CS 70 Discrete Mathematics and Probability Theory Spring 2017 Rao HW 7

1 Sundry

Before you start your homework, write down your team. Who else did you work with on this homework? List names and email addresses. (In case of homework party, you can also just describe the group.) How did you work on this homework? Working in groups of 3-5 will earn credit for your "Sundry" grade.

Please copy the following statement and sign next to it:

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up.

I certify that all solutions are entirely in my words and that I have not looked at another student's solutions. I have credited all external sources in this write up. (Signature here)

2 More Countability

Given:

- A is a countable set, non-empty set. For all $i \in A$, S_i is an uncountable set.
- B is an uncountable set. For all $i \in B$, Q_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) $\bigcup_{i\in A} S_i$
- (b) $\bigcap_{i \in A} S_i$
- (c) $\bigcup_{i\in B} Q_i$
- (d) $\bigcap_{i \in B} Q_i$
- (e) $A \cap B$

Solution:

- (a) Always uncountable. Let a be any element of A. S_a is uncountable. Thus, $\bigcup_{i \in A} S_i$, a superset of S_a , is uncountable.
- (b) Sometimes countable, sometimes uncountable.

Countable: When the S_i are disjoint, the intersection is empty, and thus countable. Formally, let $A = \mathbb{N}$, let $S_i = \{i\} \times \mathbb{R} = \{(i,x) \mid x \in \mathbb{R}\}$. Then, $\bigcap_{i \in A} S_i = \emptyset$.

Uncountable: When the S_i are identical, the intersection is uncountable. Let $A = \mathbb{N}$, let $S_i = \mathbb{R}$ for all i. $\bigcap_{i \in A} S_i = \mathbb{R}$ is uncountable.

(c) Sometimes countable, sometimes uncountable.

Countable: Make all the Q_i identical. Formally, let $B = \mathbb{R}$, and $Q_i = \mathbb{N}$. Then, $\bigcup_{i \in B} Q_i = \mathbb{N}$ is countable.

Uncountable: Let $B = \mathbb{R}$. Let $Q_i = \{i\}$. Then, $\bigcup_{i \in B} Q_i = \mathbb{R}$ is uncountable.

- (d) Always countable. Let b be any element of B. Q_b is countable. Thus, $\bigcap_{i \in B} Q_i$, a subset of Q_b , is also countable.
- (e) Always countable. $A \cap B \subseteq A$, which is countable.
- 3 Counting Cartesian Products

For two sets *A* and *B*, define the Cartesian Product as $A \times B = \{(a,b) : a \in A, b \in B\}$.

- (a) Given two countable sets A and B, prove that $A \times B$ is countable.
- (b) Given a finite number of countable sets A_1, A_2, \dots, A_n , prove that $A_1 \times A_2 \times \dots \times A_n$ is countable.
- (c) Consider an infinite number of countable sets: B_1, B_2, \ldots Under what condition(s) is $B_1 \times B_2 \times \cdots$ countable? Prove that if this condition is violated, $B_1 \times B_2 \times \cdots$ is uncountable.

Solution:

- (a) As shown in lecture, $\mathbb{N} \times \mathbb{N}$ is countable by creating a zigzag map that enumerates through the pairs: $(0,0), (1,0), (0,1), (2,0), (1,1), \ldots$ Since A and B are both countable, there exists a bijection between each set and a subset of \mathbb{N} . Thus we know that $A \times B$ is countable because there is a bijection between a subset of $\mathbb{N} \times \mathbb{N}$ and $A \times B : f(i,j) = (A_i, B_j)$. We can enumerate the pairs (a,b) similarly.
- (b) Proceed by induction.

Base Case: n = 2. We showed in part (a) that $A_1 \times A_2$ is countable since both A_1 and A_2 are countable.

