Category theory

Q: What is category theory?

A: Study of objects based on how they interact with each other.

$\rightsquigarrow$ External perspective

$\rightsquigarrow$ Usually as powerful and more than an internal perspective.

A: Since we take high-level view, can compare whole categories of things (Curry-Howard)

Q: What is a category?

A: Roughly:
- a collection (set, type, etc) of <u>objects</u>
- a collection of <u>morphisms</u> between objects

<u>Ex</u>. The category Set has objects: sets
morphisms: functions

<u>Ex</u>. The category Prop has objects: propositions
morphisms: implication / derivation

<u>Ex</u>. The category Ty has objects: types
morphisms: functions

<u>Ex</u>. Grp objects: groups
morphisms: group homomorphisms

Vect objects: vector spaces
morphism: linear transformations

<u>Ex</u>. ProgSpec has objects: program specifications
morphisms: way to produce a program meeting one specification from any program meeting another

<u>Ex</u>. Lang has objects: programming languages
morphisms: translations / compilations

(1)

...

More formally:

<u>Def</u>. A <u>category</u> $\mathcal{C}$ consists of
- a collection $\operatorname{ob}\mathcal{C}$ of the <u>objects</u> (usually denoted $X, Y, Z, \ldots$)
- for each pair $X, Y \in \operatorname{ob}\mathcal{C}$ a collection
  $\hom_{\mathcal{C}}(X,Y)$ of <u>morphisms</u> (a map) from X, Y
  (usually denoted $f : X \to Y$ or $g, h, \ldots : X \to Y$)

together with
- for each object X, an <u>identity</u> morphism $id_X : X \to X$
- for each triple of objects X, Y, Z, morphisms $f : X \to Y$, $g : Y \to Z$, a <u>composition</u> morphism $g \circ f : X \to Z$
  (note the order of $g \circ f$!)

such that
- for every $f : X \to Y$, we have
  $f \circ id_X = f = id_Y \circ f$ (left and right identity)
- for every $f : X \to Y$, $g : Y \to Z$, $h : Z \to A$, we have
  $h \circ (g \circ f) = (h \circ g) \circ f$ (associativity)

(2)

<u>Ex</u>. (identity) Usually $\lambda x. x$. In Prop, given axiomatically.

<u>Ex</u>. Associativity in Set: $g \circ f$ is $\lambda x. g(f(x))$

so $h \circ (g \circ f)(x) = h((g \circ f)(x))$
$= h(g(f(x)))$

$(h \circ g) \circ f = (h \circ g)(f(x))$
$= h(g(f(x)))$ ✓

In Prop: Say $\hom(P, Q) = \begin{cases} \{T\} & \text{if } P \to Q \\ \emptyset & \text{otherwise.} \end{cases}$

So if $T : P \to Q$, $T : Q \to R$, $T : R \to S$, can derive first that $T : P \to R$ and then $T : P \to S$ or first that $T : Q \to S$ and then $T : P \to S$.
In either case it is $T : P \to S$.
- In other words $T \circ (T \circ T)$, $(T \circ T) \circ T : P \to S$
  But there is (at most) one element in $P \to S$, so these must be equal.

<u>Notn</u>. Category theory becomes more readable via diagrams.

left identity:

$$\begin{tikzcd} X \arrow[r, "id_X"] \arrow[dr, equal] & X \arrow[d, "f"] \arrow[dl, phantom, "\circlearrowleft"] \\ & Y \end{tikzcd}$$

more arbitrary example:

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "h"'] \arrow[dr, phantom, "\circlearrowleft"] & Y \arrow[d, "g"] \\ W \arrow[r, "i"] & Z \end{tikzcd}$$

This square/diagram <u>commutes</u> if $g \circ f = i \circ h$.

Can write $\circlearrowleft$, but usually omitted.

See also <u>string diagrams</u> (MLL proof-nets are very similar)

(3)

Terminal and initial objects

We not only understand objects by their relations, but define/specify them via their relations.

<u>Def</u>. Let $\mathcal{C}$ be a category, and $X \in \mathcal{C}$. Say that X is <u>terminal</u> if there is exactly one morphism $Z \to X$ for all $Z \in \mathcal{C}$.

<u>Ex</u>. What is the terminal object of Set?

(These proofs are usually guess-and-check.)

