CIT 5920

Discrete Math for Computer Science

Author
Arvind Bhusnurmath
University of Pennsylvania
Fall 2014

Sets

Sets - Basic operations

A set is just a collection of things. The things inside a set are called the elements of the set.

2 important points about a set

1. the order of elements does not matter.
2. repetition does not matter (some people do not allow repetition). When you talk about the number of elements of a set, it is implicit that you are talking about the distinct elements.

Examples- A set of numbers $\{1, 2, 3\}$

A set of MCIT lecturers $\{$"*Chris*", "*Arvind*", "*Eric*", "*Tom*"$\}$

A set does not have to make any logical sense. As far as mathematics is concerned $\{3, 5.6, \frac{8}{2.7}\}$ is a set.

Notation

If S is a set and x is something in the set, we say $x \in S$.

So far, the notation we have introduced is called set-roster notation where we are listing out elements of the set. If you know each and every element of your set, this is a convenient notation i.e. just list them out between two braces. Often, sets are large and we do not want to spend time writing each element. $\{1, 2, 3, \ldots, 42\}$ is valid notation and is understood to mean all the integers from 1 to 42.

Remember that $\{0\}$ *is different from the number* 0*. One of them is a set containing the single element 0, the other is just the number 0.*

Conventionally, the variable used for representing a set is in upper case. The set S, the set T etc. The elements of the set are generally represented in lower case. Although not incorrect, it would be surprising to see anyone write $S \in x$.

Think of these conventions to be similar to the way you name your variables in Java. They are established so that readers of your mathematical statements (and you will be writing a lot of them in this course!) are less confused.

Question - How many elements are there in the set $\{1, \{1, 2\}\}$.

Answer - 2. The set has two things in it. One of them is a number and the other is a set. It does not matter how many elements are in the set inside the set.

The **empty set**(the set containing nothing) is denoted by the Greek letter $\emptyset$. Please do not confuse the empty set with the set that contains 0. One of them has a single element, the other has no elements at all.

The special sets

Certain sets are used so often that they have special notation.

- $\mathbb{N}$ - natural numbers. $\{0, 1, 2, \ldots\}$.
- $\mathbb{Z}$ - integers. These are 0, positive negative whole numbers that you are familiar with.
- $\mathbb{R}$ - real numbers (basically any number we touch in this course).
- $\mathbb{Q}$ - rational numbers. These are numbers of the form $\frac{p}{q}$ where $q \neq 0$ and $p \in \mathbb{Z}$, $q \in \mathbb{Z}$. Remember that numbers with a decimal point can be expressed in this form as long as they either have a terminating decimal like 2.56 or they have a recurring decimal like 0.3333333.

  Examples of real numbers that are not rational numbers include $\sqrt{2}, \pi, e$

Often to narrow down on the positive or negatives, the special set will have a superscript on it such as $\mathbb{Z}^+$, which means the positive integers. Or for instance, $\mathbb{R}^-$, which would mean the negative real numbers.

Set-builder

An alternative way of specifying a set is to start with one of the special sets and then pick elements that only satisfy certain properties.

What does $\{x \in \mathbb{Z} | -2 < x < 5\}$ correspond to?
It is equivalent to the set $\{-1, 0, 1, 2, 3, 4\}$.

The syntax is to first introduce the 'main' set (generally called the universe), then put down a | and then some kind of conditional.

Subset

If A and B are sets, then A is called a subset of B, $A \subseteq B$ if and only if, every element in A is also an element in B.

If $x \in A$, this means that $x \in B$.

Importantly, if $A \subseteq B$, this does not necessarily mean $B \subseteq A$.

The (true) statement 'Every rational number is a real number' gets mathematically expressed as $\mathbb{Q} \subseteq \mathbb{R}$. But, we know that there are real numbers that are not rational numbers. Among others, π and e and $\sqrt{2}$. So, $\mathbb{R} \not\subseteq \mathbb{Q}$.

Element of and subset of

A very common confusion is that between the $\in$ and $\subseteq$. Here is an attempt at clarifying that via an example (the zybook has some too).

Consider the set $S = \{CIT, \pi, \{\text{apples}, \text{bananas}\}\}$

What are the elements of this set?

An element of the set is a single item inside the set. So the elements are

1. CIT
2. π
3. {apples, bananas}

And YES, one of the elements is this weird set of some fruit.

What are the subsets of this set?

This questions becomes easier to answer now that we have answered the elements question. Subsets after all are made by collecting none, some or all of these elements and putting them between { and }, that is, making a set out of them.

So here are all the subsets

1. $\emptyset$
2. $\{CIT\}$
3. $\{\pi\}$
4. $\{\{apples, bananas\}\}$ - this one is slightly tricky, but just think of the apples and bananas set as this one single entity.
5. $\{CIT, \pi\}$
6. $\{\pi, \{\text{apples}, \text{bananas}\}\}$
7. $\{CIT, \{\text{apples}, \text{bananas}\}\}$
8. $\{CIT, \pi, \{\text{apples}, \text{bananas}\}\}$

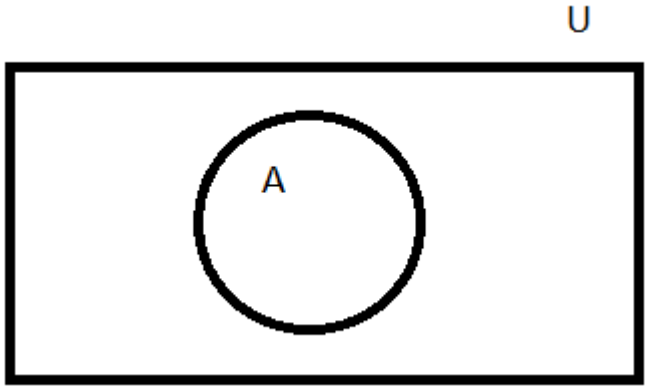
Venn diagrams

A Venn diagram is just a pictorial representation of a set.

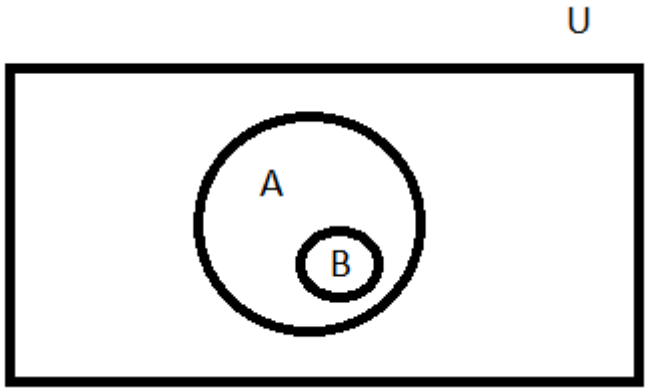
Generally speaking, we draw one big box for the universe. The universe can be different depending upon the context being used. For instance, with the numbers (especially in this course), it makes sense to think of the universe as $\mathbb{R}$.

All sets that we speak of are contained in this universe. So a set A will be be as subset of U. $A \subseteq U$.

In a Venn diagram, the set A will be represented like this.



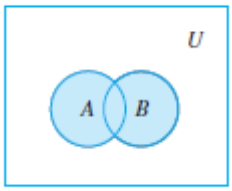
Using a Venn diagram, it becomes easy to represent a lot of things in set theory. For example, if $B \subseteq A$, it can be drawn like this



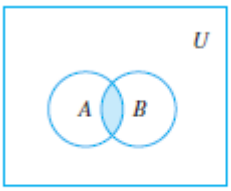
Set operations

In the following, the shaded regions in each Venn diagram represents the operation we are talking about.

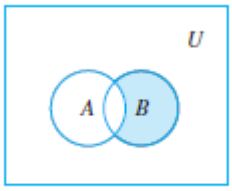
- Union - The union of two sets A and B is defined as the set of elements that are either in A or in B (elements that are in both sets are included!). It is represented as $A \cup B$



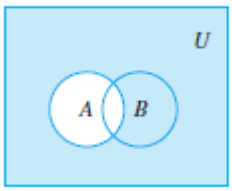
- Intersection - this is defined as the set of all elements that are in both A and in B. It is represented as $A \cap B$.



- Difference - The difference of B minus A, denoted $B - A$ refers to the set consisting of elements in B that are not in A.



- Complement - the complement of A, denoted as $\bar{A}$ is the set of all elements in U that are not in A.



It is also a good time to get used to writing these in mathematical manner.

$$A \cup B = \{x \in U | x \in A \text{ or } x \in B\}$$
$$A \cap B = \{x \in U | x \in A \text{ and } x \in B\}$$
$$B - A = \{x \in B | x \notin A\}$$
$$\bar{A} = \{x \in U | x \notin A\}$$

Properties

- $A \cup B = B \cup A$ - commutative law
- $A \cap B = B \cap A$ - commutative law
- $A \cap (B \cap C) = (A \cap B) \cap C$ - associative law
- $A \cup (B \cup C) = (A \cup B) \cup C$ - associative law
- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ - distributive law
- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ - distributive law
- $A \cap \emptyset = \emptyset$
- $A \cup U = U$
- $A \cap U = A$
- $A \cup \emptyset = A$

De-Morgan's laws

De-Morgan's laws which you might have seen as part of logic, translate very easily into set theory.

$$\overline{(A \cup B)} = \bar{A} \cap \bar{B}$$
$$\overline{(A \cap B)} = \bar{A} \cup \bar{B}$$

Examples

Show that $\overline{\bar{A} \cap U} = A$

We simplify this using the properties of sets

$$\begin{aligned}
\overline{\bar{A} \cap U} &= \bar{\bar{A}} \cup \bar{U} && \text{de-morgan's} \\
&= A \cup \bar{U} && \text{complement of a complement is the set itself} \\
&= A \cup \emptyset && \text{complement of the universe = empty!} \\
&= A
\end{aligned}$$

Show that $(A \cap B) \cup \overline{(A \cup \bar{B})} = B$

$$\begin{aligned}
\text{the left side} &= (A \cap B) \cup (\bar{A} \cap B) && \text{de-morgan's} \\
&= (A \cup \bar{A}) \cap B && \text{distributive property} \\
&= U \cap B && \text{the union of a set and its complement will be the universal set} \\
&= B
\end{aligned}$$

Cardinality

The number of elements in a finite set is called the cardinality of that set. A is a set, the cardinality is denoted by $|A|$.

What is the cardinality of the set $S = \{\{1\}, \{1, 2\}\}$?

Be careful, this is a set of sets! Cardinality is 2 and not 3.

Power set

The power set of a set A is the set of all subsets of A and is denoted P(A).

An important thing to remember about the power set of A is that the empty set $\emptyset$ and the set A itself, are both subsets of A.

What do you think $|P(A)|$ is?

CS application

Databases - A lot of query syntax involves very basic set theory.

Made up Examples

There is an retirement agency that is interested in knowing the employees in MS and Apple that are close to retirement.

Actual SQL syntax would look something like

```
SELECT Name, EmployeeId
FROM MSEmployees
WHERE age > 58

UNION

SELECT Name, EmployeeId
FROM AppleEmployees
WHERE age > 58
```

Tell me what I ate for lunch but did not eat for dinner. Set difference!

```
SELECT item FROM Lunch
WHERE consumer = 'Arvind'
EXCEPT
SELECT item FROM Dinner
WHERE consumer = 'Arvind'
```

Set Operations

Common mistakes

One of the common mistakes in set theory is to confuse 'element of' and 'subset of'. Confuse $\in$ and $\subset$.

Is it $2 \in \{1, 2, 3\}$ or $2 \subset \{1, 2, 3\}$? - 2 is just a number. It is not a set. It is an element of the set.

Is it $\{2\} \in \{1, 2, 3\}$ or $\{2\} \subset \{1, 2, 3\}$? - Now $\{2\}$ represents a set. Is this set completely contained inside the set $\{1, 2, 3\}$. Yes. Therefore we have a subset in this case.

It gets more interesting (or confusing :)) when you are dealing with sets of sets.

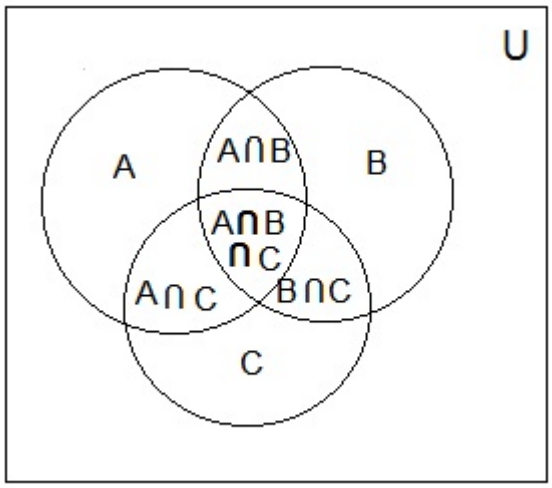
Is $\{2\} \in \{\{2\}, \{3\}\}$? Yes! Because, in this example, the set's individual elements are sets themselves.

So what would be a subset of the set above? - $\{\{2\}\}$ would be one example. There are 3 more subsets. What are they?

More unions and intersections

Set unions and intersections are not restricted to just 2 sets. In fact, expressing operations with 3 sets using Venn diagrams is quite easy.

In general a 3 set Venn diagram will look something like this.



Using just the Venn diagram, these properties can be clearly demonstrated.

- Commutative laws
  $A \cap B = B \cap A$ and $A \cup B = B \cup A$
- Associative laws
  $(A \cap B) \cap C = A \cap (B \cap C)$ and $(A \cup B) \cup C = A \cup (B \cup C)$
- Distributive laws
  $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Very often when wanting to express a number of sets, we use subscripts. Generally, each individual set will have something to do with the subscript.

Examples

1. Let N_i represent the positive integers that are divisible by i. Then
   $N_4 = \{4, 8, 12, \ldots\}$ $N_{11} = \{11, 22, 33, \ldots\}$
2. Let B_i represent the set $\{i, -i\}$. Then $B_2 = \{2, -2\}$.

If you have a collection of sets, it becomes convenient to introduce notation for the union and intersection.

$\bigcup_{i=1}^{n} A_i$ - union of the sets A_1 through A_n.

$\bigcap_{i=1}^{n} A_i$ - intersection of the sets A_1 through A_n.

Assume P_i is used to represent all the people in the world who are i years old. Then something like $\bigcup_{i=0}^{130} P_i$ would give you all the people in the world since no one is older than 130.

In the same example, note that $P_i \cap P_j = \emptyset$, for any pair of $i \neq j$. After all, no one can be both i years old and j years old if the i and j are distinct.

Cartesian Product

$A \times B$, called the cartesian product of A and B consists of ordered pairs of the form (a,b) where $a \in A$ and $b \in B$.

Say $A = \{1, 2, 3\}$ and $B = \{2, 5\}$ then the Cartesian product is

$A \times B = \{(1, 2), (1, 5), (2, 2), (2, 5), (3, 2), (3, 5)\}$

Common mistake:- In general the Cartesian product operator is not commutative. That means $B \times A \neq A \times B$. Why? Because the order matters!

Note that the Cartesian product produces another **set**. But the elements of the set are pairs.

If you have seen coordinate geometry then it should be easy to see that coordinates of the form (x, y) are just elements of the set $\mathbb{R}^2$.

Generalized Cartesian products

The Cartesian product can be applied to more than two sets as well. Consider a sequence of n sets, $A_1, A_2, \ldots, A_n$. The elements of the Cartesian product of $A_1, A_2, \ldots, A_n$ are ordered sequences of n entries, called n-tuples. The ith entry in each n-tuple is an element of A_i. The Cartesian product of $A_1, A_2, \ldots, A_n$, denoted $A_1 \times A_2 \times \ldots \times A_n$, is defined to be:

$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, a_2, \ldots, a_n) : a_i \in A_i \text{ for all integers } i \text{ such that } 1 \leq i \leq n\}$

In general, the idea of a Cartesian product is extremely useful when you want to refer to ordered groupings. If you have to make a sandwich where you first pick bread from a set like $B = \{\text{white}, \text{wheat}\}$ and pick cheese from a set like $C = \{\text{edam}, \text{camembert}\}$ then every sandwich can be expressed as an ordered pair of the form (bread, cheese). So how many possible sandwiches are there?

The number of sandwiches is the same as $|B \times C| = 2 \times 2 = 4$.

Notation:- As with 'usual' math, $P \times P \times P$ is often denoted by P^3.

If $|A| = 2$ and $|B| = 4$, how many elements does $A \times B$ have?

The Cartesian product will pair each element with A with each element of B. For the first element of A, it gets paired with each one of the 4 elements of B. For the second element of A, it gets paired with each one of the 4 elements of B. So it is $2 \times 4 = 8$.

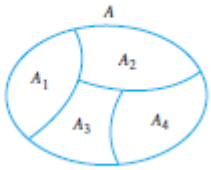
In general, if you take the cross product of a set with cardinality m and a set with cardinality n, you get a set of cardinality mn.

Partitioning a set

A set A is set to be partitioned by a family of subsets of A, $A_1, A_2, \ldots, A_n$ if the following are satisfied

$A = \bigcup_{i=1}^{n} A_i$ and

$A_i \cap A_j = \emptyset$ for every $i \neq j$. This condition is also called being mutually disjoint.



Why is this useful?

If you know that a set A is partitioned by $A_i, 1 \leq i \leq n$, then we can easily deduce that

$|A| = \sum_{i=1}^{n} |A_i|$.

