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

   Solution: True. Define set S as the infinite subset, and set T as the countable set. $S \subseteq T$, which means $|S| \leq |T|$. This means S is countable. S is also an infinite subset, so it is infinite. Therefore, S is countably infinite.

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

Solution: True. Can draw a bijection where first elem of A maps to 1, first elem of B maps to 2, second elem of A maps to 3, etc. This will include $A \times B$ at the end, and because there is a bijection from A to $\mathbb{N}$ and B to $\mathbb{N}$, there is a bijection from $A \times B$ to $\mathbb{N}$. There is clearly a mapping from $A \times B$ to $\mathbb{N} \times \mathbb{N}$, and $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$, so there is a bijective mapping from $A \times B$ to $\mathbb{N}$.

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

Solution: We can give a kind of a "collapsing" argument. We know that $A_1 \times A_2$ is countable, so we can just define $A_{12} = A_1 \times A_2$, then we have $A_{12} \times A_3$.. and so on.

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

Solution: False. Consider the real numbers - each "digit" is picked from a finite countable set, crossed a countable number of times, but the entire set is uncountable.

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

Solution: True. Let A be an uncountable subset of B. Assume that B is countable. $f : \mathbb{N} \to B$ is a bijection. There must be a subset M of $\mathbb{N}$ such that $f : M \to A$ is a bijection. Then A is countable. This is a contradiction. So B must be 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.

Solution: Finite. There are 2 choices (0 or 1) for each bit, and n bits, so there are $2 \times 2 \times \ldots \times 2 = 2^n$ such bit strings.

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

Solution: Finite. By part (a), there are 2^1 bit strings of length 1, 2^2 of length 2, etc. Thus, there are $2^1 + 2^2 + \ldots + 2^n = 2^{n+1} - 2$.

(c) All finite bit strings

Solution: Countably infinite. We can list these strings as follows: $\{0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, 0000, \ldots\}$. This gives us a bijection with the (countable) natural numbers, so these are countably infinite.

(d) All infinite bit strings

Solution: Uncountably infinite. We can construct a bijection between this set and the set of real numbers between 0 and 1. We can represent these real numbers using binary e.g. they are of the form $0.0110001010110\ldots$. By diagonalization, the set of real numbers between 0 and 1 is uncountably infinite; therefore, so is this set.

(e) All finite or infinite bit strings.

Solution: Uncountably infinite. This is the union of a countably infinite set (part c) and an uncountably infinite set (part d), so it is uncountably infinite.

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

Solution: The set of integers congruent to 1 mod n is $A = \{1 + kn \mid k \in \mathbb{Z}\}$. Define $g : \mathbb{Z} \to A$ by $g(x) = 1 + x \times n$; this is a bijection because it is clearly one-to-one, and is onto by the definition of A. We can combine this with the bijective mapping $f : \mathbb{N} \to \mathbb{Z}$ from the notes, defined by $f(x) = \begin{cases} \frac{x}{2} & \text{if } x \text{ is even} \\ \frac{-(x+1)}{2} & \text{if } x \text{ is odd} \end{cases}$. Then $f \circ g$ is a function from $\mathbb{N}$ to A, which is a bijection.

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

Solution: The power sets of a countably infinite set are uncountably infinite. There is a bijection between the set $2^{\mathbb{N}}$ and 2^S, as S and $\mathbb{N}$ have the same cardinality. The set $2^{\mathbb{N}}$ is uncountable. We prove this through contradiction. We assume the set $2^{\mathbb{N}}$ is countably infinite. This means we can list the subsets of $\mathbb{N}$ such that every subset is N_i for some i. We define another set $A = \{i | i \geq 0 \text{ and } i \notin N_i\}$ which contains integers i not part of N_i. But, N is a subset of $\mathbb{N}$ so we must have $N = N_j$ for some j. This means that if $j \in N$, then $j \notin N$, and if $j \notin N$, then $j \in N$. This is a contradiction since j is either in N or not, so the set is not countably infinite.

