Category Theory

Andrew Pitts

University of Cambridge
Computer Science Tripos
Part II Unit of Assessment
Part III and MPhil. ACS Module L108
Michaelmas Term 2022

Course web page

Go to

https://www.cl.cam.ac.uk/teaching/2223/CAT/

https://www.cl.cam.ac.uk/teaching/2223/L108/

for

- these slides and lecture recordings
- exercise sheets and details of examples classes

  (trying the exercises is essential!)
- pointers to some additional material

Recommended text for the course is:

[Awodey] Steve Awodey, *Category theory*,
Oxford University Press (2nd ed.), 2010.

Assessment

- **A graded exercise sheet** (25% of the final mark).
  issued in lecture 10 with a one week deadline
- **A take-home test** (75% of the final mark).
  issued after the end of the course

There will be two one-hour example class sessions to provide help with the exercises.

See course web page for dates and deadlines.

Links to the lectures

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

Lecture 1

What is category theory?

> What we are probably seeking is a “purer” view of functions: a theory of functions in themselves, not a theory of functions derived from sets. What, then, is a pure theory of functions? Answer: category theory.
>
> Dana Scott, *Relating theories of the λ-calculus*, p406

set theory gives an “element-oriented” account of mathematical structure, whereas

category theory takes a ‘function-oriented” view – understand structures not via their elements, but by how they transform, i.e. via morphisms.

(Both theories are part of Logic, broadly construed.)

GENERAL THEORY OF NATURAL EQUIVALENCES

BY

SAMUEL EILENBERG AND SAUNDERS MacLANE

CONTENTS

Introduction. The subject matter of this paper is best explained by an example, such as that of the relation between a vector space L and its "dual"

Presented to the Society, September 8, 1942; received by the editors May 15, 1945.

Category Theory emerges

1945 Eilenberg† and MacLane†
General Theory of Natural Equivalences,
Trans AMS 58, 231–294

(algebraic topology, abstract algebra)

1950s Grothendieck† (algebraic geometry)

1960s Lawvere (logic and foundations)

1970s Joyal and Tierney† (elementary topos theory)

1980s Dana Scott, Plotkin

(semantics of programming languages)

Lambek† (linguistics)

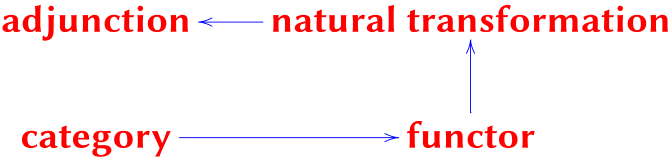
Category Theory and Computer Science

> "Category theory has…become part of the standard "tool-box" in many areas of theoretical informatics, from programming languages to automata, from process calculi to Type Theory."
>
> Dagstuhl Perpectives Workshop on *Categorical Methods at the Crossroads*
> April 2014

See http://www.appliedcategorytheory.org/events for recent examples of category theory being applied (not just in computer science).

This course

basic concepts of category theory



adjunction

natural transformation

category

functor

applied to $\left\{\begin{array}{l}\textbf{typed lambda-calculus}\\ \textbf{functional programming}\end{array}\right.$

Definition

A category $\mathbf{C}$ is specified by

- a set $\boxed{\mathtt{obj}\,\mathbf{C}}$ whose elements are called $\mathbf{C}$-objects
- for each $X, Y \in \mathtt{obj}\,\mathbf{C}$, a set $\boxed{\mathbf{C}(X, Y)}$ whose elements are called $\mathbf{C}$-morphisms from X to Y
- a function assigning to each $X \in \mathtt{obj}\,\mathbf{C}$ an element $\boxed{\mathtt{id}_X \in \mathbf{C}(X, X)}$ called the identity morphism for the $\mathbf{C}$-object X
- a function assigning to each $f \in \mathbf{C}(X, Y)$ and $g \in \mathbf{C}(Y, Z)$ (where $X, Y, Z \in \mathtt{obj}\,\mathbf{C}$) an element $\boxed{g \circ f \in \mathbf{C}(X, Z)}$ called the composition of $\mathbf{C}$-morphisms f and g and satisfying…

Definition, continued

satisfying…

- associativity: for all $X, Y, Z, W \in \mathtt{obj}\,\mathbf{C}$, $f \in \mathbf{C}(X, Y)$, $g \in \mathbf{C}(Y, Z)$ and $h \in \mathbf{C}(Z, W)$

$$\boxed{h \circ (g \circ f) = (h \circ g) \circ f}$$

- unity: for all $X, Y \in \mathtt{obj}\,\mathbf{C}$ and $f \in \mathbf{C}(X, Y)$

$$\boxed{\mathtt{id}_Y \circ f = f = f \circ \mathtt{id}_X}$$

Example: category of sets, **Set**

- `obj` **Set** = some fixed universe of sets

  (more on universes later)

- $\mathbf{Set}(X, Y) =$

  $\{f \subseteq X \times Y \mid f$ is single-valued and total$\}$

Cartesian product of sets X and Y is the set of all ordered pairs (x, y) with $x \in X$ and $y \in Y$.

Equality of ordered pairs:

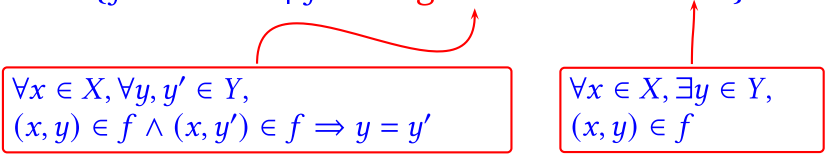
$(x, y) = (x', y') \Leftrightarrow x = x' \wedge y = y'$

Example: category of sets, **Set**

- `obj` **Set** = some fixed universe of sets

  (more on universes later)

- $\mathbf{Set}(X, Y) =$
  $\{f \subseteq X \times Y \mid f$ is single-valued and total$\}$



$\forall x \in X, \forall y, y' \in Y,$
$(x, y) \in f \wedge (x, y') \in f \Rightarrow y = y'$

$\forall x \in X, \exists y \in Y,$
$(x, y) \in f$

Example: category of sets, **Set**

- $\mathtt{obj}\,\mathbf{Set}$ = some fixed universe of sets

  (more on universes later)

- $\mathbf{Set}(X, Y) =$
  $\{f \subseteq X \times Y \mid f$ is single-valued and total$\}$

- $\mathtt{id}_X = \{(x, x) \mid x \in X\}$

- composition of $f \in \mathbf{Set}(X, Y)$ and $g \in \mathbf{Set}(Y, Z)$ is

$$\begin{aligned} g \circ f = \{(x, z) \mid \\ &\exists y \in Y,\ (x, y) \in f \wedge (y, z) \in g\} \end{aligned}$$

(check that associativity and unity properties hold)

Example: category of sets, **Set**

Notation. Given $f \in \mathbf{Set}(X, Y)$ and $x \in X$, it is usual to write $\boxed{f\,x}$ (or $f(x)$) for the unique $y \in Y$ with $(x, y) \in f$.

Thus

$$\begin{aligned} \mathtt{id}_X\, x &= x \\ (g \circ f)\, x &= g(f\, x) \end{aligned}$$

Domain and codomain

Given a category $\mathbf{C}$,

write $\boxed{f : X \to Y}$ or $\boxed{X \xrightarrow{f} Y}$

to mean that $f \in \mathbf{C}(X, Y)$,

in which case one says

object X is the domain of the morphism f
object Y is the codomain of the morphism f

and writes

$$X = \mathtt{dom}\, f \qquad Y = \mathtt{cod}\, f$$

(Which category $\mathbf{C}$ we are referring to is left implicit with this notation.)

Commutative diagrams

in a category $\mathbf{C}$:

a diagram is

a directed graph whose vertices are $\mathbf{C}$-objects and whose edges are $\mathbf{C}$-morphisms

and the diagram is commutative (or commutes) if

any two finite paths in the graph between any two vertices determine equal morphisms in the category under composition

Commutative diagrams

Examples:

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, "f"'] & Y \arrow[d, "\mathtt{id}_Y" description] \arrow[dr, "g"] & \\
& Y \arrow[r, "g"'] & Z
\end{tikzcd}$$

$$\begin{tikzcd}
X \arrow[r, "h\circ(g\circ f)"] \arrow[d, "f"'] & W \\
Y \arrow[ur, "h\circ g" description] \arrow[r, "g"'] & Z \arrow[u, "h"']
\end{tikzcd}$$

Alternative notations

I will often just write

$\mathbf{C}$ for $\mathtt{obj}\,\mathbf{C}$

$\mathtt{id}$ for $\mathtt{id}_X$

Some people write

$\mathtt{Hom}_{\mathbf{C}}(X, Y)$ for $\mathbf{C}(X, Y)$

1_X for $\mathtt{id}_X$

$g\,f$ for $g \circ f$

I use "applicative order" for morphism composition; other people use "diagrammatic order" and write

$f\,;g$ (or $f\,g$) for $g \circ f$

Alternative definition of category

The definition given here is “dependent-type friendly”.

See [Awodey, Definition 1.1] for an equivalent formulation:

One gives the whole set of morphisms $\mathtt{mor}\,\mathbf{C}$

(in bijection with $\sum_{X,Y\in\mathtt{obj}\,\mathbf{C}}\mathbf{C}(X,Y)$ in my definition)

plus functions

$$\mathtt{dom}\,,\mathtt{cod}\;:\mathtt{mor}\,\mathbf{C}\longrightarrow\mathtt{obj}\,\mathbf{C}$$
$$\mathtt{id}:\mathtt{obj}\,\mathbf{C}\longrightarrow\mathtt{mor}\,\mathbf{C}$$

and a *partial* function for composition

$$\_\circ\_:\mathtt{mor}\,\mathbf{C}\times\mathtt{mor}\,\mathbf{C}\rightharpoonup\mathtt{mor}\,\mathbf{C}$$

defined at (f,g) iff $\mathtt{cod}\,f=\mathtt{dom}\,g$

and satisfying the associativity and unity equations.