Induction Hypothesis: Assume that for some $n \in \mathbb{N}$, $A_1 \times A_2 \times \cdots \times A_n$ is countable. Induction Step: Consider $A_1 \times \cdots \times A_n \times A_{n+1}$. We know from our hypothesis that $A_1 \times \cdots \times A_n$ is countable, call it $C = A_1 \times \cdots \times A_n$. We proved in part (a) that since C is countable and A_{n+1} are countable, $C \times A_{n+1}$ is countable, which proves our claim. (c) Trivially, if any of the B_i are empty, then the infinite Cartesian Product is also empty. Hence, we assume that none of the B_i are empty.

A necessary and sufficient condition for the infinite Cartesian Product to be countable is for all but finitely many of the B_i to have exactly one element.

Necessary: Suppose not; then, infinitely many B_i have at least two elements. Notice that for any B_i with only one element, every element of the infinite Cartesian Product must necessarily use the single element of B_i . Therefore, we can ignore each B_i with only one element and assume that every B_i has at least two elements. Proceed with a diagonalization argument by assuming for the sake of contradiction that $B_1 \times B_2 \times \cdots$ is countable and its elements can be enumerated in a list:

```
(b_{1,1},b_{2,1},b_{3,1},b_{4,1},\dots)
(b_{1,2},b_{2,2},b_{3,2},b_{4,2},\dots)
(b_{1,3},b_{2,3},b_{3,3},b_{4,3},\dots)
(b_{1,4},b_{2,4},b_{3,4},b_{4,4},\dots)
\vdots
```

where $b_{i,j}$ represents the item from set B_i that is included in the jth element of the Cartesian Product. Now consider the element $(\overline{b_{1,1}}, \overline{b_{2,2}}, \overline{b_{3,3}}, \overline{b_{4,4}}, \dots)$, where $\overline{b_{i,j}}$ represents any item from set B_i that differs from $b_{i,j}$ (i.e. any other element in the set). This is a valid element that should exist in the Cartesian Product B_1, B_2, \dots , yet it is not in the enumerated list. This is a contradiction, so B_1, B_2, \dots must be uncountable.

Notice that this relies on the fact that each set B_i has some other item we can choose when constructing our diagonal element.

Sufficient: If all but finitely many of the B_i have one element, then there are only a finite number of B_i which have more than one element. Again, we can ignore the B_i with only one element. By the previous part, we showed that the Cartesian Product of a finite number of countable sets is countable, so the infinite Cartesian Product in this case is countable.

4 Impossible Programs

Show that none of the following programs can exist.

- (a) Consider a program P that takes in any program F, input x and output y and returns true if F(x) outputs y and returns false otherwise.
- (b) Consider a program P that takes in any program F and returns true if F(F) halts and returns false if it doesn't halt.
- (c) Consider a program P that takes in any programs F and G and returns true if F and G halt on all the same inputs and returns false otherwise.

Solution:

(a) If P exists, we can solve the halting problem. We show this by constructing $\mathtt{HALT}(F,x)$ where F is a program and x is the input. We will use P as a subroutine to derive a contradiction.

```
def HALT(F, x):
    y = 0 # arbitrarily chosen
    def F_prime(x):
      F(x)
      return y
    return P(F_prime, x, y)
```

We modify F to create F that runs F(x) and if F(x) halts it outputs an arbitrarily chosen output y. We then call P(f,x,y) and if it returns true then F(x) halts and if it returns false then F(x) must not halt. Therefore we have solved the halting problem. This is a contradiction because the halting problem is uncomputable. Therefore the program P cannot exist.

(b) If P exists, we can solve the halting problem. We show this by constructing HALT(F,x) where F is a program and x is the input. We will use P as a subroutine to derive a contradiction.

```
def HALT(F, x):
   def F_prime(ignore):
     return F(x)
   return P(F_prime)
```

We construct a function F which ignores its input and simply runs F(x). We then call P(F). If P(F') returns true then F(x) must have halted, otherwise P(F') will have returned false. Therefore, we have solved the halting problem. This is a contradiction because the halting problem is uncomputable. Therefore, the program P cannot exist.

