A diagram for the Yoneda Lemma
(In which each node and arrow can be interpreted precisely as a “term”, and most of the interpretations are “obvious”; plus dictionaries!!!)

Eduardo Ochs (UFF, Rio das Ostras, Brazil)
http://angg.twu.net/#intro-tys-lfc

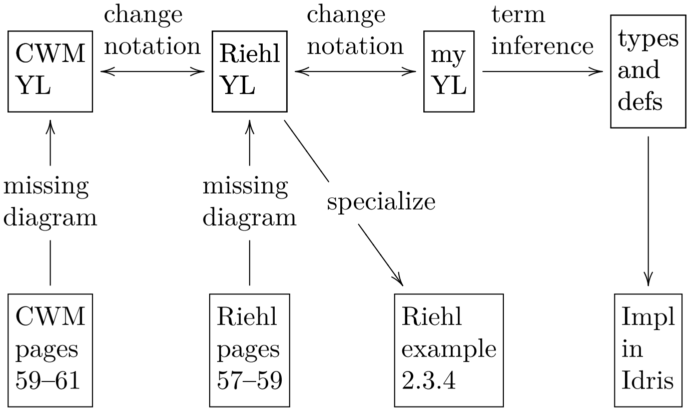
$$\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C\to_) \arrow[r, "T"'] & (A\to R_)
\end{tikzcd}$$

$T_0 := \lambda D.\lambda g.(\gamma; Rg)$
$\gamma := TC(\mathrm{id}_C)$

The Big Picture

The Big Picture
The rectangles in the next page represent:
lots of text in very small type (several pages of a book), or
a big diagram (a Yoneda Lemma in any of several notations),
or several typings and definitions in Type Theory,
or these typings and definitions converted to code in Idris.

Following it in this way ${}_{\uparrow}{}^{\to\to\to}{}_{\downarrow}$ means:
We start with a standard presentation of the YL ([CWM]),
then we draw its "missing diagrams", then we convert them
to the notation of another book (Emily Riehl's), then we
convert them to a notation that "looks more like logic",
then we complete its the details using term inference,
then we translate our formalization to Idris.


CWM YL
change notation
Riehl YL
change notation
my YL
term inference
types and defs
missing diagram
missing diagram
specialize
CWM pages 59–61
Riehl pages 57–59
Riehl example 2.3.4
Impl in Idris

The Big Picture: the four diagrams

$$\begin{tikzcd}
\boxed{\begin{array}{l}\text{CWM}\\ \text{YL}\end{array}} \arrow[r, leftrightarrow, "\begin{array}{l}\text{change}\\ \text{notation}\end{array}"] & \boxed{\begin{array}{l}\text{Riehl}\\ \text{YL}\end{array}} \arrow[r, leftrightarrow, "\begin{array}{l}\text{change}\\ \text{notation}\end{array}"] \arrow[dr, "\text{specialize}"] & \boxed{\begin{array}{l}\text{my}\\ \text{YL}\end{array}} \\
& & \boxed{\begin{array}{l}\text{Riehl}\\ \text{example}\\ \text{2.3.4}\end{array}}
\end{tikzcd}$$

$$\begin{tikzcd}
 & * \arrow[d, "u"] \\
r \arrow[r, mapsto] & Kr \\
D \arrow[r, "K"] & \mathbf{Set} \\
D(r,-) \arrow[r] \arrow[dr, "\psi"'] \arrow[from=4-1, to=2-2, bend left, leftrightarrow] & \mathbf{Set}(*,K-) \arrow[d, leftrightarrow] \\
 & K
\end{tikzcd}$$

$$\begin{tikzcd}
 & 1 \arrow[d, "\ulcorner x \urcorner"] \\
c \arrow[r, mapsto] & Fc \\
\mathsf{C} \arrow[r, "F"] & \mathsf{Set} \\
\mathsf{C}(c,-) \arrow[r] \arrow[dr, "\alpha"'] \arrow[from=4-1, to=2-2, bend left, leftrightarrow] & \mathsf{Set}(1,F-) \arrow[d, leftrightarrow] \\
 & F
\end{tikzcd}$$

$$\begin{tikzcd}
 & 1 \arrow[d, "\ulcorner x \urcorner"] \\
C \arrow[r, mapsto] & RC \\
\mathbf{C} \arrow[r, "R"] & \mathbf{Set} \\
(C \to _) \arrow[r, "T"'] \arrow[dr, "T'"'] \arrow[from=4-1, to=2-2, bend left, leftrightarrow] & (1 \to R_) \arrow[d, leftrightarrow] \\
 & R
\end{tikzcd}$$

$$\begin{tikzcd}
 & & 1 \arrow[d, "\ulcorner x \urcorner"] \arrow[ddd, bend left, "\forall \ulcorner \varphi(x) \urcorner"] \\
 \mathbb{Z}[x] \arrow[rr, mapsto] \arrow[d, "\exists ! \varphi"'] & & U(\mathbb{Z}[x]) \arrow[d, "U\varphi"] \\
 \forall R \arrow[rr, mapsto] & \mapsto & UR \\
 & & \\
 \mathsf{Ring} \arrow[rr, mapsto, "U"] & & \mathsf{Set} \\
 \mathsf{Ring}(\mathbb{Z}[x],-) \arrow[rr, leftrightarrow] \arrow[drr, leftrightarrow] & & \mathsf{Set}(1,U-) \arrow[d, leftrightarrow] \\
 & & U
\end{tikzcd}$$

MISSING
DIAGRAMS

A quote from Eilenberg and Steenrod
From *Foundations of Algebraic Topology* ([ES52, p.ix]):

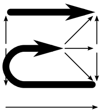
> The diagrams incorporate a large amount of information. Their use provides extensive savings in space and in mental effort. In the case of many theorems, the setting up of the correct diagram is the major part of the proof. We therefore urge that the reader stop at the end of each theorem and attempt to construct for himself the relevant diagram before examining the one which is given in the text. Once this is done, the subsequent demonstration can be followed more readily; in fact, **the reader can usually supply it himself.**

A quote from [IDARCT]

1. Mental Space and Diagrams

My memory is limited, and not very dependable: I often have to rededuce results to be sure of them, and I have to make them fit in as little "mental space" as possible...

Different people have different measures for "mental space"; someone with a good algebraic memory may feel that an expression like $\mathsf{Frob} : \Sigma_f(P \wedge f^*Q) \cong \Sigma_f P \wedge Q$ is easy to remember, while I always think diagramatically, and so what I do is that I remember this diagram,



and I reconstruct the formula from it.

Another quote from [IDARCT]

12. Skeletons of proofs

Let's call the "projected" version of a mathematical object its "skeleton". The underlying idea in this paper is that for the right kinds of projections, and for some kinds of mathetical objects, it should be possible to reconstruct enough of the original object from its skeleton and few extra clues — just like paleontologists can reconstruct from a fossil skeleton the look of an animal when it was alive.

A quote from my submission to the ACT2019
My extended abstract was called "On Some Missing Diagrams in The Elephant" ([MDE]). It was rejected.

> Imagine two category theorists, Aleks and Bob, who both think very visually and who have exactly the same background. One day Aleks discovers a theorem, T_1, and sends an e-mail, E_1, to Bob, stating and proving T_1 in a purely algebraic way; then Bob is able to reconstruct by himself Aleks's diagrams for T_1 exactly as Aleks has thought them. We say that Bob has reconstructed the *missing diagrams* in Aleks's e-mail.

(Cont...)

> Now suppose that Carol has published a paper, P_2, with a theorem T_2. Aleks and Bob both read her paper independently, and both pretend that she thinks diagrammatically in the same way as them. They both "reconstruct the missing diagrams" in P_2 in the same way, even though Carol has never used those diagrams herself.

The Yoneda Lemma involves three categories.
It's hard to find good conventions for "drawing the missing diagrams" when there are several categories involved. Here I will use my current favorite conventions.

Missing diagrams: wishlist

What are "good conventions" for
drawing missing diagrams?

Some ideas:

1. The notation in the diagram should be similar to the one in the text that we are trying to complement.
2. We should have good *positional conventions* — for example, drawing $A \to B$ above $\mathbf{C}$ could *usually* mean that the morphism $A \to B$ is *in* the category $\mathbf{C}$... so *above* usually means *inside*.
3. It should be possible to infer lots of *typings* and *definitions* from the diagrams.
4. Each node and arrow in our diagrams should have a meaning that we know how to formalize.

Missing diagrams: wishlist (2)

Also:

5. The diagram should contain all entities mentioned in the text.

...or almost all.

