# Category Theory

# Category Theory

FOR THE UNEMPLOYED MATH UNDERGRAD

# So what is a category?

# Stage 0. First, let's talk about sets

- A set, intuitively, is any collection of things
  - "What's a collection then?" you ask.
  - "Don't worry about it" I condescendingly respond.

- The intuitive idea is that if A is a set, everything in the world -- including the number 5.4, the symbol ©, the meaning of happiness, and the set containing the set containing the empty set – is either in A or not in A.
  - Just, try to avoid self referential stuff like "the set of all sets that don't contain themselves"

# Now, let's talk about functions

- If A and B are sets, we often want to define a function f from A to B. A function is just a rule for assigning a single element of B to every element of A.
- So, if a is in A, then f(a) is in B
  - Note, two different things in A can be assigned to the same thing in B.
  - Also, there could be something in B that nothing in A gets assigned to.

- The one thing you do know is that every input gets exactly one output.
- Also, if f: A -> B and g: B -> C, then we can “compose” and get gf: A -> C
- Last minor point. You can always define id: A -> A by the rule id(a) = a

# Ok, now let's talk about the set $\{2\}$

- My favorite number is 2, so obviously my favorite set is $\{2\}$.

- Is there anything special we can say about the set $\{2\}$ just by looking at functions coming into or out of it?

# Yeah

- Let S be any other set. What to the functions from S to {2} look like?
  - Let f: S -> {2}. For any element x in S, f(x) is an element of {2}. But there's only one thing it can be, since the only element of {2} is 2.
  - So if f: S -> {2}, then f(x) = 2 for all x in S.
  - Existence: well, this rule indeed defines a function from S to {2}
  - Uniqueness: two functions from the same place to the same place are equal if and only if they do the same thing to all of the input elements. This is actually the definition of equality between functions. So this f is the only function from S to {2}, since any such function must follow the same rule that f does, and they would end up doing the same thing to all elements of S.

# New definition of {2} (…?)

- Lets try to do the following: my new definition of {2} is that it's the set such that, for any set S, there exists a unique function from S to {2}. I'm going to call this the "***final***" property. So {2} is the set with the final property. Does this work?

- Kind of, but not really. It clearly doesn't completely work since I never used the fact that the one element of {2} was 2, and not 4 or pi or the empty set or ©.

- Any other one element set satisfies my definition for the same reasons on the last slide. ***But*** any set with more than one element, like {2,3}, can't satisfy the definition, since for example we can make more than one function f, g: {5} -> {2,3} where f(5) = 2 and g(5) = 3

# More abstractly, why does this work?

- We can look at the sets directly and conclude that a set has the final property if and only if it has one element. But there’s an indirect way as well, and it ends up being more powerful.

- Let's say you had two sets, A and B, and they both had the final property.

- Because B has the final property:
  - There exists a unique function f: A -> B
- Because A has the final property:
  - There exists a unique function g: B -> A

# So we have f: A -> B and g: B -> A

- We can compose and get gf: A -> A. Is there anything else from A to A?
  - Yeah, the identity id: A -> A defined by the rule id(a) = a, for all a in A.
- But A is final! For any set S, there is a unique function from S to A
  - Let S be A. So, since id and gf are both from A to A, they must be the same
  - Technical Note: when we say A is final, that means any set, *including itself,* has a unique function to it. There’s no reason why we can’t let S be A! Look at the proof that there's a unique function from X to {2}: it didn't matter whether or not X = {2}, just that there’s only one choice of output! this confuses people sometimes.
- Since fg: B -> B, and so is id, then since B is final, fg equals id

# So, $gf = id$ and $fg = id$

- These functions going back and forth from A to B cancel each other out. We say they are inverses. So, we can match up the elements of A with the elements of B by using this relationship.
  - Given some a in A, match it with f(a) in B.
  - Then, for any b in B, it will be matched up to g(b) in A.
- Basically, f is a relabeling, and A and B have the same size!
- (in rigorous set theory, having these inverse functions is actually the definition of two sets being the same size, because they're just relablings )

# Let’s recap this final property of singletons

- We started with a definition of something called the final property for sets. The property talked about functions, and not elements.
- Just using abstract reasoning, and just by talking about functions, and not elements, we proved that if two sets both satisfied the final property, there must be inverse functions going between them.
- So, reinterpreting using elements, the final property indirectly characterized the sets with one element.
- You just learned a “universal property”.  They tend to define objects indirectly, and “up to isomorphism”, or in this case, up to bijection.

# What’s the universal property of AxB?

- If A and B are sets, then we can form their cartesian product AxB
  - AxB = { (a,b) | a in A, b in B}. This is the definition that uses elements

- Is there a way of talking about AxB just using functions?
- What are some special functions associated with AxB?
  - Projection into the first coordinate, fst: AxB -> A, defined by fst(a,b) =a
  - Projection into the second coordinate, snd: AxB -> A, defined by snd(a,b) =b
- What’s so “special” about them? Are they even really special?