Example

Let $A = \{1, 2, 3, 4, 5, 6\}$ and $A_1 = \{1, 2\}, A_2 = \{3, 4\}$ and $A_3 = \{5, 6\}$. Is $\{A_1, A_2, A_3\}$ a partition of A.

Yes.

Is $\{\{a, d, e\}, \{b, c\}, \{d, f\}\}$ a partition of $\{a, b, c, d, e, f\}$?

No. d is in two of the sets.

Application of Cartesian products

Databases - The JOIN syntax in databases is fundamentally using Cartesian products.

Example taken from http://en.wikipedia.org/wiki/Join_(SQL)

Application of partitions

One of the first steps in designing a parallel program is to break the problem into discrete "chunks" of work that can be distributed to multiple tasks. This is known as decomposition or partitioning.

Often useful for counting things as we will see soon.

Functions

Definition of a relation

A relation between set A and set B is a subset of $A \times B$. While for the purposes of pure math there does not need to be any underlying property that governs the relation, for most practical purposes, you will find that there will be some property.

For example: Consider the set $A = \{1, 2, 3\}$ and $B = \{4, 5, 6\}$ and let us define the relation R to consist of tuples $\{(a, b) | a \in A, b \in B \text{ and } b = a + 3\}$. Then the relation $R = \{(1, 4), (2, 5), (3, 6)\}$.

If $(a, b) \in R$, then this is very often denoted as aRb. That notation is similar to the way we write out relations like greater than, less than etc.

Properties

- A relation R on set A is reflexive if **for all** elements a in set A, aRa.
- A relation R from set A to set B is said to be symmetric if aRb means that bRa (implicitly we are assuming $a \in A$, $b \in B$).
- A relation R on a set A is said to be transitive if whenever aRb and bRc then aRc.
- A relation R is said to be an equivalence relation if and only if it is reflexive, symmetric and transitive.
- A relation R is said to be anti-symmetric if aRb and bRa can only happen when $a = b$.

  Alternatively we can define anti-symmetric as $x \neq y$ should imply that either x is not related to y or y is not related to x.

Examples

1. Given the set $A = \{1, 2, 4\}$, define the relation R to be the $\leq$ relation. That is aRb whenever $a \leq b$.

   Is this reflexive? Yes. For $a \leq a$ is always true. In particular, it is true for elements of A.

   Is this symmetric? No. For instance $1 \leq 2$ but it is not the case that $2 \leq 1$.

Is this transitive? Yes. If $a \leq b$ and $b \leq c$ then it is clearly true that $a \leq c$.

Is this anti-symmetric? Yes. If $a \leq b$ and $b \leq a$ then it must be the case that $a = b$.

2. Given the set U = the set of all the states in the USA.

   Let us define a relation as follows. Two states are related if and only if they are adjacent (they share a land or water border).

   Is this reflexive? That is an interesting question because it really depends on the answer to a question like 'does Pennsylvania border Pennsylvania'. And really that argument can go both ways. This is a classic case of English and Math not being totally in sync. It is also a reason why after a while, we have to resort to more mathematical examples!

   Is this symmetric? This is clear. YES.

   Is this transitive? No. While we have not covered the concept of a mathematical proof yet, it is important to learn that in order to disprove a statement that says 'for every ....', all you need to demonstrate is one counter example. Think of it as the one rotten apple!

   Where does this fall apart? Virginia borders North Carolina. North Carolina borders South Carolina. But VA does not border SC.

   Is this anti-symmetric? No. The same example as above works as a counter example.

Is it possible for a relation to be symmetric and anti-symmetric at the same time?
This will lead to the concept of a vacuously true statement.
Consider the set $\{1\}$ and the relation $\{(1, 1)\}$. While seemingly trivial and stupid, this relation turns out to be both symmetric and anti-symmetric.

Definition of a function

A function f from a set X to a set Y, denoted $f : X \to Y$ is a relation from X, the domain to Y the co-domain that satisfies two properties

1. every element of X is related to some element in Y
2. no element of X is related to more than one element in Y

The zybook refers to the co-domain as target.
The set of all values of f taken together - range.
You are probably used to functions in calculus. The functions we do in this course are exactly the same (yay! math definitions do not just change randomly). The big advantage is we can draw what is generally referred as arrow diagrams since the sets we deal with are discrete.

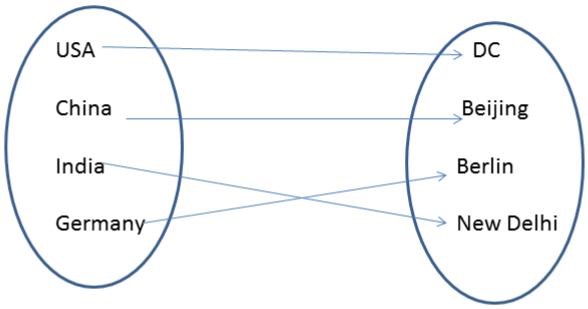
Note:- A function definition is incomplete unless the domain of the function is specified. It is also generally a good idea specify the co-domain, but if not specified it is just considered to be the range of all possible values that the function takes.

Note: - Every function is a relation but only some relations are functions. In particular it is easy to define the relation $R = \{(x, f(x)) \text{ where } x \in A\}$.

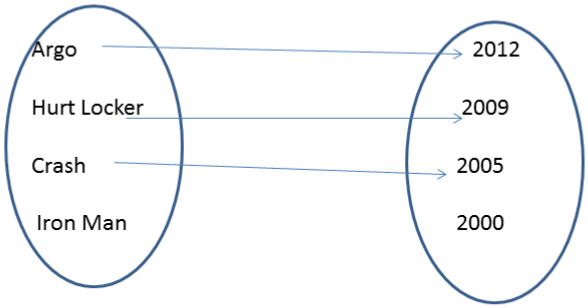
Here is an example of a relation which is not a function $R = \{(1,1),(1,3)\}$. Why is this not a function? Because 1 is being related to both 1 and 3. To be a function 1 can be mapped to only a single thing!

Examples

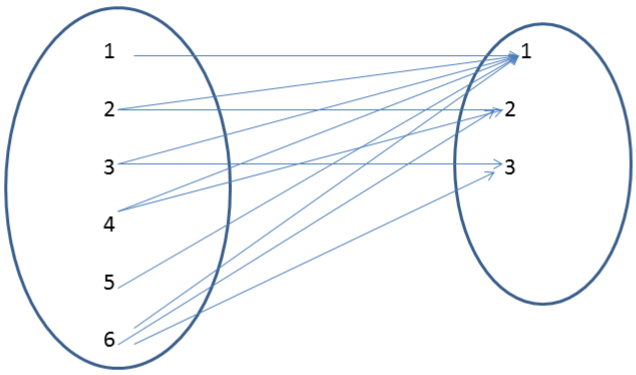
Here are a few examples of mappings between 2 sets that either are functions or are not functions depending upon the



In this first example we are mapping between countries in the left side to the capitals of countries on the right. This is a function because every country has a capital and every country (in this diagram!) has only 1 capital. Trivia: - Name a country that has more than 1 capital.

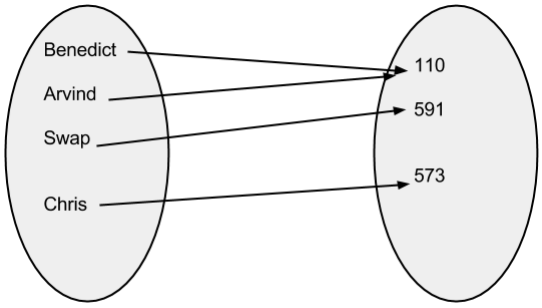


In this second example we are mapping between movies in the left side to years in which they won the best picture award at the oscars. No movie wins the best picture award for more than 1 year, so that condition is satisfied. But Iron Man did not win the best picture award ever, and hence this is not a function.



In this third example we have integers from 1 to 6 on the left side and integers from 1 to 3 on the right side. The integers in the left set are mapped to integers on the right side if the right integer is a factor of the left integer. Remember that b is a factor of a if b divides a. Also recall that 1 is a factor for every integer.

This example fails to be a function because a number of elements in the left set are being mapped to more than 1 element on the right.



In this example although both Benedict and Arvind are teaching CIS110, it is still a function! Note that no where in the definition of a function does it say that each element of the first set must be mapped to a different element of the second set.

Two functions $F : X \to Y$ and $G : X \to Y$ are said to be equal iff $F(x) = G(x)$ for every $x \in X$.

Ceiling and Floor

There are a few functions which show up a lot in Computer Science, especially once you start analyzing algorithms. The extremely common one is the log function, which you might have seen before but we will cover in a little bit more detail once we get to analyzing algorithms. Two others which are used extensively to approximate things are the ceiling and the floor function.

The ceiling function, denoted as $\lceil x \rceil$, is the smallest integer y such that $y \geq x$.

The floor function, denoted as $\lfloor x \rfloor$ is the greatest integer y such that $y \leq x$.

Is the floor function defined on $\mathbb{R}$ one-one? When a statement like the above one is made, the implicit assumption is that both the domain and the co-domain are $\mathbb{R}$.

Consider these two $\lfloor \pi \rfloor$ and $\lfloor 3.5 \rfloor$. They are both 3! So here we have two distinct element of $\mathbb{R}$ mapping to the same element. The floor function is not one-one on $\mathbb{R}$.

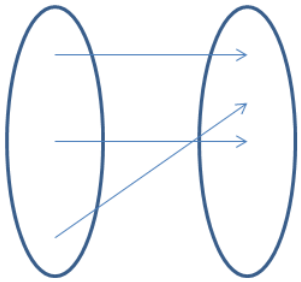
Is the floor function defined on $\mathbb{Z}$ one-one? This function IS one-one because what is the floor of any integer? It is the integer itself! So you will not have the previous situation.

One-One

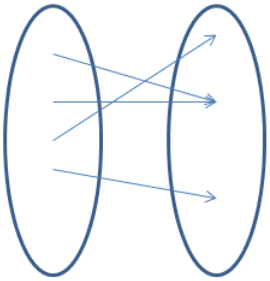
A function is said to be one to one if $x \neq y$ means $f(x) \neq f(y)$. Equivalently, $f(x) = f(y)$ means $x = y$.

Of the two functions above, which is one to one.

From an arrow-diagram identifying whether a function is one-one is immediate. You just have to make sure that you do not have two arrow heads incident on the same point



One - one



Not
one – one

Example

Is the function $f(x) = x^2$ one-one when the domain and co-domain are the set of real numbers?

Not one-one. For example $f(1) = f(-1)$, but obviously 1 is not equal to -1.

Is the same function one-one when the domain and co-domain are the set of natural numbers?

Now the function is one-one because

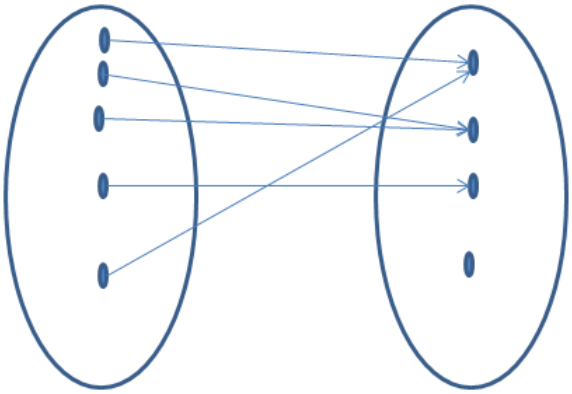
$$\begin{aligned} x^2 = y^2 & \\ \implies & x^2 - y^2 = 0 \\ \implies & (x-y)(x+y) = 0 \\ \implies & x = y \text{ or } x = -y \end{aligned}$$

But both x and y are natural numbers and therefore are positive so it must be the case that $x = y$. Therefore the function is one to one.

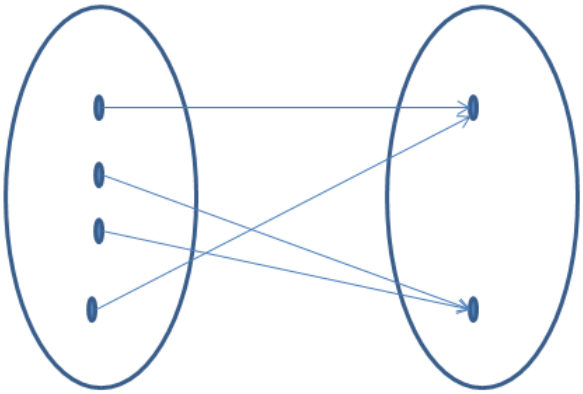
Onto

A function is said to be onto if the entire co-domain is 'covered'. More formally, $f : X \to Y$ will be onto if for every $y \in Y$, there is some $x \in X$ such that $f(x) = y$.

From an arrow-diagram perspective it just means that there is some arrow incident on every single point in the co-domain.



Not onto
(also not one – one)



Onto
(this is not one-one though)

For instance, the function $f : \mathbb{Z} \to \mathbb{Z}$ defined as $f(x) = x^2$ is not onto. Why?
Is there any integer whose square is equal to 2?

Bijection

A function that is both one to one and onto.

Examples-

Let $F = \{`Arvind', `Nitay', `Olivia'\}$ and
Let $L = \{`Bhusnurmath', `Caspi', `Sun'\}$.

Every first name can be mapped to a unique last name. Also there is no last name that nothing is mapped to.

If $f : X \to Y$ is a bijection from X to Y, what can you say about $|X|$ and $|Y|$.
$|X| = |Y|$.

Do not worry about a rigorous proof of that statement yet. Just understand it intuitively as follows.

Bijection means both onto and one-one. As a consequence of being onto, every element of Y has some arrow incident on it. This immediately means $|X| \geq |Y|$ (f is a function so two arrows cannot come out of the same point in X).

As a consequence of being one-one, every element of X has to shoot an arrow to a different element of Y. That immediately means $|Y| \geq |X|$.

Put everything together and you get $|X| = |Y|$.

Application of bijections

If you have a bijection between 2 sets you can use elements of one set to exactly represent the other.

If you have seen binary numbers, you know that every positive integer can be expressed in binary, which can basically be considered as a string of 0s and 1s.

Encoding and decoding are also examples of bijections. You can think of an encoding as the application of a function to the message in order to produce a message that is hard to decipher. Given a secret message, you would want your decoding to be unique otherwise the ambiguity would be annoying.

Basics of Counting

Notes.
Main concepts

- Sum rule
- Product rule
- Bijection principle

Product rule

Often when counting something, it is useful to think of what we are counting as a selection from one set **AND** then a selection from another set. In such situations the product rule is useful.

The product rule stems from the concept of Cartesian products. $|A \times B| = |A||B|$.

Example, a customer goes to Dunkin Donuts and orders a small coffee. They are then asked if they want cream and sugar. How many different types of coffee are possible.

A simple way of dealing with this problem is to say that the customer has to first choose from a set that has the elements $\{cream, nocream\}$ and then from another set that has the elements $\{sugar, nosugar\}$.

So there are 4 possibilities.

Important clarification - We are not distinguishing between option1 -adding cream first and then adding sugar) and option2 - (adding sugar first and then adding cream). That, hopefully, is a reasonable assumption!

In general, if you are choosing from a set A_1, then choosing from a set A_2, then from a set A_3 and so on until A_n, the total number of choices you have is $|A_1| \cdot |A_2| \cdots |A_n|$. But, you do need to be careful that the number of choices you have in subsequent steps does not depend on things you do in the previous steps.

Another way this gets stated (a more formal way!) is

If an operation has k steps and

the first step can be performed in n_1 ways

the second step can be performed in n_2 ways (regardless of how the first step was performed)

the kth step can be performed in n_k (regardless of how the previous steps were performed)

then the entire operation can be performed in $n_1 n_2 \cdots n_k$ ways

Example - How many possible 3 digit numbers are there? The first digit cannot be 0 so 9 possibilities. The next two digits can be anything. Hence - 9 * 10 * 10 = 900. Also easily done as we have to count all the numbers from 100 to 999, both inclusive. 999-100+1 elements.

Fun example - How many possible drinks can you get at Starbucks? We'll figure out the answer in class.

Sum rule

If instead of making choices one after the other, you are counting something where you choose from one set OR from another set, then the sum rule is applied.

The keyword to look out for over here is **OR**.

A database has two tables. One of them has information about faculty - F, the other has information about administrative staff - S. The payroll program is trying to figure out how many people to pay. Can it add the number of people in faculty and the number of people in staff? Yes - because these two sets are disjoint.

Consider n sets, A_1, A_2, ..., A_n. If the sets are mutually disjoint, $A_i \cap A_j = \emptyset$, then $|A_1 \cup A_2 \cup \ldots \cup A_n| = |A_1| + |A_2| + \ldots + |A_n|$.

Consider two sets U - united states citizens. C - canadian citizens. If I want to count the number of people who are either United States citizens OR Canadian citizens, can we do it just by saying $|U| + |C|$. Why or why not?

What about her? - http://en.wikipedia.org/wiki/Alanis_Morissette.

When applying the sum rule, be careful that your sets truly are disjoint.

Applying the sum and product rule together

How many passwords can be made using just upper and lower case letters that are between 4 and 8 characters long?

Let us use some notation first

P_4 - passwords of length 4.

P_5 - passwords of length 5.

P_6 - passwords of length 6

and so on.

Then consider $P = P_4 \cup P_5 \cup P_6 \cup P_7 \cup P_8$. What we want to find is $|P|$. The sum rule can be applied since you cannot possibly be of length 5 and of length 8 for instance. The sets are mutually disjoint (nothing in common between any pairs of them!).