An exercise: a missing diagram in CWM
Read the beginning of section III.2 in [CWM].
That section is about the Yoneda Lemma.
Check that practically all the entities in Proposition 1 and in its proof appear in the diagram below.

$$
\begin{tikzcd}
 & c \arrow[d, "u"] \\
r \arrow[r, mapsto] \arrow[d, "f'"'] & Sr \arrow[d, "Sf'"] \\
d \arrow[r, mapsto] \arrow[d, "g'"'] & Sd \arrow[d, "Sg'"] \\
d' \arrow[r, mapsto] & Sd \\
D \arrow[r, "S"] & C
\end{tikzcd}
$$

$$D(r,-) \xrightarrow{\varphi} C(c,S-)$$

$$
\begin{tikzcd}
d \arrow[d, "f'"'] & D(r,d) \arrow[r, "\varphi_r"] \arrow[d, "D(r,f')"'] & C(c,Sd) \arrow[d, "C(c,Sf')"] & f' \arrow[r, mapsto] \arrow[d, mapsto] & Sf' \circ u \arrow[d, mapsto] \\
 & & & & Sg' \circ (Sf' \circ u) \\
d' & D(r,d') \arrow[r, "\varphi_d"] & C(c,Sd') & g' \circ f' \arrow[r, mapsto] & S(g \circ f) \circ u
\end{tikzcd}
$$

$$D(r,-) \xrightarrow{\varphi} C(c,S-)$$

A smaller diagram

The right side of the diagram of the previous page is "obvious" in a precise sense: it is just the internal view of the natural transformation φ. So we can omit it.

The least obvious, and most important, part of the diagram is the bijection between 'u's and 'φ's. So we will stress it.

We will sometimes omit the middle part.

$$\begin{tikzcd}
 & c \arrow[d, "u"] \\
r \arrow[r, mapsto] \arrow[d, "f'"'] & Sr \arrow[d, "Sf'"] \\
d \arrow[r, mapsto] \arrow[d, "g'"'] & Sd \arrow[d, "Sg'"] \\
d' \arrow[r, mapsto] & Sd \\
D \arrow[r, "S"] & C \\
D(r,-) \arrow[r, "\varphi"] & C(c, S-)
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
 & c \arrow[d, "u"] \\
r \arrow[r, mapsto] & Sr \\
D \arrow[r, "S"] & C \\
D(r,-) \arrow[r, "\varphi"'] & C(c, S-)
\end{tikzcd}$$

The Yoneda Lemma in CWM

Left: Proposition 1 in [CWM, p.59].

Right: Proposition 2 in [CWM, p.60].

We will see how to obtain functors $D(r,-)$, $C(c,S-)$, and $\mathbf{Set}(*,K-)$ that "deserve these names", but it will be easier to do that in another notation first — and then translate.

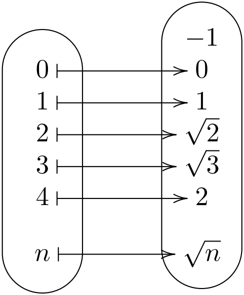
$$\begin{tikzcd}
 & c \arrow[d, "u"] \\
r \arrow[r, mapsto] & Sr \\
D \arrow[r, "S"] & C \\
D(r,-) \arrow[r, "\varphi"'] & C(c,S-)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & * \arrow[d, "u"] \\
r \arrow[r, mapsto] & Kr \\
D \arrow[r, "K"] & \mathbf{Set} \\
D(r,-) \arrow[r] \arrow[dr, "\psi"'] & \mathbf{Set}(*,K-) \arrow[d, leftrightarrow] \\
 & K
\end{tikzcd}$$

INTERNAL AND EXTERNAL VIEWS

Motivation: blob-sets

From the introduction of [PH1], a.k.a.
"Planar Heyting Algebras for Children":

$$\begin{array}{rccc} \sqrt{\ } : & \mathbb{N} & \to & \mathbb{R} \\ & n & \mapsto & \sqrt{n} \end{array}$$



$$\mathbb{N} \xrightarrow{\sqrt{\ }} \mathbb{R}$$

Motivation: blob-sets (2)

"Internal diagrams are a tool that lets us lower the level of abstraction. (...) Look at the figure at the left in the previous slide and compare its '$\mathbb{N} \to \mathbb{R}$' in the upper line (the external view), with the '$n \mapsto \sqrt{n}$' in the lower line (the internal view); the $n \mapsto \sqrt{n}$ shows a (generic) element and its image.

The figure at the right shows the external view at the bottom and an internal view at the top; note that all elements in the blobs for $\mathbb{N}$ and $\mathbb{R}$ are named, but only a few of the elements are shown (...) the arrows like $3 \mapsto \sqrt{3}$ and $4 \mapsto 2$, that show elements and their images, are substitution instances of the generic $n \mapsto \sqrt{n}$, maybe after some calculations (or "reductions" in λ-calculus terminology)."

Blob-categories and functors

When we draw the internal views of functors we don't draw the blobs.
In the diagram below the morphism $g : A \to B$ is above $\mathbf{C}$ because it is in $\mathbf{C}$, and $Fg : FA \to FB$ is above $\mathbf{D}$ because it is in $\mathbf{D}$.
The '$\mapsto$' arrows above F are actions of F on objects ('F_0's) and on morphisms ('F_1').

$$\begin{tikzcd} A \arrow[r, mapsto, "F_0"] \arrow[d, "g"'] \arrow[dr, phantom, "\overset{F_1}{\longmapsto}"] & FA \arrow[d, "Fg"] \\ B \arrow[r, mapsto, "F_0"] & FB \end{tikzcd}$$

$$\mathbf{C} \xrightarrow{F} \mathbf{D}$$

“The”

Most texts in Category Theory (“CT”)
are full of expressions like this:

“Let’s write $(A\times)$ for the functor that takes
each object B to $A \times B$”

I was absolutely fascinated by this “the”.
A functor — say, $(A\times)$ — has an action on objects,
an action on morphisms, and guarantees, or proofs,
that it respects identities and compositions.

That “the functor” implies that the reader should be able
to figure out by himself the action on morphisms, i.e.,
the precise meaning for $(A\times)f$ when $f : B \to C$, and to
check that this $(A\times)$ respects identities and compositions.

"The" (2)

Formally, a functor $(A\times) : \mathbf{Set} \to \mathbf{Set}$
is a 4-uple:

$$(A\times) = ((A\times)_0, (A\times)_1, \mathsf{respids}_{(A\times)}, \mathsf{respcomp}_{(A\times)})$$

The "the" in

"$(A\times)$ is the functor that takes each object B to $A\times B$"

suggests that learning CT transforms you in a certain way...
you become a person who can infer $(A\times)_1$, $\mathsf{respids}_{(A\times)}$,
and $\mathsf{respcomp}_{(A\times)}$ from just $(A\times)_0$...

...you become a person who can define functors in a very
compact way, and the other CT people will understand you.

(I wanted to become like that when I'd grow up)

“The” (3)

...you become a person who can define functors in a very compact way, and the other CT people will understand you.

I wanted to become like that when I’d grow up, but
I also wanted to be able to explain my tricks!!!

In short: the trick for inferring the meaning for $(A\times)f$
is slightly related to Type Inference...
it is a (sloppy kind of) Term Inference, composed by a
Proof Search followed by Curry-Howard.

I will explain that with lots of diagrams, but the general idea is: we will try to build an object of type $A \times B \to A \times C$ from a “hypothesis” $f : B \to C$.