# Yeah

- So AxB has a function into A and another into B. Indulge me, what if we have *another* set X with some functions f: X -> A and g: X -> B?

- Lets see if there's any connection between X and AxB, like last time.

- If there happened to be a function from X to AxB, then there would be two ways of getting from X to A:
  - You can go right from X to A using f
  - You can go from X to AxB, followed by fst
  - Likewise, there would be functions from X to B

# Lets try to get a special h: X -> AxB

- I want an h so that it doesn’t matter which path we take from X to A, or from X to B.
  - Basically, I want an h so that: fst h = f and snd h = g
  - Does h exist? If it did, would it be unique? Can you see what I’m getting at?

- Let’s look at the elements of these sets and try to come up with an h.
  - For any x in X, h(x) is in AxB. So h(x) looks like some (a,b).
  - We want fst h = f, so fst h(x) = fst (a,b) = a should be the same as f(x)
  - We want snd h = g, so snd h(x) = snd (a,b) = b should be the same as g(x)
  - So if we want a = f(x) and b = g(x) for all x in X
  - Let’s just define h(x) = ( f(x) , g(x) ) for all x in X.

$$\begin{tikzcd} & X \arrow[dl, "f_1"'] \arrow[d, dashed, "f"] \arrow[dr, "f_2"] & \\ A & P \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & B \end{tikzcd}$$

# Universal property of AxB

- We can say that if A and B are two sets, then “a” product AxB, along with the projection maps fst and snd, is a set such that:
- for any set X and for any functions f: X -> A and g: X -> B, there exists a unique function h: X -> AxB such that this diagram “commutes”.
- Instead of h however, we usually call it <f,g>
- For the same reason as with the final property, if two different sets P1 and P2 both satisfy this definition of a product of A and B, then we get (unique) inverse functions between P1 and P2. So, from a function perspective, P1 and P2 “work the same way”. They’re indistinguishable

# Indistinguishable to whom? To a category theorist!

- Category theory is about putting emphasis on relationships between objects, not necessarily on their internal properties. Like, “what can I say about this object *relative to other objects of the same type*?

- Category theory grew out of observations that there were deep similarities between very different areas in math, like abstract algebra and topology. Even though the objects themselves are very different, the way they “connect to each other” is similar

- So, roughly, category theory is the mathematical study of mathematical objects. Lets explore the idea of “object” further.

# Big theme: idea becomes encapsulated by an object

- Linearity encapsulated by vector spaces
- Continuity encapsulated by topological spaces
- Symmetry encapsulated by groups
- Smoothness encapsulated by manifolds
- “Volume” encapsulated by measurable spaces (and sigma algebras)
- Connectivity encapsulated by graphs
- Relatedness encapsulated by posets and lattices
- Composition encapsulated by a category
- Things that might happen encapsulated by a probability space

# Big theme: idea becomes encapsulated by an object

- Linearity encapsulated by vector spaces
- Continuity encapsulated by topological spaces
- Symmetry encapsulated by groups
- Smoothness encapsulated by manifolds
- “Volume” encapsulated by measurable spaces (and sigma algebras)
- Connectivity encapsulated by graphs
- Relatedness encapsulated by posets and lattices
- Composition encapsulated by a category
- Things that might happen encapsulated by a probability space
- Redundancy vs Independence encapsulated by a matroid

# Big Theme: we care about functions that "preserve" the object's structure

- Linear transformations preserve vector addition and scalar multiplication
- Continuous functions preserve open sets…kind of.
- Group homomorphisms preserve the group operation.
- Smooth (infinitely differentiable) functions preserve manifold structure.
- Measurable functions…do stuff too? Ask Jason
- Graph homomorphisms preserve the "vertex touching edge" relation.
- Monotone functions preserve relatedness in a poset or lattice. Lattice homomorphisms also preserve joins and meets.
- Functors preserve "category structure" like domains, codomains, and identities
- Apparently a probability space is just a measure space with total measure 1

# Big Theme: we care about functions that “preserve” the object’s structure

- Linear transformations preserve vector addition and scalar multiplication
- Continuous functions preserve open sets…kind of.
- Group homomorphisms preserve the group operation.
- Smooth (infinitely differentiable) functions preserve manifold structure.
- Measurable functions…do stuff too? Ask Jason
- Graph homomorphisms preserve the “vertex touching edge” relation.
- Monotone functions preserve relatedness in a poset or lattice. Lattice homomorphisms also preserve joins and meets.
- Functors preserve “category structure” like domains, codomains, and identities
- Apparently a probability space is just a measure space with total measure 1
- Matroid homomorphisms… might be a thing?