So $|P| = |P_4| + |P_5| + \ldots + |P_8|$. Now what is $|P_4|$?

Assume A is the set of all the upper case and lower case alphabet in English. $|A| = 26 \cdot 2 = 52$. For a 4 letter password, we can think of it as a 4-tuple. In particular $|P_4| = |A \times A \times A \times A|$. That means $|P_4| = 52^4$.

That logic extends to every P_i, so the total number of passwords becomes

$|P| = 52^4 + 52^5 + 52^6 + 52^7 + 52^8$.

Tricky Example (can be found in zybook as well)

Three officers - a president, a treasurer and a secretary - are to be chosen from among four people. A, B, C and D. Suppose for whatever reason A cannot be president and either C or D must be secretary. How many ways can these officers be chosen.

This question is tricky because the order of our choosing operations becomes more critical.

If we pick the president, then pick the treasurer and then pick the secretary, we run into an issue because the number of choices for secretary depends on who was picked for president and treasurer. If C or D were picked in either of the first two operations, the number of choices becomes less than 2. If we want to do this via the product principle we need to realize that we just need to pick these 3 different officials. We can reorder the order of choosing the officials so that it looks more like a situation where the product principle can be applied.

Reorder to do the choosing in the following manner - pick secretary first, then pick the president, then pick the treasurer. Now there are 2 people that can be chosen from for the secretary (C and D). Regardless of who gets chosen, when it comes to picking the president next, we realize that there are only 2 choices. B and whoever was not picked out of C and D. Also regardless of who we pick as president, there are 2 choices for the treasurer - A and whoever was not picked as either the president or the secretary.

So it is $2 \times 2 \times 2 = 8$

It is also interesting to try and do this same question in a different manner

1. Pick a secretary - you can do this in 2 ways
2. Pick a treasurer - You cannot pick the person who was picked as secretary, so this can be done in 3 ways.
3. Pick a president - But this now depends on who was picked as treasurer. If A gets picked as treasurer we have 2 choices. If anyone else gets picked as treasurer, we have only 1 choice since A cannot be president. Since we are depending upon the result of the previous step, we cannot use the product principle.

However since we are making a choice in the second stage we can look at it in this manner. Either we pick A or do not. Since the set of choices that include A are obviously disjoint from the choices that do not include A, we can use the sum rule and break this up into 2 case

- Number of ways to choose the officers, if A is to be picked - 2 ways to pick the secretary, A is the treasurer and 2 ways to pick the president. So 4 ways.
- Number of ways to choose the officers, if A is not to be picked - Still 2 ways to pick the secretary, 2 ways to pick the treasurer (either it will be B or the person who was

not picked to be the secretary) and now there is only 1 way to pick the president since A is not being picked at all. So 4 ways again.

So, the number of ways in total just becomes $4 + 4 = 8$ ways.

Example

How many different 4-letter radio station call letters can be made if the first letter can be a K or a W and no letters can be repeated. A radio station call letter is something like WKYP or KXPN etc.

Either the radio station begins with a
K - $25 \cdot 24 \cdot 23$ possibilities (cannot repeat the K hence begins with 25).
OR
it begins with a W - $25 \cdot 24 \cdot 23$ possibilities
So a total of $2 \cdot 25 \cdot 24 \cdot 23$ possibilities.

Example A wedding photographer wants to take a bunch of pictures of Ann and Bob's wedding. There are 10 people in total (including Ann and Bob) but the photographer can only fit 6 of them into a picture at any given time. How many different pictures are there if Ann must be in the picture? Remember that pictures are considered different if different people are placed in different locations.

One way of doing this is to split it up into 6 possibilities
Ann is the first position, Ann is in the second position, and so on. These are mutually exclusive.

When Ann is in the first position the second position can be filled with 9 possibilities, the third with 8 possibilities etc. So with Ann in the first position we have $9 \times 8 \times 7 \times 6 \times 5$.

For all the other possibilities for Ann, the number remains the same. There are 6 possible places Ann could be.

Therefore the total number of photographs are $6 \times 9 \times 8 \times 7 \times 6 \times 5$

Example -
How many functions are there from a 3 element set to a 4 element set? How many of them are one to one?

Bijection Principle

A set X and a set Y have the same number of elements if and only if, there is some bijection between the two sets.

Remember that a bijection means one to one and onto.

This principle is surprisingly useful at times when it is tough to count the elements of a set directly, but the elements of the set can be mapped to an easier set.

Example -

30 teams qualify for a new form of the World Cup where every single game is a knockout. If you lose, you are out. How many games need to be played before we declare a winner?

The initial and obvious approach would be to start constructing a format of the World Cup where in the first stage, you have 15 matches and then drawing out a full bracket (or tree if you prefer). The problem is once the first stage is done, we have 15 winners. How exactly do we play them. Does it matter which order the knockouts are held in?

Instead of counting this by looking at the matches, let us see if there is any interesting mapping that can be done. In every game, there is a winner and a loser. So let us map the set of games to the set of losers. Clearly, each game can be identified uniquely by its loser. The game to loser mapping is one-one and onto and therefore a bijection.

Since we have a bijection, we know that the number of losers is the same as the number of games. So how many losers are there? There are 29 losers! Therefore there are 29 games.

This result should feel a little surprising because regardless of how you arrange the matches, the counting is done so very simply.

Applications

So why is it important to know how to count?

1. Discrete probability, which we will cover in this course, is essentially an exercise in counting twice. Count all possible scenarios and then count the specific event you are interested in. Divide the two and get the probability.

2. There are lots of applications of computer science (or programming if you will) to sequencing problems. One of the more famous ones is the Traveling Salesperson Problem. Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city.

   Every possible solution can be mapped to a sequence of cities. But how many such sequences are there? That is a counting problem. For n cities, you have n choices first, then $n-1$, then $n-2$ and so on. This quantity $n \times (n-1) \times (n-2) \cdots 1$ is called n factorial and represented as $n!$.

   We will talk more about factorials and arrangements in upcoming lectures.

Permutations and Combinations

Notes.
Main concepts

- Order matters - permutation
- Choosing - combinations
- More applications

Permutations

Permutations occur when counting sequences without repetitions.

If Ash, Bo, and Carmen have to be lined up in a straight line the number of ways in which this can be done can be calculated using the product rule.

The first position can be filled any of the 3. The second position can be filled by any of the remaining 2. Finally there is only choice for the final position. Therefore this can be done in $3 \cdot 2 \cdot 1 = 6$

In general, n items can be arranged in an n sequence in $n! = n \times (n-1) \times (n-2) \times 1$ ways.

That n with an exclamation point is the notation for what is called n factorial.

What if instead of using all the elements in a set, we just want to pick r of them and arrange them in a straight line?

The product principle can still be applied

- pick the first - n choices
- pick the second - $n-1$ choices
- pick the third - $n-2$ choices
- pick the r^{th} - $n-r+1$ choices

Putting that all together you get what is called the number of r-permutations over a set with n elements which is often denoted as $P(n,r)$. An r-permutation is a sequence of r items with no repetitions, all taken from the same set.

$P(n,r) = n \cdot (n-1) \cdots (n-r+1) = \frac{n!}{(n-r)!}$

Examples - Permutations

How many functions are there from $\{1, 2, 3\}$ to $\{a, b\}$?

The key here is recognizing that each function can be identified as an ordered sequence of 3 with each place being occupied by an a or a b. Why? Something (a,a,b) would just mean that 1 is being mapped to a, 2 is being mapped to a and so on. So the total is $2 \times 2 \times 2 = 8$.

It is important to observe that in this case, since repetition is allowed in the sequence, you actually do not use the r-permutation formula.

Example There are 8 people who make it to the 100 m final at the Olympics. In how many ways can the gold, silver and bronze be handed out?

This is the same as the number of 3-permutations in a set of 8 since we have to assume that any of the 8 people can win a medal. Also, the obvious constraint here is that a person who manages to win the gold medal cannot also at the same time manage to win silver or bronze.

So $8 \cdot 7 \cdot 6$ ways.

Example A wedding party consists of 3 groomsmen, 3 bridesmaids and of course the bride and groom. The photographer wants to take pictures by arranging people on a bench. The bridesmaids have to be together, the groomsmen have to be together and the newly married couple would be together as well. How many different photographs are possible.

There are basically 3 distinct groups here. Consider the groomsmen as one unit (tie them up!), bridesmaids as one unit and the couple as one unit. These 3 units can be arranged in 3! ways. Now within the groups, you can arrange the bridesmaids in 3! ways. The groomsmen in 3! ways and finally the couple can be arranged in 2 ways. Therefore, 3!3!3!2! = 432 pictures.

Example

In how many ways can the letters of the word 'LEADING' be arranged so that the vowels are always together.

In a manner similar to the example above, we tie the vowels together. That means we have the characters L, D, N, G and then the block of A,E,I. So 5 different things to be arranged first. That can be done in 5! ways.

But that is not all. We can 'zoom' in to the block of vowels and then rearrange those. 3 of those. They can be arranged in 3! ways.

So overall, you have 5!3! ways or arranging these letters as per the constraints.

Combinations - Choice without order mattering

Assume someone tells you that Christopher Nolan is a great director. You want to watch some of his movies but you really do not have the time to watch them all. You just want to watch 3 of them before you make your decision whether he is good or not. The question is in how many ways can you make your choice.

Here are his movies - Following, Memento, Insomnia, Batman Begins, The Prestige, The Dark Knight, Inception, The Dark Knight Rises, Interstellar. 9 movies in total.

Now let's first look at this as a sequencing/permutation question and say we have 3 blanks to fill

_____ _____ _____

The first blank can be filled in 9 ways, the second in 8 ways, the third in 7 ways (you're not going to repeat a movie). So your initial idea might be to say $9 \cdot 8 \cdot 7$.

But the order does not matter! You are just making a choice of 3 movies to watch. It does not matter if you watch Memento, Insomnia, Dark Knight in that order or Insomnia, Dark Knight, Memento in that order. If someone was to ask you which 3 movies you watched, in both cases, you would just say 'I watched Insomnia, Memento and Dark Knight'.

So this means we are over counting in our above answer. How much are we over counting by??

For each choice of 3 movies, there are 3! ways in which you can order them.

For the choice mentioned above for instance, you could arrange them as

(Memento, Insomnia, Dark Knight) or (Memento, Dark Knight, Insomnia) or (Dark Knight, Insomnia, Memento) or (Dark Knight, Memento, Insomnia) or (Insomia, Dark Knight, Memento) or (Insomnia, Memento, Dark Knight).

That is 6 ways.

So for each choice, we have 6 ways to do the arrangements. That means that

$$\text{Number of choices} \times 6 = 9 \cdot 8 \cdot 7$$
$$\text{Number of choices} = \frac{9 \cdot 8 \cdot 7}{6}$$

This is basically what is called a combination problem or a problem that involves counting the number of choices.

There is a formula and terminology associated with this. We write $\binom{n}{r}$ and say 'n choose r' and this denotes the number of ways in which you can choose a set of r items from a set of n items.

$$\binom{n}{r} = \frac{n!}{r!(n-r)!}$$

Now that you have seen this notation and formula, you can answer the previous question even more directly by saying

The number of choices is the same as picking a set of 3 from a set of 9. That is $\binom{9}{3}$.

The k to 1 rule

Let $f : X \to Y$ be a function with the property that for every $y \in Y$, there are exactly k different $x \in X$ such that $f(x) = y$

Then to count the number of elements in X, you can simply say $|Y| = \frac{|X|}{k}$.

Example - How many binary strings of length 16 have exactly 4 1s.

Think of this question in the following manner. Once you know which of the 16 spots contain the 1s, you are done!

In how many ways can 4 spots be picked for the 1s. Out of the 16 possible spots, you have to pick 4.

That is $\binom{16}{4}$.

Here is another way of looking at it. Let us define a mapping from the set of 16 bit strings to a subset of the set $\{1, 2, \ldots 16\}$ of size 4 in the following manner. Map the string to the locations where a 1 can be found.

0110100010000000 gets mapped to $\{2, 3, 5, 9\}$.

0000011001010000 gets mapped $\{6, 7, 9, 11\}$.

It is clear that given a string you can uniquely map it to a 4 element set.

It is also clear that given a 4 element subset, you can form a 16 bit string. Just put the 1s in the locations provided by the 4 element subset.

So this is a bijection between 16 bit strings and 4 element subsets. Therefore the number of 16 bit strings with 4 1s is the same as the number of 4 element subsets of the set $\{1, 2, \ldots 16\}$.

How many 4 element subsets are there? Again, that is $\binom{16}{4}$.

Tricky Combinatorics

Notes.
Main concepts

- Arrangements when some objects are alike
- Counting integer solutions

Arrangements of non-distinct

- Example - How many ways can the letters of the word WILL be rearranged?

If this question was posed instead as in how many ways can the letters of the word WILD be rearranged, then it becomes quite simple. After all, in that case, it is the arrangement of 4 distinct objects and by this stage you know the answer to that is just 4!

So let us just consider one of the Ls to be in lowercase. Then again, the answer will be 4! But now we are overcounting. We are overcounting because it doesn't matter if the L is in uppercase or lowercase, that was just a step we made in order to make our counting easier.

something like lWIL is the same as LWIl when we do the counting. This is same as as 2 of these map to 1 word.

So if we use the k to 1 concept, all we need to do is $\frac{4!}{2}$.

What is generalization? If instead of 2 Ls we had a word with 3, then think of the 3 Ls as being interchangeable in any word. So any of the 3! arrangements would map to the same word!

Theorem 1. *If there are k objects in total. k_1 of them are of type 1, k_2 are of type 2 and so on until k_n are of type n, then the total number of possible arrangements of these objects in a straight line are*

$$\frac{k!}{k_1!k_2!\cdots k_n!}$$

Quick example

In how many ways can the letters of the word PHILIPPINES be rearranged?

There are PHI LIP PIN ES (Spaces just added to make counting easy) 11 characters in total. 3 Ps and 3 Is.

So using the previous result $\frac{11!}{3!3!}$

Counting integer solutions to certain types of equations

A very popular and powerful combinatorics problem is to count the number of non-negative integer solutions to certain types of equations.

For instance, how many non-negative integer solutions can be found for the following equation

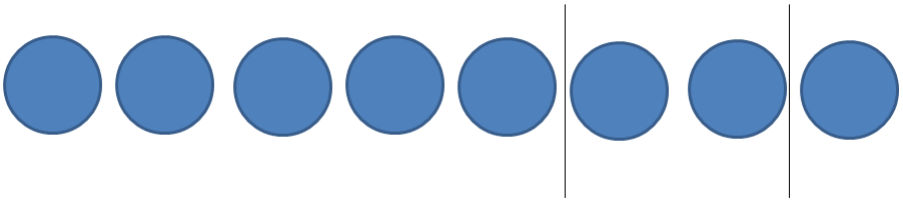
$x_1 + x_2 + x_3 = 8$

We want each of the x_is to take on a non-negative value.

At least initially, this question looks quite different from anything related to the arrangement of objects. But there is a small trick that reduces it to one of the problems seen before.

Consider the following mapping between solutions of the equations and arrangements of identical balls.

Given a solution of the equation like 5, 2, 1 ($x_1 = 5$, $x_2 = 2$ and $x_3 = 1$) we can map it to



That is an arrangement of identical balls with 2 separators placed in there to distinguish between which set corresponds to x_1, which to x_2 and the rest correspond to x_3.

So our mapping is defined now between the set of non-negative solutions to the equation and the set of arrangements of 8 identical balls and 2 identical separators (while the separators can be placed at different locations, they look exactly the same).

Is this mapping a bijection?

Given a solution to the equation, we have demonstrated how to construct an arrangement - put down as many balls as the value of x_1, then put down a separator. Then put down as many balls as the value of x_2. Then put down another separator. Then put down the remaining balls.

Similarly given any arrangement of balls and separators, to get the value of x_1, count the number of balls to the left of the first separator. That is x_1. Then count the number of balls between the first two separators. That is the value of x_2. Finally, count the number of balls to the right of the last separator. That is the value of x_3.

Since we have demonstrated how to go from one set to the other in a unique manner (yes this is a hand wavy proof, but having not done mathematical proof yet, we are ok with this), we have a bijection.

So how does one count the number of arrangments of identical balls and separators?

There are 8 balls and 2 separators. That is 10 objects, 8 of 1 kind, 2 of another kind.

$\frac{10!}{8!2!}$

You could also think of it as 10 blank spots to be filled. We have to choose 2 spots to place the separators. That is $\binom{10}{2}$.

Counting multisets

Typically one thinks of sets as collections of distinct items. However, sets can also be defined in a way that allows them to have multiple instances of the same kind of item. Such sets are called multisets. Multiset is another way of saying a set that repetitions are allowed.

Given a set $\{1, 2, 3, 4\}$, how many multisets can be formed from it?

The absurdity of the question should hopefully strike you readily. If you are allowed to repeat elements then surely $\{1, 1, 1\}$ is a multiset. So too $\{1, 1, 2, 3, 4, 4\}$. There are infinitely many.

But what if we add a constraint on the number of elements. How many 4 element multisets are there?

This again might initially seem to be unrelated to other problems but what if we did the following. Let x_1 represent the number of copies of 1 in our multiset. Let x_2 represent the number of copies of 2 and so on.

Then we want $x_1 + x_2 + x_3 + x_4 = 4$. And we know how to solve those now! 4 balls, 3 separators. That means $\frac{7!}{4!3!}$.