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:

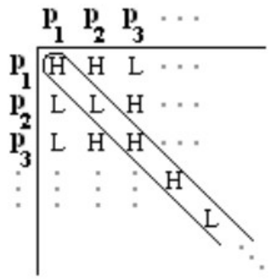
```
Turing(P)
  if TestHalt(P,P) = "yes" then loop forever
  else halt
```

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?

        Solution: Define a program that takes programs as inputs, and prints Hello World at the end of the execution of the program:
        ```
        def reduce(input):
           execute(input)
           print("Hello World")
        ```
        Now if we have some arbitrary program input x, if we call `testHelloWorld(reduce(x))`, we will be able to solve the Halting Problem, so we know that this is impossible.

    2. You want to determine whether a program P prints "Hello World!" before run-

ning the kth line of the program.

Solution: Similar to the last program, we can write a reduction program:

```
def reduce(input):
   execute(input)
   print("Hello World")
```

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?

Solution: Yes, you can run the program until k steps are executed. If P has halted by then, return true. If not, return false.

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?

   Solution: $3 \cdot 2 \cdot 4 \cdot 2$ = 48 (# necklaces · # vests · # sweaters · wearing watch or not). These are all independent choices, so we can simply multiply the number of ways to make each choice together.

   (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?

   Solution: $3 \cdot 2 \cdot 4 \cdot 2 \cdot 3! = 288$. We cannot simply multiply the outcomes like we did in the previous problem, because then we don't differentiate between (vest v, necklace n, sweater s) and (necklace n, vest v, sweater s) - we count them as the same case. We must count the number of ways to select a combination of 3 items, and then the number of ways to order each selection - 3!.

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:

1. How many ways are there to arrange the letters of the word SUPERMAN
   (a) On a straight line?

Order matters, with replacement	**Order matters, without 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	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: $\mathrm{P}(n,k) = \frac{n!}{(n-k)!}$
Order doesn't matter, without replacement	**Special note: Sequencing**
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: $\binom{n}{k} = \frac{n!}{(n-k)!k!}$	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: $\frac{10!}{2!2!3!}$

Solution: 8!

(b) On a straight line, such that SUPER occurs as a substring?

Solution: 4! Treat SUPER as one character.

(c) On a circle?

Solution: 7! Anchor one element, arrange the other 7 around in a line.

(d) On a circle, such that SUPER occurs as a substring?

Solution: 3! Treat SUPER as a single character, anchor one element, and arrange the other 3 around in a line.

2. (a) You have 15 chairs in a room and there are 9 people. How many different ways can everyone sit down?

Solution: $\frac{15!}{6!}$ There are 15 places to put the first person, then 14 places to put the second person, 13 places to put the third person, etc. all the way to the last person who has 7 places to sit. Another way to think about this is like the anagram example above. We have 9 unique letters and 6 repeats (our empty spaces). We divide by the number of repeats giving us: $15 * 14 * 13 * 12 * 11 * 10 * 9 * 8 * 7 = \frac{15!}{6!}$

(b) How many ways are there to fill 9 of the 15 chairs? (We do not care who sits in them.)

Solution: $\binom{15}{9} = \frac{15!}{9!(15-9)!}$ In this example, we do not care about the uniqueness of each person, so we can just count each person as a repeat. So like the anagram example we will divide for every repeat. We have 9 human repeats, and 6 empty space repeats. Hence $\frac{15!}{9!6!}$.

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?

Solution: Case 1: the identical digits are 1 (e.g. $11xy, 1x1y, 1xy1$)
Since there can only be two numbers that are identical, x and y cannot be 1 and $x \neq y$.
So [Possible formats] * [Possible x values] * [Possible y values] $= 3 * 9 * 8 = 216$
Case 2: identical digits are not 1 (e.g. $1xxy, 1xyx, 1yxx$)
So [Possible formats] * [Possible x values] * [Possible y values] $= 3 * 9 * 8 = 216$
Add both cases to arrive at the final result: $216 + 216 = 432$

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)?