Lecture 2

Recall

A category $\mathbf{C}$ is specified by

- a set $\boxed{\texttt{obj}\,\mathbf{C}}$ whose elements are called $\mathbf{C}$-objects
- for each $X, Y \in \texttt{obj}\,\mathbf{C}$, a set $\boxed{\mathbf{C}(X, Y)}$ whose elements are called $\mathbf{C}$-morphisms from X to Y
- a function assigning to each $X \in \texttt{obj}\,\mathbf{C}$ an element $\boxed{\texttt{id}_X \in \mathbf{C}(X, X)}$ called the identity morphism for the $\mathbf{C}$-object X
- a function assigning to each $f \in \mathbf{C}(X, Y)$ and $g \in \mathbf{C}(Y, Z)$ (where $X, Y, Z \in \texttt{obj}\,\mathbf{C}$) an element $\boxed{g \circ f \in \mathbf{C}(X, Z)}$ called the composition of $\mathbf{C}$-morphisms f and g and satisfying associativity and unity properties.

Example:
category of pre-orders, **Preord**

- objects are sets P equipped with a pre-order $\_ \sqsubseteq \_$ i.e. a binary relation on P that is
  reflexive: $\forall x \in P,\ x \sqsubseteq x$
  transitive: $\forall x, y, z \in P,\ x \sqsubseteq y \wedge y \sqsubseteq z \Rightarrow x \sqsubseteq z$

A partial order is a pre-order that is also
anti-symmetric: $\forall x, y \in P,\ x \sqsubseteq y \wedge y \sqsubseteq x \Rightarrow x = y$

Example:
category of pre-orders, **Preord**

- objects are sets P equipped with a pre-order $\_ \sqsubseteq \_$
- morphisms: $\mathbf{Preord}((P_1, \sqsubseteq_1), (P_2, \sqsubseteq_2)) \triangleq \{f \in \mathbf{Set}(P_1, P_2) \mid f \text{ is monotone}\}$

$$\boxed{\forall x, x' \in P_1,\ x \sqsubseteq_1 x' \Rightarrow f\,x \sqsubseteq_2 f\,x'}$$

Example: category of pre-orders, **Preord**

- objects are sets P equipped with a pre-order $\_ \sqsubseteq \_$
- morphisms: $\mathbf{Preord}((P_1, \sqsubseteq_1), (P_2, \sqsubseteq_2)) \triangleq \{f \in \mathbf{Set}(P_1, P_2) \mid f \text{ is monotone}\}$
- identities and composition: as for **Set**

Q: why is this well-defined?
A: because the set of monotone functions contains identity functions and is closed under composition.

Example: category of pre-orders, **Preord**

- objects are sets P equipped with a pre-order $\_ \sqsubseteq \_$
- morphisms: $\mathbf{Preord}((P_1, \sqsubseteq_1), (P_2, \sqsubseteq_2)) \triangleq \{f \in \mathbf{Set}(P_1, P_2) \mid f \text{ is monotone}\}$
- identities and composition: as for **Set**

Pre- and partial orders are relevant to the denotational semantics of programming languages (among other things).

Example:
category of monoids, **Mon**

- objects are monoids $(M, \cdot, e)$ — set M equipped with a binary operation $\_ \cdot \_ : M \times M \to M$ which is associative $\forall x, y, z \in M,\ x \cdot (y \cdot z) = (x \cdot y) \cdot z$ has e as its unit $\forall x \in M,\ e \cdot x = x = x \cdot e$

  CS-relevant example of a monoid: $(\mathtt{List}\,\Sigma, @, \mathtt{nil})$ where

$$\begin{array}{rcl} \mathtt{List}\,\Sigma & = & \text{set of finite lists of elements of set } \Sigma \\ @ & = & \text{list concatenation} \\ & & \quad \mathtt{nil} \mathbin{@} \ell = \ell \\ & & \quad (a :: \ell) \mathbin{@} \ell' = a :: (\ell \mathbin{@} \ell') \\ \mathtt{nil} & = & \text{empty list} \end{array}$$

Example:
category of monoids, $\mathbf{Mon}$

- objects are monoids $(M, \cdot, e)$
- morphisms: $\mathbf{Mon}((M_1, \cdot_1, e_1), (M_2, \cdot_2, e_2)) \triangleq$
$$\{f \in \mathbf{Set}(M_1, M_2) \mid f\, e_1 = e_2 \wedge \\ \qquad \forall x, y \in M_1,\ f(x \cdot_1 y) = (f\, x) \cdot_2 (f\, y)\}$$

It's common to denote a monoid $(M, \cdot, e)$ just by its underlying set M, leaving $\_ \cdot \_$ and e implicit (hence the same notation gets used for different instances of monoid operations).

Example:
category of monoids, **Mon**

- objects are monoids $(M, \cdot, e)$
- morphisms: $\mathbf{Mon}((M_1, \cdot_1, e_1), (M_2, \cdot_2, e_2)) \triangleq$
  $\{f \in \mathbf{Set}(M_1, M_2) \mid f\ e_1 = e_2 \wedge$
  $\quad \forall x, y \in M_1,\ f(x \cdot_1 y) = (f\ x) \cdot_2 (f\ y)\}$
- identities and composition: as for **Set**

Q: why is this well-defined?
A: because the set of functions that are monoid morphisms contains identity functions and is closed under composition.

Example: category of monoids, **Mon**

- objects are monoids $(M, \cdot, e)$
- morphisms: $\mathbf{Mon}((M_1, \cdot_1, e_1), (M_2, \cdot_2, e_2)) \triangleq$ $\{f \in \mathbf{Set}(M_1, M_2) \mid f\ e_1 = e_2 \wedge$ $\forall x, y \in M_1,\ f(x \cdot_1 y) = (f\ x) \cdot_2 (f\ y)\}$
- identities and composition: as for **Set**

Monoids are relevant to automata theory (among other things).

Example: each pre-order determines a category

Given a pre-ordered set $(P, \sqsubseteq)$,
we get a category $\mathbf{C}_P$ by taking

- objects $\mathtt{obj}\,\mathbf{C}_P = P$
- morphisms $\mathbf{C}_P(x, y) \triangleq \begin{cases} 1 & \text{if } x \sqsubseteq y \\ \emptyset & \text{if } x \not\sqsubseteq y \end{cases}$

  (where 1 is some fixed one-element set and $\emptyset$ is the empty set)

Example: each pre-order determines a category

Given a pre-ordered set $(P, \sqsubseteq)$, we get a category $\mathbf{C}_P$ by taking

- objects $\mathtt{obj}\,\mathbf{C}_P = P$
- morphisms $\mathbf{C}_P(x, y) \triangleq \begin{cases} 1 & \text{if } x \sqsubseteq y \\ \emptyset & \text{if } x \not\sqsubseteq y \end{cases}$
- identity morphisms and composition are uniquely determined (why?)

Example: each pre-order determines a category

Given a pre-ordered set $(P, \sqsubseteq)$,
we get a category $\mathbf{C}_P$ by taking

- objects $\mathtt{obj}\,\mathbf{C}_P = P$
- morphisms $\mathbf{C}_P(x, y) \triangleq \begin{cases} 1 & \text{if } x \sqsubseteq y \\ \emptyset & \text{if } x \not\sqsubseteq y \end{cases}$
- identity morphisms and composition are uniquely determined (why?)

E.g. when $(P, \sqsubseteq)$ has just one element 0

$\mathbf{C}_P =$

$$\begin{tikzcd} 0 \arrow[loop right, "\mathtt{id}_0"] \end{tikzcd}$$

one object, one morphism

Example: each pre-order determines a category

Given a pre-ordered set $(P, \sqsubseteq)$,
we get a category $\mathbf{C}_P$ by taking

- objects $\mathtt{obj}\,\mathbf{C}_P = P$
- morphisms $\mathbf{C}_P(x, y) \triangleq \begin{cases} 1 & \text{if } x \sqsubseteq y \\ \emptyset & \text{if } x \not\sqsubseteq y \end{cases}$
- identity morphisms and composition are uniquely determined (why?)

E.g. when $(P, \sqsubseteq)$ has just two elements $0 \sqsubseteq 1$

$$\mathbf{C}_P = \boxed{\begin{tikzcd} 0 \arrow[loop left, "\mathtt{id}_0"] \arrow[r] & 1 \arrow[loop right, "\mathtt{id}_1"] \end{tikzcd}}$$

two objects, one non-identity morphism

Example: each pre-order determines a category

Given a pre-ordered set $(P, \sqsubseteq)$,
we get a category $\mathbf{C}_P$ by taking

- objects $\mathtt{obj}\,\mathbf{C}_P = P$
- morphisms $\mathbf{C}_P(x, y) \triangleq \begin{cases} 1 & \text{if } x \sqsubseteq y \\ \emptyset & \text{if } x \not\sqsubseteq y \end{cases}$
- identity morphisms and composition are uniquely determined (why?)

Example of a finite category that does not arise from a pre-ordered set:

$$\begin{tikzcd}
0 \arrow[loop left, "\mathtt{id}_0"] \arrow[r, bend left] \arrow[r, bend right] & 1 \arrow[loop right, "\mathtt{id}_1"]
\end{tikzcd}$$

two objects, two non-identity morphisms

Example: each monoid determines a category

Given a monoid $(M, \cdot, e)$,
we get a category $\mathbf{C}_M$ by taking

- objects: $\mathtt{obj}\,\mathbf{C}_M = 1 = \{0\}$ (one-element set)
- morphisms: $\mathbf{C}_M(0, 0) = M$
- identity morphism: $\mathtt{id}_0 = e \in M = \mathbf{C}_M(0, 0)$
- composition of $f \in \mathbf{C}_M(0, 0)$ and $g \in \mathbf{C}_M(0, 0)$ is $g \cdot f \in M = \mathbf{C}_M(0, 0)$

Definition of isomorphism

Let $\mathbf{C}$ be a category. A $\mathbf{C}$-morphism $f : X \to Y$ is an isomorphism if there is some $g : Y \to X$ for which

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, "\mathtt{id}_X"'] & Y \arrow[d, "g"] \arrow[dr, "\mathtt{id}_Y"] & \\
 & X \arrow[r, "f"'] & Y
\end{tikzcd}$$

is a commutative diagram.