Examples dealing with constraints

This problem shows up in many different forms. I will just refer to it as the girls and boys counting problem. Suppose you have 5 girls and 3 boys. You want to arrange them (in a line) in such a way that you do not have 2 boys standing next to each other.

The classic way of solving this problem is to first arrange the girls - 5! ways.

Now think of any girl arrangement as G G G G G, that is with these gaps in between them.

Where do the boys go? In the gaps! How many gaps are there? 6 gaps (remember the gaps at the start and end of the line).

The boys choose the gaps in $\binom{6}{3}$ and then can be rearranged in 3! ways. Alternatively, we can do it directly by saying in how many ways can a set of 3 (boys) be mapped to a set of 6 (gaps).

Putting it all together we get $5! \cdot 6 \cdot 5 \cdot 4$.

Basic Probability

What is an event

An experiment is a procedure that results in one out of a number of possible outcomes. The set of all possible outcomes is called the sample space of the experiment. A subset of the sample space is called an event.

Examples are

- Consider a simple example of an experiment in which a red and a blue die are thrown. We will denote a single outcome by an ordered pair where the first number denotes the outcome on the blue one and second number denotes the outcome on the red one. Then the sample space is $S = \{(x, y) | 1 \leq x \leq 6, 1 \leq y \leq 6\,, x \in \mathbb{Z}, y \in \mathbb{Z}\}$. Any subset of the sample space is called an event. So for instance the event E or having doubles show up on the dice can be written explicitly as

  $E = \{(1,1), (2,2), (3,3), (4,4), (5,5), (6,6)\}$.

- Another common experiment is playing cards. Let us say we are playing poker and we get a 5 card hand. Obviously there are several possible outcomes. Every single outcome is one of the 5 element subsets of the set of 52 cards.

  The sample space therefore is of size $\binom{52}{5}$.

  A particular event in this case for instance is a royal flush, which means A, K, Q, J and 10, all of the same suit. Let's call that event

$$R = \{\{A\clubsuit, K\clubsuit, Q\clubsuit, J\clubsuit, 10\clubsuit\}, \{A\spadesuit, K\spadesuit, Q\spadesuit, J\spadesuit, 10\spadesuit\}, \{A\heartsuit, K\heartsuit, Q\heartsuit, J\heartsuit, 10\heartsuit\}, \{A\diamondsuit, K\diamondsuit, Q\diamondsuit, J\diamondsuit, 10\diamondsuit\}\}$$

  There are 4 such royal flushes. Therefore the probability of getting one of these becomes

$$\frac{4}{\binom{52}{5}}$$

Basic probability definition

A natural question with regards to an experiment is to say, what is the likelihood (the probability) that a particular event occurs.

A probability distribution over an experiment with a sample space of S is a function p from S to the set $[0, 1]$.

Certain properties need to be satisfied by this probability distribution.

$$\sum_{s \in S} p(s) = 1$$

The probability of the single outcome s is $p(s)$. If $E \subseteq S$ is an event, then the probability of the event E is

$$p(E) = \sum_{s \in E} p(s)$$

This definition is especially useful when dealing with situations where not every outcome is equally likely.

Loaded dice for instance. (More detail in the zybook)

Let's say a die is twice as likely to show up 4 as anything else. To make this work, we basically want to assign a value of $2/7$ to $p(4)$ and a value of $1/7$ to the probability of every other outcome.

Given these individual outcome probabilities, you can now work out the probabilities of any event

For example the event that the die shows a number greater than 3 becomes p(4) + p(5) + p (6) which is 4/7.

Uniform distribution

In many scenarios, the probability of every outcome in the sample space is the same. The probability distribution in which every outcome has the same probability is called the uniform distribution.

Since there are $|S|$ outcomes and their probabilities sum to 1, under the uniform distribution, for each $s \in S$, $p(s) = 1/|S|$. The uniform distribution reduces questions about probabilities to questions about counting because for every event E,

$$p(E) = \frac{|E|}{|S|}$$

Poker probabilities

A great review of both probability and permutations and combinations can be achieved via this wikipedia entry that has the probabilities of various poker hands

https://en.wikipedia.org/wiki/Poker_probability

A zybook example

Consider a situation in which files are stored on a distributed network. Multiple copies of each files are stored around the network so that if one or more computers crash, the data is more likely to be available from at least one source. Suppose that three copies of a files are stored at different locations in a network of 30 computers and that at a particular moment, five random computers fail. Each subset of 5 computers are equally likely to be the five that have failed. What is the probability that there are no copies left of the file?

The experiment over here is choosing 5 computers that fail. So total number of distinct outcomes is $\binom{30}{5}$. Every computer is equally likely to fail. That means every subset of 5 is equally likely to fail.

How many of these subsets have all 3 of the computers that contain this file? The file is in 3 of the computers, so we have to have all 3 of these computers being picked. But we still have a choice for the remaining 2 computers. 27 other computers exist. 2 of them have to be picked.

therefore $\frac{\binom{27}{2}}{\binom{30}{5}}$

Birthdays! - Complementary probability

One of the cool applications of Probability is in the computation of how many people you need to have in a group before at least 2 of them have the same birthday.

Let's keep things simple and say that the year has 365 days (which works for most applications).

Now obviously, if you have 366 people, the probability that at least 2 of them share the same birthday is 1. That sort of argument is what is generally referred to as the Pigeon hole principle (something that we do not cover much in this course).

Let's say you have m people. Now the number of ways in which those m people get birthdays assigned to them is - the first person has 365 choices, so does the second, so does the third and so on. So if we wrote out the sample space it would look like

$\{(b_1, b_2, \ldots, b_m) | 1 \leq b_i \leq 365\}$.

We know how to count that. Product rule or the idea of Cartesian products. So the cardinality of this set is 365^m.

How do we count the number of cases where at least 2 of these m people have the same birthday? One way of doing it would be to say let's count the cases for exactly 2 having the

same birthday then the cases for exactly 3 having the same birthday, and so on. But if m is large, this is just going to be a huge pain!

Instead, we just look at what is called the complementary event and use this important idea

$P(A^c) = 1 - P(A)$

because after all, every outcome is either part of the event you want or not. And the probabilities have to total to 1.

So what is the complementary event in this case. The event that NONE of the m people share a birthday. In how many ways can we have these m people not share birthdays at all.

Think of it again as a sequencing/arrangement problem. Person 1 has 365 available to him/her as potential birthdays. Person 2 is not allowed to have a birthday on the same day, so that means 364 possible days. And now you get the idea...

That means the probability of the complement event =

$$\frac{365 \times 364 \times \cdots (365 - m + 1)}{365^m}$$

Therefore the probability of the event itself is

$$1 - \frac{365 \times 364 \times \cdots (365 - m + 1)}{365^m}$$

This initially might seem like an uninteresting geeky expression until you start plugging numbers in and noticing the really quickly increasing nature.

Application - Hashing and hashmaps

Assume we have n values and they are being hashed to 20 locations. Assume also that we do not have a hash function in mind yet, but just pick one completely at random. What is the probability that our hash function has no collisions!

If you think about this, it is actually the same thing as people and birthdays. Except the year now has 20 days instead of 365!!

So the probability that a randomly chosen hash function has no collisions is

$$\frac{P(20, n)}{20^n} = \frac{20 \cdot 19 \cdot 18 \cdots (20 - n + 1)}{20^n}$$

Similar to the birthday problem it might be somewhat surprising to realize that the probability of some collision is actually fairly high.

Birthday attack!

This is just an application of the birthday probabilities. It is called the birthday attack and describes how the above probability can be used to create a devious deception.

Click crytpo notes for notes on this.

Probability of Unions

Probability of unions

Two events are mutually exclusive if they are disjoint (which is another way of saying that their intersection is empty).

Having done set theory we know how to count the elements of a set if the two are mutually disjoint. Just applying that idea

$P(A \cup B) = P(A) + P(B)$.

Example

Consider the situation in which files are stored on a distributed network that has 30 computers. Three copies of File 1 are stored at three distinct locations in the network, and three copies of File 2 are stored at three different locations in the network (locations for File 1 are different from locations for File 2). Suppose that there are 6 random computers that have failed. What is the probability that either file has been wiped out? Let F_1 be the event that all three copies of File 1 are gone and F_2 the event that all three copies of File 2 are gone. What is $P(F_1 \cup F_2)$?

First let us compute $P(F_1)$. Our sample space consists of all the possible ways in which 6 out of 30 computers can fail. That is the same as choosing 6 from 30 or $\binom{30}{6}$. For the event described, we need all 3 computers that hold F_1 to be wiped out but the remaining 3 failed computer could be any of the rest. So that is $\binom{27}{3}$ ways.

$P(F_1) = \frac{\binom{27}{3}}{\binom{30}{6}}$

F_2 in terms of the probability calculation is exactly the same. So $P(F_2) = P(F_1)$.

But is there a chance that both F_1 and F_2 happen? Indeed. If the 6 computers that fail are the F_1 and F_2 computers. Which can only happen in this 1 disastrous way.

Therefore,

$$P(F_1 \cup F_2) = \frac{2 \cdot \binom{27}{3}}{\binom{30}{6}} - \frac{1}{\binom{30}{6}}$$

Conditional Prob

Conditional Probability

Often times, given the occurrence of an event, the probability of another event changes. The general formula for conditional probability is

$$P(E|F) = \frac{P(E \cap F)}{P(F)}$$

Example - rolling a red die and a blue die. What is the probability that the two numbers on the dice sum to at least 11 given the information that the blue die has the value 5.

Define the events

E - event of the two dice summing to 11.

F - blue die has the value of 5.

$P(E \cap F) = \frac{1}{36}$; the only way this happens is if the red die has the value 6.

$P(F) = \frac{6}{36}$; when the blue die is 6, the red die could be anything.

Therefore the conditional probability is $P(E|F) = 1/6$.

Using conditional probability to reason about an event

Conditional probability can be used to split up the probability of an event conditioned by the occurrence of some other event.

A student knows 80% of the material on a true-false exam. If the student knows the material, she has a 95% chance of getting it right. If the student does not know the material she just guesses and as expected has just a 50% chance of getting it right.

What is the probability of getting the question right?

In this type of problem we can basically define the following events

R - the event of getting the question right

K - the event of knowing the question

$P(R) = P(R \cap K) + P(R \cap K^c)$

$P(R \cap K) = P(R|K)P(K) = 0.95 * 0.8 = 0.76$

$P(R \cap K^c) = P(R|K^c)P(K^c) = 0.5 * 0.2 = 0.1$

Adding those you get a probability of 86% to get a question right.

Independence

The following conditions correspond to the criteria for declaring 2 events to be independent.

$$P(E|F) = P(E)$$
$$P(E \cap F) = P(E)P(F)$$
$$P(F|E) = P(F)$$

As a really small example consider the event of rolling two dice of different color (red and yellow). We want to show the event "total number of dots on top is odd" is independent of the event "the red die has an odd number of dots on top".

Truel

For discussion on the Truel see
http://www.mathgoespop.com/2009/10/martin-gardner-and-the-three-way-duel.html

Bayes Theorem

Suppose that F and X are events from a common sample space and $P(F) \neq 0$ and $P(X) \neq 0$. Then

$$P(F|X) = \frac{P(X|F)P(F)}{P(X|F)P(F) + P(X|F^c)P(F^c)}$$

where F^c is just the event of F not happening.

Example - Taken from UWash notes

In Orange County, 51% of the adult population is above the age of 35. One adult is randomly selected for a survey involving credit card usage.

1. Find the prior probability that the selected person is above 35.
2. It is later learned that the selected survey subject was smoking a cigar. Also, 9.5% of adults above 35 smoke cigars, whereas only 1.7% of adults 35 and below smoke cigars . Use this additional information to find the probability that the selected subject is above 35.

Let us use some notation to simplify the analysis
T = above 35 . Thereby T^c = 35 and below.
S = cigar smoker. And of course this means S^c = not a cigar smoker.
$P(T) = 0.51$

The second part is basically asking us to compute $P(T|S)$.

$$P(T|S) = \frac{P(TP(S|T)}{P(S|T^c)P(T^c) + P(S|T)P(T)}$$

We are given $P(S|T)$ since 9.5% over 35 smoke. Similarly $P(S|T^c)$ has been provided as 1.7% .

$$P(T|S) = \frac{0.51 * 0.095}{0.51 * 0.095 + 0.49 * 0.017} = 0.853$$

So there is an 85.3% chance that it is an older person if you observe cigars!

Bayes Theorem and medical tests

Bayes theorem is used to evaluate probabilities for medical tests. Here is a questions as an example

There is a medical test that has been designed to screen for a disease that affects 5 in 1000 people. Suppose the false positive rate is 3% and the false negative rate is 1%.

What is the probability that a randomly chosen person who tests positive actually has the disease?

Solution

We first need to clearly understand what false positive and false negative mean. Both these terms are used in detection systems and it is terminology that is commonly used for classifiers in machine learning as well.

False positive - the test says the person has the disease when the person actually does not have it. The word 'when' is the same as the word 'given'.

False negative - the test says the person does not have the disease when the person actually does have it. The word 'when' is the same as the word 'given'.

Now to solve this question, let us define some events

- Let A be the the event that the person tests positive for the disease.
- B_1 be the event that a randomly chosen person has the disease.
- B_2 be the event that a randomly chosen person does not have the disease.

Now the question is asking $P(B_1|A)$.

$$P(B_1|A) = \frac{P(A|B_1)P(B_1)}{P(A|B_1)P(B_1) + P(A|B_2)P(B_2)}$$

Now $P(B_1)$ is just the probability that a randomly selected person has a disease. So 0.005. Also $P(B_2)$ is just the probability that a randomly selected person does not have the disease. So 0.995.

$P(A|B_1)$ = 1 - probability of a false negative = 1- 0.01= 0.99

$P(A|B_2)$ = probability of a false positive = 0.03

Plugging all these values back into the original

$$\begin{aligned} P(B_1|A) &= \frac{0.99 * 0.005}{0.099 * 0.005 + 0.03 * 0.995} \\ &= 0.1422 \\ &= 14.22\% \end{aligned}$$

What is the probability that a randomly chosen person who tests negative does not actually have disease

This is now asking us to compute $P(B_2|A^c)$ which can again be broken by Bayes theorem as

$$P(B_2|A^c) = \frac{P(A^c|B_2)P(B_2)}{P(A^c|B_2)P(B_2) + P(A^c|B_1)P(B_1)}$$

Similar to our previous reasoning

$P(A^c|B_2)$ = 1 - probability of a false positive = 1 - 0.03 = 0.97

$P(A^c|B_1)$ = probability of a false negative = 0.01

Plugging these values into the above equation we get a probability of 99.99%.

Therefore we have a very high probability that someone who tests negative actually does not have the disease. We have a low probability that someone who is testing positive actually has the disease. The rationale behind this type of test is that as a screening test it is more important to make sure that the negative results are truly negative.

Those that test positive are asked to take a test where the probability of detecting the disease when the person has the disease is much higher. But those tests might be expensive and the cheaper screening test ensures that not all of the population has to pay for those expensive tests.

Application - Bayesian Machine Learning

The primary goal of machine learning is learning models of data. The Bayesian framework for machine learning states that you start out by enumerating all reasonable models of the data and then assign what is called a prior probability $P(M)$ to each of these models. Then you go out and collect some data (that part has become easy these days!).

Apply Bayes Theorem to get

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

Once you have observed the data D, you evaluate how probable the data was under each of these models to compute $P(D|M)$. Multiplying this probability by the prior and renormalizing gives you what is called the posterior probability, $P(M|D)$, which encapsulates everything that you have learned from the data regarding the models under consideration.

How do you compare two models M and M'? We compute their relative probability given the data: $P(M)P(D|M)$ and $P(M')P(D|M')$.

This leads to a very common technique called MAP estimation which stands for maximum a posteriori. Taken from wikipedia - MAP.

Expectations

Definition of a random variable

A random variable X is a function from the sample space S of an experiment to the real numbers. X(S) denotes the range of the function X.

The random variable is basically some number that we want to associate with outcomes of the experiment. For instance, roll two dice - red and blue and define the random number D to be the sum of the two dice. In more mathematical terms let x be the value on the blue die and y be the value on the red die, then $D(x, y) = x + y$.

If X is a random variable defined on the sample space S of an experiment and r is a real number, then X = r is an event. The event X = r consists of all outcomes s in the sample space such that X(s) = r. p(X = r) is the sum of p(s) for all s such that X(s) = r.

Expressed in more verbose language, the probability that the random variable X takes on the value r, is the sum of the probabilities for all outcomes s for which the value of the random variable is r.

Distribution of a random variable

The distribution of a random variable is the set of all pairs (r, p(X = r)) such that $r \in X(S)$.

A histogram (a visualization that most of you have seen), is nothing more than a plot of these points.

Expectation

The expected value of a random variable is what in layperson's terms we just refer to as the average value of the random variable. Formally in math, if you have a random variable X, the expected value of the random variable is

$$E[X] = \sum_{s \in S} X(s)p(s)$$

The sum is being taken over every outcome in the sample space.

```
sum = 0
for every point in the sample space {
     sum += value of the random variable for that outcome * probability of that outcome
}
```

Example - Lottery (taken from Zybook)

A lottery is run in which every ticket has six numbers. Each of the six numbers is in the range from 1 through 50. The person purchasing a lottery ticket can select the numbers on their ticket. On each Friday of the week, the state lottery commission holds a drawing in which six random numbers are generated. Each number in the range from 1 through 50 is equally likely. In order to win the jackpot, the ticket must match each number selected in order. If a person has a winning ticket, they win $100,000,000. What is the expected winning from a single ticket?