(c) If P exists, we can solve the halting problem. We show this by constructing $\mathtt{HALT}(F,x)$ where F is a program and x is the input. We will use P as a subroutine to derive a contradiction.

```
def HALT(F, x):
    def F_prime(y):
        F(x)
        while x != y:
        pass
        return

def G(y):
    while x != y:
        pass
    return

return P(F_prime, G)
```

We construct functions G and F'. Both functions loop forever unless the input is x. Additionally, F' runs F(x) and so only halts if F(x) halts. We then call P(F',G), and if the answer is true, then the F halts on x. Otherwise, it does not halt on x. Therefore, we have solved the halting problem. This is a contradiction because the halting problem is uncomputable. Therefore, the program P cannot exist.

5 Printing All x Where M(x) Halts

Prove that it is possible to write a program *P* which:

- takes as input M, a Java program,
- runs forever, and prints out strings to the console,
- for every x, if M(x) halts, then P(M) eventually prints out x,
- for every x, if M(x) does NOT halt, then P(M) never prints out x.

Solution:

```
let S = \{\}
for i = 1 to \infty:
   let N_i = a new machine loaded with program M and input i
S = S \cup \{N_i\}

simulate every machine in S for 1 cycle

foreach N_x in S that has halted:
   print out X
   remove N_x from S
```

Consider any x, such that M(x) halts after n steps. For some k, at stage k, M(x) is added to S. At state k+n+1, M(x) halts, and x is then printed out.

For any x where M(x) does not halt, x is never printed out.

6 Counting, Counting, and More Counting

The only way to learn counting is to practice, practice, practice, so here is your chance to do so. For this problem, you do not need to show work that justifies your answers. We encourage you to leave your answer as an expression (rather than trying to evaluate it to get a specific number).

(a) How many 10-bit strings are there that contain exactly 4 ones?

- (b) How many ways are there to arrange n 1s and k 0s into a sequence?
- (c) A bridge hand is obtained by selecting 13 cards from a standard 52-card deck. The order of the cards in a bridge hand is irrelevant.
 How many different 13-card bridge hands are there? How many different 13-card bridge hands are there that contain no aces? How many different 13-card bridge hands are there that contain all four aces? How many different 13-card bridge hands are there that contain exactly 6 spades?
- (d) How many 99-bit strings are there that contain more ones than zeros?
- (e) An anagram of FLORIDA is any re-ordering of the letters of FLORIDA, i.e., any string made up of the letters F, L, O, R, I, D, and A, in any order. The anagram does not have to be an English word.
 - How many different anagrams of FLORIDA are there? How many different anagrams of ALASKA are there? How many different anagrams of ALABAMA are there? How many different anagrams of MONTANA are there?
- (f) If we have a standard 52-card deck, how many ways are there to order these 52 cards?
- (g) 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?
- (h) We have 9 balls, numbered 1 through 9, and 27 bins. How many different ways are there to distribute these 9 balls among the 27 bins? Assume the bins are distinguishable (e.g., numbered 1 through 27).
- (i) We throw 9 identical balls into 7 bins. How many different ways are there to distribute these 9 balls among the 7 bins such that no bin is empty? Assume the bins are distinguishable (e.g., numbered 1 through 7).
- (j) How many different ways are there to throw 9 identical balls into 27 bins? Assume the bins are distinguishable (e.g., numbered 1 through 27).
- (k) 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?
- (l) Let (1, 1) be the bottom-left corner and (8, 8) be the upper-right corner of a chessboard. If you are allowed to move one square at a time and can only move up or right, what is the number of ways to go from the bottom-left corner to the upper-right corner?
- (m) What is the number of ways to go from the bottom-left corner to the upper-right corner of the chesssboard, if you must pass through the square (6, 2), where (i, j) represents the square in the ith row from the bottom and the jth column from the left? (Again, you are only allowed to move up or right.)
- (n) How many solutions does $x_0 + x_1 + \cdots + x_k = n$ have, if each x must be a non-negative integer?
- (o) How many solutions does $x_0 + x_1 = n$ have, if each x must be a *strictly positive* integer?

