Countability, Computability, Counting, Combinatorial Proofs 6

Computer Science Mentors 70

October 8 to 12, 2018

1 Countability

1.1 Introduction

(a.) Cardinality is the number of elements in a set. We define a set as countably infinite if it has the same cardinality as the natural numbers (or any countable set).
(b.) We can prove a set is countable by finding a bijection between it and any countable set. A few classic examples are the hotel argument to show that $\mathbb{Z}^+$ is countable, and the spiral argument to show that $\mathbb{Q}$ is countable, both included in your notes.
(c.) To prove a set is uncountable, we can either find a bijection between it and an uncountable set or use the Cantor Diagonalization proof, included in your notes.

1.2 Questions

1. **True/False**

   (a) Every infinite subset of a countable set is countable

   (b) If A and B are both countable, then $A \times B$ is countable

   (c) If A_i is countable, then $A_1 \times A_2 \times A_3 .... \times A_N$ for N finite is countable.

   (d) If A_i is countable, then $A_1 \times A_2 \times A_3 ...$ a countably infinite number of times is countable.

(e) Every infinite set that contains an uncountable set is uncountable.

2. Are these sets countably infinite/uncountably infinite/finite? If finite, what is the order of the set?

   (a) Finite bit strings of length n.

   (b) All finite bit strings of length 1 to n.

   (c) All finite bit strings

   (d) All infinite bit strings

   (e) All finite or infinite bit strings.

3. Find a bijection between N and the set of all integers congruent to $1 \mod n$, for a fixed n.

4. Are the power sets S of a countably infinite set are finite, countably infinite, or uncountably infinite? Provide a proof for your answer.

2 Computability

2.1 Introduction

The Halting Problem: Does a given program ever halt when executed on a given input? This given input has to be general.

$$\texttt{TestHalt(P,x)} = \begin{cases} \text{“yes"}, & \text{if program } P \text{ halts on input } x \\ \text{“no"}, & \text{if program } P \text{ loops on input } x \end{cases}$$

To prove `TestHalt` doesn't exist, we assume it does, and hope to reach a contradiction. We define another program:

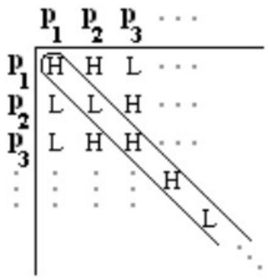
```
Turing(P)
  if TestHalt(P,P) = "yes" then loop forever
  else halt
```

What happens when we call `Turing(Turing)`?

Case 1 : It halts. If `Turing(Turing)` halts then `TestHalt(Turing, Turing)` must have returned no. But `TestHalt(Turing Turing)` calls `Turing(Turing)` and calling `Turing(Turing)` must loop. But we assumed that `Turing(Turing)` halted. Contradiction.

Case 2 : It loops. This implies that `TestHalt(Turing, Turing)` returned yes, which by the way that `TestHalt` is defined implies that `Turing` halted. But we assumed that `Turing(Turing)` looped. Contradiction.

How is this just a reformulation of proof by diagonalization?



List all possible programs as rows and columns. The rows are the programs and the columns are the inputs. Turing must be one of the rows, say row n. If entry (n, n) is H then Turing will loop by definition. If entry (n, n) is L then Turing will halt by definition. Therefore Turing cannot be on the list of all programs and therefore it does not exist. Therefore the Halting Problem is unsolvable. We can use this to prove that other problems are also unsolvable.

Say we are asked if program M is solvable. To prove it is not, we just need to prove the following claim: If we can compute program M, then we could also compute the halting problem. This would then prove that M can not exist, since the halting problem is not computable. This amounts to proof by contradiction.

2.2 Questions

1. Determine the computability of the following tasks. If it's not computable, write a reduction or self-reference proof. If it is, write the program.

    1. You want to determine whether a program P on input x prints "Hello World!" Is there a computer program that can perform this task? Justify your answer?

    2. You want to determine whether a program P prints "Hello World!" before running the kth line of the program.

    3. You want to determine whether a program P prints "Hello World!" in the first k steps of its execution. Is there a computer program that can perform this task? Justify your answer?

3 Intro to Counting

Counting:
In this class, the basic premise of counting is determining the total number of possible ways something can be done. Reaching a particular outcome requires a number of specific choices to be made. To figure out the total number of possible outcomes, we multiply together the number of potential choices at each step.

1. You're getting ready in the morning, and you have to choose your outfit for the day.
   (a) You need to wear a necklace, a vest, and a sweater. Depending on the day, you decide whether it is worth wearing your watch. If you have 3 necklaces, 2 vests, and 4 sweaters, how many different combinations do you choose from each morning?

   (b) Now the order in which you put on your necklace, vest, and sweater matters. Specifically, your look after putting on necklace n first, vest v, and then sweater s is different than if you put on vest v first, necklace n, and then sweater s. When you put on your watch is irrelevant. Now how many options do you have?