Definition of isomorphism

Let $\mathbf{C}$ be a category. A $\mathbf{C}$-morphism $f : X \to Y$ is an isomorphism if there is some $g : Y \to X$ with $g \circ f = \mathtt{id}_X$ and $f \circ g = \mathtt{id}_Y$.

- Such a g is uniquely determined by f (why?) and we write $\boxed{f^{-1}}$ for it.
- Given $X, Y \in \mathbf{C}$, if such an f exists, we say the objects X and Y are isomorphic in $\mathbf{C}$ and write $\boxed{X \cong Y}$

  (There may be many different f that witness the fact that X and Y are isomorphic.)

Theorem. A function $f \in \mathbf{Set}(X, Y)$ is an isomorphism in the category $\mathbf{Set}$ iff f is a bijection, that is

- injective: $\forall x, x' \in X,\ f\,x = f\,x' \Rightarrow x = x'$
- surjective: $\forall y \in Y, \exists x \in X,\ f\,x = y$

Proof…

Theorem. A function $f \in \mathbf{Set}(X, Y)$ is an isomorphism in the category **Set** iff f is a bijection, that is

- ► injective: $\forall x, x' \in X,\ f\,x = f\,x' \Rightarrow x = x'$
- ► surjective: $\forall y \in Y, \exists x \in X,\ f\,x = y$

Proof…

Theorem. A monoid morphism $f \in \mathbf{Mon}((M_1, \cdot_1, e_1), (M_2, \cdot_2, e_2))$ is an isomorphism in the category **Mon** iff $f \in \mathbf{Set}(M_1, M_2)$ is a bijection.

Proof…

Define **Poset** to be the category whose objects are posets = pre-ordered sets for which the pre-order is anti-symmetric, but is otherwise defined like the category **Preord** of pre-ordered sets.

Define **Poset** to be the category whose objects are posets = pre-ordered sets for which the pre-order is anti-symmetric, but is otherwise defined like the category **Preord** of pre-ordered sets.

Theorem. A monotone function $f \in \mathbf{Poset}((P_1, \sqsubseteq_1), (P_2, \sqsubseteq_2))$ is an isomorphism in the category **Poset** iff $f \in \mathbf{Set}(P_1, P_2)$ is a surjective function satisfying

- reflective: $\forall x, x' \in P_1,\ f\,x \sqsubseteq_2 f\,x' \Rightarrow x \sqsubseteq_1 x'$

Proof…

Define **Poset** to be the category whose objects are posets = pre-ordered sets for which the pre-order is anti-symmetric, but is otherwise defined like the category **Preord** of pre-ordered sets.

Theorem. A monotone function $f \in \mathbf{Poset}((P_1, \sqsubseteq_1), (P_2, \sqsubseteq_2))$ is an isomorphism in the category **Poset** iff $f \in \mathbf{Set}(P_1, P_2)$ is a surjective function satisfying

- reflective: $\forall x, x' \in P_1,\ f\,x \sqsubseteq_2 f\,x' \Rightarrow x \sqsubseteq_1 x'$

Proof…

(Why does this characterisation not work for **Preord**?)

Lecture 3

Category-theoretic properties

Any two isomorphic objects in a category should have the same category-theoretic properties – statements that are provable in a formal logic for category theory, whatever that is.

Instead of trying to formalize such a logic, we will just look at examples of category-theoretic properties.

Here is our first one…

Terminal object

An object T of a category $\mathbf{C}$ is terminal if for all $X \in \mathbf{C}$, there is a unique $\mathbf{C}$-morphism from X to T, which we write as $\boxed{\langle\rangle_X : X \to T}$.

So we have $\begin{cases} \forall X \in \mathbf{C},\ \langle\rangle_X \in \mathbf{C}(X,T) \\ \forall X \in \mathbf{C}, \forall f \in \mathbf{C}(X,T),\ f = \langle\rangle_X \end{cases}$

(So in particular, $\mathtt{id}_T = \langle\rangle_T$)

Sometimes we just write $\langle\rangle_X$ as $\langle\rangle$.

Some people write $!_X$ for $\langle\rangle_X$ – there is no commonly accepted notation; [Awodey] avoids using one.

Examples of terminal objects

- In **Set**: any one-element set.
- Any one-element set has a unique pre-order and this makes it terminal in **Preord** (and **Poset**)
- Any one-element set has a unique monoid structure and this makes it terminal in **Mon**.

Examples of terminal objects

- In **Set**: any one-element set.
- Any one-element set has a unique pre-order and this makes it terminal in **Preord** (and **Poset**)
- Any one-element set has a unique monoid structure and this makes it terminal in **Mon**.
- A pre-ordered set $(P, \sqsubseteq)$, regarded as a category $\mathbf{C}_P$, has a terminal object iff it has a greatest element $\top$, that is: $\forall x \in P,\ x \sqsubseteq \top$
- When does a monoid $(M, \cdot, e)$, regarded as a category $\mathbf{C}_M$, have a terminal object?

Terminal object

Theorem. In a category $\mathbf{C}$:

(a) If T is terminal and $T \cong T'$, then T' is terminal.

(b) If T and T' are both terminal, then $T \cong T'$ (and there is only one isomorphism between T and T').

In summary: terminal objects are unique up to unique isomorphism.

Proof…

Terminal object

Theorem. In a category $\mathbf{C}$:

(a) If T is terminal and $T \cong T'$, then T' is terminal.

(b) If T and T' are both terminal, then $T \cong T'$ (and there is only one isomorphism between T and T').

In summary: terminal objects are unique up to unique isomorphism.

Proof…

Notation: from now on, if a category $\mathbf{C}$ has a terminal object we will write that object as $\boxed{1}$

Opposite of a category

Given a category $\mathbf{C}$, its opposite category $\boxed{\mathbf{C}^{\mathtt{op}}}$ is defined by interchanging the operations of $\mathtt{dom}$ and $\mathtt{cod}$ in $\mathbf{C}$:

- $\mathtt{obj}\,\mathbf{C}^{\mathtt{op}} \triangleq \mathtt{obj}\,\mathbf{C}$
- $\mathbf{C}^{\mathtt{op}}(X, Y) \triangleq \mathbf{C}(Y, X)$, for all objects X and Y
- identity morphism on $X \in \mathtt{obj}\,\mathbf{C}^{\mathtt{op}}$ is $\mathtt{id}_X \in \mathbf{C}(X, X) = \mathbf{C}^{\mathtt{op}}(X, X)$
- composition in $\mathbf{C}^{\mathtt{op}}$ of $f \in \mathbf{C}^{\mathtt{op}}(X, Y)$ and $g \in \mathbf{C}^{\mathtt{op}}(Y, Z)$ is given by the composition $f \circ g \in \mathbf{C}(Z, X) = \mathbf{C}^{\mathtt{op}}(X, Z)$ in $\mathbf{C}$ (associativity and unity properties hold for this operation, because they do in $\mathbf{C}$)

The Principle of Duality

Whenever one defines a concept / proves a theorem in terms of commutative diagrams in a category $\mathbf{C}$,

one obtains another concept / theorem, called its dual,

by reversing the direction or morphisms throughout, that is, by replacing $\mathbf{C}$ by its opposite category $\mathbf{C}^{\mathrm{op}}$.

For example…

Initial object

is the dual notion to "terminal object":

An object 0 of a category $\mathbf{C}$ is initial if for all $X \in \mathbf{C}$, there is a unique $\mathbf{C}$-morphism $0 \to X$, which we write as $\boxed{[\,]_X : 0 \to X}$.

So we have $\begin{cases} \forall X \in \mathbf{C},\ [\,]_X \in \mathbf{C}(0, X) \\ \forall X \in \mathbf{C}, \forall f \in \mathbf{C}(0, X),\ f = [\,]_X \end{cases}$

(So in particular, $\mathtt{id}_0 = [\,]_0$)

By duality, we have that initial objects are unique up to isomorphism and that any object isomorphic to an initial object is itself initial.
(**N.B.** "isomorphism" is a self-dual concept.)

Examples of initial objects

- The empty set is initial in **Set**.
- Any one-element set has a uniquely determined monoid structure and is initial in **Mon**. (why?)

  So initial and terminal objects co-incide in **Mon**

  An object that is both initial and terminal in a category is sometimes called a zero object.
- A pre-ordered set $(P, \sqsubseteq)$, regarded as a category $\mathbf{C}_P$, has an initial object iff it has a least element $\bot$, that is: $\forall x \in P, \bot \sqsubseteq x$

Example: free monoids as initial objects

(relevant to automata and formal languages)

The free monoid on a set Σ is $(\mathtt{List}\,\Sigma, @, \mathtt{nil})$ where

$$\begin{aligned} \mathtt{List}\,\Sigma &= \text{set of finite lists of elements of } \Sigma \\ @ &= \text{list concatenation} \\ \mathtt{nil} &= \text{empty list} \end{aligned}$$

Example: free monoids as initial objects

(relevant to automata and formal languages)

The free monoid on a set Σ is $(\mathtt{List}\,\Sigma, @, \mathtt{nil})$ where

$$\begin{array}{rcl} \mathtt{List}\,\Sigma & = & \text{set of finite lists of elements of } \Sigma \\ @ & = & \text{list concatenation} \\ \mathtt{nil} & = & \text{empty list} \end{array}$$

The function

$$\begin{array}{rcl} \eta_\Sigma : \Sigma & \to & \mathtt{List}\,\Sigma \\ a & \mapsto & [a] = a :: \mathtt{nil} \text{ (one-element list)} \end{array}$$

has the following “universal property”…

Example:
free monoids as initial objects

(relevant to automata and formal languages)

Theorem. For any monoid $(M, \cdot, e)$ and function $f : \Sigma \longrightarrow M$, there is a unique monoid morphism $\overline{f} \in \mathbf{Mon}((\mathtt{List}\,\Sigma, @, \mathtt{nil}), (M, \cdot, e))$ making

$$\begin{tikzcd}
\Sigma \arrow[r, "\eta_\Sigma"] \arrow[dr, "f"'] & \mathtt{List}\,\Sigma \arrow[d, "\overline{f}"] \\
& M
\end{tikzcd}$$

commute in **Set**.