A diagrammatic convention for functors

Let me introduce a diagrammatic convention.
The obvious way of drawing $(A\times) : \mathbf{Set} \to \mathbf{Set}$
is the diagram at the left.
When I draw it as at the right I am saying that
$(A\times)_0 B := A \times B$,
$(A\times)_0 C := A \times C$,
$(A\times)_1 f := \lambda p.(\pi p, f(\pi' p))$.

$$\begin{tikzcd}
B \arrow[r, mapsto, "(A\times)_0"] \arrow[d, "f"'] \arrow[dr, phantom, "\overset{(A\times)_1}{\longmapsto}"] & (A\times)_0 B \arrow[d, "(A\times)_1 f"] \\
C \arrow[r, mapsto, "(A\times)_0"] & (A\times)_0 B \\
\mathbf{Set} \arrow[r, "(A\times)"] & \mathbf{Set}
\end{tikzcd}
\qquad
\begin{tikzcd}
B \arrow[r, mapsto] \arrow[d, "f"'] \arrow[dr, phantom, "\longmapsto"] & A\times B \arrow[d, "\lambda p.(\pi p, f(\pi' p))"] \\
C \arrow[r, mapsto] & A\times C \\
\mathbf{Set} \arrow[r, "(A\times)"] & \mathbf{Set}
\end{tikzcd}$$

Finding a meaning for $(A\times)f$

How did I discover that $(A\times)_1 f = \lambda p.(\pi p, f(\pi' p))$
in the previous slide?
Let's go back one step. Can we find a precise meaning
for the $(A\times)f$ at the right below? Or: do we have a
natural way to construct a function $(A\times)f : A\times B \to A\times C$?
We are allowed to use $f : B \to C$ in the construction.
What are natural constructions? Ta-daaa: λ-terms!

$$\begin{tikzcd}
B \arrow[r, mapsto, "(A\times)_0"] \arrow[d, "f"'] \arrow[dr, phantom, "\overset{(A\times)_1}{\longmapsto}"] & (A\times)_0 B \arrow[d, "(A\times)_1 f"] \\
C \arrow[r, mapsto, "(A\times)_0"] & (A\times)_0 B
\end{tikzcd}
\qquad
\begin{tikzcd}
B \arrow[r, mapsto] \arrow[d, "f"'] \arrow[dr, phantom, "\longmapsto"] & A\times B \arrow[d, "(A\times)f"] \\
C \arrow[r, mapsto] & A\times C
\end{tikzcd}$$

$$\mathbf{Set} \xrightarrow{(A\times)} \mathbf{Set} \qquad\qquad \mathbf{Set} \xrightarrow{(A\times)} \mathbf{Set}$$

Curry-Howard in one slide
(Suggested reading: [GLT89], [WadlerPaT])
Compare:

$$\dfrac{\dfrac{\dfrac{[P \wedge Q]^1}{P} \qquad \dfrac{\dfrac{[P \wedge Q]^1}{Q} \qquad Q \to R}{R}}{P \wedge R}}{P \wedge Q \to P \wedge R}\ 1$$

$$\dfrac{\dfrac{\dfrac{[p : A{\times}B]^1}{\pi p : A} \qquad \dfrac{\dfrac{[p : A{\times}B]^1}{\pi' p : B} \qquad f : B \to C}{f(\pi' p) : C}}{(\pi p, f(\pi' p)) : A{\times}C}}{(\lambda p{:}A{\times}B.(\pi p, f(\pi' p))) : A{\times}B \to A{\times}C}\ 1$$

Proof search

A double bar in a proof means
"several steps here, details omitted", or
"several steps here, the details are obvious"...

$$\dfrac{Q \to R}{\overline{P \land Q \to P \land R}} \qquad \Rightarrow \qquad \dfrac{\dfrac{\dfrac{[P \land Q]^1}{P} \qquad \dfrac{\dfrac{[P \land Q]^1}{Q} \qquad Q \to R}{R}}{P \land R}}{P \land Q \to P \land R}\ 1$$

Proof search, translated to sets (term inference)

$$\dfrac{f : B \to C}{(A\times)f : A \times B \to A \times C} \qquad \Rightarrow$$

$$\dfrac{\dfrac{\dfrac{[p : A \times B]^1}{\pi p : A} \qquad \dfrac{\dfrac{[p : A \times B]^1}{\pi' p : B} \qquad f : B \to C}{f(\pi' p) : C}}{(\pi p, f(\pi' p)) : A \times C}}{(\lambda p{:}A \times B.(\pi p, f(\pi' p))) : A \times B \to A \times C}\ 1$$

Alternative names:
term search, term inference...

Two kinds of sloppiness

We can "pronounce" the trees below as:

$$\begin{array}{c} Q \to R \\ \hline\hline P \land Q \to P \land R \end{array} \qquad \begin{array}{c} f : B \to C \\ \hline\hline (A\times)f : A \times B \to A \times C \end{array}$$

If I know that $P \to Q$ is true
then I know that $P \land Q \to P \land R$ is true.

If I know an element of type $B \to C$ (call it 'f')
then I know an element of type $A \times B \to A \times C$
(call it '$(A\times)f$').

I can ignore the 'f :' and '$(A\times)f$:' and think/say just this:
if I know an element of type $B \to C$
then I know an element of type $A \times B \to A \times C$.

Two kinds of sloppiness (2)

If I know an element of type $B \to C$
then I know an element of type $A \times B \to A \times C$.

The diagrams that I will show for the Yoneda Lemma(s)
permit the kind of thinking above as a first step.
In it we don't care if there are several different,
non-equivalent ways of building a result of the
desired output type from the inputs.

We can start by playing just with the types —
and then we introduce the terms, as if they were
"obvious", or a "trivial exercise".

(I can do this "playing with types" visually very quickly
by looking at the diagrams without writing anything down).

Two kinds of sloppiness (3)

Our diagrams for the Yoneda Lemmas also permit a second kind of sloppiness: the “Syntactical World” of [IDARCT, section 19], in which we do all constructions first, and all the parts that involve checking equations are left to a second moment.

But this is just a curiosity!
The details are not relevant now.
I will explain this very briefly in the last slides.

A diagrammatic convention on NTs

If $F, G : \mathbf{C} \to \mathbf{D}$ are functors and
$T : F \to G$ is a natural transformation between them,
we will usually draw the internal view of $T : F \to G$
as a square above its external view, and at the right
of the "input morphism" $f : A \to B$ in $\mathbf{C}$:

$$
\begin{tikzcd}
A \arrow[dr, mapsto] \arrow[drr, mapsto] \arrow[dd, "f"'] & & \\
& FA \arrow[r, "TA"'] \arrow[dd, "Ff"] & GA \arrow[dd, "Gf"] \\
B \arrow[dr, mapsto] \arrow[drr, mapsto] & & \\
& FB \arrow[r, "TB"'] & GB \\
& F \arrow[r, "T"'] & G
\end{tikzcd}
\quad \Longrightarrow \quad
\begin{tikzcd}
A \arrow[d, "f"'] & FA \arrow[r, "TA"] \arrow[d, "Ff"'] & GA \arrow[d, "Gf"] \\
B & FB \arrow[r, "TB"] & GB \\
& F \arrow[r, "T"] & G
\end{tikzcd}
$$

Another diagrammatic convention on NTs

If $F, G : \mathbf{C} \to \mathbf{Set}$ are functors going to **Set** and
$T : F \to G$ is a natural transformation between them,
we draw everything as before, but we add
at the right an internal view of the internal view.

$$
\begin{tikzcd}
A \arrow[d, "f"'] & FA \arrow[r, "TA"] \arrow[d, "Ff"'] & GA \arrow[d, "Gf"] & x \arrow[r, mapsto] \arrow[d, mapsto] & (TA)(x) \arrow[d, mapsto] \\
B & FB \arrow[r, "TB"] & GB & (Ff)(x) \arrow[r, mapsto] & (Gf \circ TA)(x) \\
 & & & & (TB \circ Ff)(x)
\end{tikzcd}
$$

$$
\begin{tikzcd}
F \arrow[r, "T"] & G
\end{tikzcd}
$$

A diagrammatic convention on adjunctions

I draw adjunctions like this — see [IDARCT].

$$
\begin{tikzcd}
LA' \arrow[d] & A' \arrow[l, mapsto] \arrow[d] \\
LA \arrow[d] \arrow[r, phantom, "\mapsfrom"] & A \arrow[l, mapsto] \arrow[d] \\
B \arrow[d] \arrow[r, mapsto] \arrow[ur, phantom, "\substack{\mapsfrom \\ \mapsto}"] & RB \arrow[d] \\
B' \arrow[r, mapsto] \arrow[ur, phantom, "\mapsto"] & RB'
\end{tikzcd}
\qquad
\begin{tikzcd}
A \times C \arrow[d] & A \arrow[l, mapsto] \arrow[d] \\
B \times C \arrow[d] \arrow[r, phantom, "\mapsfrom"] & B \arrow[l, mapsto] \arrow[d] \\
D \arrow[d] \arrow[r, mapsto] \arrow[ur, phantom, "\substack{\mapsfrom \\ \mapsto}"] & (C \to D) \arrow[d] \\
E \arrow[r, mapsto] \arrow[ur, phantom, "\mapsto"] & (C \to E)
\end{tikzcd}
$$

$$
\begin{tikzcd}
\mathbf{B} \arrow[r, shift right, "R"'] & \mathbf{A} \arrow[l, shift right, "L"']
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{Set} \arrow[r, shift right, "(C\to)"'] & \mathbf{Set} \arrow[l, shift right, "(\times C)"']
\end{tikzcd}
$$

Left: general case. Right: my favorite particular case.

