The set of infinite-length strings over the English alphabet.

2 Countability Proof Practice

(a) A disk is a 2D region of the form $\{(x,y) \in \mathbb{R}^2 : (x-x_0)^2 + (y-y_0)^2 \leq r^2\}$, for some $x_0, y_0, r \in \mathbb{R}$, $r > 0$. Say you have a set of disks in $\mathbb{R}^2$ such that none of the disks overlap. Is this set always countable, or potentially uncountable?
(*Hint*: Attempt to relate it to a set that we know is countable, such as $\mathbb{Q} \times \mathbb{Q}$.)

(b) A circle is a subset of the plane of the form $\{(x,y) \in \mathbb{R}^2 : (x-x_0)^2 + (y-y_0)^2 = r^2\}$ for some $x_0, y_0, r \in \mathbb{R}$, $r > 0$. Now say you have a set of circles in $\mathbb{R}^2$ such that none of the circles overlap. Is this set always countable, or potentially uncountable?
(*Hint*: The difference between a circle and a disk is that a disk contains all of the points in its interior, whereas a circle does not.)

(c) Is the set containing all increasing functions $f : \mathbb{N} \to \mathbb{N}$ (i.e., if $x \geq y$, then $f(x) \geq f(y)$) countable or uncountable? Prove your answer.

(d) Is the set containing all decreasing functions $f : \mathbb{N} \to \mathbb{N}$ (i.e., if $x \geq y$, then $f(x) \leq f(y)$) countable or uncountable? Prove your answer.

3 Unprogrammable Programs

Prove whether the programs described below can exist or not.

(a) A program $P(F,x,y)$ that returns true if the program F outputs y when given x as input (i.e. $F(x) = y$) and false otherwise.

(b) A program P that takes two programs F and G as arguments, and returns true if F and G halt on the same set of inputs (or false otherwise).

4 The Complexity Hierarchy

The complexity hierarchy is a monument to our collective understanding of computation and its limitations. In fact, you may already be familiar with the classes P and NP from CS61B. In this problem, we will focus on decision problems like the Halting Problem, where the output is "Yes" (True) or "No" (False), and explore the classes RE, coRE, and R.

(a) A problem is recursively enumerable (RE) if there exists a program P that can print out all the inputs for which the answer is "Yes", and no inputs for which the answer is "No". The program P can print out a given input multiple times, so long as every input gets printed eventually. The program P can run forever, so long as every input which should be printed is at a finite index in the printed output.

Prove that the Halting Problem belongs in RE. Namely, prove that it is possible to write a program P which:

- runs forever over all possible programs M and inputs x, and prints out strings to the console,
- for every (M,x), if $M(x)$ halts, then P eventually prints out (M,x),
- for every (M,x), if $M(x)$ does NOT halt, then P never prints out (M,x).

In this context, P is called an *enumerator*. (Hint: Consider the tail of a dove.)

(b) An equivalent definition of RE is as follows: A problem belongs in RE if there exists a program P' that will output "Yes" when given an input x for which the answer is "Yes". If the answer is "No", then $P'(x)$ may output "No" or loop forever. As an optional exercise, you should be able to convince yourself that this is indeed an equivalent definition.

Prove that the Halting Problem belongs in RE using this equivalent definition. Namely, prove that it is possible to write a program P' which:

- takes as input a program M and input x.
- if M halts on input x, then P' should print "Yes".
- if M does not halt on input x, then P' may output "No" or loop forever.

In this context, P' is called a *recognizer*.

(c) As you might suspect, a problem is co-recursively enumerable (coRE) if its complement is in RE. The complement of a decision problem A is another problem A' where $A'(x)$ is "Yes" iff $A(x)$ is "No", and $A'(x)$ is "No" iff $A(x)$ is "Yes". State the complement of the Halting Problem.

(d) Finally, a problem belongs in the class R if it is computable, meaning there exists a program P that answers "Yes" when the answer is "Yes", and answers "No" when the answer is "No". By definition then, the problem is a computable function if it is computable.

We know that the TestHalt is not computable, and that the Halting Problem belongs in RE. Prove by contradiction that the Halting Problem cannot belong in coRE.

5 Probability Warm-Up

(a) Suppose that we have a bucket of 30 green balls and 70 orange balls. If we pick 15 balls uniformly out of the bucket, what is the probability of getting exactly k green balls (assuming $0 \leq k \leq 15$) if the sampling is done **with** replacement, i.e. after we take a ball out the bucket we return the ball back to the bucket for the next round?

(b) Same as part (a), but the sampling is **without** replacement, i.e. after we take a ball out the bucket we **do not** return the ball back to the bucket.

(c) If we roll a regular, 6-sided die 5 times. What is the probability that at least one value is observed more than once?

6 Past Probabilified

In this question we review some of the past CS70 topics, and look at them probabilistically.
For the following experiments,

i. Define an appropriate sample space Ω.

ii. Give the probability function $\mathbb{P}[\omega]$.

iii. Compute $\mathbb{P}[E_1]$.

iv. Compute $\mathbb{P}[E_2]$.

(a) Fix a prime $q > 2$, and uniformly sample twice with replacement from $\{0, \ldots, q-1\}$ (assume we have two $\{0, \ldots, q-1\}$-sided fair dice and we roll them). Then multiply these two numbers

with each other in $\pmod q$ space.
$E_1 =$ The resulting product is 0.
$E_2 =$ The product is $(q-1)/2$.

(b) Make a graph on v vertices by sampling uniformly at random from all possible edges, (assume for each edge we flip a coin and if it is head we include the edge in the graph and otherwise we exclude that edge from the graph).
$E_1 =$ The graph is complete.
$E_2 =$ vertex v_1 has degree d.

(c) Create a random stable matching instance by having each person's preference list be a random permutation of the opposite entity's list (make the preference list for each individual job and each individual candidate a random permutation of the opposite entity's list). Finally, create a uniformly random pairing by matching jobs and candidates up uniformly at random (note that in this pairing, (1) a candidate cannot be matched with two different jobs, and a job cannot be matched with two different candidates (2) the pairing does not have to be stable).
$E_1 =$ All jobs have distinct favorite candidates.
$E_2 =$ The resulting pairing is the candidate-optimal stable pairing.