Proof…

Example:
free monoids as initial objects

(relevant to automata and formal languages)

Theorem. $\forall M \in \mathbf{Mon}, \forall f \in \mathbf{Set}(\Sigma, M), \exists! \overline{f} \in \mathbf{Mon}(\mathtt{List}\,\Sigma, M),\ \overline{f} \circ \eta_\Sigma = f$

The theorem just says that $\eta_\Sigma : \Sigma \to \mathtt{List}\,\Sigma$ is an initial object in the category $\Sigma/\mathbf{Mon}$:

- objects: $((M, \cdot, e), f)$ where $(M, \cdot, e) \in \mathtt{obj}\,\mathbf{Mon}$ and $f \in \mathbf{Set}(\Sigma, M)$
- morphisms in $\Sigma/\mathbf{Mon}(((M_1, \cdot_1, e_1), f_1), ((M_2, \cdot_2, e_2), f_2))$ are $f \in \mathbf{Mon}((M_1, \cdot_1, e_1), (M_2, \cdot_2, e_2))$ such that $f \circ f_1 = f_2$
- identities and composition as in $\mathbf{Mon}$

Example: free monoids as initial objects

(relevant to automata and formal languages)

Theorem. $\forall M \in \mathbf{Mon}, \forall f \in \mathbf{Set}(\Sigma, M), \exists! \overline{f} \in \mathbf{Mon}(\mathtt{List}\,\Sigma, M),\ \overline{f} \circ \eta_\Sigma = f$

The theorem just says that $\eta_\Sigma : \Sigma \longrightarrow \mathtt{List}\,\Sigma$ is an initial object in the category $\Sigma/\mathbf{Mon}$:

So this “universal property” determines the monoid $\mathtt{List}\,\Sigma$ uniquely up to isomorphism in $\mathbf{Mon}$.

We will see later that $\Sigma \mapsto \mathtt{List}\,\Sigma$ is part of a functor (= morphism of categories) which is left adjoint to the “forgetful functor” $\mathbf{Mon} \longrightarrow \mathbf{Set}$.

Lecture 4

Binary products

In a category $\mathbf{C}$, a product for objects $X, Y \in \mathbf{C}$ is a diagram $X \xleftarrow{\pi_1} P \xrightarrow{\pi_2} Y$ with the universal property:

For all $X \xleftarrow{f} Z \xrightarrow{g} Y$ in $\mathbf{C}$, there is a unique $\mathbf{C}$-morphism $h : Z \to P$ such that the following diagram commutes in $\mathbf{C}$:

$$\begin{tikzcd}
 & Z \arrow[dl, "f"'] \arrow[d, "h"] \arrow[dr, "g"] & \\
X & P \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & Y
\end{tikzcd}$$

Binary products

In a category $\mathbf{C}$, a product for objects $X, Y \in \mathbf{C}$ is a diagram $X \xleftarrow{\pi_1} P \xrightarrow{\pi_2} Y$ with the universal property:

For all $X \xleftarrow{f} Z \xrightarrow{g} Y$ in $\mathbf{C}$, there is a unique $\mathbf{C}$-morphism $h : Z \to P$ such that $f = \pi_1 \circ h$ and $g = \pi_2 \circ h$

So (P, π_1, π_2) is a terminal object in the category with

- objects: (Z, f, g) where $X \xleftarrow{f} Z \xrightarrow{g} Y$ in $\mathbf{C}$
- morphisms $h : (Z_1, f_1, g_1) \to (Z_2, f_2, g_2)$ are $h \in \mathbf{C}(Z_1, Z_2)$ such that $f_1 = f_2 \circ h$ and $g_1 = g_2 \circ h$
- composition and identities as in $\mathbf{C}$

So if it exists, the binary product of two objects in a category is unique up to (unique) isomophism.

Binary products

In a category $\mathbf{C}$, a product for objects $X, Y \in \mathbf{C}$ is a diagram $X \xleftarrow{\pi_1} P \xrightarrow{\pi_2} Y$ with the universal property:

For all $X \xleftarrow{f} Z \xrightarrow{g} Y$ in $\mathbf{C}$, there is a unique $\mathbf{C}$-morphism $h : Z \longrightarrow P$ such that $f = \pi_1 \circ h$ and $g = \pi_2 \circ h$

N.B. products of objects in a category do not always exist. For example in the category

$$\begin{tikzcd} 0 \arrow[loop left, "\mathtt{id}_0"] & & 1 \arrow[loop right, "\mathtt{id}_1"] \end{tikzcd}$$

two objects, no non-identity morphisms

the objects 0 and 1 do not have a product, because there is no diagram of the form $0 \longleftarrow ? \longrightarrow 1$ in this category.

Notation for binary products

Assuming $\mathbf{C}$ has binary products of objects, the product of $X, Y \in \mathbf{C}$ is written

$$\boxed{X \xleftarrow{\pi_1} X \times Y \xrightarrow{\pi_2} Y}$$

and given $X \xleftarrow{f} Z \xrightarrow{g} Y$, the unique $h : Z \to X \times Y$ with $\pi_1 \circ h = f$ and $\pi_2 \circ h = g$ is written

$$\boxed{\langle f\,, g\rangle : Z \to X \times Y}$$

Examples of products

In **Set**, category-theoretic products are given by the usual cartesian product of sets (set of all ordered pairs)

$$
\begin{aligned}
X \times Y &= \{(x, y) \mid x \in X \wedge y \in Y\} \\
\pi_1(x, y) &= x \\
\pi_2(x, y) &= y
\end{aligned}
$$

because…

Examples of products

In **Preord**, can take product of $(P_1, \sqsubseteq_1)$ and $(P_2, \sqsubseteq_2)$ to be

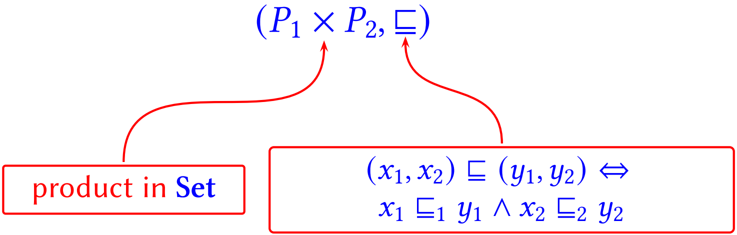
$$(P_1 \times P_2, \sqsubseteq)$$

product in **Set**

$$(x_1, x_2) \sqsubseteq (y_1, y_2) \Leftrightarrow$$
$$x_1 \sqsubseteq_1 y_1 \wedge x_2 \sqsubseteq_2 y_2$$

Examples of products

In **Preord**, can take product of $(P_1, \sqsubseteq_1)$ and $(P_2, \sqsubseteq_2)$ to be



$$(P_1 \times P_2, \sqsubseteq)$$

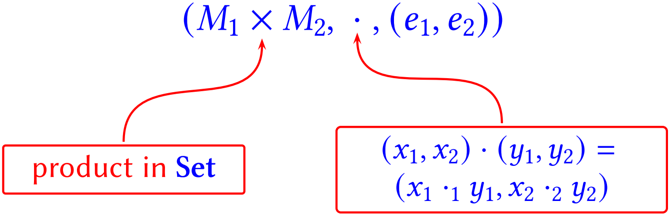
product in **Set**

$$(x_1, x_2) \sqsubseteq (y_1, y_2) \Leftrightarrow x_1 \sqsubseteq_1 y_1 \wedge x_2 \sqsubseteq_2 y_2$$

The projection functions $P_1 \xleftarrow{\pi_1} P_1 \times P_2 \xrightarrow{\pi_2} P_2$ are monotone for this pre-order on $P_1 \times P_2$ and have the universal property needed for a product in **Preord** (check).

Examples of products

In **Mon**, can take product of $(M_1, \cdot_1, e_1)$ and $(M_2, \cdot_2, e_2)$ to be



$$(M_1 \times M_2, \ \cdot \ , (e_1, e_2))$$

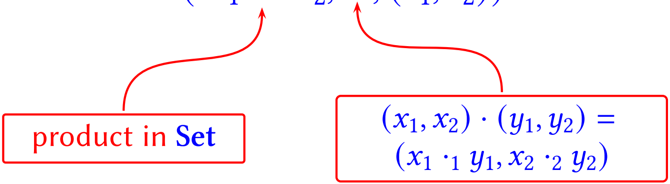
product in **Set**

$$(x_1, x_2) \cdot (y_1, y_2) = (x_1 \cdot_1 y_1, x_2 \cdot_2 y_2)$$

Examples of products

In **Mon**, can take product of $(M_1, \cdot_1, e_1)$ and $(M_2, \cdot_2, e_2)$ to be

$$(M_1 \times M_2,\ \cdot\ , (e_1, e_2))$$



product in **Set**

$(x_1, x_2) \cdot (y_1, y_2) =$
$(x_1 \cdot_1 y_1, x_2 \cdot_2 y_2)$

The projection functions $M_1 \xleftarrow{\pi_1} M_1 \times M_2 \xrightarrow{\pi_2} M_2$ are monoid morphisms for this monoid structure on $M_1 \times M_2$ and have the universal property needed for a product in **Mon** (check).