# Examples

- If T: V -> W is a linear transformation between vector spaces, then
  - T(x + y) = T(x) + T(y)    and    T(cx) = cT(x)

- If f: X -> Y is a continuous function between topological spaces, then
  - Whenever V is an open subset of Y, the preimage of V under f is an open subset of X

- If m: P -> Q is a monotone map between posets, then
  - For all x and y in P, if x <= y, then m(x) <= m(y)

# Notice, they are all designed to compose

- If T: U -> V and S: V -> W are linear transformations between vector spaces
  - ST(x + y) := S(T(x + y)) = S(T(x) + T(y)) = S(T(x)) + S(T(y)) =: ST(x) + ST(y)

- If f: X -> Y and g: Y -> Z are continuous functions between topological spaces
  - W an open subset of Z ==> (g inverse (W)) an open subset of Y ==>
  - (f inverse (g inverse (W))) an open subset of X
  - But (f inverse (g inverse (W))) is just (gf inverse (W)), so this means gf is continuous

# So it's not just sets and functions!

- In some sense, we now have to define what a category is so that we can ***apply*** all of our universal property ideas to objects other than sets. A category will be kind of like an “environment”, or “universe” that we will live in, and in which we can wield universal properties to describe stuff.

- The definition of a category is meant to convey the idea of “a collection of things, and some composable things between the things”.

- So lets abstract all of the situations where we have objects and composable functions between them and make a general definition.

# Definition of a category

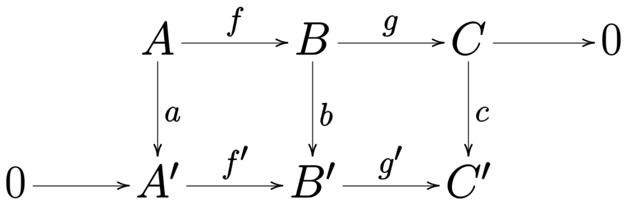
- A category, call it C, is 2 things
  - A collection of objects, obj(C)
  - For any 2 objects, A and B in obj(C), a collection Hom(A,B).
    - These are the "morphisms" or "arrows" or "maps" from A to B.
    - These are disjoint, so if anything is in both Hom(A,B) and Hom(C,D), then A=C and B=D. That's just saying every arrow has an unambiguous starting and ending object
  - We have certain things we can do with these objects and morphisms:
  - If A, B, and C are objects, and f is in Hom(A,B) and g is in Hom(B,C) then there exists something called gf in Hom(A,C). We call it the composition of f and g
  - Every object A has a so called identity morphism, id, which is in Hom(A,A)

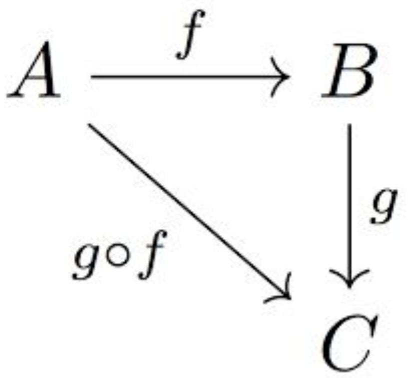
# Rules a category has to satisfy

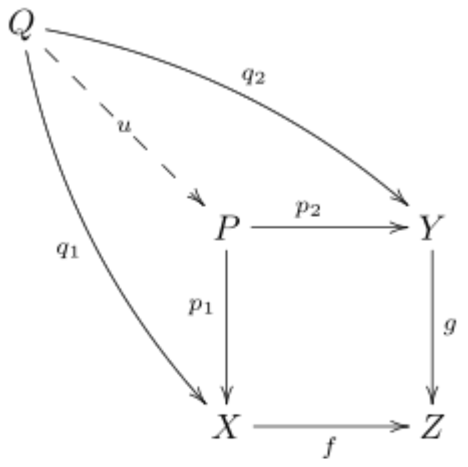
- Composition and identities have to work well for this definition to be of any use to us. So we require:
  - Composition is associative. That is, $h(gf) = (hg)f$ whenever that expression makes sense (meaning, g starts where f ends, and h starts where g ends, so composition can actually be defined).
  - Identities don't affect composition. So if $f: A \to B$, then $f\,id = id\,f = f$
    - Same as identities in a group, like 0 for addition and 1 for multiplication
    - Only difference is you need to keep in mind what the domains and codomains are.