Solution: There are $4!$ ways to distribute the aces to the 4 players, $4!$ ways to distribute the twos, and so on, so the number of ways to deal the cards in this manner is $4!^{13} = 24^{13}$.

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).

Solution: $= 13 * \binom{4}{2} * \binom{12}{3} * 4^3$. There are 13 value options for a (2, 3, 4, $\ldots$, K, A). We then need to choose 2 out of the 4 possible suits. Now we need to choose b, c, and d. There are 12 values left (must be different from a). Finally, there are 4 suit options for each of the values chosen for b, c, and 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).

Solution: $= 13 * 12 * 4$

(c) How many ways can we have a straight? A straight is 5 consecutive cards, that do not all necessarily have the same suit.

Solution: A straight can begin at any number from 2-10: (2, 3, 4, 5, 6); (3, 4, 5, 6, 7)...(10, J, Q, K, A). That gives us 9 possibilities. Each number in hand has 4 possibilities (suits) $= 9 * (4^5)$ total possibilities.

(d) How many ways can we have a hand of all of the same suit?

Solution: $4 * \binom{13}{5}$. For each of the 4 suits, there are $\binom{13}{5}$ different combinations of 5 cards among 13 to choose from.

(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.

Solution: For each of 4 suits, there are 9 number combinations (as shown in c, starting from 2 to starting from 10). Each number combination is unique, because there is only one number per suit. $= 4 * 9 = 36$.

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)?

Solution: 3! * $\binom{8}{3}$ This reduces to a stars and bars problem–the S U P E R are bars, and we want to put M A N somewhere in the sequence. Once we do so, there can be any permutation of M A N within the bars. Equivalently, we can arrange the letters of SUPERMAN 8! ways, but divide by 5! because we have arranged SUPER in any of 5! ways, when we only want one way. This gives us 8! / 5!, which is equal to 3! * 8! / (5! 3!) = 3! * $\binom{8}{3}$.

(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)?

Solution: 3! * $\binom{7}{3}$. Anchor one element (for simplicity, choose M, A, or N). Then follow the same procedures as earlier.

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

Solution: Count the number of ways to give 10 cookies to 4 friends if some can get no cookies. The number of ways is $\binom{13}{3} = 286$.

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

Solution: We can think of this in terms of stars and bars. We have two bars between the variables x, y, and z, and our stars are the 10 1s we have to distribute among them. Since all variables must be positive integers, x, y, and z will each be at least 1. So, we have 7 1s left to distribute. So $n = 7$ stars, $k = 2$ bars. Answer = $\binom{n+k}{k} = \binom{9}{2} = 36$.

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

Solution: This can be framed as a stars and bars problem. We have 9 bars between the numbers 0 through 9 and must place 5 stars in these slots. The location of a star represents the value of its associated digit. This ensures the 5 numbers in our sequence are either repeated or increasing. So, the answer is $\binom{14}{9} = 2002$. Note that if the question asked about increasing order, we would not use the stars and bars approach.

5 Combinatorial Proofs

5.1 Questions

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

> **Solution:** Choose a team of k players where one of the players is the captain.
> LHS: Pick a team with k players. This is $\binom{n}{k}$. Then make one of the players the captain. There are k options for the captain so we get $k \times \binom{n}{k}$.
> RHS: Pick the captain. There are n choices for the captain. Now pick the last $k-1$ players on the team. There are now $n-1$ people to choose from. So we get $n \times \binom{n-1}{k-1}$.

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

> **Solution:** Suppose that you have a group of n players.
> LHS: Number of ways to pick a team of a of these players, designate one member of the team as captain, and then pick one reserve player from the remaining $n-a$ people.
> RHS: The right-hand side is the number of ways to pick the captain, then the reserve player, and then the other $a-1$ members of the team.