A diagrammatic convention on universals
I draw universals like this.
Left: general case. Right: my favorite particular case.

$$
\begin{tikzcd}
 & A \arrow[d, "\eta"] & {} \arrow[dd, "\forall f"] \\
B \arrow[r, mapsto] \arrow[d, "\exists! g"'] \arrow[dr, phantom, "\longmapsto"] & RB \arrow[d, "Rg"] & \\
B' \arrow[r, mapsto] & RB' & {}
\end{tikzcd}
\qquad
\begin{tikzcd}
 & B \arrow[d, "\eta"] & {} \arrow[dd, "\forall f"] \\
B \times C \arrow[r, mapsto] \arrow[d, "\exists! g"'] \arrow[dr, phantom, "\longmapsto"] & (C \to B \times C) \arrow[d, "Rg"] & \\
D \arrow[r, mapsto] & (C \to D) & {}
\end{tikzcd}
$$

$$
\mathbf{B} \xrightarrow{R} \mathbf{A} \qquad\qquad \mathbf{Set} \xrightarrow{(C \to \_)} \mathbf{Set}
$$

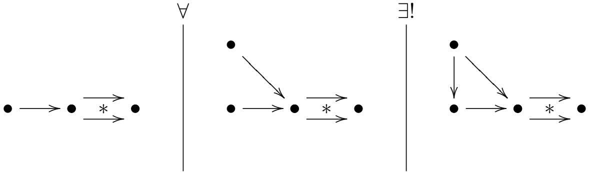
My convention on how to draw the Yoneda Lemma
is based on that shape.

A notation that looks like Logic

Origins: Freyd76

The diagrams here are based on the ones from [Freyd76], but with several changes.

[Freyd76] uses bars with quantifiers, that mean, for example, "for all extensions of the previous diagram to this one"...



Origins: Freyd76 (2)
We can draw them more compactly
if we use numbered quantifiers.
A category *has equalizers* iff:

$$\begin{tikzcd}
\bullet \arrow[d, "\exists!_4"'] \arrow[dr, "\forall_3"] & & \\
\bullet \arrow[r, "\exists_2"'] & \bullet \arrow[r, shift left=2, "\forall_1"] \arrow[r, shift right=2, "\forall_1"'] \arrow[r, phantom, "*"] & \bullet
\end{tikzcd}$$

Origins: Freyd76 (3)
At some point I started to draw
the “$\forall\exists!$”s as functions and
this gave me arrows between arrows...

$$
\begin{tikzcd}
\bullet \arrow[d, "k"'] \arrow[dr, "h"] \arrow[d, phantom, shift left=4ex, "\overset{\longleftarrow}{\varphi}"] & & \\
\bullet \arrow[r, "e"'] & \bullet \arrow[r, shift left, "f"] \arrow[r, shift right, "g"'] \arrow[r, phantom, "*"] & \bullet
\end{tikzcd}
$$

The “equalizerness” of the lower $\bullet \longrightarrow \bullet \rightrightarrows \bullet$
is the small horizontal arrow inside the triangle.

Arrows between arrows

Some of my diagrams with arrows between arrows look like Logic.

For example:

$$\begin{tikzcd}
X \arrow[r, mapsto] \arrow[d, "f"'] \arrow[dr, phantom, "\longmapsto"] & \mathrm{Hom}_{\mathbf{C}}(A,X) \arrow[d, "\mathrm{Hom}_{\mathbf{C}}(A,f)"] \\
Y \arrow[r, mapsto] & \mathrm{Hom}_{\mathbf{C}}(A,Y)
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[d] \arrow[dr, phantom, "\longrightarrow"] & (A \to X) \arrow[d] \\
Y & (A \to Y)
\end{tikzcd}$$

$$(X \to Y) \to ((A \to X) \to (A \to Y))$$

Arrows between arrows (2)
Some of my diagrams with arrows between arrows look like Logic.

This motivated me to try to define a certain (incomplete!) system of Natural Deduction for Categories...

More on this in the last slides.

Diagrams induce typings

...and this lets us omit some typings! An example:

$$\begin{tikzcd}
& A \arrow[d, "\eta"] \arrow[dd, bend left=60, "\forall f"] \\
B \arrow[r, mapsto] \arrow[d, "\exists! g"'] \arrow[dr, phantom, "\longmapsto"] & RB \arrow[d, "Rg"] \\
\forall B' \arrow[r, mapsto] & RB'
\end{tikzcd}$$

$$\mathbf{B} \xrightarrow{R} \mathbf{A}$$

$\forall B' : \mathrm{Objs}(\mathbf{B}).$
$\forall f : A \to RB'.$
$\exists! g : B \to B'.$
$f = Rg \circ \eta$

$$\Downarrow$$

$\forall B'.$
$\forall f.$
$\exists! g.$
$f = Rg \circ \eta$

Diagrams with quantifiers induce propositions

In [Freyd76] every extension of a diagram to the next one commutes by default, and non-commuting cells have to be indicated... if we follow this convention,
"$\exists! g.\ f = Rg \circ \eta$" can be rewritten as:
"$\exists! g$ (such that everything commutes)".
Another convention: "$\forall$"s come before "$\exists!$"s. So:

$$\begin{tikzcd}
 & A \arrow[d, "\eta"] \arrow[dd, bend left, "\forall f"] \\
B \arrow[r, mapsto] \arrow[d, "\exists! g"'] \arrow[dr, phantom, "\longmapsto"] & RB \arrow[d, "Rg"] \\
\forall B' \arrow[r, mapsto] & RB'
\end{tikzcd}$$

$$\mathbf{B} \xrightarrow{R} \mathbf{A}$$

$\forall B'.\ \forall f.\ \exists! g.$
$f = Rg \circ \eta$

$\Downarrow$

$\forall B'.\ \forall f.\ \exists! g$

$\Downarrow$

$\forall$___.$\exists$__

$\Downarrow$

(nothing!)

Diagrams with quantifiers induce propositions (2)

With these conventions the default proposition associated to the diagram at the left is the one at the right.
We don't need to write down the proposition — it can be extracted from the diagram.
Note that the "$\forall B'$" must come before the "$\forall f$".

$$
\begin{tikzcd}
& A \arrow[d, "\eta"] \arrow[dd, bend left=60, "\forall f"] \\
B \arrow[r, mapsto] \arrow[d, "\exists! g"'] \arrow[dr, phantom, "\longmapsto"] & RB \arrow[d, "Rg"] \\
\forall B' \arrow[r, mapsto] & RB'
\end{tikzcd}
\qquad \Longrightarrow \qquad
\begin{array}{l}
\forall B' : \mathrm{Objs}(\mathbf{B}). \\
\forall f : A \to RB'. \\
\exists! g : B \to B'. \\
f = Rg \circ \eta
\end{array}
$$

$$\mathbf{B} \xrightarrow{R} \mathbf{A}$$

TODO: Ask Tom Leinster and Paolo Perrone
They both have books on basic CT that are available online and that use some of these diagrammatic conventions...
See: [Leinster, Introduction]
and [Perrone, Chapter 4].

Are these conventions common?
Where are they formalized?
(I don't know!!! I'm an outsider!!!)
=(

Motivation

People who are very visual often remember statements, constructions and proofs via shapes and movement ([IDARCT]). For them the diagrams are the "skeletons" of formal proofs ([IDARCT, sec.12]).

The "shape" of the Yoneda Lemma that I will present in these slides is my favorite way for remembering both the statement and the proof of the YL — and its variants!!!

In order to work with diagrams *formally* we need to be able to translate our diagrams to languages that are *precise* and *well-known*. One way to do that is to treat our diagrams as skeletons for their formalizations in Type Theory.

For an introduction to using TT as foundations, see [HOTT, introduction and chapter 1].

I'm currently trying to implement/program this in Idris.

BACK TO THE YONEDA LEMMA

Three Yoneda Lemmas

These are our diagrams for three "Yoneda Lemmas".

$$
\begin{tikzcd}
& A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC
\end{tikzcd}
\qquad
\begin{tikzcd}
& 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC
\end{tikzcd}
\qquad
\begin{tikzcd}
& 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & (B\to C)
\end{tikzcd}
$$

$$
\begin{tikzcd}
(C\to\_) \arrow[r, "T"] & (A\to R\_)
\end{tikzcd}
\qquad
\begin{tikzcd}
(C\to\_) \arrow[r, "T"] \arrow[dr, "T'"'] & (1\to R\_) \arrow[d, leftrightarrow] \\
& R
\end{tikzcd}
\qquad
\begin{tikzcd}
(C\to\_) \arrow[r, "T"] \arrow[dr, "T'"'] & (1\to(B\to\_)) \arrow[d, leftrightarrow] \\
& (B\to\_)
\end{tikzcd}
$$

In all cases we have bijections between 'γ's and 'T's.
The 'γ's are morphisms, the 'T's are natural transformations.
Right: the most concrete and familiar YL, a bijection
$\mathrm{Hom}(B,C) \leftrightarrow \mathrm{Nat}((C\to\_),(B\to\_))$.
Left: the most abstract YL.