If we think of the random variable X as representative of the winnings from a lottery drawing, then X takes on only two values. 0, which happens most of the time, corresponding to not winning the lottery and then $100,000,000 which corresponds to winning the lottery. To compute the expected value we need to find the probabilities associated with the event of winning and the event of losing.

$$E[X] = (100,000,000) \cdot P(winning) + 0 \cdot (1 - P(winning))$$

What is the probability of winning? Our sample space consists of all possible arrangements of 6 numbers, where each number ranges from 1 to 50. The first number has 50 options, the second number has 50 options and so on. And of course, numbers could be repeated. So the sample space has the size 50^6.

How do you win? Only 1 way. When all the numbers match! So $P(winning) = \frac{1}{50^6}$.

Therefore

$$E[X] = (100,000,000) \cdot \frac{1}{50^6} + 0 \cdot (1 - \frac{1}{50^6}) = 0.0064$$

So the expected winnings in this lottery for a single ticket is 0.64 cents.

Example - Roulette

A roulette wheel has 38 slots. These slots consist of the number 1 through 36. Half of them are red and half of them are black. You also have 0 and 00.

Let us say you bet $1 on Black. If a black number comes up, you receive your dollar back plus $1; otherwise you lose your dollar. Let X be your net winnings in one game. Then X can take on the values +1 and -1. What are your expected winnings.

$P[X = 1] = 18/38$ and $P[X = (-1)] = 20/38$.

Thus

$$E[X] = 1 \cdot \frac{18}{38} + -1 \cdot \frac{20}{38} = \frac{18}{38} - \frac{20}{38} = -\frac{1}{19}$$

Again, you are expected to lose some money on each turn (about a nickel).

Linearity of Expectation

Quite possibly, one of the nicest results when it comes to computing Expectations is what is called Linearity of Expectations. We will state it formally and then see how a lot of calculations become really simple once you use this idea.

If X and Y are random variables defined on the same sample space S and a and b are two real numbers, then the following result is true

$$E[aX + bY] = aE[X] + bE[Y]$$

The applications of this are immediate. For instance going back to the roulette question, if instead of playing roulette just once, you decide to spin 100 times, how much do you stand to gain or lose? Let X_1 be the random variable corresponding to the earning the first time around. X_2 be the random variable corresponding to the earnings the second time around and so on.

If X is the total amount earned in this process, then $X = X_1 + X_2 + \ldots + X_{100}$.

Now thanks to linearity of expectations, you get the $E[X] = 100 \cdot \frac{-1}{19} = 5.26$. So the more you play, the more you lose.

Example of coin flips

Suppose you flip a coin n times and the random variable X is used to denote the number of heads. What the expected value of this random variable?

Intuition suggests that about half the time the number will be heads and as we will see over here, intuition will be right. But there are two ways to approach this problem and one is a lot easier.

Method 1

Let X_i denote the number of heads we see on the ith flip. Obviously X_i can take values 0 or 1 only.

then $X = X_1 + X_2 + \ldots + X_n$. This immediately means

$$E(X) = E(X_1 + X_2 + \ldots X_n) = 0.5n$$

Method 2

Suppose instead of breaking the problem up into each individual coin flip and then using linearity of expectations, you decide to do it directly. X being the number of heads can take on any value from 0 to n. What is the probability that X takes on a specific value, let's call it i.

This basically amounts to answering the question, in how many ways can we get i coins to be heads. Each coin flip is independent, so the chance of getting heads at any point is 0.5. To get i heads we just need to choose i to the coin flips to land as heads. The other $(n - i)$ times, we get tails and this event has a probability of 0.5 as well. Therefore $P(X = i) = \binom{n}{i}(0.5)^i(0.5)^{(n-i)}$.

Now that we know the probability of getting (X=i), we can compute the expectation as follows

$$E[X] = \sum_{i=0}^{n} i \cdot P(X = i) = \sum_{i=0}^{n} i \cdot \binom{n}{i} (0.5)^i (0.5)^{(n-i)}$$
$$= (0.5)^n \sum_{i=0}^{n} i \cdot \binom{n}{i}$$

The summation is not particularly easy to compute until you realize that you can use the binomial theorem and some basic calculus to show that it is in fact $n2^{n-1}$. Putting everything together you get the same result as before that the expected value will be 0.5n.

This example shows that while expectation can be calculated using the direct formula, it is usually better to try and see if there is any possible way that linearity of expectations can be used. It leads to tremendous simplifications.

Indicator random variables

Indicator random variables are random variables that take the value 1 if the event happens and 0 if the event does not happen. While seemingly trivial, these random variables have a nice property that makes them invaluable in expectation calculations. $E(X_i) = P(X_i = 1) =$ probability of the event happening.

Derangements problem

A bunch of people go to a bar. Each one of them has to check their hats in. When they leave the bar they ask for their hats back. However, they are too drunk to be careful about picking up the correct hat. Compute the expected number of people who get their (own) hats back?

Let X_i be the indicator random variable for the ith person getting their hat back. So X_i is 1 if person i gets their hat back and 0 otherwise. Given this setup, the total number of people who get their hats back will be $X_1 + X_2 + \ldots X_n$.

$E(X_i)$, since it is an indicator variable, is the same thing as the probability that the ith person gets their hat. That is a $\frac{1}{n}$ chance. Obviously this expectation is independent of i, so every single indicator variable has the expected value of $\frac{1}{n}$.

The expected number of people getting their hats will be $E(X_1) + E(X_2) + \ldots + E(X_n)$. That means the expectation is $nE(X_1)$ if we use linearity of expectation and the fact that each of the expectations is basically the same.

This means the expected number of people who get their hats back is $n\frac{1}{n}$ and that gives us the somewhat surprising result that on average only person gets their hat back. Rather surprisingly, the number does not depend upon n at all.

Usage of indicator random variables

One area in algorithms that has gained a lot of popularity is the concept of randomized algorithms. While we will visit this topic in more detail either at the end of this course or at the beginning of 596, the methods of analyzing a randomized algorithm are all rooted in probability and expectation.

So here is an example of using this with some actual code.

```
def findMin(arr):
    minimum = arr[0]
    for i in range(0,len(arr)):
        if arr[i] < minimum :
            minimum = arr[i]
    return minimum
```

We are interested in knowing the expected number of times the variable minimum gets reassigned (number of updates).

When asked to analyze an algorithm like this, the assumption really is that you are dealing with a random input array. Because anything else and you are biasing towards some answer.

So let's say we have a random array arr, with elements arr[0] through arr[n].

Let us define indicator variables X_i to correspond to the event that arr[i] becomes the minimum.

What is $E(X_i) = P(arr[i]$ becomes the minimum while iterating through this loop)

That is the same as P(arr[i] is the minimum in the set $\{arr[0], arr[1], arr[2], ...arr[i-1]\}$)

What is the probability of that?

One way of thinking of it is that there is a $\frac{1}{i}$ chance that when you randomly take i elements, the lowest one is at the end.

What is the expected number of updates. Since we want to use linearity of expectation, if we just say let $X = \sum_{i=0}^{n-1} X_i$ then clearly the value of X represents the number of times the variables minimum gets updated.

So

$$E[X] = \sum_{i=0}^{n-1} \frac{1}{i+1}$$

Unfortunately there is no closed form expression for this (a formula). But there is an approximation which comes from the theory of Harmonic numbers that basically approximates this to be close to the natural logarithm $ln(n)$. Intuition might suggest that you actually have a linearly increasing number of updates that take place as the size of your array goes up. Applying the theory of expectation, we see that actually the increase is a lot slower.

Bernoulli Trial

An experiment which has a single outcome of success or of failure. Generally, the probability of success is represented with p.

A coin-flip is the most traditional variant of a Bernoulli trial.

The generalized version of computing the probability of a certain number of heads in n coin flips is this theorem.

Theorem 1. *The probability of exactly k successes in a sequence of n independent Bernoulli trials, with probability of success p and probability of failure $q = 1 - p$ is*

$$\binom{n}{k} p^k q^{n-k}$$

Basic Logic

Proposition

A proposition is a statement that is either true or false. For instance 'Philadelphia is the capital of Pennsylvania' is a proposition, since that is a false statement.

Things that do not have a true-false answer are not propositions. For instance, the most commonly asked 'What is today's weather going to be like'.

Propositional variables such as p, q and r can be used to denote arbitrary propositions, as in "p: January has 31 days".

Boolean operators

This section will be assumed knowledge. The only reason we go through this is to introduce the syntax we will use for our course.

Conjunction operator (AND) will be denoted by $\wedge$ - use \wedge in latex.

Disjunction operator (OR) will be denoted by $\vee$ - use \vee in latex.

Exclusive or (XOR) will be denoted by $\oplus$ - use \oplus in latex. Remember that $p \oplus q$ is true only when exactly one of the propositions is true.

NOT will be denoted by $\neg$ - use \neg in latex.

One of the first things to learn in logic is expressing English sentences in a more mathematical form using logical connectives.

As an example consider the statement - 'Dinner is eaten in our house at 7pm or at 8pm'. If we wanted to express this in terms of logical connectives, we can break this down into

$$p : \text{I eat dinner at 7 pm}$$
$$q : \text{I eat dinner at 8 pm}$$

Then it seems like the most logical way of expressing the statement would be to write $p \vee q$. But if we pause for a moment we realize that one aspect of the English sentence is not really being captured. If we eat dinner at 7pm, can we say something about dinner being eaten at 8pm? Generally speaking, the English language uses the word or in the sense of exclusive or. So really if we wanted to capture the full meaning of our English sentence we should be saying $p \oplus q$.

This example might seem like an exaggeration of English ambiguities, but it is important to understand that if we want to make a computer work in a normal deterministic fashion,

then the more clearly we can express thoughts and commands, the simpler it is to get the computer to do its job.

The order of operations for Boolean algebra. The normal order of operations for Boolean algebra is the following

1. $\neg$
2. $\wedge$
3. $\vee$

Truth Tables

Truth tables are definitely things that we assume you have seen by this point. Either as a past course or in CIT593 (Note: will be covered in class if someone says they have not seen them at all). If you do not know what a truth table is, please set up a 10-15 min meeting with a TA or the instructor. They are not hard!

De-Morgan

For the simplification of Boolean expressions, the only rule that we need you to know is De-Morgan's law for booleans

- $\neg(p \vee q) = \neg p \wedge \neg q$
- $\neg(p \wedge q) = \neg p \vee \neg q$

Conditional statements

The "if-then" type of statement has specific rules when it comes to logic. Generally called an implication, it is expressed as $p \rightarrow q$ where p and q are propositions.

Very importantly, $p \rightarrow q$ is itself a proposition and it has a truth value which is determined from the following truth table.

p	q	$p \rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

The first part of a conditional is called the hypothesis and the second is called the conclusion. The hypothesis being false gives you no evidence for the truth value of the implication and the way that mathematical logic works, you go with the 'innocent until proven guilty' idea.

We have been using this already when we have discussed whether or not the less-than relationship is anti-symmetric.

Expressing conditionals in English

The following all express $p \to q$ in English

1. If p then q
2. If p, q
3. q if p
4. p only if q
5. p implies q
6. p is sufficient for q
7. q is necessary for p

Also important to remember this

$$p \to q \text{ is equivalent to } \neg p \vee q$$

This formula(or equivalence) can be verified by means of a truth table.

Contrapositive, Converse and Inverse

There are 3 statements that are closely related to $p \to q$.
They are

- Contrapositive $\neg q \to \neg p$
- Converse $q \to p$
- Inverse $\neg p \to \neg q$

Predicates and Quantifiers

Many mathematical statements contain variables. Statements that contain variables are different from propositions because their truth value depends on the value of the variable. A logical statement that is a function of variable(s) is called a predicate. A predicate always has an underlying domain associated with it. Variables are only allowed to take values from within the domain.

Universal quantifier

Sometimes we want to assert that a predicate is true regardless of the value taken from the domain. This is a universally quantified predicate and is represented with the $\forall$.

$$\forall x \in \mathbb{R}^+ \ \frac{1}{x+1} < 1$$

Proving a universally quantified statement is **True** requires showing that it is true regardless of the value of any variables that are present in the statement.

$$\begin{aligned} \frac{1}{x+1} &= 1 - \frac{x}{x+1} \\ x \in \mathbb{R}^+ &\rightarrow x+1 \in \mathbb{R}^+ \\ \text{Therefore } \frac{x}{x+1} &\in \mathbb{R}^+ \\ \text{Therefore } 1 - \frac{x}{x+1} &< 1, \ \forall x \in \mathbb{R}^+ \end{aligned}$$

Proving a universally quantified statement is **False** is easier. It just requires one counter example.

For example $\forall x \in \mathbb{R}$, $x^2 > x$ is a false statement because for $x = 0$, $x^2 = x$.

Existential quantifier

If there is at least one value in the domain that causes the predicate to evalute to true, that is expressed using an existential quantifier $\exists$.

To prove an existentially quantified statement, you just need to show one single value that makes the statement true.

For instance, to prove the statement 'There is a natural number n such that $2^n > n!$' all you have to do is say that for $n = 2$, $2^2 > 2!$ because after all $4 > 2$.

What will proving an existential statement is False involve? That is the same as proving that no value in the domain will make the statement True.

For instance to prove that $\exists x \in \mathbb{R}^+$, $x+1 < x$ is a false statement the following approach can be used.

$$\begin{aligned} & x + 1 < x \\ \implies & \ 1 < 0 \qquad \text{(by subtracting x from both sides)} \end{aligned}$$

And $1 \not< 0$ therefore proved.

Writing English statements with quantifiers

Now that we have seen quantifiers, we can express more English sentences using them.

Every CIT592 student has taken the midterm.

Assume x is a variable that comes from the domain of all MCIT first year students.
Then this would be one way of writing the above statement.
Let P(x): x is a student in 592.
Let Q(x): x has taken the midterm.
$\forall x P(x) \rightarrow Q(x)$

Negating quantified statements

It is important to be able to express the negation of a quantified statement.

Consider the statement 'All birds fly'. To write this as a quantified statement you will probably consider x to come from the domain of all birds and let $F(x)$ be a predicate that is true when x can fly. $\forall x$, $F(x)$.

What if we wanted to negate that statement. In English we would simply say 'Not all birds fly'. Which actually can be equivalently stated as 'There is at least one bird that does not fly'. That brings us to more familiar territory as far as expressing things with quantifiers goes.

$\neg(\forall x\, F(x)) \equiv \exists x\, \neg F(x)$

The sign placed between the two quantified statements is the logical equivalence sign. That is saying that whenever the left statement is true, the right statement is true and whenever the left statement is false then the right statement is false.

Similarly consider the statement 'There is some student that proved Fermat's last theorem'. Let us assume the domain is the set of all students. And let FLT(x) be true if the student proved Fermat's last theorem. Then the statement is saying $\exists x FLT(x)$.

What if we negate the statement. In English we might simply say there is no student that proved Fermat's last theorem. That is the equivalent to saying for every student, it is true that they have not proven Fermat's last theorem. Or in other words,

$$\neg(\exists x\ FLT(x)) \equiv \forall x\ \neg FLT(x)$$

In general, here are the two rules to remember when it comes to negating a quantified statement

$$\neg\forall x\ P(x) \equiv \exists\ \neg P(x)$$
$$\neg\exists x\ P(x) \equiv \forall\ \neg P(x)$$

English with Logic

Free variables and bound variables

A variable x in the predicate P(x) is said to be a free variable because the variable is free to take on any value in the domain.

The variable x in the statement $\forall x P(x)$ is a bound variable because the variable is bound to the universal quantifier.

A statement with no free variables is a proposition because the statement's truth value can actually be determined.

While we started off with predicates of just one variable, you can easily extend the definition to predicates with any number of variables.

$B(x, y)$: x beat y in a tournament, where x and y are professional tennis players.

$D(x, y)$: x drinks y, where x is from the set of professors and y from the set of beers available at Monks.

You can combine quantifiers and then have the concept of free and bound variables show up even more.

$\forall x P(x, y)$	(x is a bound variable, but y is free)
$\forall x \exists y Q(x, y)$	(x and y are both bound)

Nested quantifiers

Same quantifiers

Let the domain be the set of students who are in MCIT first year.

Define a predicate H to be:

F(x,y): y and x are friends on facebook.

Consider the following quantified statement

$$\forall x \forall y F(x, y)$$

What is this in English? Perhaps something like 'In the MCIT program, everyone is everyone's friend on facebook.'

But does this miss something? What about the case when x and y are the same?

Generally speaking, if we do not expect everyone to be friending themselves on facebook, the mathematical statement is better expressed as

$$\forall x \forall y((x \neq y) \rightarrow F(x, y))$$

and the English translation now would be. "In the MCIT program, everyone has friended every other person on facebook."

Let's retain the same domain. Another predicate that can be made is

P(x,y): x has pair programmed with y

Now the proposition

$$\exists x \exists y P(x, y)$$

is basically then saying "There is some MCIT student that has pair programmed with some MCIT student."

Example

Assume the domain is $\mathbb{Q}$

What is the truth value of $\forall x \ \forall y \ ((x + y \neq x) \vee (y = 0))$

Alternating quantifiers

Let's try and express the following in English. The predicates used are the ones defined in the previous subsection.

$$\forall x \exists y P(x, y)$$

This is now saying "Every MCIT student has pair programmed with some MCIT student".

Does the order of those quantifiers matter?

$$\exists x \forall y P(x, y)$$

This is now saying "There is some MCIT student that has pair programmed with every MCIT student".