Examples of products

Recall that each pre-ordered set $(P, \sqsubseteq)$ determines a category $\mathbf{C}_P$.

Given $p, q \in P = \mathtt{obj}\,\mathbf{C}_P$, the product $p \times q$ (if it exists) is a greatest lower bound (or glb, or meet) for p and q in $(P, \sqsubseteq)$:

lower bound:

$$p \times q \sqsubseteq p \ \wedge \ p \times q \sqsubseteq q$$

greatest among all lower bounds:

$$\forall r \in P,\ r \sqsubseteq p \ \wedge \ r \sqsubseteq q \ \Rightarrow \ r \sqsubseteq p \times q$$

Notation: glbs are often written $\boxed{p \wedge q}$ or $\boxed{p \sqcap q}$

Duality

A binary coproduct of two objects in a category $\mathbf{C}$ is their product in the category $\mathbf{C}^{\mathrm{op}}$.

Duality

A binary coproduct of two objects in a category $\mathbf{C}$ is their product in the category $\mathbf{C}^{\mathrm{op}}$.

Thus the coproduct of $X, Y \in \mathbf{C}$
if it exists,
is a diagram $X \xrightarrow{\mathtt{inl}} X + Y \xleftarrow{\mathtt{inr}} Y$
with the universal property:

$$\begin{aligned}
&\forall\ (X \xrightarrow{f} Z \xleftarrow{g} Y),\\
&\quad \exists!\ (X + Y \xrightarrow{h} Z),\\
&\qquad f = h \circ \mathtt{inl} \wedge g = h \circ \mathtt{inr}
\end{aligned}$$

Duality

A binary coproduct of two objects in a category $\mathbf{C}$ is their product in the category $\mathbf{C}^{\mathrm{op}}$.

E.g. in $\mathbf{Set}$, the coproduct of X and Y

$$X \xrightarrow{\mathtt{inl}} X + Y \xleftarrow{\mathtt{inr}} Y$$

is given by their disjoint union (tagged sum)

$$\begin{aligned} X + Y &= \{(0, x) \mid x \in X\} \cup \{(1, y) \mid y \in Y\} \\ \mathtt{inl}(x) &= (0, x) \\ \mathtt{inr}(y) &= (1, y) \end{aligned}$$

(prove this)

Lecture 5

Exponentials

Given $X, Y \in \mathbf{Set}$, let $Y^X \in \mathbf{Set}$ denote the set of all functions from X to Y.

$$Y^X = \mathbf{Set}(X, Y) = \{f \subseteq X \times Y \mid f \text{ is single-valued and total}\}$$

Aim to characterise Y^X category theoretically.

Exponentials

Given $X, Y \in \mathbf{Set}$, let $Y^X \in \mathbf{Set}$ denote the set of all functions from X to Y.

Aim to characterise Y^X category theoretically.

Function application gives a morphism $\mathsf{app} : Y^X \times X \to Y$ in $\mathbf{Set}$.

$$\mathsf{app}(f, x) = f\,x \qquad (f \in Y^X, x \in X)$$

so as a set of ordered pairs, app is
$\{((f, x), y) \in (Y^X \times X) \times Y \mid (x, y) \in f\}$

Exponentials

Given $X, Y \in \mathbf{Set}$, let $Y^X \in \mathbf{Set}$ denote the set of all functions from X to Y.

Aim to characterise Y^X category theoretically.

Function application gives a morphism $\mathtt{app} : Y^X \times X \to Y$ in $\mathbf{Set}$.

Currying operation transforms morphisms $f : Z \times X \to Y$ in $\mathbf{Set}$ to morphisms $\mathtt{cur}\, f : Z \to Y^X$

$$\mathtt{cur}\, f\, z\, x = f(z, x) \qquad (f \in Y^X, z \in Z, x \in X)$$

$$\boxed{\begin{gathered}\mathtt{cur}\, f\, z = \{(x, y) \mid ((z, x), y) \in f\} \\ \mathtt{cur}\, f = \{(z, g) \mid g = \{(x, y) \mid ((z, x), y) \in f\}\}\end{gathered}}$$

Haskell Curry

Haskell Brooks Curry (/ˈhæskəl/; September 12, 1900 – September 1, 1982) was an American mathematician and logician. Curry is best known for his work in combinatory logic; while the initial concept of combinatory logic was based on a single paper by Moses Schönfinkel,[1] much of the development was done by Curry. Curry is also known for Curry's paradox and the Curry–Howard correspondence. There are three programming languages named after him, Haskell, Brook and Curry, as well as the concept of *currying*, a

Haskell Brooks Curry	
Born	September 12, 1900 Millis, Massachusetts
Died	September 1, 1982 (aged 81) State College, Pennsylvania
Nationality	American
Alma mater	Harvard University
Known for	Combinatory logic Curry–Howard correspondence

For each function $f : Z \times X \to Y$ we get a commutative diagram in $\mathbf{Set}$:

$$\begin{tikzcd}
Y^X \times X \arrow[r, "\mathtt{app}"] & Y \\
Z \times X \arrow[u, "\mathtt{cur}\, f \times \mathtt{id}_X"] \arrow[ur, "f"']
\end{tikzcd}$$

$$\begin{tikzcd}
(\mathtt{cur}\, f\, z, x) \arrow[r, mapsto] & \mathtt{cur}\, f\, z\, x = f(z, x) \\
(z, x) \arrow[u, mapsto] \arrow[ur, mapsto]
\end{tikzcd}$$

For each function $f : Z \times X \to Y$ we get a commutative diagram in **Set**:

$$\begin{tikzcd}
Y^X \times X \arrow[r, "\mathtt{app}"] & Y \\
Z \times X \arrow[u, "\mathtt{cur}\, f \times \mathtt{id}_X"] \arrow[ur, "f"'] &
\end{tikzcd}$$

Furthermore, if any function $g : Z \to Y^X$ also satisfies

$$\begin{tikzcd}
Y^X \times X \arrow[r, "\mathtt{app}"] & Y \\
Z \times X \arrow[u, "g \times \mathtt{id}_X"] \arrow[ur, "f"'] &
\end{tikzcd}$$

then $g = \mathtt{cur}\, f$, because of function extensionality…

Function Extensionality

Two functions $f, g \in Y^X$ are equal if (and only if) $\forall x \in X,\ f\,x = g\,x$.

This is true of the set-theoretic notion of function, because then

$$\begin{aligned} & \{(x, f\,x) \mid x \in X\} = \{(x, g\,x) \mid x \in X\} \\ \text{i.e.} \quad & \{(x, y) \mid (x, y) \in f\} = \{(x, y) \mid (x, y) \in g\} \\ \text{i.e.} \quad & f = g \end{aligned}$$

(in other words it reduces to the extensionality property of sets: two sets are equal iff they have the same elements).

Exponential objects

Suppose a category $\mathbf{C}$ has binary products, that is, for every pair of $\mathbf{C}$-objects X and Y there is a product diagram $X \xleftarrow{\pi_1} X \times Y \xrightarrow{\pi_2} Y$.

Notation: given $f \in \mathbf{C}(X, X')$ and $f' \in \mathbf{C}(Y, Y')$, then $\boxed{f \times f' : X \times Y \to X' \times Y'}$ stands for $\langle f \circ \pi_1 \, , f' \circ \pi_2 \rangle$,
that is, the unique morphism $g \in \mathbf{C}(X \times Y, X' \times Y')$ satisfying

$\pi_1 \circ g = f \circ \pi_1$ and $\pi_2 \circ g = f' \circ \pi_2$.

Exponential objects

Suppose a category $\mathbf{C}$ has binary products.

An exponential for $\mathbf{C}$-objects X and Y is specified by

$$\text{object } Y^X + \text{morphism } \mathtt{app} : Y^X \times X \to Y$$

satisfying the universal property

for all $Z \in \mathbf{C}$ and $f \in \mathbf{C}(Z \times X, Y)$, there is a unique $g \in \mathbf{C}(Z, Y^X)$ such that

$$\begin{tikzcd}
Y^X \times X \arrow[r, "\mathtt{app}"] & Y \\
Z \times X \arrow[u, "g\times\mathtt{id}_X"] \arrow[ur, "f"'] &
\end{tikzcd}$$

commutes in $\mathbf{C}$.

Notation: we write $\boxed{\mathtt{cur}\, f}$ for the unique g such that $\mathtt{app} \circ (g \times \mathtt{id}_X) = f$.

Exponential objects

The universal property of $\mathtt{app} : Y^X \times X \to Y$ says that there is a bijection

$$\begin{aligned} \mathbf{C}(Z, Y^X) &\cong \mathbf{C}(Z \times X, Y) \\ g &\mapsto \mathtt{app} \circ (g \times \mathtt{id}_X) \\ \mathtt{cur}\, f &\leftarrow\!\shortmid f \\ \mathtt{app} \circ (\mathtt{cur}\, f \times \mathtt{id}_X) &= f \\ g &= \mathtt{cur}(\mathtt{app} \circ (g \times \mathtt{id}_X)) \end{aligned}$$

Exponential objects

The universal property of $\mathtt{app} : Y^X \times X \to Y$ says that there is a bijection…

It also says that $(Y^X, \mathtt{app})$ is a terminal object in the following category:

- objects: (Z, f) where $f \in \mathbf{C}(Z \times X, Y)$
- morphisms $g : (Z, f) \to (Z', f')$ are $g \in \mathbf{C}(Z, Z')$ such that $f' \circ (g \times \mathtt{id}_X) = f$
- composition and identities as in $\mathbf{C}$.