Three Yoneda Lemmas (and their names)

$$
\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC
\end{tikzcd}
\qquad
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC
\end{tikzcd}
\qquad
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & (B \to C)
\end{tikzcd}
$$

$$
\begin{tikzcd}
(C \to \_) \arrow[r, "T"] & (A \to R\_)
\end{tikzcd}
\qquad
\begin{tikzcd}
(C \to \_) \arrow[r, "T"] \arrow[dr, "T'"'] & (1 \to R\_) \arrow[d, leftrightarrow] \\
 & R
\end{tikzcd}
\qquad
\begin{tikzcd}
(C \to \_) \arrow[r, "T"] \arrow[dr, "T'"'] & (1 \to (B \to \_)) \arrow[d, leftrightarrow] \\
 & (B \to \_)
\end{tikzcd}
$$

Left: an (obscure?) lemma from adjunctions.
Middle: the Yoneda Lemma.
Right: the Yoneda Embedding.

Left: $\mathrm{Hom}(A, RC) \cong \mathrm{Nat}(\mathrm{Hom}(C, -), \mathrm{Hom}(A, R-))$
Middle: $RC \cong \mathrm{Nat}(\mathrm{Hom}(C, -), R)$
Right: $\mathrm{Hom}(B, C) \cong \mathrm{Nat}(\mathrm{Hom}(C, -), \mathrm{Hom}(B, -))$

Some notational conventions

We will follow the conventions for internal and external diagrams from [PH1, introduction], with some slight changes.

In the diagram below we have the external view of the functor $F : \mathbf{C} \to \mathbf{A}$ below and its internal view above.

The object C over $\mathbf{C}$ is in the category $\mathbf{C}$.

The morphism $\gamma : A \to FC$ above $\mathbf{A}$ is in the category $\mathbf{A}$.

The arrow $C \mapsto FC$ shows the functor F acting on an object. Note the difference between '$\to$' and '$\mapsto$'!

$$\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & FC \\
\mathbf{C} \arrow[r, "F"] & \mathbf{A}
\end{tikzcd}$$

Some notational conventions (2)

We will sometimes omit the external view.

1 is an object of **Set** — the singleton set.
In this diagram there's a morphism $\ulcorner x \urcorner : 1 \to FC$,
so 1 and FC are in the same category, so FC is in **Set**.

$\ulcorner x \urcorner$ is pronounced "the name of x".
The image of a map $\ulcorner \alpha \urcorner : 1 \to A$ is the element $\alpha \in A$.

$$\begin{tikzcd}
 & 1 \arrow[d, "\ulcorner x \urcorner"] \\
C \arrow[r, mapsto] & FC \\
\mathbf{C} \arrow[r, "F"] & \mathbf{Set}
\end{tikzcd}$$

Some notational conventions (3)

The lower triangle is made of three objects
in another category. What category is that?

$R : \mathbf{C} \to \mathbf{Set}$, so R is an object of $\mathbf{Set}^{\mathbf{C}}$,
and $(C{\to}\_)$ and $(1{\to}R\_)$ are objects of $\mathbf{Set}^{\mathbf{C}}$ too.

R, $(C{\to}\_)$, and $(1{\to}R\_)$ are functors, so T and T' are NTs.

$$
\begin{tikzcd}
& 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
\mathbf{C} \arrow[r, "R"] & \mathbf{Set}
\end{tikzcd}
$$

$$
\begin{tikzcd}
(C{\to}\_) \arrow[r, "T"] \arrow[dr, "T'"'] & (1{\to}R\_) \arrow[d, leftrightarrow] \\
& R
\end{tikzcd}
$$

Some notational conventions (4)
We can use parallel diagrams to compare notations.
At the left: our notation for the Yoneda Lemma.
At the right: the same diagram in the notation of [Riehl].
She states the Yoneda Lemma as $\mathrm{Hom}(\mathsf{C}(c,-),F) \cong Fc$.
It's her theorem 2.2.4, in page 57.

$$
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
\mathbf{C} \arrow[r, "R"] & \mathbf{Set}
\end{tikzcd}
\qquad
\begin{tikzcd}
 & 1 \arrow[d, "\ulcorner x \urcorner"] \\
c \arrow[r, mapsto] & Fc \\
\mathsf{C} \arrow[r, "F"] & \mathsf{Set}
\end{tikzcd}
$$

$$
\begin{tikzcd}
(C{\to}\_) \arrow[r, "T"] \arrow[dr, "T'"'] & (1{\to}R\_) \arrow[d, leftrightarrow] \\
 & R
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathsf{C}(c,-) \arrow[r] \arrow[dr, "\alpha"'] & \mathsf{Set}(1, F(-)) \arrow[d, leftrightarrow] \\
 & F
\end{tikzcd}
$$

The details
(as types and terms)

The first (most abstract) Yoneda Lemma
Choose (locally small) categories $\mathbf{A}$ and $\mathbf{C}$,
objects $A \in \mathbf{A}$ and $C \in \mathbf{C}$, a functor $R : \mathbf{C} \to \mathbf{A}$,
and a morphism $\gamma : A \to RC$.

$$
\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC
\end{tikzcd}
$$

$$(C \to \_) \xrightarrow{T} (A \to R\_)$$

We need to understand the functors $(C \to \_) : \mathbf{C} \to \mathbf{Set}$
and $(A \to R\_) : \mathbf{C} \to \mathbf{Set}$ and see how the morphism
$\gamma : A \to RC$ induces a natural transformation T...

The two functors: internal views

To understand the functors $(C{\to}\_)$ and $(A{\to}R\_)$ we
1) draw an auxiliar diagram (left),
2) draw their internal views (middle, right).

$$\begin{tikzcd}
 & A \arrow[d, "\gamma"] \arrow[dd, bend left, "\delta"] \\
C \arrow[r, mapsto] \arrow[d, "g"'] & RC \arrow[d, "Rg"] \\
D \arrow[r, mapsto] \arrow[d, "h"'] & RD \arrow[d, "Rh"] \\
E \arrow[r, mapsto] & RE
\end{tikzcd}
\qquad
\begin{tikzcd}
D \arrow[r, mapsto] \arrow[d, "h"'] & (C{\to}D) \arrow[d, "\lambda g.(g;h)"] & g \arrow[d, mapsto] \\
E \arrow[r, mapsto] & (C{\to}E) & g;h
\end{tikzcd}
\qquad
\begin{tikzcd}
D \arrow[r, mapsto] \arrow[d, "h"'] & (A{\to}RD) \arrow[d, "\lambda \delta.(\delta;Rh)"] & \delta \arrow[d, mapsto] \\
E \arrow[r, mapsto] & (A{\to}RE) & \delta;Rh
\end{tikzcd}$$

$$\mathbf{C} \xrightarrow{(C\to\_)} \mathbf{Set} \qquad \mathbf{C} \xrightarrow{(A\to R\_)} \mathbf{Set}$$

$$\mathbf{C} \xrightarrow{R} \mathbf{A}$$

$(C{\to}\_)_0(D) = \mathrm{Hom}_{\mathbf{C}}(C, D)$, $(C{\to}\_)_1(h) = \lambda g.(g; h)$,
$(A{\to}R\_)_0(D) = \mathrm{Hom}_{\mathbf{A}}(A, RD)$, $(A{\to}R\_)_1(h) = \lambda\delta.(\delta; Rh)$.

The natural transformation: internal view

To understand the NT $T : (C{\to}\_) \to (A{\to}R\_)$
we start by seeing how it produces, for objects $D, E \in \mathbf{C}$,
morphisms TD and TE...

$$\begin{tikzcd}
 & A \arrow[d, "\gamma"] \arrow[dd, shift left=6, "\delta"] \\
C \arrow[r, mapsto] \arrow[d, "g"'] & RC \arrow[d, "Rg"] \\
D \arrow[r, mapsto] \arrow[d, "h"'] & RD \arrow[d, "Rh"] \\
E \arrow[r, mapsto] & RE
\end{tikzcd}
\qquad
\begin{array}{lcc}
D & (C{\to}D) \xrightarrow{TD} (A{\to}RD) & g \longmapsto \gamma; Rg \\
E & (C{\to}E) \xrightarrow{TE} (A{\to}RE) & g; h \longmapsto \gamma; R(g; h) \\
 & (C{\to}_) \xrightarrow{T} (A{\to}R_) &
\end{array}$$

$$\mathbf{C} \xrightarrow{R} \mathbf{A}$$

So $TD = \lambda g.(R; g)$,
$T_0 = \lambda D.\lambda g.(R; g)$.