The singleton is terminal: there is indeed exactly one map from any set to the singleton.

<u>Ex</u>. Prop?

T, the true proposition

<u>Ex</u>. Ty? unit

Isomorphism

We regard two objects in a category $\mathcal{C}$ as "the same" if they are isomorphic. (We disregard equality.) ↑ justified momentarily

<u>Def</u>. Consider $f : X \to Y$ in a category $\mathcal{C}$. Say f is an <u>isomorphism</u> if there is another map $g : Y \to X$ s.t. $g \circ f = id_X$ and $f \circ g = id_Y$.

Often write f^{-1} for g, $f : X \cong Y$, or $X \cong Y$. Say X and Y are isomorphic.

<u>Ex</u>. In Set isomorphisms are bijections.
In Prop isomorphisms are biimplications

<u>Exercise</u>. Show that if $f : X \to Y$ has an inverse f^{-1}, then this is unique.

(4)

<u>Lem</u>. If $X, Y \in \mathcal{C}$ are isomorphic, then

$$\hom(Z, X) \cong \hom(Z, Y) \text{ and } \hom(X, Z) \cong \hom(Y, Z)$$

for every $Z \in \mathcal{C}$.

(Thus, we cannot tell them apart via their relationship with other objects.)

<u>Pf</u>. Define $\hom(Z, X) \rightleftarrows \hom(Z, Y)$ by

$x \longmapsto f \circ x$

$f^{-1} \circ y \longleftarrow y$

Since $f^{-1} f x = x$ and $f f^{-1} y = y$, these give an isomorphism / bijection.
The other bijection is similar. □

<u>Lem</u>. Terminal objects are unique up to unique isomorphism.

<u>Pf</u>. Consider two terminal objects $T, T' \in \mathcal{C}$.

There is a unique $T \xrightarrow{t'} T'$ and $T' \xrightarrow{t} T$.

Composing, we get $t t' : T \to T$, but there is only one morphism $T \to T$, so this must be the identity. Similarly, $t' t = id_{T'}$.

Thus, $T \cong T'$ and this isomorphism is unique.

(Thus, there is only one terminal object up to isomorphism and we thus talk about <u>the</u> terminal object if it exists.)

Duality

(5)

I wrote the word 'similarly' above to avoid doing a part of the proof that was basically the same but 'opposite'.

This can be made formal, in which case we write 'dually' instead of similarly.

<u>Def</u>. Given a category $\mathcal{C}$, define the category $\mathcal{C}^{op}$, the <u>opposite</u> of $\mathcal{C}$, by
- $\operatorname{ob}\mathcal{C}^{op} := \operatorname{ob}\mathcal{C}$
- $\hom_{\mathcal{C}^{op}}(X, Y) := \hom_{\mathcal{C}}(Y, X)$
- $\underset{\hom_{\mathcal{C}^{op}}(X,X)}{\underset{\uparrow}{id_X}} := \underset{\hom_{\mathcal{C}}(X,X)}{\underset{\uparrow}{id_X}}$
- $g \circ_{\mathcal{C}^{op}} f := f \circ_{\mathcal{C}} g \in \hom_{\mathcal{C}^{op}}(X, Z) = \hom_{\mathcal{C}}(Z, X)$

$$\begin{array}{cc} \text{in } \mathcal{C}^{op} & X \xrightarrow{f} Y \xrightarrow{g} Z \\ \mathcal{C} & X \xleftarrow{f} Y \xleftarrow{g} Z \end{array}$$

<u>Exercise</u>. Check that this forms a category.

<u>Exercise</u>. Check that if $X \cong Y$ in $\mathcal{C}$, then $X \cong Y$ in $\mathcal{C}^{op}$.

Initial objects

<u>Def</u>. Say that $I \in \mathcal{C}$ is <u>initial</u> if it is terminal in $\mathcal{C}^{op}$.

In other words, I is initial if for all $Z \in \mathcal{C}$, there is a unique map $I \to Z$.

<u>Exercise</u>. Check these two definitions are equivalent.

<u>Lem</u>. Initial objects are unique up to unique isomorphism.

(6)

<u>Pf</u>. This is the dual statement of the lemma above.

<u>Ex</u>. In Set, the initial object is $\emptyset$.

In Prop, the initial object is $\bot$.

In Ty, the initial object is empty.