Now let us get back to math and explore the truth value of some of these quantified statements.

Assume the domain to be integers

$$\forall x \exists y \ x + y = 0$$

Is this true? A fun way of reasoning about it is to treat it as a two player game. Player 1 - the universal player is focussing on the universally quantified x and is trying to make the statement false. Player 2 - the existential player is focusing on the existentially quantified y and is trying to make the statement true.

In the above statement, regardless of what value of x the universal player chooses, the existential player just sets y to be $-x$ and the proposition is true.

So this is a true statement.

Let us switch the order of quantifiers

$$\exists x \forall y \; x + y = 0$$

In the two person game method of reasoning, the players go in the order of quantifiers. So the existential player picks an x this time. The universal player happily gets to be evil and picks a y that is not $-x$ and gets the statement to be false. Thus the universal player can always win and therefore the statement $\exists x \forall y \; x + y = 0$ turns out to be false.

Negations with multiple quantifiers

We have negated quantified statements and noticed how the universal quantifier turns into the existential quantifier and vice-versa. That logic can be extended to statements that involve multiple quantifiers as well. Let us try to negate the boxed statement below

There is a student who has eaten at every food cart.

It is safest to turn these into our language of quantifiers and then do the negation.

Let the variable x come from the domain of students. Let the variable y come from the domain of food carts.

Define the predicate $E(x, y)$ as true when student x has eaten at foodcart y

then the boxed statement becomes

$$\exists x \forall y E(x, y)$$

Now that we have this in nice quantified form, we can go ahead and do the negation

$$\begin{aligned} & \neg \exists x \forall y \;\; E(x, y) \\ = \; & \forall x \exists y \;\; \neg E(x, y) \end{aligned}$$

Now to express this back in English

Every student has some food cart that they have not eaten at.

Domain expansion

Consider the statement

All mammals have hair.

Let M be the set of mammals.

Let H(x) be the predicate that is true when x has hair.

$$\forall x \in M \ (H(x))$$

Consider how we would express this if we wanted to change the domain and make it larger. Make it larger to encompass all animals. Let us call the set of all animals A.

$$\forall x \in A \ (x \in M \rightarrow H(x))$$

Consider the statement

Some politicians are honest.

Let P be the set of politicians

Let $H(x)$ be the predicate that is true when x is honest

$$\exists x \in P \ (H(x))$$

Consider how we would express this if we wanted to change the domain to make it all working humans.

$$\exists x \in W \ ((x \in P) \wedge H(x))$$

Example of expressing uniqueness

Since mathematical theorems very often use the expression 'there exists a unique', it is important to understand how to express that idea via predicates, logical connectives etc.

Assuming we start with the domain of humans.

Everyone has exactly one best friend

Let us break this into a couple of steps

1. $\forall x$(x has exactly one best friend)
2. $\forall x \exists y$ BFF(x, y) and x does not have any other best friend. Where BFF(x, y) is a predicate that is true whenever y is the best friend of x.
3. $\forall x \exists y$ BFF(x, y) and all other z satisfy $\neg$BFF(x, z)
4. $\forall x \exists y$ BFF$(x, y) \wedge \forall z$ If $z \neq y$ then $\neg$BFF(x, z)
5. $\forall x \exists y (BFF(x, y) \wedge \forall z (z \neq y) \rightarrow \neg$BFF$(x, z)))$

Common confusion

One of the more common mistakes is to interchange the $\wedge$ and the $\rightarrow$. As an amusing illustration of this, we'll use a classic from Lewis Carroll (or C.L. Dodgson if you prefer his geeky side).

Here are a set of logical statements. Remember that right now we are only trying to express the statements as opposed to trying to prove anything.

1. All lions are fierce
2. Some lions do not drink coffee
3. Some fierce creatures do not drink coffee

Let the domain be the set of all creatures. Let us use $L(x)$ to denote x is a lion. $F(x)$ for x is fierce and $C(x)$ for coffee drinking.

Here are the statements in logic

1. $\forall x(L(x) \rightarrow F(x))$
2. $\exists x(L(x) \wedge \neg C(x))$
3. $\exists x(F(x) \wedge \neg C(x))$

So why can't we try writing $\exists x(L(x) \rightarrow \neg C(x))$? Consider just the $L(x) \rightarrow \neg C(x)$ part. This implication becomes try anytime the $L(x)$ is false. What does it mean to say $L(x)$ is false? That just means we have a creature that is not a lion.

Therefore, if we write the statement using an implication, the moment we find a creature that is not a lion, it becomes a true statement. More importantly it becomes a true statement even in the case when every single lion is a coffee drinker. Hence, writing it as an implication is not correct.

Expressing theorems using logical symbols

We have dealt with smaller sentences being expressed using predicates. Now that we have seen universal quantifiers and existential quantifiers, we can combine all our knowledge in order to make more logical expressions out of the English statements of mathematical theorems.

Here for instance is the statement of what is called the division theorem -

For every positive integer n and every non-negative integer m , there are non-negative integers q and r with $0 \leq r < n$ such that $m = qn + r$.

Here is one way of expressing this using quantifiers and logical connectors. We introduce the notation $\mathbb{W}$ to represent the set of non-negative integers. This is because we went with the convention of not including 0 in our natural numbers.

$$\forall n \in \mathbb{N}(\forall m \in \mathbb{W}(\exists q \in \mathbb{W}(\exists r \in \mathbb{W}((r < n) \wedge (m = qn + r)))))$$

The parentheses are used over here just to make things clearer. It would not be wrong to use the phrase 'such that' between the various pieces.

Basics of Proofs

For this particular part of the course we will rely on sections from Scheinerman's book which has a good explanation of how a proof needs to be written. While it might be too detailed initially, it is important to establish the fundamentals.

Example of a proof

Consider the following statement of a result that we want to prove.

Theorem 1. *The sum of two even integers is even*

A possible proof goes as follows

1. We show that if x and y are even, then $x + y$ is even.
2. Let x and y be any even integers
3. Since x is even, by the definition of even integers we know that $2|x$
4. Similarly y being even means $2|y$
5. Since $2|x$ we know $\exists b \in \mathbb{Z}$ such that $x = 2b$
6. Similarly $\exists c \in \mathbb{Z}$ such that $y = 2c$
7. Now observe $x + y = 2b + 2c = 2(b + c)$
8. Therefore there is an integer a (which happens to equal $b + c$) such that $x + y = 2a$.
9. Therefore $2|(x + y)$
10. Therefore $x + y$ is even

The steps of a proof

The first step is to convert to statement of the theorem into logic. In this case we are able to convert it into an 'if-then' statement.

We also need to introduce some notation in order to begin the proof. For a universally quantified statement like this, we need to show that the statement holds for any value. Therefore it begins with x and y being even integers, but there is nothing special about the choice of these. It would be just as fine to say, 'Consider any x and y even integers.'

Once we have our initial step of the proof, we write down what we need to show at the end of the proof. In this case, we need to show $x + y$ is even.

The rest of the proof consists of filling the space from the starting point to the end point.

Generally, it is a good idea to use definitions. We unravel the definition of even and of the word divisible. Once you have seen this a few times, you can jump directly to the step of saying that x is even implies $\exists a \in \mathbb{Z}$ such that $2a = x$. If you unravel the definitions all the way through, you get to step 6 and then seemingly you are stuck.

That is when you have to unravel the definitions that are in the final statement. You work your way backwards and get to step 8.

That gets to stage where you have to connect the top part of the proof with the bottom part of the proof. This middle-part of the proof generally the hard part. You have to establish a connection between what you've got and what you need.

In this particular case, manufacturing an a is not hard since we are able to produce one just by adding x and y and observing the result.

Proof for sets

There are two basic things to remember for proving things in set theory from first principles

Showing $A \subseteq B$

The technique is to consider any element $x \in A$ and then logically work out why $x \in B$.

Example Prove that $A = \{x \in \mathbb{Z} | 18 \text{ divides } x\}$ is a subset of $B = \{x \in \mathbb{Z} | 6 \text{ divides } x\}$

Proof:

Consider any $x \in A$. Then because $18|x$ this means $x = 18y$ for some $y \in \mathbb{Z}$. But that can be written as $x = 6(3y)$, so that means x is divisible by 6.

Therefore $x \in B$.

Since any $x \in A$ is also going to be in B, therefore $A \subseteq B$.

Showing A = B

The technique is to show $A \subseteq B$ and $B \subseteq A$.

Example: To show the distributive property of sets from first principle

$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Proof:

First we need to show $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$

Consider any $x \in A \cup (B \cap C)$

By definition $x \in A$ or $x \in B \cap C$

Case 1: $x \in A$

Then by definition of union $x \in A \cup B$ and $x \in A \cup C$.

By definition of interesection this would mean $x \in (A \cup B) \cap (A \cup C)$

That means $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$

Case 2: $x \in B \cap C$
By definition this means $x \in B \wedge x \in C$
By definition of union $x \in (A \cup B) \wedge x \in (A \cup C)$.
This would then mean $x \in (A \cup B) \cap (A \cup C)$.
That means $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$
So in both cases $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$

Now to show $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$
Consider any $x \in (A \cup B) \cap (A \cup C)$
By definition $x \in A \cup B$ and $x \in A \cup C$
Consider $x \in A \cup B$. Then $x \in A$ or $x \in B$.
When $x \in A \cup C$. Then $x \in A$ or $x \in C$.
Combining everything there are two possibilities either $x \in A$ or $x \in B \cap C$.
Using the definition of $\cup$ this can just be written as $x \in A \cup (B \cap C)$.
We have therefore shown $(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$.
By showing $A \cup (B \cap C) \subseteq (A \cup B) \cap (A \cup C)$
and
$(A \cup B) \cap (A \cup C) \subseteq A \cup (B \cap C)$.
we can now conclude that
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Introduction to Induction

Recognizing a pattern

Consider the following pattern

$$1+3=4$$
$$1+3+5=9$$
$$1+3+5+7=16$$

The pattern at this point is hopefully becoming clear. Let us do one more
$1+3+5+7+9=25$

Looks like the sum of odd numbers is equal to some perfect square. It is time to formalize this. Every odd number is of the form $2k+1$ for some integer k. That should lead us to this

$$\sum_{i=1}^{n} 2i-1 = n^2$$

where $n \in \mathbb{N}$.

Whenever you have a result that follows a pattern like this, in order to prove it you have to show the formula/pattern holds for every single value of n. Since n takes values in 'discrete' steps (1,2,3 etc), the powerful method called induction can be applied to prove it.

Idea behind induction

Induction is used to prove that a theorem holds for all natural numbers (sometimes we will include 0).

If we are able to show the following.

- the theorem holds for 1 (or some small number)
- If the theorem holds for $n-1$, then it will hold for n.

Now let's say we want to show the theorem is going to hold for 5. Here's how we could do it

- it holds for 1

- Since it holds for 1, it holds for 2.
- Since it holds for 2, it holds for 3.
- Since it holds for 3, it holds for 4.
- Since it holds for 4, it holds for 5.

This idea of using the previous to prove the current is the crux of induction. If you have seen recursion in programming, this methodology of solving sub problems and using their solution to solve the big problem should look familiar. It is closely closely related to recursion.

Outline of an induction proof

1. Write the statement of the theorem in terms of a predicate that has an input of a single number. Your goal in proving the result/theorem is to show the predicate will always return the value true regardless of what number is used as an input. So your theorem is now stated as

   To show P(n) is true for all positive n (or for all $n \geq 42$ or something like that) where P(n) is defined as ....

2. Prove the theorem for the smallest possible n. Generally this will be something like 0 or 1 but it depends on the theorem. This is called the base case.

3. Show $P(n-1) \implies P(n)$

   Remember that to prove the above implication we need to assume $P(n-1)$ to be true and use that in some manner when we are proving $P(n)$.

   It is common convention to call the $P(n-1)$ is true assumption the *induction hypothesis*.

Proof of summations

Let us attempt an inductive proof of the first result.

$$\sum_{i=1}^{n} 2i - 1 = n^2$$

Define the predicate $P(n)$ as a function which returns true or false depending

Base case: When $n = 1$, there is only term in the summation which is $2 - 1 = 1$ and the right side is also 1. So it is true for 1.

Assume the result is true for $n - 1$ So

$$\sum_{i=1}^{n-1} 2i - 1 = (n-1)^2$$

To show if $P(n-1)$ then $P(n)$.
Consider the sum

$$\begin{aligned}
\sum_{i=1}^{n} 2i - 1 &= \sum_{i=1}^{n-1} 2i - 1 + 2n - 1 \\
&= (n-1)^2 + 2n - 1 && \text{(using P(n-1) true)} \\
&= n^2 - 2n + 1 + 2n - 1 \\
&= n^2
\end{aligned}$$

which shows that the predicate is true for n.

Proof of inequalities

Which is greater $n!$ or 2^n?

3! is 6 and 2^3 is 8. 4! is 24 and 2^4 is 16. 5! is 120 and $2^5 = 32$. So looks like the inequality $n! > 2^n$ holds for $n \geq 4$

To prove this claim by induction.
Let the predicate be defined as $n! > 2^n, n \geq 4$.
The base case is $n = 4$ which we have already shown to be true.
Let the statement hold true for $n - 1$. Meaning

$$(n-1)! > 2^{n-1} \tag{1}$$

To prove this inductively we need to use this to show $n! > 2^n$.
Multiply both sides of 1 by n since $n! = (n-1)!n$.
So we get $n! > 2^{n-1} \cdot n$. Since $n > 4$, we know that
$2^{n-1} \cdot n > 2^{n-1} \cdot 2$
But $2^{n-1} \cdot 2 = 2^n$
Hence proved

Example involving sets

Generalized De-Morgan's law.

We will do this in class and the steps are all in the Zybook as part of participation activity 10.5.6

Induction puzzle

In Josephine's kingdom every woman has to pass a logic exam before being allowed to marry. Every married woman knows about the fidelity of every man in the kingdom except for her own husband, and etiquette demands that no woman should tell another about the fidelity of her husband. Also, a gunshot fired in any house in the kingdom will be heard in any other house. Queen Josephine announced that unfaithful men had been discovered in the kingdom, and that any woman knowing her husband to be unfaithful was required to shoot him at midnight following the day after she discovered his infidelity. How did the wives manage this?

This is a classic puzzle and the solution can actually be found on wikipedia. We will discuss in class or recitation.

Divisibility proof

$2^{n+2} + 3^{2n+1}$ is divisible by 7 for all positive integers

Proof by induction

Let $P(n)$ be the predicate that returns true when $2^{n+2} + 3^{2n+1}$ is divisible by 7.

Base case: P(1) says $2^3 + 3^3 = 8 + 27 = 35$ and 35 is 5×7.

To show $P(n-1) \rightarrow P(n)$.

We are given that $2^{(n-1)+2} + 3^{2(n-1)+1}$ is divisible by 7. That is $2^{n+1} + 3^{2n-1}$ is divisible by 7.

We now consider $2^{n+2} + 3^{2n+1}$ and simplify it to the point where we can use our induction hypothesis and prove the theorem.

$$
\begin{aligned}
&2^{n+2} + 3^{2n+1} \\
&\qquad = 2 \cdot (2^{n+1} + 3^{2n-1}) + 3^{2n+1} - 2 \cdot 3^{2n-1} \\
&\qquad = 2 \cdot (2^{n+1} + 3^{2n-1}) + 3^{2n-1}(3^2 - 2) \\
&\qquad = 2 \cdot (2^{n+1} + 3^{2n-1}) + 3^{2n-1} \cdot 7
\end{aligned}
$$

The first term is divisible by 7 because of the induction hypothesis. The second term is clearly divisible by 7.

Hence we have shown that $P(n-1) \rightarrow P(n)$, which along with the base case completes a proof by induction.

Breaking down or building up

The most controversial aspect of induction happens to be whether to build the problem up or break the problem down. It is universally true that breaking the problem down is the

correct way to go. The issue is that in a lot of the algebraic questions, building the problem up works as well. So it is easy to fall into the trap of using that as your approach always.

The place where this concept of building the problem up definitely breaks down is actually closer to computer science which is the reason for stressing it so much.

Autocities

One example of the build up versus breaking down aspect of induction proofs is the following

Define an autocity to be a city that has a road that leads to some other(different) city.

Claim: In any collection of n autocities there is a way (obviously via a collection of roads) to get from any city to any other city.

Let us try a proof by induction

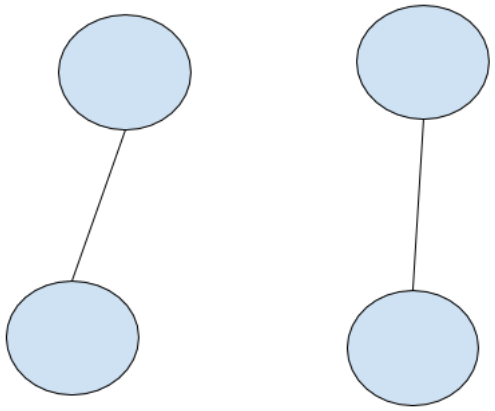
Base case: We need at least 2 cities for the definition of autocity to work. With 2 cities, the only way that both of them can be autocities is if there is a road connecting them. So the statement of the claim is obviously true when we have 2 cities.

Induction hypothesis: In any collection of $n-1$ autocities there is a path between any pair of cities.