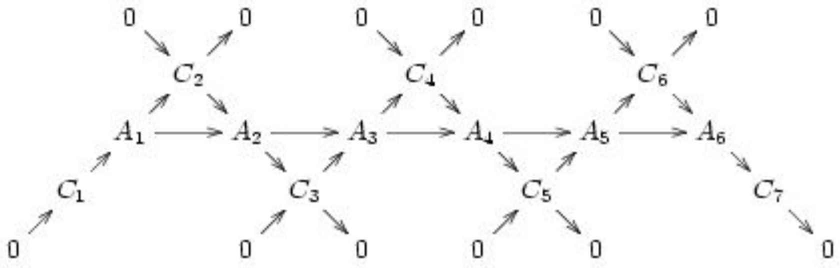
# We can use diagrams to draw all of this out

- Diagrams are awesome. Diagrams are to category theory what equations are to algebra. Diagrams can provide a lot of information at once. To assert that a diagram “commutes” is kind of like saying an equation is actually true, and not just an expression that’s either true of false.
- Objects are letters, and morphisms are labeled arrows



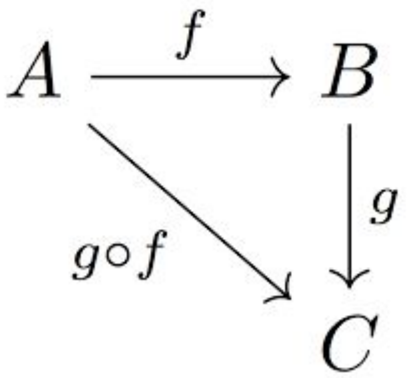



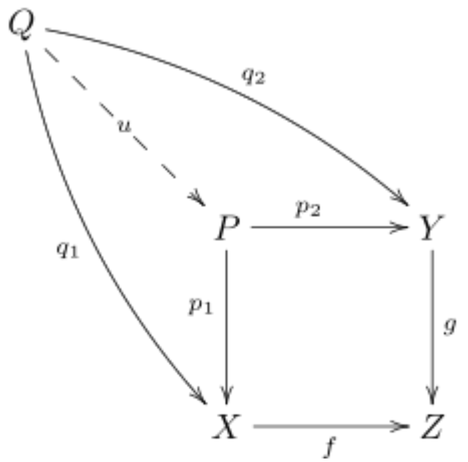



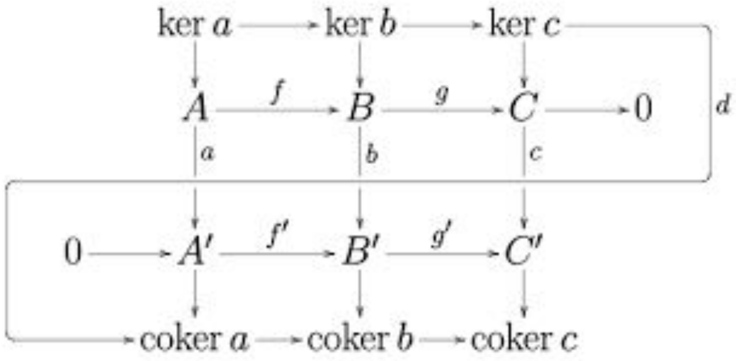



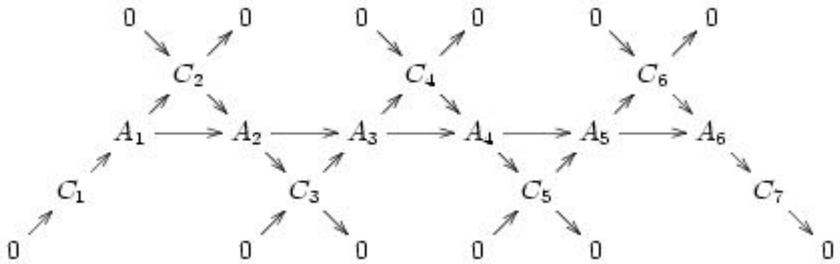
# We can use diagrams to draw all of this out

- Diagrams are awesome. Diagrams are to category theory what equations are to algebra. Diagrams can provide a lot of information at once. To assert that a diagram “commutes” is kind of like saying an equation is actually true, and not just an expres true of false.
- Objects are letters, and morphisms are labeled arrows



# Great, we’re at Stage 1. We know the definition. What can we do with this?

- We can go out and immediately test to see if certain types of objects form a category.
- The category whose objects are sets and whose morphisms are normal functions is called **Set**.
- The category whose objects are vector space (over the real numbers), and whose morphisms are linear transformations is called **Vect**.
- Can you guess what **Top** is? How about **Man**? What about **Pos**? **Grp**? **Grph**?
- What about a category whose objects are metric spaces (a set with a distance function that satisfies positive definiteness, symmetry, and the triangle inequality)? What should the morphisms between metric spaces be? There’s at least 4 options that various people care about.

# We can learn more definitions! And sound smart!

- How many prefixes for morphism do you know? Iso? Homeo? Endo? Auto?

- An isomorphism is a morphism f: A -> B such that there exists another morphism g: B -> A where gf is the identity on A and fg is the identity on B. We say g is f inverse, and vice versa, and we call A and B isomorphic.

- An endomorphism is a fancy word for a morphism from some object to itself. An automorphism is just an endomorphism that's also an isomorphism.