So when they exist, exponential objects are unique up to (unique) isomorphism.

Cartesian closed category

Definition. C is a cartesian closed category (ccc) if it is a category with a terminal object, binary products and exponentials of any pair of objects.

This is a key concept for the semantics of lambda calculus and for the foundations of functional programming languages.

Notation: an exponential object Y^X is often written as $\boxed{X \to Y}$

Cartesian closed category

Definition. $\mathbf{C}$ is a cartesian closed category (ccc) if it is a category with a terminal object, binary products and exponentials of any pair of objects.

Examples:

- **Set** is a ccc — as we have seen.
- **Preord** is a ccc: we already saw that it has a terminal object and binary products; the exponential of $(P_1, \sqsubseteq_1)$ and $(P_2, \sqsubseteq_2)$ is $(P_1 \to P_2, \sqsubseteq)$ where

$$\begin{aligned} P_1 \to P_2 &= \mathbf{Preord}((P_1, \sqsubseteq_1), (P_2, \sqsubseteq_2)) \\ f \sqsubseteq g &\Leftrightarrow \forall x \in P_1,\ f\,x \sqsubseteq_2 g\,x \end{aligned}$$

(check that this is a pre-order and does give an exponential in **Preord**)

Lecture 6

Course assessment—heads up

Graded exercise sheet (Ex.Sh.#4) for 25% credit

- issued 12:00 on Friday 28 October 2022 via Moodle
- your answers are due (via Moodle) by 12:00 on Friday 4 November 2022

Take-home exam, 75% credit, will be available via Moodle from 12:00 on Friday 25 November 2022, with solutions to be submitted by 12:00 on Friday 2 December 2022.

ccc

Recall:

Definition. $\mathbb{C}$ is a cartesian closed category (ccc) if it is a category with a terminal object, binary products and exponentials of any pair of objects.

Non-example of a ccc

The category **Mon** of monoids has a terminal object and binary products, but is not a ccc

because of the following bijections between sets, where 1 denotes a one-element set and the corresponding one-element monoid:

$$\begin{aligned}\mathbf{Set}(1, \mathtt{List}\,1) &\cong \mathbf{Mon}(\mathtt{List}\,1, \mathtt{List}\,1)\\ &\cong \mathbf{Mon}(1 \times \mathtt{List}\,1, \mathtt{List}\,1)\end{aligned}$$

by universal property of the free monoid $\mathtt{List}\,1$ on the one-element set 1

by Ex.Sh. 2, qu. 2 (1 is terminal in **Mon**)

Non-example of a ccc

The category $\mathbf{Mon}$ of monoids has a terminal object and binary products, but is <u>not</u> a ccc

because of the following bijections between sets, where $\mathtt{1}$ denotes a one-element set and the corresponding one-element monoid:

$$
\begin{aligned}
\mathbf{Set}(\mathtt{1}, \mathtt{List}\,\mathtt{1}) &\cong \mathbf{Mon}(\mathtt{List}\,\mathtt{1}, \mathtt{List}\,\mathtt{1}) \\
&\cong \mathbf{Mon}(\mathtt{1} \times \mathtt{List}\,\mathtt{1}, \mathtt{List}\,\mathtt{1})
\end{aligned}
$$

Since $\mathbf{Set}(\mathtt{1}, \mathtt{List}\,\mathtt{1})$ is countably infinite, so is $\mathbf{Mon}(\mathtt{1} \times \mathtt{List}\,\mathtt{1}, \mathtt{List}\,\mathtt{1})$.

Since the one-element monoid is initial (see Lect. 3) in $\mathbf{Mon}$, for any $M \in \mathbf{Mon}$ we have that $\mathbf{Mon}(\mathtt{1}, M)$ has just one element and hence

$$
\mathbf{Mon}(\mathtt{1} \times \mathtt{List}\,\mathtt{1}, \mathtt{List}\,\mathtt{1}) \ \ncong \ \mathbf{Mon}(\mathtt{1}, M)
$$

Therefore no M can be the exponential of the objects $\mathtt{List}\,\mathtt{1}$ and $\mathtt{List}\,\mathtt{1}$ in $\mathbf{Mon}$.

Cartesian closed pre-order

Recall that each pre-ordered set $(P, \sqsubseteq)$ gives a category $\mathbf{C}_P$. It is a ccc iff P has

- a greatest element $\top$: $\forall p \in P,\ p \sqsubseteq \top$
- binary meets $p \wedge q$:
  $\forall r \in P,\ r \sqsubseteq p \wedge q \iff r \sqsubseteq p \wedge r \sqsubseteq q$
- Heyting implications $p \to q$:
  $\forall r \in P,\ r \sqsubseteq p \to q \iff r \wedge p \sqsubseteq q$

Cartesian closed pre-order

Recall that each pre-ordered set $(P, \sqsubseteq)$ gives a category $\mathbf{C}_P$. It is a ccc iff P has

- a greatest element $\top$: $\forall p \in P,\ p \sqsubseteq \top$
- binary meets $p \wedge q$:
  $\forall r \in P,\ r \sqsubseteq p \wedge q \ \Leftrightarrow\ r \sqsubseteq p \wedge r \sqsubseteq q$
- Heyting implications $p \rightarrow q$:
  $\forall r \in P,\ r \sqsubseteq p \rightarrow q \ \Leftrightarrow\ r \wedge p \sqsubseteq q$

E.g. any Boolean algebra (with $p \rightarrow q = \neg p \vee q$).

E.g. $([0,1], \leq)$ with $\top = 1$, $p \wedge q = \min\{p, q\}$ and $p \rightarrow q = \begin{cases} 1 & \text{if } p \leq q \\ q & \text{if } q < p \end{cases}$

Intuitionistic Propositional Logic (IPL)

We present it in “natural deduction” style and only consider the fragment with conjunction and implication, with the following syntax:

Formulas of IPL: $\varphi, \psi, \theta, \ldots ::=$

$p, q, r, \ldots$	propositional identifiers
$\texttt{true}$	truth
$\varphi \mathbin{\&} \psi$	conjunction
$\varphi \Rightarrow \psi$	implication

Sequents of IPL:

Φ	$::=$	$\diamond$	empty
		Φ, ϕ	non=empty

(so sequents are finite snoc-lists of formulas)

IPL entailment $\Phi \vdash \varphi$

The intended meaning of $\Phi \vdash \varphi$ is “the conjunction of the formulas in Φ implies the formula φ”. The relation $\_ \vdash \_$ is inductively generated by the following rules:

$$\frac{}{\Phi, \varphi \vdash \varphi}\ (\textsc{ax}) \qquad \frac{\Phi \vdash \varphi}{\Phi, \psi \vdash \varphi}\ (\textsc{wk}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi, \varphi \vdash \psi}{\Phi \vdash \psi}\ (\textsc{cut})$$

$$\frac{}{\Phi \vdash \mathtt{true}}\ (\textsc{true}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi \vdash \psi}{\Phi \vdash \varphi \mathbin{\&} \psi}\ (\&\textsc{i}) \qquad \frac{\Phi, \varphi \vdash \psi}{\Phi \vdash \varphi \Rightarrow \psi}\ ({\Rightarrow}\textsc{i})$$

$$\frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \varphi}\ (\&\textsc{e}_1) \qquad \frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \psi}\ (\&\textsc{e}_2) \qquad \frac{\Phi \vdash \varphi \Rightarrow \psi \qquad \Phi \vdash \varphi}{\Phi \vdash \psi}\ ({\Rightarrow}\textsc{e})$$

For example, if $\Phi = \diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta$, then $\Phi \vdash \varphi \Rightarrow \theta$ is provable in IPL, because:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Phi \vdash \psi \Rightarrow \theta}\,(\textsc{ax})}{\Phi, \varphi \vdash \psi \Rightarrow \theta}\,(\textsc{wk}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\diamond, \varphi \Rightarrow \psi \vdash \varphi \Rightarrow \psi}\,(\textsc{ax})}{\Phi \vdash \varphi \Rightarrow \psi}\,(\textsc{wk})}{\Phi, \varphi \vdash \varphi \Rightarrow \psi}\,(\textsc{wk}) \qquad \dfrac{}{\Phi, \varphi \vdash \varphi}\,(\textsc{ax})}{\Phi, \varphi \vdash \psi}\,(\Rightarrow\textsc{e})}{\Phi, \varphi \vdash \theta}\,(\Rightarrow\textsc{e})}{\Phi \vdash \varphi \Rightarrow \theta}\,(\Rightarrow\textsc{i})$$

Semantics of IPL

in a cartesian closed pre-oder $(P, \sqsubseteq)$

Given a function M assigning a meaning to each propositional identifier p as an element $M(p) \in P$, we can assign meanings to IPL formula φ and sequents Φ as element $M[\![\varphi]\!], M[\![\Phi]\!] \in P$ by recursion on their structure:

$$
\begin{array}{rcll}
M[\![p]\!] & = & M(p) & \\
M[\![\mathtt{true}]\!] & = & \top & \text{greatest element} \\
M[\![\varphi \mathbin{\&} \psi]\!] & = & M[\![\varphi]\!] \wedge M[\![\psi]\!] & \text{binary meet} \\
M[\![\varphi \Rightarrow \psi]\!] & = & M[\![\varphi]\!] \rightarrow M[\![\psi]\!] & \text{Heyting implication} \\
M[\![\diamond]\!] & = & \top & \text{greatest element} \\
M[\![\Phi, \varphi]\!] & = & M[\![\Phi]\!] \wedge M[\![\varphi]\!] & \text{binary meet}
\end{array}
$$

Semantics of IPL

in a cartesian closed pre-oder $(P, \sqsubseteq)$

Soundness Theorem. If $\Phi \vdash \varphi$ is provable from the rules of IPL, then $M[\![\Phi]\!] \sqsubseteq M[\![\varphi]\!]$ holds in any cartesian closed pre-order.