The natural transformation: internal view (2)
Now we want to check that this T obeys sqcond_T, i.e. that for every morphism $h : D \to E$ the "obvious" induced square commutes.

$$
\begin{tikzcd}
& A \arrow[d, "\gamma"] \arrow[dd, bend left, "\delta"] \\
C \arrow[r, mapsto] \arrow[d, "g"'] & RC \arrow[d, "Rg"] \\
D \arrow[r, mapsto] \arrow[d, "h"'] & RD \arrow[d, "Rh"] \\
E \arrow[r, mapsto] & RE \\
\mathbf{C} \arrow[r, "R"] & \mathbf{A}
\end{tikzcd}
\qquad
\begin{tikzcd}
D \arrow[d, "h"'] \\
E
\end{tikzcd}
\qquad
\begin{tikzcd}
(C\to D) \arrow[r, "TD"] \arrow[d, "(C\to\_)h"'] & (A\to RD) \arrow[d, "(A\to R\_)h"'] \\
(C\to E) \arrow[r, "TE"] & (A\to RE) \\
(C\to\_) \arrow[r, "T"] & (A\to R\_)
\end{tikzcd}
\qquad
\begin{tikzcd}
g \arrow[r, mapsto] \arrow[d, mapsto] & \gamma;Rg \arrow[d, mapsto] & \delta \arrow[d, mapsto] \\
g;h \arrow[r, mapsto] & (\gamma;Rg);Rh \quad \gamma;R(g;h) & \delta;Rh
\end{tikzcd}
$$

It commutes because we have
$\forall g.((\gamma;Rg);Rh = \gamma;R(g;h))$.

The diagram of the first Yoneda Lemma

We now understand all nodes and arrows in this diagram...

Remember that γ induced T.

$$\begin{tikzcd} & A \arrow[d, "\gamma"] \\ C \arrow[r, mapsto] & RC \end{tikzcd}$$

$$(C{\to}\_) \xrightarrow{T} (A{\to}R\_)$$

$A \in \mathbf{A}$
$C \in \mathbf{C}$
$R : \mathbf{A} \to \mathbf{C}$
$\gamma : A \to RC$

$(C{\to}\_) : \mathbf{C} \to \mathbf{Set}$
$(C{\to}\_)_0(D) = \mathrm{Hom}_{\mathbf{C}}(C, D)$
$(C{\to}\_)_1(h) = \lambda g.(g; h)$

$(A{\to}R\_) : \mathbf{C} \to \mathbf{Set}$
$(A{\to}R\_)_0(D) = \mathrm{Hom}_{\mathbf{A}}(A, RD)$
$(A{\to}R\_)_1(h) = \lambda \delta.(\delta; Rh)$

$T : (C{\to}\_) \to (A{\to}R\_)$
$T_0(D) := \lambda g.(\gamma; Rg)$

The diagram of the first Yoneda Lemma (2)

We started with a morphism γ and defined T from it.
We can also do the inverse!

$$\begin{tikzcd} & A \arrow[d, "\gamma"] \\ C \arrow[r, mapsto] & RC \end{tikzcd}$$

$$(C{\to}\_) \xrightarrow{T} (A{\to}R\_)$$

$A \in \mathbf{A}$
$C \in \mathbf{C}$
$R : \mathbf{A} \to \mathbf{C}$
$\gamma : A \to RC$
$\gamma := TC(\mathrm{id}_C)$

$(C{\to}\_) : \mathbf{C} \to \mathbf{Set}$
$(C{\to}\_)_0(D) = \mathrm{Hom}_{\mathbf{C}}(C, D)$
$(C{\to}\_)_1(h) = \lambda g.(g; h)$

$(A{\to}R\_) : \mathbf{C} \to \mathbf{Set}$
$(A{\to}R\_)_0(D) = \mathrm{Hom}_{\mathbf{A}}(A, RD)$
$(A{\to}R\_)_1(h) = \lambda \delta.(\delta; Rh)$

$T : (C{\to}\_) \to (A{\to}R\_)$

The bijection
Fact: the operations
$T := \lambda D.\lambda g.(\gamma; Rg)$ and
$\gamma := TC(\mathrm{id}_C)$
are inverses to one another.
Let's rewrite them as "T_γ" and "γ_T"...

$$
\begin{aligned}
&T_\gamma = \lambda D.\lambda g.(\gamma; Rg)\\
&\gamma_T = TC(\mathrm{id}_C)\\
&T_{(\gamma_T)} = \lambda D.\lambda g.(\gamma_T; Rg) = \lambda D.\lambda g.((TC(\mathrm{id}_C)); Rg)\\
&\gamma_{(T_\gamma)} = T_\gamma C(\mathrm{id}_C) = (\lambda D.\lambda g.(\gamma; Rg))C(\mathrm{id}_C)
\end{aligned}
$$

We want to check that $\gamma_{(T_\gamma)} = \gamma$ (easy)
and that $T_{(\gamma_T)} = T$ (harder).

The bijection (2)

It's easy to check that $\gamma_{(T_\gamma)} = \gamma$:

$$\begin{array}{rcl}
\gamma_{(T_\gamma)} & = & T_\gamma C(\mathrm{id}_C) \\
& = & (\lambda D.\lambda g.(\gamma; Rg))C(\mathrm{id}_C) \\
& = & (\lambda g.(\gamma; Rg))(\mathrm{id}_C) \\
& = & \gamma; R(\mathrm{id}_C) \\
& = & \gamma; \mathrm{id}_{RC} \\
& = & \gamma
\end{array}$$

The bijection (3)
Remember that
$T_{(\gamma_T)} = \lambda D.\lambda g.((TC(\mathrm{id}_C)); Rg)$, and so
$T_{(\gamma_T)}D(g) = TC(\mathrm{id}_C); Rg$.

We want to check this:
$\forall D.\forall g.(T_{(\gamma_T)}D(g) = TD(g))$, i.e.,
$\forall D.\forall g.(TC(\mathrm{id}_C); Rg = TD(g))$...

This is a consequence of sqcond_T!

The bijection (4)
We want to check this:
$\forall D. \forall g. (TC(\mathrm{id}_C); Rg = TD(g))$...

This is a consequence of sqcond_T!
Look:

$$
\begin{tikzcd}
C \arrow[d, "g"'] & (C \to C) \arrow[r, "TC"] \arrow[d, "(C\to\_)g"'] & (A \to RC) \arrow[d, "(A\to R\_)g"'] & \mathrm{id}_C \arrow[r, mapsto] \arrow[dd, mapsto] & TC(\mathrm{id}_C) \arrow[d, mapsto] \\
 & & & & TC(\mathrm{id}_C); Rg \\
D & (C \to D) \arrow[r, "TD"] & (A \to RD) & g \arrow[r, mapsto] & TD(g)
\end{tikzcd}
$$

$$(C\to\_) \xrightarrow{T} (A \to R\_)$$

Drawing the bijection
A honest way to draw the bijection between 'γ's and 'T's would be diagram with the curved arrow in the middle... But we will commit an abuse of (diagrammatical) language and use a vertical arrow, as in the diagram at the right.

$$
\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"] & (A{\to}R\_)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"'] & (A{\to}R\_)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] \arrow[d, phantom, ""] & RC \\
(C{\to}\_) \arrow[r, "T"'] \arrow[u, phantom, "\updownarrow" {xshift=2.5em}] & (A{\to}R\_)
\end{tikzcd}
$$

Now we have a shape for the (first) Yoneda Lemma and we can use it to compare several notations... But it's better to do that with the tree YLs at once, so let's prove the other two.

Changing the category $\mathbf{A}$ to $\mathbf{Set}$

Remember that $A, RC \in \mathbf{A}$ and $C \in \mathbf{C}$...
This is not shown in the diagram, but it appears
in the terms and types in lots of places.
Let's take a particular case: $\mathbf{A}$ becomes $\mathbf{Set}$...
In the notation of (simultaneous) substitution: $[\mathbf{A} := \mathbf{Set}]$.
The diagram does not change, but we can now
take a particular case of A too: $[A := 1]$. We get:

$$
\left(
\begin{tikzcd}
 & A \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"'] & (A{\to}R\_)
\end{tikzcd}
\right)
\begin{bmatrix} \mathbf{A} := \mathbf{Set} \\ A := 1 \end{bmatrix}
\quad = \quad
\left(
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"'] & (1{\to}R\_)
\end{tikzcd}
\right)
$$

Getting rid of the '1's

Convention: 1 is the singleton set,
with single element $*$: $* \in 1 \in \mathbf{Set}$, $1 = \{*\}$.
If $B \in \mathbf{Set}$ then an arrow $\beta : 1 \to B$
"selects" an element $b \in B$...
We have a bijection between elements of $b \in B$
and arrows $\beta : 1 \to B$,
that we write as $B \leftrightarrow (1 {\to} B)$,
or as two operations as $b := \beta(*)$, $\beta := \lambda *.b$ ...