(p) How many solutions does $x_0 + x_1 + \cdots + x_k = n$ have, if each x must be a *strictly positive* integer?

Solution:

- (a) This is just the number of ways to choose 4 positions out of 10 positions to place the ones, and so is $\binom{10}{4}$.
- (b) $\binom{n+k}{k}$
- (c) We have to choose 13 cards out of 52 cards, so this is just $\binom{52}{13}$.

We now have to choose 13 cards out of 48 non-ace cards. So this is $\binom{48}{13}$.

We now require the four aces to be present. So we have to choose the remaining 9 cards in our hand from the 48 non-ace cards, and this is $\binom{48}{9}$.

We need our hand to contain 6 out of the 13 spade cards, and 7 out of the 39 non-spade cards, and these choices can be made separately. Hence, there are $\binom{13}{6}\binom{39}{7}$ ways to make up the hand.

(d) **Answer 1:** There are $\binom{99}{k}$ 99-bit strings with k ones and 99-k zeros. We need k > 99-k, i.e. $k \ge 50$. So the total number of such strings is $\sum_{k=50}^{99} \binom{99}{k}$. This expression can however be simplified. Since $\binom{99}{k} = \binom{99}{99-k}$, we have

$$\sum_{k=50}^{99} \binom{99}{k} = \sum_{k=50}^{99} \binom{99}{99-k} = \sum_{l=0}^{49} \binom{99}{l}$$

by substituting l = 99 - k. Now $\sum_{k=50}^{99} \binom{99}{k} + \sum_{l=0}^{49} \binom{99}{l} = \sum_{m=0}^{99} \binom{99}{m} = 2^{99}$. Hence, $\sum_{k=50}^{99} \binom{99}{k} = (1/2) \cdot 2^{99} = 2^{98}$.

Answer 2: Since the answer from above looked so simple, there must have been a more elegant way to arrive at it. Since 99 is odd, no 99-bit string can have the same number of zeros and ones. Let A be the set of 99-bit strings with more ones than zeros, and B be the set of 99-bit strings with more zeros than ones. Now take any 99-bit string x with more ones than zeros i.e. $x \in A$. If all the bits of x are flipped, then you get a string y with more zeros than ones, and so $y \in B$. This operation of bit flips creates a one-to-one and onto function (called a bijection) between A and B. Hence, it must be that |A| = |B|. Every 99-bit string is either in A or in B, and since there are 2^{99} 99-bit strings, we get $|A| = |B| = (1/2) \cdot 2^{99}$. The answer we sought was $|A| = 2^{98}$.

(e) This is the number of ways of rearranging 7 distinct letters and is 7!.

In this 6 letter word, the letter A is repeated 3 times while the other letters appear once. Hence, the number 6! overcounts the number of different anagrams by a factor of 3! (which is the

number of ways of permuting the 3 A's among themselves). Hence, there are 6!/3! different anagrams.

In this 7 letter word, the letter A is repeated 4 times while the other letters appear once. Hence, the number 7! overcounts the number of different anagrams by a factor of 4! (which is the number of ways of permuting the 4 A's among themselves). Hence, there are 7!/4! anagrams.

In this 7 letter word, the letter A and N are each repeated 2 times while the other letters appear once. Hence, the number 7! overcounts the number of different anagrams by a factor of $2! \times 2!$ (one factor of 2! for the number of ways of permuting the 2 A's among themselves and another factor of 2! for the number of ways of permuting the 2 N's among themselves). Hence, there are $7!/(2!)^2$ different anagrams.