Proof. exercise (show that $\{(\Phi, \varphi) \mid M[\![\Phi]\!] \sqsubseteq M[\![\varphi]\!]\}$ is closed under the rules defining IPL entailment and hence contains $\{(\Phi, \varphi) \mid \Phi \vdash \varphi\}$)

Example

Peirce's Law $\diamond \vdash ((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi$

is not provable in IPL.

(whereas the formula $((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi$ is a classical tautology)

Example

Peirce's Law $\diamond \vdash ((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi$

is <u>not</u> provable in IPL.

(whereas the formula $((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi$ is a classical tautology)

For if $\diamond \vdash ((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi$ were provable in IPL, then by the Soundness Theorem we would have
$\top = M[\![\diamond]\!] \sqsubseteq M[\![((\varphi \Rightarrow \psi) \Rightarrow \varphi) \Rightarrow \varphi]\!]$.

But in the cartesian closed partial order $([0, 1], \leq)$, taking $M(p) = 1/2$ and $M(q) = 0$, we get

$$\begin{aligned} M[\![((p \Rightarrow q) \Rightarrow p) \Rightarrow p]\!] &= ((1/2 \to 0) \to 1/2) \to 1/2 \\ &= (0 \to 1/2) \to 1/2 \\ &= 1 \to 1/2 \\ &= 1/2 \\ &\not\geq 1 \end{aligned}$$

Semantics of IPL

in a cartesian closed pre-oder $(P, \sqsubseteq)$

Completeness Theorem. Given Φ, φ, if for all cartesian closed pre-orders $(P, \sqsubseteq)$ and all interpretations M of the propositional identifiers as elements of P, it is the case that $M[\![\Phi]\!] \sqsubseteq M[\![\varphi]\!]$ holds in P, then $\Phi \vdash \varphi$ is provable in IPL.

Semantics of IPL

in a cartesian closed pre-oder $(P, \sqsubseteq)$

Completeness Theorem. Given Φ, φ, if for all cartesian closed pre-orders $(P, \sqsubseteq)$ and all interpretations M of the propositional identifiers as elements of P, it is the case that $M[\![\Phi]\!] \sqsubseteq M[\![\varphi]\!]$ holds in P, then $\Phi \vdash \varphi$ is provable in IPL.

Proof. Define

$$\begin{aligned} P &\triangleq \{\text{formulas of IPL}\} \\ \varphi \sqsubseteq \psi &\triangleq \diamond, \varphi \vdash \psi \text{ is provable in IPL} \end{aligned}$$

Then one can show that $(P, \sqsubseteq)$ is a cartesian closed pre-ordered set.
For this $(P, \sqsubseteq)$, taking M to be $M(p) = p$, one can show that $M[\![\Phi]\!] \sqsubseteq M[\![\varphi]\!]$ holds in P iff $\Phi \vdash \varphi$ is provable in IPL. □

Lecture 7

IPL entailment $\Phi \vdash \varphi$

Recall the rules:

$$\frac{}{\Phi, \varphi \vdash \varphi}\ (\textsc{ax}) \qquad \frac{\Phi \vdash \varphi}{\Phi, \psi \vdash \varphi}\ (\textsc{wk}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi, \varphi \vdash \psi}{\Phi \vdash \psi}\ (\textsc{cut})$$

$$\frac{}{\Phi \vdash \mathtt{true}}\ (\textsc{true}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi \vdash \psi}{\Phi \vdash \varphi \mathbin{\&} \psi}\ (\&\textsc{i}) \qquad \frac{\Phi, \varphi \vdash \psi}{\Phi \vdash \varphi \Rightarrow \psi}\ ({\Rightarrow}\textsc{i})$$

$$\frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \varphi}\ (\&\textsc{e}_1) \qquad \frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \psi}\ (\&\textsc{e}_2) \qquad \frac{\Phi \vdash \varphi \Rightarrow \psi \qquad \Phi \vdash \varphi}{\Phi \vdash \psi}\ ({\Rightarrow}\textsc{e})$$

Proof theory

Two IPL proofs of $\diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta \vdash \varphi \Rightarrow \theta$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\Phi, \varphi \vdash \psi \Rightarrow \theta}\,(\textsc{wk}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Phi, \varphi \vdash \varphi \Rightarrow \psi}\,(\textsc{wk}) \quad \dfrac{}{\Phi, \varphi \vdash \varphi}\,(\textsc{ax})}{\Phi, \varphi \vdash \psi}\,(\Rightarrow\textsc{e})}{\Phi, \varphi \vdash \theta}\,(\Rightarrow\textsc{e})}{\Phi \vdash \varphi \Rightarrow \theta}\,(\Rightarrow\textsc{i})$$

where $\Phi \triangleq \diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Psi \vdash \varphi \Rightarrow \psi}\,(\textsc{wk}) \quad \dfrac{}{\Psi \vdash \varphi}\,(\textsc{ax})}{\Psi \vdash \psi}\,(\Rightarrow\textsc{e}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Psi, \psi \vdash \psi \Rightarrow \theta}\,(\textsc{wk}) \quad \dfrac{}{\Psi, \psi \vdash \psi}\,(\textsc{ax})}{\Psi, \psi \vdash \theta}\,(\Rightarrow\textsc{e})}{\Psi \vdash \theta}\,(\textsc{cut})}{\diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta \vdash \varphi \Rightarrow \theta}\,(\Rightarrow\textsc{i})$$

where $\Psi \triangleq \diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta, \varphi$

Proof theory

Two IPL proofs of $\diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta \vdash \varphi \Rightarrow \theta$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\Phi, \varphi \vdash \psi \Rightarrow \theta}\,(\textsc{wk}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Phi, \varphi \vdash \varphi \Rightarrow \psi}\,(\textsc{wk}) \quad \dfrac{}{\Phi, \varphi \vdash \varphi}\,(\textsc{ax})}{\Phi, \varphi \vdash \psi}\,(\Rightarrow\textsc{e})}{\Phi, \varphi \vdash \theta}\,(\Rightarrow\textsc{e})}{\Phi \vdash \varphi \Rightarrow \theta}\,(\Rightarrow\textsc{i})$$

where $\Phi \triangleq \diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Psi \vdash \varphi \Rightarrow \psi}\,(\textsc{wk}) \quad \dfrac{}{\Psi \vdash \varphi}\,(\textsc{ax})}{\Psi \vdash \psi}\,(\Rightarrow\textsc{e}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Psi, \psi \vdash \psi \Rightarrow \theta}\,(\textsc{wk}) \quad \dfrac{}{\Psi, \psi \vdash \psi}\,(\textsc{ax})}{\Psi, \psi \vdash \theta}\,(\Rightarrow\textsc{e})}{\Psi \vdash \theta}\,(\textsc{cut})}{\diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta \vdash \varphi \Rightarrow \theta}\,(\Rightarrow\textsc{i})$$

where $\Psi \triangleq \diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta, \varphi$

Why is the first proof simpler than the second one?

Proof theory

$$\frac{}{\Phi, \varphi \vdash \varphi}\ (\textsc{ax}) \qquad \frac{\Phi \vdash \varphi}{\Phi, \psi \vdash \varphi}\ (\textsc{wk}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi, \varphi \vdash \psi}{\Phi \vdash \psi}\ (\textsc{cut})$$

$$\frac{}{\Phi \vdash \mathtt{true}}\ (\textsc{true}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi \vdash \psi}{\Phi \vdash \varphi \mathbin{\&} \psi}\ (\&\textsc{i}) \qquad \frac{\Phi, \varphi \vdash \psi}{\Phi \vdash \varphi \Rightarrow \psi}\ ({\Rightarrow}\textsc{i})$$

$$\frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \varphi}\ (\&\textsc{e}_1) \qquad \frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \psi}\ (\&\textsc{e}_2) \qquad \frac{\Phi \vdash \varphi \Rightarrow \psi \qquad \Phi \vdash \varphi}{\Phi \vdash \psi}\ ({\Rightarrow}\textsc{e})$$

FACT: if an IPL sequent $\Phi \vdash \phi$ is provable from the rules, it is provable without using the (CUT) rule.

Proof theory

$$\frac{}{\Phi, \varphi \vdash \varphi}\ (\textsc{ax}) \qquad \frac{\Phi \vdash \varphi}{\Phi, \psi \vdash \varphi}\ (\textsc{wk}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi, \varphi \vdash \psi}{\Phi \vdash \psi}\ (\textsc{cut})$$

$$\frac{}{\Phi \vdash \texttt{true}}\ (\textsc{true}) \qquad \frac{\Phi \vdash \varphi \qquad \Phi \vdash \psi}{\Phi \vdash \varphi \mathbin{\&} \psi}\ (\&\textsc{i}) \qquad \frac{\Phi, \varphi \vdash \psi}{\Phi \vdash \varphi \Rightarrow \psi}\ (\Rightarrow\textsc{i})$$

$$\frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \varphi}\ (\&\textsc{e}_1) \qquad \frac{\Phi \vdash \varphi \mathbin{\&} \psi}{\Phi \vdash \psi}\ (\&\textsc{e}_2) \qquad \frac{\Phi \vdash \varphi \Rightarrow \psi \quad \Phi \vdash \varphi}{\Phi \vdash \psi}\ (\Rightarrow\textsc{e})$$

FACT: if an IPL sequent $\Phi \vdash \phi$ is provable from the rules, it is provable without using the (CUT) rule.