My favourite way to represent a bijection $A \underset{g}{\overset{f}{\rightleftarrows}} B$
in a type system is as a 6-uple $(A, B, f, g, \mathsf{wdl}, \mathsf{wdr})$,
where $f : A \to B$, $g : B \to A$, and wdl and wdr assure
that $\forall a \in A.(g \circ f)(a) = a$ and $\forall b \in A.(f \circ g)(b) = b$
respectively.

Bijections and isos in type systems

One of my reasons for writing these notes was to show how these diagrams can be interpreted in a formal way in type systems and in proof assistants, so let me be type-ish for a moment...

Thorsten Altenkirch — in his book chapter "Naïve Type Theory" (from 2018(?), available from this home page) — uses the notation $[\![P]\!]$ for the "set of evidence" for the proposition P.

I prefer to call $[\![P]\!]$ the "set of proofs" of P (which *suggests* that we are in the BHK interpretation), or the "set of witnesses" of P (which *suggests* a model with proof-irrelevance and every $[\![P]\!]$ being either empty or a singleton)...

So...

Bijections and isos in type systems (2)
So:

$$
\begin{array}{rcl}
(A \underset{g}{\overset{f}{\rightleftarrows}} B) & = & (A, B, f, g, \mathsf{wdl}, \mathsf{wdr}) \\
 & & \text{where:} \\
 & & A \text{ is a set,} \\
 & & B \text{ is a set,} \\
 & & f : A \to B, \\
 & & g : B \to A, \\
 & & \mathsf{wdl} : [\![\forall a \in A. (g \circ f)(a) = a]\!], \\
 & & \mathsf{wdr} : [\![\forall b \in B. (f \circ g)(b) = b]\!].
\end{array}
$$

This is easy to adapt to define isos in a category.

$(A \overset{f}{\longleftrightarrow} B)$ is interpreted as $(A \underset{f^{-1}}{\overset{f}{\rightleftarrows}} B)$.

Getting rid of the '1's (2)
The (nameless) bijection $(1{\to}B) \leftrightarrow B$ can be interpreted as:

$$
\begin{array}{ccc}
\begin{tikzcd}
(1{\to}B) \arrow[d, leftrightarrow] & \beta \arrow[d, mapsto, shift right] \\
B & b \arrow[u, mapsto, shift right]
\end{tikzcd}
& \qquad &
\begin{array}{l}
b \in B \\
\beta \in (1{\to}B) \\
b := \beta(*) \\
\beta := \lambda *.b
\end{array}
\end{array}
$$

and written as:

$$
B \underset{\lambda\beta.\beta(*)}{\overset{\lambda b.\lambda *.b}{\rightleftarrows}} (1{\to}B) \qquad \text{or} \qquad (1{\to}B) \underset{\lambda b.\lambda *.b}{\overset{\lambda\beta.\beta(*)}{\rightleftarrows}} B
$$

The components wdl and wdr of the 6-uples are treated as "obvious", and are omitted.

Getting rid of the '1's (3)

If $R : \mathbf{C} \to \mathbf{Set}$ then we have a (nameless) natural transformation $(1{\to}R\_) \leftrightarrow R$ between these functors:

$$
\begin{tikzcd}
D \arrow[r, mapsto] \arrow[d, "h"'] & RD \arrow[d, "Rh"] \\
E1 \arrow[r, mapsto] & RE
\end{tikzcd}
\quad \longmapsto \quad
\begin{tikzcd}
D \arrow[r, mapsto] \arrow[d, "h"'] & (1{\to}RD) \arrow[d, "{\begin{array}{l}(1{\to}R\_)(h) \\ := \lambda\delta.(\delta;Rh)\end{array}}"] \\
E \arrow[r, mapsto] & (1{\to}RE)
\end{tikzcd}
$$

$$
\mathbf{C} \xrightarrow{R} \mathbf{Set} \qquad\qquad \mathbf{C} \xrightarrow{(1{\to}R\_)} \mathbf{Set}
$$

Note that in type theory $R = (R_0, R_1, \ldots)$, $(1{\to}R\_) = ((1{\to}R\_)_0, (1{\to}R\_)_1, \ldots)$, and the diagrams above give us enough information to let us build $(1{\to}R\_)$ as a term.

Getting rid of the '1's (4)

If $R : \mathbf{C} \to \mathbf{Set}$ then we have a (nameless)
natural isomorphism $(1{\to}R\_) \leftrightarrow R$
between the functors defined in the previous page...
If $F, G : \mathbf{A} \to \mathbf{B}$ then a natural transformation $T : F \to G$
is formalized in TT as a pair (T_0, sqcond_T),
where T_0 is its "action on objects"
and sqcond_T is its "square condition".
The nameless natual iso $(1{\to}R\_) \leftrightarrow R$ can be interpreted as
a nameless NT $(1{\to}R\_) \to R$,
a nameless NT $R \to (1{\to}R\_)$,
and guarantees that their composites are identity functors...

Getting rid of the '1's (5)

The nameless natual iso $(1{\to}R\_) \leftrightarrow R$ can be interpreted as
a nameless NT $(1{\to}R\_) \to R$,
a nameless NT $R \to (1{\to}R\_)$,
and guarantees that their composites are identity functors...

Their actions on objects can be defined from this:
$((1{\to}R\_) \to R)_0(D) : (1{\to}RD) \to RD)$
$((1{\to}R\_) \to R)_0(D) = \lambda\delta.\delta(*)$
$(R \to (1{\to}R\_))_0(D) : D \to (1{\to}RD)$
$(R \to (1{\to}R\_))_0(D) = \lambda d.\lambda *.d$

(I will omit the details)

Changing the category A to Set (2)

With the nameless natural iso $(1{\to}R\_) \leftrightarrow R$
we can add an extra level to the basement our diagram,
and this yields an “obvious” bijection between ‘γ’s and ‘T'’s.
This new diagram “is” our Second Yoneda Lemma.

$$
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"'] & (1{\to}R\_)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"'] \arrow[dr, "T'"'] & (1{\to}R\_) \arrow[d, leftrightarrow] \\
 & R
\end{tikzcd}
$$

Changing R to $(B{\to})$
Choose an object $B \in \mathbf{C}$.
It induces a functor $(B{\to}) : \mathbf{C} \to \mathbf{Set}$.
Several slides ago we did this substitution
on the diagram of the first Yoneda Lemma:

$$\begin{bmatrix} \mathbf{A} & := & \mathbf{Set} \\ A & := & 1 \end{bmatrix}$$

Now we will do this substitution
on the diagram of the second Yoneda Lemma:

$$\begin{bmatrix} R & := & (B{\to}) \end{bmatrix}$$

very little will change in the diagram,
but a lot will change in the terms and types.

Changing R to $(B{\to})$ (2)
After the substitution $\big[R \;:=\; (B{\to})\big]$
the diagram for the Second Yoneda Lemma (left) becomes
the diagram for the Third Yoneda Lemma (right):

$$
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & RC \\
(C{\to}\_) \arrow[r, "T"'] \arrow[dr, "T'"'] & (1{\to}R\_) \arrow[d, leftrightarrow] \\
 & R
\end{tikzcd}
\qquad
\begin{tikzcd}
 & 1 \arrow[d, "\gamma"] \\
C \arrow[r, mapsto] & (B{\to}C) \\
(C{\to}\_) \arrow[r, "T"'] \arrow[dr, "T'"'] & (1{\to}(B{\to}\_)) \arrow[d, leftrightarrow] \\
 & (B{\to}\_)
\end{tikzcd}
$$

Our Third Yoneda Lemma is usually stated as this bijection:
$(B{\to}C) \leftrightarrow ((C{\to}\_) \to (B{\to}\_))$, where the right side is
the space of natural transformations from $(C{\to}\_)$ to $(B{\to}\_)$).

HELP
NEEDED

Help needed: proof assistants

I was never able to learn enough Coq or Agda...