- We don't talk about homeomorphisms…

# We can also look for universal properties

- The “final” definition defined an object “up to isomorphism”. Meaning, two different objects could satisfy it, but they are forced to be isomorphic.

- Generally, a universal property is some definition, typically of the form “an object is a blablabla if for all other objects satisfying blabla, there exists a unique morphism such that blablablablabla diagram commutes.”

- The final object definition was a particularly simple example, but it illustrates the point that universal properties are always crafted so that they define objects up to isomorphism.

- FYI, if a category has final objects, we usually refer to them with the symbol 1. Up to isomorphism, we can talk about “the” object 1 in obj(C).

# A word on isomorphisms in particular

- In the category of sets, these are just bijections. In the category of groups, these are bijective group homomorphisms (this isn’t immediately obvious). In the category of topological spaces, these are homeomorphisms.

- Different areas of math study different properties of objects. To a geometer, angles and lengths matter, but a topologist only cares about “deeper” properties, like whether your space is connected, or has holes.

- Even is some areas (graph theory) don’t care as much about morphisms, they definitely care about isomorphisms. Every area of math has it’s own notion of isomorphism. E.g, homeomorphic really means “indistinguishable from the perspective of topology”.

- Sometimes, morphisms are defined just so that isomorphism mean “the right thing”.

# We can also look for weird categories! Some defs for the next slide

- A group G is a set along with an operation, call it *, that takes two elements of G and returns a single element, and it satisfies:
  - 1)The operations is associative, so a*(b*c) = (a*b)*c
  - 2) There exists an element e in G such that for all g in G, we have that e*g = g*e = g
  - 3) given any g in G, there is some element h in G where g*h = h*g = e
- A monoid is any set with an operation that just satisfies the first 2 axioms
- A preorder P is a set along with a relation, call it <=, that takes two elements of P and returns either true or false, satisfying
  - Reflexivity: for all x in P, we have x<=x.
  - Transitivity: for all x, y, z in P, we have that if x<=y and y<=z, then x<=z
- A partial order is a preorder that also satisfies antisymmetry: whenever x<=y and y<=x, then x=y
- An equivalence relation is a preorder with symmetry: whenever x<=y, then y<=x

# Monoids and groups as categories

- Monoids and groups as categories themselves: suppose you had just one object, *, in obj(C). Then every morphism f is in Hom(*,*), and any two morphisms are composable. The morphisms of C form a monoid! (group without inverses). Likewise, any monoid can be converted to a category with one object (label the arrows and define composition accordingly)
  - Since C is a category, composition is already associative, and the identity on * would be the identity of the monoid, so the monoid axioms check out.

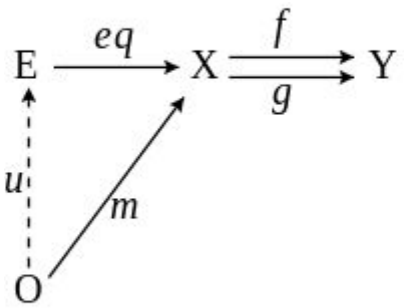
- Think, what would it take for a single object category to be a group?

# Preorders as categories

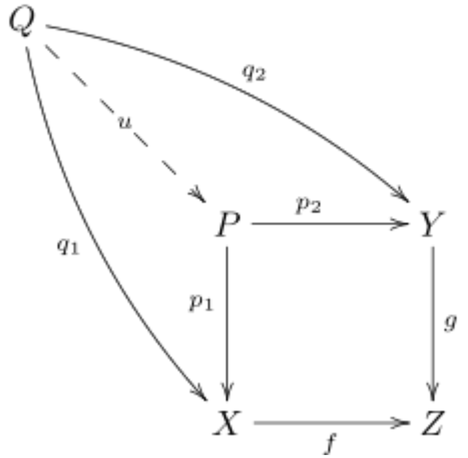
- What if we have a category where any objects have at most one arrow between them? Define a relation <= on the objects. A<=B if there's an arrow from A to B (or B to A, the direction doesn't matter much)
  - There's always an identity arrow from A to A for this to be a category, which gives us reflexivity
  - If there's an arrow from x to y, and another arrow from y to z, then composition gives us an arrow from x to z, which means the relation <= is transitive.
- Likewise, a preorder P can be converted into a category: the objects of the category are defined as elements of P, and whenever x<=y, we let there be an arrow from x to y in the category.
- So preorders are basically the same thing as "thin" categories – categories where every Hom set is either empty or a singleton
- What would it take for a category to be a partial order? What about an equivalence relation?

