CS 70 Spring 2024	Discrete Mathematics and Probability Theory Seshia, Sinclair	HW 06

Due: Saturday, 3/1, 4:00 PM
Grace period until Saturday, 3/1, 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 Unions and Intersections

Note 11

Given:

- X is a countable, non-empty set. For all $i \in X$, A_i is an uncountable set.
- Y is an uncountable set. For all $i \in Y$, B_i is a countable set.

For each of the following, decide if the expression is "Always countable", "Always uncountable", "Sometimes countable, Sometimes uncountable".

For the "Always" cases, prove your claim. For the "Sometimes" case, provide two examples – one where the expression is countable, and one where the expression is uncountable.

(a) $X \cap Y$ Always countable $X \cap Y \subseteq X$ → A countable set's subset is always countable

(b) $X \cup Y$ Always uncountable → $X \cup Y$ is a superset of Y

?

(c) $\bigcup_{i \in X} A_i$ always uncountable

(d) $\bigcap_{i \in X} A_i$ sometimes

(e) $\bigcup_{i \in Y} B_i$ ~~always uncountable~~ → sometimes → countable: Make all the B_i identical

(f) $\bigcap_{i \in Y} B_i$ ~~sometimes~~ always countable → is a subset of B_i →

2 Count It!

Note 11

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):

$g(i) = f_i(i) + 1$

(a) The integers which divide 8. finite ± 1 ± 2 ± 4 ± 8

(b) The integers which 8 divides. countably infinite. use cantor's diagonalization

(c) The functions from $\mathbb{N}$ to $\mathbb{N}$. ? uncountably infinite → proof

(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.) countably infinite.

(e) The set of finite-length strings drawn from a countably infinite alphabet, $\mathscr{A}$. ~~uncountably~~ countably infinite. set $\{0,1,2\}^*$ is countable

(f) The set of infinite-length strings over the English alphabet. uncountably ∃ bijection to R[0,1]

3 Fixed Points

Note 12 Consider the problem of determining if a program P has any fixed points. Given any program P, a fixed point is an input x such that $P(x)$ outputs x.

(a) Prove that the problem of determining whether a program has a fixed point is uncomputable.

(b) Consider the problem of outputting a fixed point of a program if it has one, and outputting "Null" otherwise. Prove that this problem is uncomputable.

(c) Consider the problem of outputting a fixed point of a program F if the fixed point exists *and* is a natural number, and outputting "Null" otherwise. If an input is a natural number, then it has no leading zero before its most significant bit.

Show that if this problem can be solved, then the problem in part (b) can be solved. What does this say about the computability of this problem? (You may assume that the set of all possible inputs to a program is countable, as is the case on your computer.)

uncomputable

4 Unprogrammable Programs

Note 12 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).

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

(a) testhalt (P, x) :

P'(x):
  if P(x) halt : return x

    else : return other value except x

if testfixedpoint (P', x) : return true → simplify

  else return false. return FixedPoint(F_Pime)

(b) getfixedpoint (P,)

testfixedpoint (P,)

  if getfixedpoint (P,) != Null
      return true

  else:
      return false.

(c) say this problem can be solved by a function A (P,)

getfixedpoint (P) : f(x) ⟷ nature number

  P'(x) : ~~← nature number~~ → if x ∉ N error
      ↓ actual input
    return P(f(x))

 if A(P') is null : return null

   else return f^{-1}(A(P'))

4(a) P(F, x, y) → F(x) = y →

Testhalt(P1, x)

F(x):

run P1(x)

return y → 0

→ 转换为停机问题

simplify

return P(F, x, 0)

if P(F, x, y) is true return true

else return false

(b) unprogramable

P(F, G) ∈ ?

Tesehalt(P1, x)

F(y):

if y == x

P(x)

else: loop

G(y)

always loop

if P(F, G): return ~~true~~

else: return false

return not P(Q, R)

$\frac{104!}{102! \times 2!} * \frac{102!}{100! \times 2!} \cdots * \frac{2!}{0! 2!}$

5 Counting, Counting, and More Counting

Note 10

The only way to learn counting is to practice, practice, practice, so here is your chance to do so. Although there are many subparts, each subpart is fairly short, so this problem should not take any longer than a normal CS70 homework problem. You do not need to show work, and **Leave your answers as an expression** (rather than trying to evaluate it to get a specific number).

(a) How many 19-digit ternary (0,1,2) bitstrings are there such that no two adjacent digits are equal? $3 * 2 * 2 * \cdots * 2 \to 3 * 2^{18}$

(b) Two identical decks of 52 cards are mixed together, yielding a stack of 104 cards. How many different ways are there to order this stack of 104 cards? $\frac{104!}{2^{52}}$

(c) An anagram of ALABAMA is any re-ordering of the letters of ALABAMA, i.e., any string made up of the letters A, L, A, B, A, M, and A, in any order. The anagram does not have to be an English word.

redundant ∪ $\frac{7!}{4!}$ $C^3_6 * 3 * 2 * 1$ $6 \times 5 \times 4$

i. How many different anagrams of ALABAMA are there?

ii. How many different anagrams of MONTANA are there? N: 2, A: 2, M: 1, O: 1, T: 1 $7 \times 6 \times 5 \times C^2_4$ $\frac{7!}{2! * 2!}$

(d) How many different anagrams of ABCDEF are there if: $\frac{6!}{2}$

i. C is the left neighbor of E $5 * 4! \to 5!$

ii. C is on the left of E (and not necessarily E's neighbor) $(5+4+3+2+1) * 4! \to 15 * 4!$

(e) We have 8 balls, numbered 1 through 8, and 25 bins. How many different ways are there to distribute these 8 balls among the 25 bins? Assume the bins are distinguishable (e.g., numbered 1 through 25). $24 + 8 = 32$ $\binom{32}{8}$ ✗ balls are different 25^8

(f) There are exactly 20 students currently enrolled in a class. How many different ways are there to pair up the 20 students, so that each student is paired with one other student? Solve this in at least 2 different ways. **Your final answer must consist of two different expressions.**

19 (f) $\frac{C^2_{20} * C^2_{18} * \cdots C^2_2}{10!}$

$\to 19 \times 17 \times 15 \times \cdots \times 1$