- (f) The first position of the ordering has 52 choices for the card, the second position has 51 choices, and so on, until the last position where there is only one choice. Thus, there are 52! ways to order the deck.
- (g) If we consider the 104! rearrangements of 2 identical decks, since each card appears twice, we would have overcounted each distinct rearrangement. Consider any distinct rearrangement of the 2 identical decks of 52 cards and see how many times this appears among the rearrangement of 104 cards where each card is treated as different. For each identical pair (such as the two Ace of spades), there are two ways they could be permuted among each other (since 2! = 2). This holds for each of the 52 pairs of identical cards. So the number 104! overcounts the actual number of rearrangements of 2 identical decks by a factor of 2^{52} . Hence, the actual number of rearrangements of 2 identical decks is $104!/2^{52}$.
- (h) Each ball has a choice of which bin it should go to. So each ball has 27 choices and the 9 balls can make their choices separately. Hence, there are 27⁹ ways.
- (i) **Answer 1:** Since each bin is required to be non-empty, let's throw one ball into each bin at the outset. Now we have 2 identical balls left which we want to throw into 7 distinguishable bins. There are 2 cases to consider:

Case 1: The 2 balls land in the same bin. This gives 7 ways.

Case 2: The 2 balls land in different bins. This gives $\binom{7}{2}$ ways of choosing 2 out of the 7 bins for the balls to land in. Note that it is *not* 7×6 since the balls are identical and so there is no order on them.

Summing up the number of ways from both cases, we get $7 + \binom{7}{2}$ ways.

Answer 2: Since each bin is required to be non-empty, let's throw one ball into each bin at the outset. Now we have 2 identical balls left which we want to throw into 7 distinguishable bins. From class (see notes 10), we already saw that the number of ways to put k identical balls into n distinguishable bins is $\binom{n+k-1}{k}$. Taking k=2 and n=7, we get $\binom{8}{2}$ ways to do this.

EASY EXERCISE: Can you give an expression for the number of ways to put k identical balls into n distinguishable bins such that no bin is empty?

- (j) Since there is no restriction on how many balls a bin needs to have, this is just the problem of throwing k identical balls into n distinguishable bins, which can be done in $\binom{n+k-1}{k}$ ways. Here k = 9 and n = 27, so there are $\binom{35}{9}$ ways.
- (k) **Answer 1:** Let's number the students from 1 to 20. Student 1 has 19 choices for her partner. Let i be the smallest index among students who have not yet been assigned partners. Then no matter what the value of i is (in particular, i could be 2 or 3), student i has 17 choices for her partner. The next smallest indexed student who doesn't have a partner now has 15 choices for her partner. Continuing in this way, the number of pairings is $19 \times 17 \times 15 \times \cdots \times 1 = \prod_{i=1}^{10} (2i-1)$.

Answer 2: Arrange the students numbered 1 to 20 in a line. There are 20! such arrangements. We pair up the students at positions 2i - 1 and 2i for i ranging from 1 to 10. You should be able to see that the 20! permutations of the students doesn't miss any possible pairing. However, it counts every different pairing multiple times. Fix any particular pairing of students. In this pairing, the first pair had freedom of 10 positions in any permutation that generated it, the second pair had a freedom of 9 positions in any permutation that generated it, and so on. There is also the freedom for the elements within each pair i.e. in any student pair (x,y), student x could have appeared in position 2i - 1 and student y could have appeared in position 2i and also vice versa. This gives 2 ways for each of the 10 pairs. Thus, in total, these freedoms cause $10! \times 2^{10}$ of the 20! permutations to give rise to this particular pairing. This holds for each of the different pairings. Hence, 20! overcounts the number of different pairings by a factor of $10! \times 2^{10}$. Hence, there are $20!/(10! \cdot 2^{10})$ pairings.

Answer 3: In the first step, pick a pair of students from the 20 students. There are $\binom{20}{2}$ ways to do this. In the second step, pick a pair of students from the remaining 18 students. There are $\binom{18}{2}$ ways to do this. Keep picking pairs like this, until in the tenth step, you pick a pair of students from the remaining 2 students. There are $\binom{2}{2}$ ways to do this. Multiplying all these, we get $\binom{20}{2}\binom{18}{2}\ldots\binom{2}{2}$. However, in any particular pairing of 20 students, this pairing could have been generated in 10! ways using the above procedure depending on which pairs in the pairing got picked in the first step, second step, ..., tenth step. Hence, we have to divide the above number by 10! to get the number of different pairings. Thus there are $\binom{20}{2}\binom{18}{2}\ldots\binom{2}{2}/10!$ different pairings of 20 students.