# We can learn more universal properties

- If f and g are both morphisms from X to Y, then the equalizer of f and g is a new object, E, along with a morphism eq: E -> X, such that f eq = g eq, and
  - E and eq are supposed to be universal with respect to this “mapping into A such that following up with f or g gives the same thing” property.
  - Explicitly, for any object O and m: O -> X, if fm = gm, then there exists a unique u:O -> E making the diagram commute (meaning eq u = m)




- If we have two morphisms, f: A -> C and g: B -> C, with the same codomain, we can form the pullback P, with morphisms p: P -> A and q: P -> B, such that f p = g q,
  - and where P, p, and q are somehow “universal” with respect to that property.
  - Try to guess the condition here



# We can also save time with duality – reverse all the arrows

- If C is a category, then so is C^op. The objects are the same, but Hom(A,B) in C^op is defined as Hom(B,A). If a definition or theorem works for all categories, then we get a new definition in C by applying the old definition to C^op

- Initial objects: by definition, A is an initial object in C if for all objects Y in C, there exists a unique morphism from A to Y.

- Notice, A is initial in C precisely when it is final in C^op. So duality says we know that any two initial objects are isomorphic to each other.

# What happens when we reverse the arrows in the “product diagram”?

- Try to reverse the arrows by hand, reinterpret what the diagram is telling you, and see if you can come up with the definition of the so called “coproduct”.
- Think about what the coproduct of A and B, called A+B, looks like in your favorite category. Does it even exist? (yes, I already expect you to have a favorite category).
- This is a good time to let everything sink in. I’ll wander around and answer questions for 10 minutes

# Problems

1. Prove products are unique up to isomorphism using the same idea as for final objects
2. Draw the diagram for products, reverse all the arrows, and guess the definition of coproducts
3. How many different categories can you list? $1 to the person who lists the most correct examples.
4. What do (co)products and initial and final objects look like in these categories, if they exist? What about in Set, Vect, and Top in particular (Top might be a hard one)?
5. Convert the partially ordered set (N,|), natural numbers with 0 under the relation divisibility. What do the (co)products and initial and final objects look like, if they exist?
6. Show that (AxB)xC is isomorphic to Ax(BxC) if A, B, C are all objects in some category with products.
7. Show that if 1 is a final object, then Ax1 is isomorphic to A. What is the dual statement?
8. Recall the definition of equalizer and pullback, and look up the definition of monic and epic arrows. What do monic and epic functions look like in Set? Why? Find an equivalent def of monic with pullbacks
9. Show that if eq: E -> A is an equalizer, then eq is monic
10. Show that eq: E -> A is an equalizer that is also epic, then it is an isomorphism
11. If f: A -> B in Set, and C is a subset of B, formulate the preimage of C using a pullback
12. Look up and prove at least 1 direction of the pullback lemma

# You did as much as you could as a Stage 1 category learner. Let’s zoom out

- You know some definitions. You can tell if stuff is a category or not. You can ask what final and initial objects look like in various categories, or what products and coproducts look like.
- You don’t know the full extent of what these definitions buy you.
- You might not realize what the potential philosophical implications of category theory are, and how powerful the language can be.
- Let’s dive into these.

# But you have yet to break free of dealing within single categories

- You hear advanced sounding phrases like
  - “A monad is a monoid in the category of endofunctors”
  - “By the Eckmann-Hilton argument, a monoid object in the monoidal category of monoids is a commutative monoid”
  - For path connected spaces, any choice of basepoint yields an isomorphic fundamental group, up to natural isomorphism if the group is abelian
  - A vector space is naturally isomorphic to its double dual space. A finite dimensional vector space is isomorphic to its dual space, but not naturally.
  - The Yoneda Lemma is a massive generalization of Cayley’s theorem, and basically says a category embeds fully faithfully into its category of presheaves
  - Metric spaces are just categories enriched over [0,infinity)
  - “Adjoint functors are everywhere” and “Everything is a Kan extension”

# You’re a solid Stage 2 category theory enthusiast. Here what I would tell you

- It was actually very “uncategorical” of us not to ask the question: what are the sensible morphisms *between* categories? They’re called functors.
- What should functors do? Preserve “category structure”? What does that mean? Since a category has objects and morphisms, a functor F from a category C to a category D should probably take objects in C to objects in D, and also take morphisms in C to morphisms in D.
- What’s going on philosophically? A famous physicist John Baez said “Every sufficiently good analogy is yearning to become a functor”…

# Philosophy of Functors

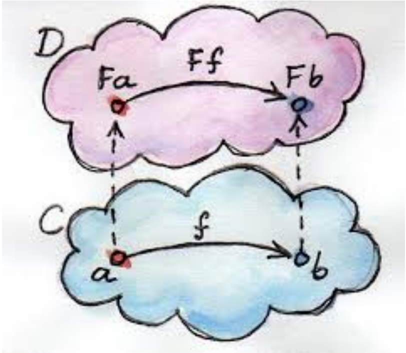
- If math is about ideas, and ideas are represented with objects, and objects live in a category with all other objects of the same type, then a category is like a “theme”, or a “type of idea”.