Ordering and Combinations:
An important idea of counting is dealing with situations in which all of our choices must be drawn from the same set. Here is a chart which walks you through how to solve problems relating to this idea:

Order matters, with replacement Example: How many 3 letter "words" can we make with the letters a, b, c, and d assuming we can repeat letters? Answer: $4^3 = 64$ General problem: From a set of n items, how many ways can we choose k of them, assuming that we can choose the same item multiple times and the order in which we choose the items matters? General Form: n^k	**Order matters, without replacement** Example: How many 3 letter "words" can we make with the letters a, b, c, d, e, and f using each letter exactly once? Answer: $\frac{6!}{(6-3)!} = 120$ General problem: From a set of n items, how many ways can we choose k of them, assuming that we can choose a given item exactly once and the order in which we choose the items matters? Answer: $P(n,k) = \frac{n!}{(n-k)!}$
Order doesn't matter, without replacement Example: How many ways can I pick a team of 3 from 7 possible people? Answer: $\frac{7!}{(7-3)!(7-4)!} = 35$ General problem: From a set of n items, how many ways can we choose k of them, assuming that we can choose a given item exactly once and the order that we choose the items doesn't matter? General Form: $\binom{n}{k} = \frac{n!}{(n-k)!k!}$	**Special note: Sequencing** Example: How many different orderings are there of the letters in "CAT"? Answer: 3! How many different orderings are there of the letters in "BOOKKEEPER?"? Answer: $\frac{10!}{2!2!3!}$

1. How many ways are there to arrange the letters of the word SUPERMAN
   (a) On a straight line?
   (b) On a straight line, such that SUPER occurs as a substring?
   (c) On a circle?
   (d) On a circle, such that SUPER occurs as a substring?
2. (a) You have 15 chairs in a room and there are 9 people. How many different ways can everyone sit down?
   (b) How many ways are there to fill 9 of the 15 chairs? (We do not care who sits in them.)
3. The numbers 1447, 1005, and 1231 have something in common. Each of them is a four digit number that begins with 1 and has two identical digits. How many numbers like this are there?
4. How many ways can you deal 13 cards to each of 4 players so that each player gets one card of each of the 13 values (A, 2, 3. . .K)?
5. We grab a deck of cards and it is poker time. Remember, in poker, order does not matter.
   (a) How many ways can we have a hand with exactly one pair? This means a hand with ranks (a, a, b, c, d).
   (b) How many ways can we have a hand with four of a kind? This means a hand with ranks (a, a, a, a, b).
   (c) How many ways can we have a straight? A straight is 5 consecutive cards, that do not all necessarily have the same suit.
   (d) How many ways can we have a hand of all of the same suit?

(e) How many ways can we have a straight flush? This means we have a consecutive-rank hand of the same suit. For example, (2, 3, 4, 5, 6), all of spades, is a straight flush, while (2, 3, 5, 7, 8), all of spades, is NOT, as the ranks are not consecutive.

4 Counting

4.1 Introduction

Balls and Bins:

a. **Distributing n distinguishable balls amongst k distinguishable bins:** Each ball has k possible bins to go into, and there are n balls. Solution: k^n

b. **Distributing n indistinguishable balls amongst k distinguishable bins:**Solution: $\binom{n+k-1}{k-1}$

Note: Distributing balls among indistinguishable bins is not covered in CS 70!

The solution for case (b) initially seems somewhat unintuitive, but can be explained through an example.

How many ways can we distribute 7 dollar bills amongst 3 students?

Approaching this with the approaches we currently know fails: There are 7 possible options for the number of bills you give to the first student, but the number of bills you choose to give the first student has a *direct* effect on the numbers of bills you can give to the second student.

To solve this problem, we need to format it slightly differently: put the dollar bills on a line, and insert 2 dividers. Everything to the left of the first divider is given to the first student. Everything in between the dividers is given to the second student. Everything to the right of the second divider is given to the third student:

There are $\frac{9!}{7!2!} = \binom{9}{2} = 36$ ways to place 2 dividers among 9 positions such that the remaining positions are filled with dollar bills, and therefore 36 ways to distribute the money. This tactic of using dividers is commonly referred to *stars and bars* or *sticks and stones*. More generally, there are $\binom{n+k-1}{k-1}$ ways to distribute n indistinguishable items amongst k people.

4.2 Questions

1. How many ways are there to arrange the letters of the word SUPERMAN
   (a) On a straight line, such that SUPER occurs as a subsequence (S U P E R appear in that order, but not necessarily next to each other)?

   (b) On a circle, such that SUPER occurs as a subsequence (S U P E R appear in that order, but not necessarily next to each other)?

2. How many ways can you give 10 cookies to 4 friends?

3. How many solutions does $x+y+z = 10$ have, if all variables must be positive integers?

4. How many 5-digit sequences have the digits in non-decreasing order?

5 Combinatorial Proofs

5.1 Questions

1. Prove $k\binom{n}{k} = n\binom{n-1}{k-1}$ by a combinatorial proof.

2. Prove $a(n-a)\binom{n}{a} = n(n-1)\binom{n-2}{a-1}$ by a combinatorial proof.