CS 70 Discrete Mathematics and Probability Theory
Fall 2026 Hug, Sabin

HW 07

Due: Wednesday, 10/21, 6:00 PM
Grace period until Friday, 10/23, 11:59 PM
Remember to show your work for all problems!

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 Countability Basics

Note 12

(a) Prove that $f : \mathbb{N} \to \mathbb{N}$, defined by $f(n) = n^2$, is an injection (one-to-one). Recall that f is injective if and only if for all $m, n \in \mathbb{N}$, $m \neq n$ implies $f(m) \neq f(n)$.

(b) Prove that $f : \mathbb{R} \to \mathbb{R}$, defined by $f(x) = x^3 + 1$, is a surjection (onto). Recall that f is surjective if and only if for every $y \in \mathbb{R}$, there exists an $x \in \mathbb{R}$ such that $f(x) = y$.

(c) In class, we showed that $(0,1)$ is uncountable. In this subpart, we will prove that $\mathbb{R}_+ = (0,\infty)$ is uncountable.

(i) Define an injection $f : (0,1) \to \mathbb{R}_+$.

(ii) Define an injection $g : \mathbb{R}_+ \to (0,1)$.

(iii) Explain why the Cantor–Bernstein theorem implies that $\mathbb{R}_+$ is uncountable.

2 Computability Basics

Note 13

Consider the problem of determining if a function F returns `TRUE`. That is, given a function F that takes inputs from some (possibly infinite) set $\mathscr{X}$, we want to know if there is any input $x \in \mathscr{X}$ such that $F(x)$ returns `TRUE`. We will prove that this problem is uncomputable.

(a) We will use proof by contradiction. Suppose there exists a program `ReturnsTrue` that determines whether a function returns `TRUE` on some input $x \in \mathscr{X}$. What are the inputs and outputs of `ReturnsTrue`?

(b) We will reach a contradiction by solving the Halting Problem. What hypothetical program would solve the Halting Problem? What are its inputs and outputs?

(c) Fix a program P and an input x, and consider the following program:

```
def F(y):
    P(x)
    return TRUE
```

Will $F(y)$ return `TRUE` if $P(x)$ halts? Will it return `TRUE` if $P(x)$ loops forever?

(d) Use `ReturnsTrue` to solve the Halting Problem, and conclude that `ReturnsTrue` cannot exist.

3 Countability and the Halting Problem

Note 12
Note 13

Using methods from countability, we will prove the Halting Problem is uncomputable.

(a) What is a reasonable representation for a computer program? Using this definition, show that the set of all programs is countable. *(Hint: Machine Code)*

(b) The Halting Problem only considers programs which take a finite length input. Show that the set of all finite-length inputs is countable.

(c) Assume that you have a program that tells you whether or not a given program halts on a specific input. Since the set of all programs and the set of all inputs are countable, we can enumerate them and construct the following table.

	x_1	x_2	x_3	x_4	$\cdots$
p_1	H	L	H	L	$\cdots$
p_2	L	L	L	H	$\cdots$
p_3	H	L	H	L	$\cdots$
p_4	L	H	L	L	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

An H (resp. L) in the ith row and jth column means that program p_i halts (resp. loops) on input x_j. Now write a program that is not within the set of programs in the table above.

(d) Find a contradiction between parts (a) and (c) to show that the Halting Problem can't be solved.

4 Count It!

Note 12

For each of the following sets, 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, $\mathscr{A}$.

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

5 Unprogrammable Programs

Note 13

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 for all inputs x, F halts on x iff G halts on x (and returns false if this equivalence is not always true).

Hint: Use P to solve the halting problem, and consider defining two subroutines to pass in to P, where one of the subroutines always halts.