- Since functors convert one type of object into another type, and they convert “connections between *these* objects” to “connections between *those* objects”, functors really are like analogies. And furthermore, they’re like good analogies, since they preserve relationships.

# Definition of a functor

- If C and D are categories, then a functor F: C -> D works as follows
  - If A is an object in C, then F(A) is an object in D
  - If g: A -> B is a morphism in C, then F(g): F(A) -> F(B) is a morphism in D.

- What would it mean to respect "category structure"? It means that F respects composition and identity arrows (and domains and codomains)
  - F(gf) = F(g)F(f) and F(id(A)) = id(F(A))



# Your first proof relating to functors

- Prove that if A and B are isomorphic objects in C, and F: C -> D is a functor, then F(A) and F(B) will be isomorphic objects in D.

- A and B are isomorphic, so there exist f: A -> B and g: B -> A where gf and fg are both identities on A and B respectively.

- We want isomorphisms from F(A) to F(B). Let’s guess F(g) F(f) works.
- F(g) F(f) = F(gf) = F( id(A) ) = id( F(A) ), so F(g) F(f) is the identity on F(A)
- Likewise, F(f) F(g) is the identity on F(B). So F(A) and F(B) are isomorphic

# Examples of functors

- Pretty much any category of sets with structure and structure preserving functions between then will have a “forgetful” functor into **Set**. It's basically just ignoring the structure, and taking the underlying set.
  - A concrete example might be U: Vect -> Set where if V is a vector space, then U(V) is the underlying set of that vector space, and if T is a linear transformation, U(T) is a normal function between sets.
- Whenever one category is a “subcategory” of another, you usually have the inclusion functor. Like I: Ab -> Grp.
- The powerset of a set is actually a functor, P: Set -> Set. It takes a set S to the set of its subsets, P(A) = { S | S is a subset of S}.
  - What would it do to functions between sets? If f: A -> B, then what should P(f): P(A) -> P(B) be?

# Examples of “nontrivial” functors

- Given a topological space X and p, a fixed element of X, then there’s a group pi(X,p), called the fundamental group of X with basepoint p, which is the set of loops in your space X going through p. The group operation is concatenation of loops (which is associative up to homotopy, so technically, pi(X,p) is the set of *homotopy classes* of loops in X based at p, but that’s not important for us).

- Given a set S, we can ask for the “free group” on that set, F(S), and this turns out to be a very special functor. As a concrete example, F({*}) = Z.

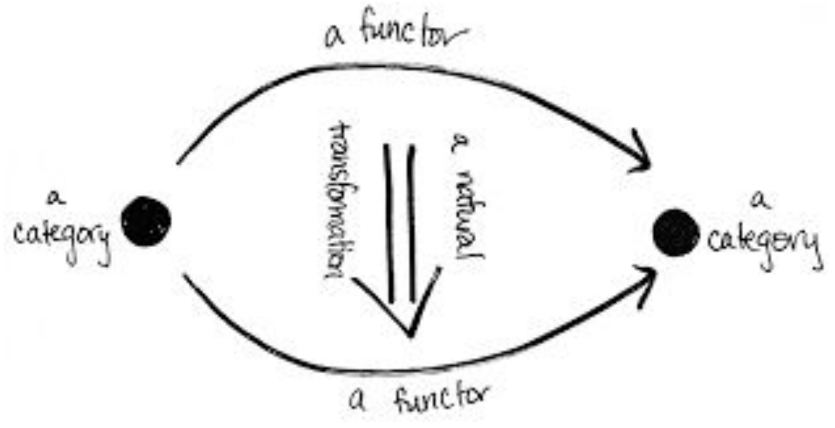
- Group abelianization: Given a group G, you can turn it into an abelian group G’ by quotienting out by commutator subgroup [xy(x^-1)(y^-1)]. This is a functor from Grp to Ab.

# Contravariant functors

- Sometimes, converting one object to another naturally reverses the order of composition. Like, F(gf) would become F(f) F(g). Is this ok?
  - A contravariant functor F from C to D can be formally expressed as a functor F: C^op -> D, so yes, this is ok.
- Examples might include the contravariant powerset functor P: Set^op -> Set that takes a set A to it's powerset P(A), but if f: A -> B is a function, it would need to take f to P(f): P(B) -> P(A). So it's a function that takes a subset of B to a subset of A. How might it be defined?
- Another example. If X is a topological space and U is an open subset of the space, then let C(U) be the set of continuous real valued functions defined on U. Convert the collection open sets of X into a poset category called O. Then, C: O^op -> Set. Why? If V is a subset of U, how do we get a relationship between C(V) and C(U)?

# Functors are morphisms between categories. What goes between functors?

- If you got the hang of the "relationship focused" mentality of category theory, you might have already asked this question to yourself.
- If F and G are both functors from C to D, there's a sensible notion of something from F to G. It's called a natural transformation. Turns out, these compose, and there are identity natural transformations, so we can talk about [C,D]! This is the category of functors from C to D, where the morphisms are natural transformations between then.