Now consider adding in the nth city. For this city to be an autocity, it must be connected via a road to one of these $n-1$ autocities. Now consider any pair of autocities in this collection of n. If the pair comes from the previous $n-1$ cities there is a path via the induction hypothesis. If the pair includes the newly added city, we can use a road to get from the nth city to whatever city it was connected to. Let's call that D. Then there is a path from D to any other city (by the induction hypothesis). So now we have a path between any pair of cities. Hence proved.

Seemingly we have proven by induction.

But let us step back and see if the claim is true. It isn't. See below!



Try doing this same proof by taking any collection of n autocities and removing an autocity. You will find that it is impossible to make the claim that by removing an autocity you still have a collection of autocities. So by the breaking down method you will not run into an incorrect proof.

Consider the example of the following question

Exercise

Prove that in a group of N people, if some of them shake hands, there are at least two who have the same number of handshakes.

The solution offered at the website has a number of issues but the biggest amongst them is that it assumes that 'a person enters the room'.

On the face of it, this does not seem like a big issue. After all, we know (or think we know) that any instance of 10 people in a room can be considered to have first been an instance of 9 people in the room and the 10th one walked in. And it is true that this would probably be the way that we would try and work it out with someone if we were discussing it informally.

Unfortunately (but only slightly so once you grasp the idea), math is not so kind. The mathematician's counter argument is that 'you have now manufactured a specific instance of n people if you take the nth person and put her into a group of $n-1$ people. What if my instance is not like that instance. What then? Sadly no amount of hand waving will convince the obstinate mathematician.

What is worse in this case is that the proof says the $n-1$ case has a pair that has shaken the same number of hands. Now the nth person comes in and the whole solution talks about the interaction of this person with this special $n-1$ pair.

But the problem is talking about n people shaking hands. So what is given to you is the case of n people. The question does not tell you that you have a group of n people with a clearly demarcated nth person and a clearly demarcated pair of people among the $n-1$ that have had the same number of hand shakes, which is what it would seem the solution relies on.

Bottom line: It is a bad idea to try and build up towards your problem. It is better to break the problem down.

Answer of exercise

Let us write the predicate down clearly. We are saying that in a group of n ($n \geq 2$) people, if people shake hands, there will always be at least one pair of people that have shaken the same number of hands.

The base case of 2 people is easily shown. They either shake hands or do not.

Now we are allowed to assume that in any group of size $n-1$ that shakes hands, there will be at least 2 people who have the same number of handshakes.

Using this assumption we have to somehow show that in a group of n people that shakes hands, there will be at least 2 people who have the same number of handshakes.

Consider the n people (you have to start from the instance of the problem that is supplied and break it down).

There are 2 cases

1. One of the people does not shake any hands. Let us call them A. Look at the $n-1$ people aside from A. That is a group of $n-1$ people with some handshakes. By the induction hypothesis, it should be possible to find someone in that group that has the same number of handshakes. Put person A back in. They have no affect whatsoever

on the handshakes. So in the instance provided to us we have found 2 people who have the same number of handshakes.

2. If we have no 0 handshakes person in the room. This means that there is nobody with 0 handshakes. Then this means that the possible values for the number of handshakes are $\{1, 2, 3, \ldots, n-1\}$. That is a grand total of $n-1$ distinct values but there are n people. This means at least one value will have to be duplicated (this type of reasoning is also called the pigeon hole principle)

In both cases, we can show that there will be pair of people with the same number of handshakes.

We have shown that assuming $P(n-1)$ we have $P(n)$ to be true.

This coupled with the base case completes the proof by induction.

Strong Induction

Strong induction template

Let P(n) be a predicate parameterized by a natural number n. Let $a \leq b$, where a and b are integers. Then $P(n)$ is true for all $n \geq a$, if the following two conditions hold:

1. $P(a), P(a+1), \ldots, P(b-1)$ are true (the base case).
2. For all $k \geq b-1$, if $P(j)$ is true for every $a \leq j < k$, then $P(k)$ is also true(the inductive step).

   Alternatively think of it as $P(a) \wedge P(a+1) \wedge P(k-1) \rightarrow P(k)$ for all values of $k \geq b$

Strong induction - intuition

Weak induction uses the following idea. Let's prove it for 1. Then let us use that to prove for 2. Then let us use that to prove for 3 and so on ..

Of course we do not want to write a billion steps to the prove, so we show $P(n-1) \implies P(n)$. Assuming $P(n-1)$ is called the induction hypothesis. And then using the induction hypothesis to show $P(n)$ is called the induction step.

Strong induction uses the following idea. Let's prove a bunch of base cases. So for instance we show the result for 1,2,3 and 4. Then when we have to show the result for 5, we exploit the fact that the result holds for 1,2,3 and 4. It may be the case that we only need two of those or three of those. The point is, we have all of them now at our disposal.

Again, the general thing we want to show is $P(m) \wedge P(m+1) \wedge \ldots P(n-1) \rightarrow P(n)$. Here m is the smallest value for which the result holds.

Division theorem

Let n be a positive integer. Then for every non negative integer m there exist unique integers q and r such that $m = nq + r$ and $0 \leq r < n$.

Proof by induction

This statement has a lot of variables floating around. The key is to recognize that if you take any arbitrary n and show that now any m can be expressed as $qn + r$ then for any other choice of n the same reasoning can be applied. So the theorem can be proven by doing induction on m.

This means we have a fixed but arbitrary value of n that we have picked

Base Case: For $m = 0$. $0 = n \cdot 0 + 0$. So the theorem holds for 0. In fact for any $m < n$, we have $m = n \cdot 0 + (n - m)$. Because $m < n$ this gives us $0 \leq n - m < n$ which satisfies the theorem.

Now to show that $P(0) \wedge P(1) \wedge \ldots P(k-1) \rightarrow P(k)$ for all $k \in \mathbb{Z}, k > n$ (we have already shown the result for $k \leq n$ by doing the base cases)

We would like to show $k = nq + r$.

The first thing to note is that it is tough to do anything with just $P(k-1)$ which is the reason we have to resort of strong induction.

Consider $k - n$, which is a positive integer which is less than k. By the induction hypothesis we know

$k - n = nq' + r'$ where $0 \leq r' < n$.

But this means

$k = n(q' + 1) + r'$ where $0 \leq r' < n$. Since q' is an integer, so is $q' + 1$.

So set $q = (q' + 1)$ and $r = r'$ and we are able to express $m = nq + r$ where $q \in \mathbb{Z}$ and $r \in \mathbb{Z}$ and $0 \leq r < n$

This proves that $P(0) \wedge P(1) \wedge P(k-1) \rightarrow P(k)$ for all $k \geq n$, $k \in \mathbb{Z}$.

Combine this with the base cases and we have our proof by strong induction.

Chocolate division

You are given a rectangular chocolate bar consisting of n chocolate squares. Your task is to split the bar into small squares (always breaking along the lines between the squares) with the minimum number of breaks. How many breaks will it take? Prove by induction that the result holds for any n.

Proof by induction.

The induction will be on the number of chocolate squares. Since we do not know what the result is in terms of n, we first do a few base cases.

Base case: If there is only 1 giant square of chocolate, no splitting is needed, so the number of breaks is 0. If there are 2 squares of chocolate, we split along the 1 dividing line and the number of breaks is 1.

So it looks like the number of break is $n - 1$. Define a predicate $P(n)$ to be true when a n square chocolate bar can be split into smaller square in $n - 1$ breaks.

We have already proven a few base cases so to complete the proof we need to show

$P(1) \wedge P(2) \wedge P(k-1) \rightarrow P(k)$

Think about any k piece chocolate bar. To break it into smaller squares there has to be first break. Now we are left with an m piece chocolate bar and an n piece chocolate bar. Clearly, $m + n = k$.

By the induction hypothesis, since both m and n are less than k we know that the m piece chocolate bar will take $m - 1$ breaks and the n piece bar will take $n - 1$ breaks.

So the total number of breaks will be $1 + m - 1 + n - 1 = m + n - 1 = k - 1$

Therefore by using the induction hypothesis we are able to show that a k square bar will take $k-1$ breaks.

Combine this with the base cases we proved earlier and we have a proof by induction that any n square chocolate will take $n-1$ breaks to be broken into every single smaller square.

Nim

In the parlour game Nim, there are two players and two piles of matches. At each turn, a player removes some (non-zero) number of matches from one of the piles. The player who removes the last match wins. If the two piles contain the same number of matches at the start of the game, then the second player can always win. Figure out what the strategy for winning is and then prove this strategy does guarantee a win by using induction.

Proof by induction on the number of matches n in each pile.

Base case If both piles contain 1 match, the first player has only one possible move: remove the last match from one pile. The second player can then remove the last match from the other pile and thereby win.

Induction hypothesis Suppose that the second player can win any game that starts with two piles of k matches, where $1 \leq k < n$. We now need to show that player 2 will win when we have n matches.

So, suppose that both piles contain n matches.

A legal move by the first player involves removing j matches from one pile, where $0 \leq j < n$. The piles then contain n matches and $n-j$ matches. The second player can now remove j matches from the other pile. This leaves us with two piles of $n-j$ matches. Two cases exist - if $j = n$, then the second player just takes all the matches from the other pile and wins. If $j \leq n$, then we're now effectively at the start of a game with $n-j$ matches in each pile. Since $1 \leq n-j < n$, we can now use the induction hypothesis and show that player 2 wins the game.

Making change

Suppose that cans of juice come in packs of 3 or 4. We would like to be able to buy n cans of juice for any n. For which values of n is this possible? We would like to show for $n \geq 6$, it is possible to buy a combination of 3-packs and 4-packs so that the total number of cans is exactly n.

Directly from the zybook (please see solution there).

How do I know I need strong induction

Why not always use strong induction?

The key is to understand that the hypothesis in strong induction already includes the hypothesis that weak induction has. So it is always ok to assume inductively that the result holds for all smaller values. Then if you find that you only need to use the $n-1$ case, that will amount to weak induction.

In some sense this is similar to the concept of a proof by contradiction. Every proof can be done by contradiction. It actually gives you the most to work with. Some proofs by contradiction might be rewritten as a direct proof or a contrapositive proof.

Recursion and Big-O

Recursion

The idea behind recursion is the same as that being used in proofs by induction. Can we show the fact that if a problem is solved (if a theorem is true) for smaller instances (for n-1, n-2 etc) then the problem is solvable (then the theorem is true) for the current instance (for n).

The canonical example used to introduce recursion in most CS classes is factorial. Factorial has this nice property that $factorial(n) = n \times factorial(n-1)$. So when you write the function recursively there is this level of trust that goes on that should not make you really question something along the lines of 'why should I believe factorial(n-1) will get computed correctly'. The correct way of thinking is to say let me assume factorial(n-1) is computed correctly. Can I do something with that?

Proving a program works!

It is sometimes important to know if a program really works. And by this we do not mean, is there a syntax error or a logical flaw. We are asking the question that in a world where the idea that you have (CS calls those algorithms) gets perfectly translated into code (in some non crazy programming language), can you prove that your program will work regardless of the input provided to it.

Now the current best practice in the industry to do this proof is to write a bunch of unit tests. But does unit testing equate to a proof of correctness?

Unfortunately, math will just laugh at that notion. Unit testing just amounts to proof by example. Wouldn't it be better if you could say it works for everything. That turns out to be a universal statement.

So for factorial this is what you want to prove

This program will compute factorial correctly regardless of input!

How do you actually write a proof? You rely on the fact that induction is recursion!

Here's a sketch of the proof.

Base case - observe that when n=0 then the program returns 1 and that happens to be the same as 0!

Induction hypothesis - we assume that factorial works just fine for 0,1,2,...,n-1

Now for computing factorial of n, see that the program computes it as $n \times factorial(n-1)$. But we know by induction that we actually have $factorial(n-1) = (n-1)!$.

So the program is just computing $n \times (n-1)!$. But that is just $n!$.

How long does it take? aka Big - O

As noted before, in most cases in the industry, you do not care about proving the correctness of your code (depends on the industry of course). But you do care about how much time your program is going to take!

While there are several ways of measuring time, the theoretical computer science methodology is to basically focus on the particular aspect of scalability.

Take for instance the case of searching through an array of size n.

The way a CS person expresses the time taken for an algorithm is to say the algorithm is O(f(n)). For instance the search problem can easily be shown to be what is called $O(n)$ (once you know the definition of O). So you will say

Searching in an unsorted array is an O of n operation.

or

Searching is a linear operation.

Definition of big-O:

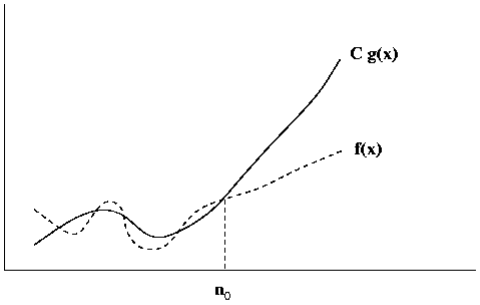
Let f and g be two functions defined on some subset of the real numbers. One writes

$$f(x) = O(g(x)) \text{ as } x \to \infty$$

if and only if there is a positive constant C such that for all sufficiently large values of x, $f(x)$ is at most C multiplied by the absolute value of $g(x)$. That is, $f(x) = O(g(x))$ if and only if there exists a positive real number C and a real number n_0 such that

$$|f(x)| \leq \ C|g(x)| \text{ for all } x \geq n_0.$$

While the formal definition is needed for math, most CS people think of it in terms of this graph



f(x) is O(g(x))

Beyond a certain point, a constant multiple of one of the functions dominates the other.

Example

Show that $n^2 + 100n$ is $O(n^2)$.

Show in class why in this case you can set C to be a number like 200 and x_0 to be 2. Obviously other values are possible but the thing about the O analysis is that you can be sloppy

Sloppiness is ok?

Algorithmic complexity is interesting in that it deals with inequalities more than equalities. So you will often look at the expression for time taken and draw conclusions very quickly by dropping terms. For instance if you compute the relationship between search time and the size of the array and say that it is $100n$, that is actually generally expressed as $0(n)$.

The constants do matter, but first it is important to make an efficient algorithm in the big - O sense. If you have two competing algorithms that are both linear, then you can worry about the constants. But if you have one which is linear and one which is quadratic, it should be clear which one is to be picked.

Therefore, remember that you are allowed to do some sloppy math when you are analyzing algorithms. The key is to begin the sloppiness only after you have written a correct expression down and done some level of evaluation.

Recurrences

A recurrence relation in the most basic sense is an equation which is trying to define a sequence recursively.

So things like $f(n) = n*f(n-1)$ (factorial) or $T(n) = 2T(n/2)+n$ (merge sort recurrence).

The analysis of algorithms that are recursive in nature boils down to solving recurrences and we will cover some theorems

First order Linear recurrences with constant coefficients

Theorem 1. *The closed form solution for*

$$T(n) = \begin{cases} rT(n-1) + g(n) & \text{if } n > 0 \\ a & \text{if } n = 0 \end{cases}$$

is given by

$$T(n) = r^n a + \sum_{i=1}^{n} r^{n-i} g(i)$$

Proof. The proof of most any property related to recursion and recurrences is best done with induction.

Base case: When $n = 0$, we have $T(0) = a$ as per the definition and as per the formula we have $r^0 a + \sum_{i=1}^{0} r^{0-i} g(i)$. But the summation just returns 0 if we are going from a higher number to a lower number. So this just reduces to $r^0 a = a$.

Assume the formula is correct for $T(n-1)$.

We know $T(n) = rT(n-1) + g(n)$.

This is where the induction kicks in. Just substitute the formula for $T(n-1)$ since we have assumed that to be true.

$$\begin{aligned} T(n) =& r(r^{n-1}a + \sum_{i=1}^{n-1} r^{n-1-i}g(i)) + g(n) \\ &= r^n a + \sum_{i=1}^{n-1} r^{n-i}g(i) + g(n) \\ &= r^n a + \sum_{i=1}^{n-1} r^{n-i}g(i) + r^{n-n}g(n) \\ &= r^n a + \sum_{i=1}^{n} r^{n-i}g(i) \end{aligned}$$

and this matches the claim being made for $T(n)$.

Combine this with the base case and we have a proof by induction.

□

Application of the theorem.

This theorem can readily be applied for a whole number of 'real world' problems.

For instance, here is a recursive way of finding the maximum element in an array

```
maximum(array) {
    if length(array) is 0 return n/a
    if length(array) is 1 return array[1]
    return max(firstelem,maximum(rest))
}
```

How long does this function take? What is the running time of this algorithm?

Both of these questions are generally meant to be answered in the big-O sense. In particular, we need to express $T(n)$ (the time taken on an input of size n) as $O(g(n))$ where $g(n)$ is one of the commonly found functions.

As we can see from the algorithm, to solve the n sized version of the problem, we make one call to the function with an $n-1$ sized version of the problem and do a single computation after that.

$$T(n) = T(n-1) + 1$$

and it is very easy to use the above theorem to see that the closed form solution for this recurrence is going to be $T(n) = n$. So the running time is $O(n)$.

The Towers of Hanoi recursion works out to be $Moves(n) = 2Moves(n-1) + 1$. Also in this case we have $Moves(0) = 0$.

By direct application of the theorem we have the result that

$$
\begin{aligned}
Moves(n) &= 2^n \cdot 0 + \sum_{i=1}^{n} 2^{n-i} \cdot 1 \\
&= 2^{n-1} + 2^{n-2} + \ldots + 1 \\
&= 2^n - 1
\end{aligned}
$$

Binary search recurrence

For mergesort we can write the recurrence as the following

$$T(n) = T(n/2) + T_{check}$$

Also as a base case, assume that for an array of size 1, it takes constant time a. $T(1) = a$.