I guess that it would be easy to formalize the figure with the three Yoneda Lemmas in Coq or Agda. We can number its objects as

$$\begin{array}{cccccc}
 & \mathrm{o11} & & \mathrm{o21} & & \mathrm{o31} \\
\mathrm{o12} & \mathrm{o13} & \mathrm{o22} & \mathrm{o23} & \mathrm{o32} & \mathrm{o33} \\
 & & & & & \\
\mathrm{o14} & \mathrm{o15} & \mathrm{o24} & \mathrm{o25} & \mathrm{o34} & \mathrm{o35} \\
 & & & \mathrm{o26} & & \mathrm{o36}
\end{array}$$

and choose some convention for the ascii names for arrows, and for the ascii names for arrows between arrows.

Help needed: proof assistants (2)

Smart proof assistants should be able to find by themselves the proofs that we said that were "obvious". Besides the obvious proofs I've said that some constructions are "obvious". Finding obvious "constructions" needs term inference instead of proof inference, and implementation of term inference are rare.

DICTIONARIES

Same shape, several notations
Now that we have a shape for the three Yoneda Lemmas we can change the notation — i.e., what is written in each of the nodes that we named o11, o12, ..., o36 a few slides ago, and also change what is written in the arrows...

For typographical reasons — I don't have good ways to put labels along curved arrows — I will have to commit the abuse of diagrammatical language explained in the slide "Drawing the bijection" (p.13), and draw the curved bijections as just their vertical-ish lower halves.

Categories for the Working Mathematician
Here is how MacLane states our YLs in his CWM.
Our first YL is implicit in his Proposition 1 in p.59:

> **Proposition 1.** *For a functor* $S : D \to C$ *a pair* $\langle r, u : c \to Sr \rangle$ *is universal from* c *to* S *if and only if the function sending each* $f' : r \to d$ *into* $Sf'u : c \to Sd$ *is a bijection of hom-sets*
>
> $$D(r, d) \cong C(c, Sd). \tag{1}$$
>
> *This bijection is natural in* d. *Conversely, given* r *and* c, *any natural isomorphism (1) is determined in this way by a unique arrow* $u : c \to Sr$ *such that* $\langle r, u \rangle$ *is universal from* c *to* S.

Categories for the Working Mathematician (2)
Our second YL appears in p.61 of CWM, as this:

> **Lemma** *(Yoneda). If $K : D \to$ **Set** is a functor from D and r an object in D (for D a category with small hom-sets), there is a bijection*
>
> $$y : \mathrm{Nat}(D(r, -), K) \cong Kr$$
>
> *which sends each natural transformation $\alpha : D(r, -) \xrightarrow{\bullet} K$ to $\alpha_r 1_r$, the image of the identity $r \to r$.*

Categories for the Working Mathematician (3)
Our third YL also appears in p.61 of CWM, as a corollary:

> **Corollary.** *For objects $r, s \in D$, each natural transformation $D(r,-) \to D(s,-)$ has the form $D(h,-)$ for a unique arrow $h : s \to r$.*

Categories for the Working Mathematician (4)

$$
\begin{tikzcd}
 & c \arrow[d, "u"] \\
r \arrow[r, mapsto] \arrow[d, shift right, mapsto] & Sr \\
D(r,-) \arrow[r, "T"'] \arrow[u, shift right, mapsto] & C(c,S-)
\end{tikzcd}
\qquad
\begin{tikzcd}
 & * \arrow[d, "u"] \\
r \arrow[r, mapsto] \arrow[d, shift right, mapsto] & Kr \\
D(r,-) \arrow[r] \arrow[dr, "T'"'] \arrow[u, shift right, mapsto, "y"'] & \mathbf{Set}(*,K-) \arrow[d, leftrightarrow] \\
 & K
\end{tikzcd}
\qquad
\begin{tikzcd}
 & * \arrow[d, "f"] \\
r \arrow[r, mapsto] \arrow[d, shift right, mapsto, "Y"'] & D(s,r) \\
D(r,-) \arrow[r] \arrow[dr, "{D(f,-)}"'] \arrow[u, shift right, mapsto] & \mathbf{Set}(*,D(s,-)) \arrow[d, leftrightarrow] \\
 & D(s,-)
\end{tikzcd}
$$

ND for Cats

$$
\dfrac{
\dfrac{\dfrac{\dfrac{C:\quad [D:]^1}{C{\to}D:}}{\lambda D.C{\to}D:}}{(C{\to}\underline{\ \ })_0:}\ \text{ren}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[:C{\to}D]^2\quad [:D{\to}E]^3}{:C{\to}E}}{:(C{\to}D){\to}(C{\to}E)}\ 2}{:(D{\to}E){\to}((C{\to}D){\to}(C{\to}E))}\ 3}{:\Pi D.(D{\to}E){\to}((C{\to}D){\to}(C{\to}E))}}{:\Pi D.\Pi E.(D{\to}E){\to}((C{\to}D){\to}(C{\to}E))}}{(C{\to}\underline{\ \ })_1:}\ \text{ren}
}{(C{\to}\underline{\ \ }):}
$$

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{[D :]^1}{RD :}
}{A \to RD :}
}{\wedge\, D.A \to RD :}
}{(A \to R\_)_0 :}\ \mathrm{ren}
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
[: A \to RD]^2 \qquad \dfrac{[: D \to E]^3}{: RD \to RE}
}{: A \to RE}
}{: (A \to RD) \to (A \to RE)}\ 2
}{: (D \to E) \to ((A \to RD) \to (A \to RE))}\ 3
}{: \Pi E.(D \to E) \to ((A \to RD) \to (A \to RE))}
}{: \Pi D.\Pi E.(D \to E) \to ((A \to RD) \to (A \to RE))}
}{(A \to R\_)_1 :}\ \mathrm{ren}
}{(A \to R\_) :}
$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{[: A \to RC]^2 \qquad \cfrac{[: C \to D]^1}{: RC \to RD}}{: A \to RD}}{: (C \to D) \to (A \to RD)}\ 1}{: \Pi D.(C \to D) \to (A \to RD)}}{((C \to \_\_) \to (A \to R\_\_))_0 :}\ \mathrm{ren}}{: (C \to \_\_) \to (A \to R\_\_)} \qquad \cfrac{\cfrac{C :}{: C \to C} \qquad \cfrac{C : \quad [: (C \to \_\_) \to (A \to R\_}{: (C \to C) \to (A \to RC)}}{: A \to RC}}{: (A \to RC) \leftrightarrow ((C \to \_\_) \to (A \to R\_\_))}\ 2$$

Bibliography

[CWM] S. MacLane. *Categories for the Working Mathematician (2nd ed.)* Springer, 1997.

[ES52] S. Eilenberg and N. Steenrod. *Foundations of algebraic topology*. Princeton, 1952.

[Freyd76] P. Freyd. "Properties Invariant within Equivalence Types of Categories". In: *Algebra, Topology and Category Theory: A Collection of Papers in Honour of Samuel Eilenberg*. Ed. by A. Heller and M. Tierney. `http://angg.twu.net/Freyd76.html`. Academic Press, 1976, pp. 55–61.

[GLT89] J.-Y. Girard, Y. Lafont, and P. Taylor. *Proofs and Types*. `http://www.paultaylor.eu/stable/prot.pdf`. Cambridge, 1989.

[HOTT] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. `http://saunders.phil.cmu.edu/book/hott-online.pdf`. Institute for Advanced Study, 2013.

[IDARCT] E. Ochs. "Internal Diagrams and Archetypal Reasoning in Category Theory". In: *Logica Universalis* 7.3 (Sept. 2013). http://angg.twu.net/math-b.html#idarct, pp. 291–321.

[Leinster] T. Leinster. *Basic Category Theory*. https://arxiv.org/pdf/1612.09375.pdf. Cambridge, 2014.

[MDE] E. Ochs. "On Some Missing Diagrams in The Elephant". http://angg.twu.net/math-b.html#missing-diagrams-elephant. 2019.

[Perrone] P. Perrone. "Notes on Category Theory with examples from basic mathematics". https://arxiv.org/abs/1912.10642. 2020.

[PH1] E. Ochs. "Planar Heyting Algebras for Children". In: *South American Journal of Logic* 5.1

(2019). http://angg.twu.net/math-b.html#zhas-for-children-2, pp. 125–164.

[Riehl] E. Riehl. *Category Theory in Context*. http://www.math.jhu.edu/~eriehl/context.pdf. Dover, 2016.

[WadlerPaT] P. Wadler. “Propositions as Types”. In: *Communications of the ACM* 58.12 (Dec. 2015). http://homepages.inf.ed.ac.uk/wadler/topics/history.html#propositions-as-types, pp. 75–84.