You may want to check for yourself that all three methods are producing the same integer, even though they are expressed very differently.

- (1) We must move up 7 times and right 7 times to reach the upper-right corner from the bottom-left corner. We can think of our movements as some permutation of exactly 7 U's and 7 R's, or UUUUUUURRRRRR. The number of ways to do this is $\binom{14}{7} = 3432$.
- (m) If we must pass through (6, 2), we can break up our movements into two parts. First, we must move up 5 times and right 1 time, some permutation of UUUUUR. Then, once we've hit (6, 2), we must move up 2 more times and right 6 more times, some permutation of UURRRRRR. The total number of ways for the entire journey is given by every combination of the first part of our journey with every combination of the second part: $\binom{6}{5} \cdot \binom{8}{2} = 168$.

- (n) $\binom{n+k}{k}$. There is a bijection between a sequence of n ones and k plusses and a solution to the equation: x_0 is the number of ones before the first plus, x_1 is the number of ones between the first and second plus, etc. A key idea is that if a bijection exists between two sets they must be the same size, so counting the elements of one tells us how many the other has.
- (o) n-1. It's easy just to enumerate the solutions here. x_0 can take values 1, 2, ..., n-1 and this uniquely fixes the value of x_1 . So, we have n-1 ways to do this. But, this is just an example of the more general question below.
- (p) $\binom{(n-(k+1))+k}{k} = \binom{n-1}{k}$. By subtracting 1 from all k+1 variables, and k+1 from the total required, we reduce it to problem with the same form as the previous problem. Once we have a solution to that we reverse the process, and adding 1 to all the non-negative variables gives us positive variables.

7 Fermat's Necklace

Let p be a prime number and let k be a positive integer. We have an endless supply of beads. The beads come in k different colors. All beads of the same color are indistinguishable.

- (a) We have a piece of string. As a relaxing study break, we want to make a pretty garland by threading *p* beads onto the string. How many different ways are there construct such a sequence of *p* beads of *k* different colors?
- (b) Now let's add a restriction. We want our garland to be exciting and multicolored. Now how many different sequences exist? (Your answer should be a simple function of k and p.)
- (c) Now we tie the two ends of the string together, forming a circular necklace which lets us freely rotate the beads around the necklace. We'll consider two necklaces equivalent if the sequence of colors on one can be obtained by rotating the beads on the other. (For instance, if we have k = 3 colors—red (R), green (G), and blue (B)—then the length p = 5 necklaces RGGBG, GGBGR, GBGRG, BGRGG, and GRGGB are all equivalent, because these are cyclic shifts of each other.)

How many non-equivalent sequences are there now? Again, the p beads must not all have the same color. (Your answer should be a simple function of k and p.)

[*Hint*: What follows if rotating all the beads on a necklace to another position produces an identical looking necklace?]

(d) Use your answer to part (c) to prove Fermat's little theorem. (Recall that Fermat's little theorem says that if p is prime and $a \not\equiv 0 \pmod{p}$, then $a^{p-1} \equiv 1 \pmod{p}$.)

Solution:

(a) k^p .

- (b) $k^p k$. You can have k sequences of a beads with only one color.
- (c) Since p is prime, rotating any sequence by less than p spots will produce a new sequence. As in, there is no number x smaller than p such that rotating the beads by x would cause the pattern to look the same. So, every pattern which has more than one color of beads can be rotated to form p-1 other patterns. So the total number of patterns equivalent with some bead sequence is p. Thus, the total number of non-equivalent patterns are $(k^p k)/p$.
- (d) $(k^p k)/p$ from above has to be an integer. Hence, $k(k^{p-1} 1)$ has to be divisible by p. If k is coprime with p, then it follows that $k^{p-1} 1$ is divisible by p which is equivalent to FLT.