T_{check} is the time taken to see whether the middle element is equal to the element that we are seeking. That should take the same amount of time as the base case.

$$
\begin{aligned}
T(n) &= T(n/2) + a \\
&= T(n/4) + a + a \\
&= T(n/8) + a + a + a \\
&= T(n/2^i) + ia \\
&= a + a \log_2 n
\end{aligned}
$$

which means that binary search is O(log n).

Misc topics in Recursion

O, Ω and Θ

Although most algorithms' time complexity will just be expressed in terms of time big -O, there are actually 3 related ways of measuring the time taken.

We have already seen that something like $O(n^2)$ just means that some constant times of n^2 effectively acts as an upper bound on the running time of the algorithm.

Big - omega (Ω) - A function $f(n)$ is said to be $\Omega(g(n))$ if and only if there exist n_0 and M such that $M|f(n)| \geq |g(n)|$ once $n \geq n_0$.

Big - theta (Θ) - The function $f(n)$ is $\Theta(g(n))$ if $f = O(g)$ and $f = \Omega(g)$.

Usually when programmers are expressing their time complexity they are talking about a big-Theta value. But they tend to say things like 'Order n squared' when it is $\Theta(n^2)$.

Fibonacci

The well known Fibonacci sequence can be defined by the following function

$$f(n) = \begin{cases} 1 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ f(n-1) + f(n-2) & \text{ow} \end{cases}$$

While the program becomes annoyingly slow if written in the most naive recursive manner, it is actually easier to analyze from a theoretical perspective when written as this recurrence.

A linear homogeneous recurrence relation of degree k has the following form

$f_n = c_1 f_{n-1} + c_2 f_{n-2} + \ldots + c_k f_{n-k}$

where the c_j's are constants that do not depend on n, and $c_k \neq 0$.

A standard way of trying to solve for these types of recurrences in closed form is to substitute x^n as f_n and see what happens.

For instance in the fibonacci case

$$\begin{aligned} f(n) &= f(n-1) + f(n-2) \\ x^n &= x^{n-1} + x^{n-2} \\ x^2 - x - 1 &= 0 \end{aligned}$$

The final equation that is obtained is called the characteristic equation. In this case it is a quadratic equation and it can be solved for roots which are

$$\frac{1+\sqrt{5}}{2} and \frac{1-\sqrt{5}}{2}$$

It is also easy to see (prove it by induction for a complete proof) that any linear combination of these two roots will satisfy the recurrence relation $f(n) = f(n-1) + f(n-2)$

So we know the closed form solution for fibonacci is

$$f(n) = c\left(\frac{1+\sqrt{5}}{2}\right)^n + s\left(\frac{1-\sqrt{5}}{2}\right)^n$$

How do we get the values of c and s? We use the base cases to say

$$c + s = 1$$
$$c\left(\frac{1+\sqrt{5}}{2}\right) + s\left(\frac{1-\sqrt{5}}{2}\right) = 1$$

Solving these two simultaneous equations we get the values of c and s and the closed form

$$f(n) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+1}$$

The amazing aspect of this is that fibonacci is a sequence of integers but the closed form contains terms that are filled with the irrational value $\sqrt{5}$.

Algorithm for solving recurrence relations in

1. convert the recurrence to a characteristic equation (can be done by claiming x^n to be the solution for the recurrence and working towards an equation)
2. find roots of the equation.
3. A theorem (proven in the zybook) shows how any linear combination of the roots will be a valid solution for the recurrence.
4. To get the values for the coefficients, use the base cases.

Recursion trees

When solving recurrences that involve dividing the initial input into halves, thirds etc, the best way to go about doing it is to to use the concept of the recursion tree.

See the handout on canvas for complete explanation. Look under the 'other resources' folder.

Master theorem

The recursion trees follow a general pattern and the more common ones can actually all be solved using a very powerful theorem called the master theorem that can be used quite easily for things like mergesort.

Theorem 1. *Let a and b be positive real numbers, with $a \geq 1$ and $b > 1$. Let $T(n)$ be defined for integers n that are powers of b by*

$$T(n) = \begin{cases} aT(n/b) + f(n) & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases}$$

Then we have the following:

1. *If $f(n) = \Theta(n^c)$, where $\log_b a < c$, then $T(n) = \Theta(n^c) = \Theta(f(n))$.*
2. *If $f(n) = \Theta(n^c)$, where $\log_b a = c$, then $T(n) = \Theta(n^c logn) = \Theta(f(n)logn)$.*
3. *If $f(n) = \Theta(n^c)$, where $\log_b a > c$, then $T(n) = \Theta(n^{\log_b a})$.*

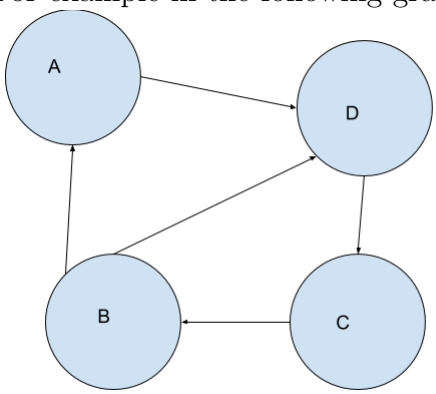
Graph Theory

What is a graph

A graph G is defined by a combination of a vertex set V and an edge set $E \subseteq V \times V$. It is common to see the notation

$G = (V, E)$

For example in the following graph



We will mathematically write $G = (V, E)$ where
$V = \{A, B, C, D\}$ and
$E = \{(A, D), (D, C), (C, B), (B, A), (B, D)\}$

Terminology

The two vertices that are part of an edge are its endpoints.

When a vertex is an endpoint of an edge we say the edge is incident on the vertex.

Note that two vertices connected by an edge is describing a relation (recall what that means from a previous part of the course).

A graph can either be directed meaning that there can be an edge from vertex u to vertex v without having an edge from vertex v to vertex u or undirected meaning that $(u, v) \in E \leftrightarrow (v, u) \in E$.

If we view this in terms of relations, an undirected graph is describing a symmetric relationship.

Loop(or sometimes self-loop) - an edge that joins a vertex to itself.

Complete graph - a graph where every vertex is connected to every other vertex.

A complete graph on n vertices is denoted as K_n.

Question
How many edges does a complete graph have?

That is the same as $\binom{n}{2}$ since for any pair of distinct vertices they have to be connected by an edge.

Paths and walks

For the remainder of the course we will be focusing on the following type of graph

- undirected
- no self loops
- no parallel edges
- finite number of edges

whenever we have to deviate from this type of graph it will be made clear.
Path - A path is an alternating sequence of vertices and edges that satisfy the following

1. it starts and ends with a vertex.
2. each edge joins the vertex before it in the sequence to the vertex after it in the sequence.
3. no vertex appears more than once.

A **walk** satisfies the first two conditions but not the third

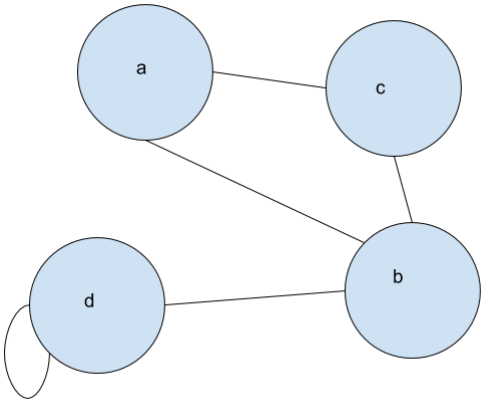
Result
If there is a walk between two vertices u and v, there has to be a path between vertices u and v.
Proof
Consider any walk between vertices u and v. There are two cases
Case 1 - There is no vertex that is repeated in the walk. In this case then by definition this walk is a path and we are done.
Case 2 - There exists a vertex x in this walk such that it is repeated. For this vertex, we can make a shorter walk by removing the part of the walk that is between the first and last occurrences of the the vertex. That is, include the last occurrence of x but not the first. Repeat the process of finding these repeated vertices and making shorter walks. Note that in each case we are eliminating one of the vertices from the list of repeated vertices. Therefore eventually we will have no repeated vertices in our walk and our walk will reduce to a path. (draw a picture to complete this proof).

The length of a path is the number of edges in the path.
There can be multiple paths between two vertices. An algorithms course is likely to spend a fair number of lectures talking about different methods to find shortest paths in graphs.

Degrees

The degree of a vertex is the number of edges that are incident on it.

Note that if you have a self-loop, it counts twice to the degree of the vertex. We are allowing self loops in the discussion of degrees.



In the above example the degree of vertex d, denoted by deg(d) = 3. deg(a) = 2 etc.

Theorem 1. *Sum of the degrees of the vertices is twice the number of edges.*

Proof

Proof by induction on the number of edges. Consider that we are given a graph with e edges and we want to prove the result holds for any e.

Base case: If the number of edges is 0 then the sum of the degrees is 0. Twice the number of edges is also 0. Done.

Induction hypothesis: Suppose that for $0 < k \leq e$ the result holds.

Now consider a graph with $e > 0$ edges. Let e_1 be some edge in this graph. Consider what happens when we delete this edge (note that deleting an edge means just deleting the edge and not deleting the associated vertices). We get a graph G' which has 1 less edge. So we can apply the induction hypothesis to it.

Applying the induction hypothesis says that the sum of degrees of the vertices in G' is $2(e-1)$.

Now let us consider 2 cases for the edge that we removed

Case 1: the removed edge was a self loop. When we add this edge back in the degree of one single vertex will go up by 2. So the sum of the degrees will be $2(e-1)+2 = 2e$.

Case 2: the removed edge joined two distinct vertices. When we add this edge back the degree of two vertices goes up by 1 each. Again the sum of the degrees will be $2e$.

Combining this with the base case of 0 edges, we have a complete proof by induction. QED!

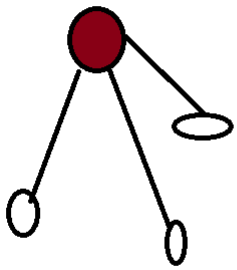
Using graphs to prove results

A famous theorem says that in any group of 6 people you are bound to find either a group of 3 distinct mutual friends or 3 distinct mutual enemies (obviously this assumes for any given pair of people they are either mutual friends or mutual enemies).

Proof

Consider one person. Call them Sally. Divide the rest of the people into 2 buckets - either they are friends of Sally's or enemies of Sally. Since we have 2 buckets and 5 people it must be the case that one of the buckets contains at least 3 people. Without loss of generality, let us assume the friends bucket has at least 3 people (if it has more than 3 it is even better for our proof).

Then let us draw the 'friendship' graph. It is going to look like this, with the red vertex being Sally.



Now consider what happens if we add any edge to this graph. That would immediately create a triangle (a K_3) which means that we have 3 distinct mutual friends and we are done with proving the result.

Consider what happens if we do not add any edge to this graph. That would mean that none of Sally's friends are friends with each other. That immediately means that we have 3 enemies.

So currently we have shown that if any person has 3 friends then the theorem is true.

The remainder of the proof is simple because we only have to worry about

a) what happens if an arbitrarily chosen person has more than 3 friends - this is easy because if that person has more than 3 friends we can pick any 3 of them and prove the result using the above argument.

b) what happens if the arbitrarily chosen person does not have at least 3 friends - this must mean that the chosen person has at least 3 enemies. That results in a similar argument to before with the words friend and enemy being interchanged.

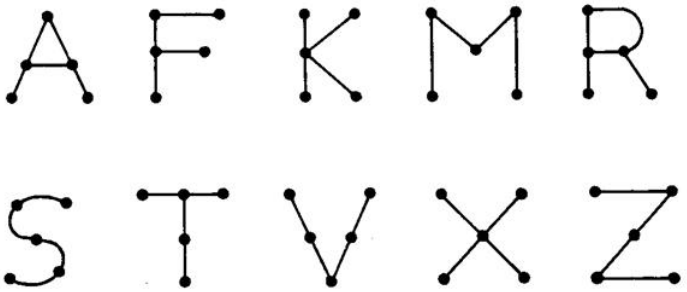
Graph Theory

Graph isomorphism

Let $G = (V, E)$ and $G' = (V', E')$. G and G' are isomorphic if there is a bijection $f : V \to V'$ such that for every pair of vertices $x, y \in V$, $(x, y) \in E$ if and only if $(f(x), f(y)) \in E'$. The function f is called an isomorphism from G to G'.

Another way of saying this is that G and G' are isomorphic if there is bijection between the vertex sets that preserves adjacency.

For an example, try an identify all pairs of isomorphic graphs in the following picture



Graph isomorphisms are actually one of the hardest problems (more on this in 596) that computer science has.

Why is that the case?

Given an mapping that claims to be an isomorphism it is easy to check if it truly is one or not.

However, what if someone asks you to find such a mapping?

How many one-one and onto mappings are there between the two vertex sets?

We know that if we have a bijection then the two vertex sets must have the same cardinality. So let us compute this in terms of $|V|$. We get $|V|!$. Obviously this is a function that grows really really fast so it is annoying to have to check all of these mappings.

There are certain tips that can help us detect whether two graphs could be isomorphic or not.

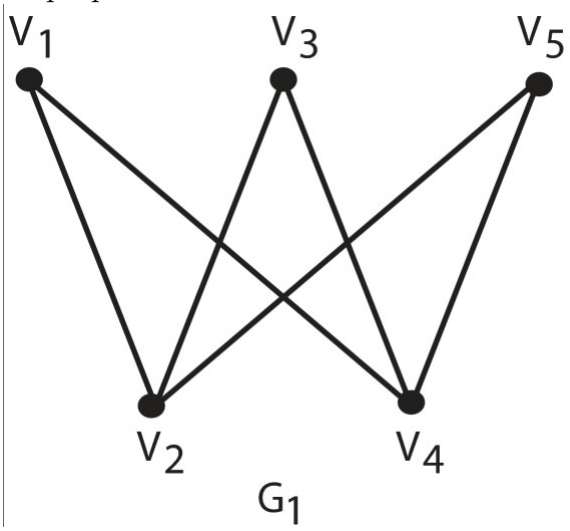
Here are some of them

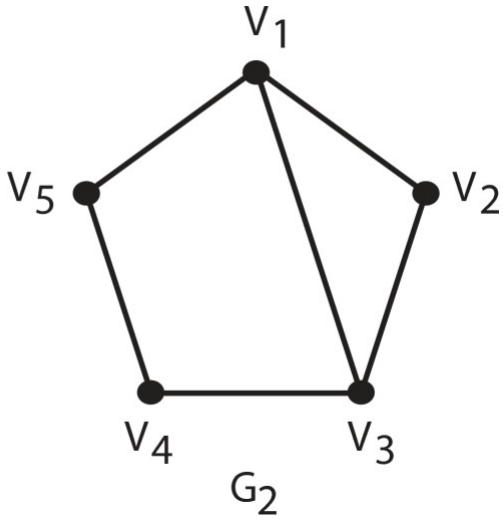
- For obvious reasons, the two graphs have to have the same number of edges and vertices.
- Under an isomorphism the degree of a vertex is preserved. So if the vertex v had a degree of k in graph G. Consider the isomorphism to be the function f. $f(v)$ which will be vertex in G' must have the same degree k.

- The degree sequence of two isomorphic graphs has to be the same. The degree sequence of a graph is a list of the degrees of all of the vertices in non-increasing order.

Here is an example of two non-isomorphic graphs which violate at least one of the properties above.

As an example here are two non-isomorphic graphs. They however are not violating any of the properties mentioned above





Circuits and cycles

A circuit is a walk in which the first vertex is the same as the last vertex. A sequence of one vertex, denoted $< a >$, is a circuit of length 0. Any circuit must have at least one repeated vertex because, by definition, the first and last vertices of a circuit are the same. A circuit is a cycle if it has length at least three and there are no additional repeated vertices, besides the first and the last.

Connectedness and connected components

A vertex v is said to be connected to vertex w in a graph G if there is a path in G from v to w. Since G is assumed to be undirected, a path from v to w implies that there is also a path from w to v. By definition, every vertex is connected to itself by a path of length 0. A vertex that is not connected with any other vertex is called an isolated vertex.

A graph is said to be connected if every vertex is connected to every other vertex in the graph and is disconnected otherwise. The property of being connected is a desirable property in many situations such as when the graph represents a road or communication network. If a graph is not connected, it has more than one connected component.

Trees

A connected acyclic (no cycles present) graph is called a tree.

Trees are one of the fundamental structures in computer science. You will learn about them in data structures. You will also learn about them in algorithms.

Properties of trees

We can show the following property

Result

Every tree with at least one edge has a leaf (vertex of degree 1).

Proof.

We do this proof by providing a method to find a leaf.

Consider any vertex of the tree. Find the longest path out of that vertex.

Claim that this longest path will terminate in a vertex of degree 1. This claim can be proved based on the fact that trees are acyclic. Wherever the longest path ends, we cannot have an edge from that terminal vertex back to any of the other vertices in the path because that would create a cycle.

Therefore it must be the case that the last vertex on this longest path will be a leaf.

Result

In any tree $e = v - 1$.

Proof by induction on the number of vertices.

Base case. 1 vertex and no edge. Easy.

Assume it works for k vertices.

Consider a $k + 1$ vertex tree.

It has a leaf because of the result above

Remove leaf and the one edge coming out of it.

We still have a connected acyclic graph (a tree). So the induction hypothesis applies.

Finish the rest of this...

Planar connected graphs

Show that v - e + f = 2