# Natural Transformation definition

- I’ll use the letter a, but it should be alpha. If a: F -> G is a natural transformation between functors from C to D, let's think about what that should mean.
- If we have an object A in C, we can get two objects in D
  - These are F(A) and G(A). So maybe, saying that a: F -> G should mean that there’s some morphism from F(A) to G(A), for all A in C.
- If we have some morphism f: A -> B in C, then there’s a lot going on!

# What should a natural transformation do to morphisms?

- Well, if a: F -> G, a natural transformation between functors going from C to D, and f: A -> B a morphism going between objects in A, then we're going to have the objects F(A), F(B), G(A), and G(B). It turns out there's two different ways of getting from F(A) to G(B) using some morphisms in D!
  - We can go from F(A) to G(A) via a(sub A) and from G(A) to G(B) via G(f)
  - We can go from F(A) to F(B) via F(f) and from F(B) to G(B) via a(sub B)

- For the transformation to be considered "natural", these have to coincide. Basically, for all morphisms f in C the resulting square commutes.
- Idea: if they were different, then this introduces an arbitrary choice! (bad)

# Naturality square drawn out

- Whenever eta: F -> G
  and F and G are functors
  from C to D, and f: A -> B
  is a morphism in C,
  then this diagram commutes

$$
\begin{tikzcd}
F(X) \arrow[r, "F(f)"] \arrow[d, "\eta_X"'] & F(Y) \arrow[d, "\eta_Y"] \\
G(X) \arrow[r, "G(f)"'] & G(Y)
\end{tikzcd}
$$

# Example: every vector space V is naturally isomorphic to its double dual

- If V is a vector space over the real numbers, then V* is its "dual", defined as V* = { f: V -> **R** |f is a linear transformation}. Sometimes called functionals over V.
- (_)* is actually a contravariant functor from Vect to Vect! What does it do to maps? If T: V -> W is a linear transformation, then how should T*: W* -> V* be defined?
  - Given some element of W*, so given some linear f: W -> R, we can precompose with T to get fT: V -> R.
- We can compose the functor (_)* with itself and get (_)**

# The idea of a morphism from V to V**

- Given an element of V, how do we get an element of V**? This will be a linear transformation that takes an element of V* and returns a real number.
- Fix x as an element of V. Define a function EVx, called evaluation at x.
  - EVx takes an f: V -> R, which is a thing in V*, and returns f(x), which is indeed a real number.
  - So EVx: V* -> R, and we can show it's linear. This means EVx is an element of V**.
  - So that means EV(_):V -> V** is defined by taking x to EVx, which takes f to f(x)
  - That's a mouthful! But if keep strait what's what, it's just EVx(f) := f(x)

- It can be shown that EV(_) is a linear transformation from V to V**
- The inverse is hard to give explicitly, but if V is finite dimensional, then for dimension reasons, it exists. If V is infinite dimensional, then V** "contains" a copy of V in a natural way, but it might contain more stuff!

# Not only is V isomorphic to V**, but it's isomorphic naturally (V finite dim)

- The categorical way of stating this that doesn't depend on ny choice of vector space is as follows:

- The identity functor on Vect and the (_)** functor on Vect are naturally isomorphic. This means there are natural transformation from one to the other that compose up to be the identity functor (i.e., they're isomorphic objects in [Vect,Vect].

$$\begin{tikzcd} \mathrm{Vect}_{\mathbb{k}} \arrow[r, bend left=50, "\mathrm{id}"{name=U}] \arrow[r, bend right=50, "(-)^{**}"'{name=D}] & \mathrm{Vect}_{\mathbb{k}} \arrow[from=U, to=D, Rightarrow, "\mathrm{eval}"'] \end{tikzcd}$$

# Stage infinity, and where to go from here

- Part 2 on adjoint functors? Examples usually look like "freely" making one type of object out of another. Good source of many universal properties.

- Special categories called topoi in which we can define an internal logic, and deal with topos semantics over a propositional logic.

- Applied category theory! I randomly went to California and saw a bunch of talks on that, and they ranged from categorical models of the neurons in our brain to functorial clustering algorithms in data science.

- Connection with functional programming languages like Haskell, and especially the idea of a monad (which is just a functor from C to C and some special natural transformations)

# Books to read and things to watch

- Algebra Chapter 0 as a grad algebra book that focuses on universal properties
- Basic Category Theory by Leinster
- Category Theory in Context by Riehl
- Category Theory for the Working Mathematician by Mac Lane
- The Catsters YouTube channel (Eugenia Cheng)
- Bartosz Milewski on Youtube, or the book Category Theory for Programmers

# Bartosz Milewski says “Thanks for coming to Alek’s talk!”