Simply-Typed Lambda Calculus provides a language for describing proofs in IPL and their properties…

Simply-Typed Lambda Calculus (STLC)

Types: $A, B, C, \ldots ::=$

$G, G', G'' \ldots$	“ground” types
$\mathtt{unit}$	unit type
$A \times B$	product type
$A \to B$	function type

Simply-Typed Lambda Calculus (STLC)

Types: $A, B, C, \ldots ::=$

$G, G', G'' \ldots$	"ground" types
$\mathtt{unit}$	unit type
$A \times B$	product type
$A \to B$	function type

Terms: $s, t, r, \ldots ::=$

c^A	constants (of given type A)
x	variable (countably many)
$()$	unit value
$(s\ ,\ t)$	pair
$\mathtt{fst}\ t \quad \mathtt{snd}\ t$	projections
$\lambda x : A.\ t$	function abstraction
$s\ t$	function application

STLC

Some examples of terms:

- $\lambda z : (A \to B) \times (A \to C).\, \lambda x : A.\, ((\mathtt{fst}\, z)\, x\, ,\, (\mathtt{snd}\, z)\, x))$

  (has type $((A \to B) \times (A \to C)) \to (A \to (B \times C))$)

- $\lambda z : A \to (B \times C).\, (\lambda x : A.\ \mathtt{fst}(z\, x)\, ,\, \lambda y : A.\ \mathtt{snd}(z\, y))$

  (has type $(A \to (B \times C)) \to ((A \to B) \times (A \to C))$)

- $\lambda z : A \to (B \times C).\, \lambda x : A.\, ((\mathtt{fst}\, z)\, x\, ,\, (\mathtt{snd}\, z)\, x)$

  (has no type)

STLC typing relation, $\Gamma \vdash t : A$

Γ ranges over typing environments

$$\Gamma ::= \diamond \mid \Gamma, x : A$$

(so typing environments are comma-separated snoc-lists of (variable,type)-pairs — in fact only the lists whose variables are mutually distinct get used)

The typing relation $\Gamma \vdash t : A$ is inductively defined by the following rules, which make use of the following notation

$\boxed{\Gamma\ \texttt{ok}}$ means: no variable occurs more than once in Γ

$\boxed{\texttt{dom}\,\Gamma}$ = finite set of variables occurring in Γ

STLC typing relation, $\Gamma \vdash t : A$

Typing rules for variables

$$\frac{\Gamma\ \mathtt{ok} \qquad x \notin \mathtt{dom}\,\Gamma}{\Gamma, x : A \vdash x : A}\ (\textsc{var})$$

$$\frac{\Gamma \vdash x : A \qquad x' \notin \mathtt{dom}\,\Gamma}{\Gamma, x' : A' \vdash x : A}\ (\textsc{var'})$$

Typing rules for constants and unit value

$$\frac{\Gamma\ \mathtt{ok}}{\Gamma \vdash c^A : A}\ (\textsc{cons})$$

$$\frac{\Gamma\ \mathtt{ok}}{\Gamma \vdash () : \mathtt{unit}}\ (\textsc{unit})$$

STLC typing relation, $\Gamma \vdash t : A$

Typing rules for pairs and projections

$$\frac{\Gamma \vdash s : A \qquad \Gamma \vdash t : B}{\Gamma \vdash (s\,,t) : A \times B}\ (\textsc{pair})$$

$$\frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \mathtt{fst}\, t : A}\ (\textsc{fst})$$

$$\frac{\Gamma \vdash t : A \times B}{\Gamma \vdash \mathtt{snd}\, t : B}\ (\textsc{snd})$$

STLC typing relation, $\Gamma \vdash t : A$

Typing rules for function abstraction & application

$$\frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x : A.\, t : A \to B}\ (\textsc{fun})$$

$$\frac{\Gamma \vdash s : A \to B \qquad \Gamma \vdash t : A}{\Gamma \vdash s\, t : B}\ (\textsc{app})$$

STLC typing relation, $\Gamma \vdash t : A$

Example typing derivation:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash g : B \to C}\,(\textsc{var})}{\Gamma, x : A \vdash g : B \to C}\,(\textsc{var'}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\diamond, f : A \to B \vdash f : A \to B}\,(\textsc{var})}{\Gamma \vdash f : A \to B}\,(\textsc{var'})}{\Gamma, x : A \vdash f : A \to B}\,(\textsc{var'}) \qquad \dfrac{}{\Gamma, x : A \vdash x : A}\,(\textsc{var})}{\Gamma, x : A \vdash f\,x : B}\,(\textsc{app})}{\Gamma, x : A \vdash g(f\,x) : C}\,(\textsc{app})}{\Gamma \vdash \lambda x : A.\,g(f\,x) : A \to C}\,(\textsc{fun})}{\diamond, f : A \to B \vdash \lambda g : B \to C.\,\lambda x : A.\,g(f\,x) : (B \to C) \to (A \to C)}\,(\textsc{fun})}{\diamond \vdash \lambda f : A \to B.\,\lambda g : B \to C.\,\lambda x : A.\,g(f\,x) : (A \to B) \to (B \to C) \to (A \to C)}\,(\textsc{fun})
$$

where $\Gamma \triangleq \diamond, f : A \to B, g : B \to C$

N.B. the STLC typing rules are "syntax-directed", by the structure of terms t and then in the case of variables x, by the structure of typing environments Γ.

Semantics of STLC types in a ccc

Given a cartesian closed category $\mathbf{C}$,

any function M mapping ground types G to objects $M(G) \in \mathbf{C}$

extends to function $A \mapsto M[\![A]\!] \in \mathbf{C}$ and $\Gamma \mapsto M[\![\Gamma]\!] \in \mathbf{C}$ from STLC types and typing environments to $\mathbf{C}$-objects, by recursion on the structure of A:

$$\begin{aligned}
M[\![G]\!] &= M(G) \\
M[\![\mathtt{unit}]\!] &= 1 && \text{terminal object in } \mathbf{C} \\
M[\![A \times B]\!] &= M[\![A]\!] \times M[\![B]\!] && \text{product in } \mathbf{C} \\
M[\![A \to B]\!] &= M[\![A]\!] \to M[\![B]\!] && \text{exponential in } \mathbf{C} \\
\\
M[\![\diamond]\!] &= 1 && \text{terminal object in } \mathbf{C} \\
M[\![\Gamma, x : A]\!] &= M[\![\Gamma]\!] \times M[\![A]\!] && \text{product in } \mathbf{C}
\end{aligned}$$

Lecture 8

Semantics of STLC terms in a ccc

Given a cartesian closed category $\mathbf{C}$,

given any function M mapping

- ground types G to $\mathbf{C}$-objects $M(G)$
  (which extends to a function mapping all types to objects, $A \mapsto M[\![A]\!]$, as we have seen)

Semantics of STLC terms in a ccc

Given a cartesian closed category $\mathbf{C}$,

given any function M mapping

- ground types G to $\mathbf{C}$-objects $M(G)$
- constants c^A to $\mathbf{C}$-morphisms $M(c^A) : \mathtt{1} \to M[\![A]\!]$
  (In a category with a terminal object $\mathtt{1}$, given an object $X \in \mathbf{C}$, morphisms $\mathtt{1} \to X$ are sometimes called global elements of X.)

Semantics of STLC terms in a ccc

Given a cartesian closed category $\mathbf{C}$,

given any function M mapping

- ground types G to $\mathbf{C}$-objects $M(G)$
- constants c^A to $\mathbf{C}$-morphisms $M(c^A) : 1 \to M[\![A]\!]$

we get a function mapping provable instances of the typing relation $\Gamma \vdash t : A$ to $\mathbf{C}$-morphisms

$$\boxed{M[\![\Gamma \vdash t : A]\!] : M[\![\Gamma]\!] \to M[\![A]\!]}$$

defined by recursing over the proof of $\Gamma \vdash t : A$ from the typing rules (which follows the structure of t):

Semantics of STLC terms in a ccc

Variables:

$$M[\![\Gamma, x : A \vdash x : A]\!] = M[\![\Gamma]\!] \times M[\![A]\!] \xrightarrow{\pi_2} M[\![A]\!]$$

$$M[\![\Gamma, x' : A' \vdash x : A]\!] =$$

$$M[\![\Gamma]\!] \times M[\![A']\!] \xrightarrow{\pi_1} M[\![\Gamma]\!] \xrightarrow{M[\![\Gamma \vdash x : A]\!]} M[\![A]\!]$$

Constants:

$$M[\![\Gamma \vdash c^A : A]\!] = M[\![\Gamma]\!] \xrightarrow{\langle\rangle} 1 \xrightarrow{M(c^A)} M[\![A]\!]$$

Unit value:

$$M[\![\Gamma \vdash () : \mathtt{unit}]\!] = M[\![\Gamma]\!] \xrightarrow{\langle\rangle} 1$$

Semantics of STLC terms in a ccc

Pairing:

$$M[\![\Gamma \vdash (s\,,\,t) : A \times B]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{\langle M[\![\Gamma\vdash s:A]\!], M[\![\Gamma\vdash t:B]\!]\rangle} M[\![A]\!] \times M[\![B]\!]$$

Projections:

$$M[\![\Gamma \vdash \texttt{fst}\, t : A]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{M[\![\Gamma\vdash t:A\times B]\!]} M[\![A]\!] \times M[\![B]\!] \xrightarrow{\pi_1} M[\![A]\!]$$

Semantics of STLC terms in a ccc

Pairing:

$$M[\![\Gamma \vdash (s\,,t) : A \times B]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{\langle M[\![\Gamma \vdash s:A]\!], M[\![\Gamma \vdash t:B]\!]\rangle} M[\![A]\!] \times M[\![B]\!]$$

Projections:

Given that $\Gamma \vdash \mathtt{fst}\, t : A$ holds, there is a unique type B such that $\Gamma \vdash t : A \times B$ already holds.

$$M[\![\Gamma \vdash \mathtt{fst}\, t : A]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{M[\![\Gamma \vdash t : A \times B]\!]} M[\![A]\!] \times M[\![B]\!] \xrightarrow{\pi_1} M[\![A]\!]$$

Lemma. If $\Gamma \vdash t : A$ and $\Gamma \vdash t : B$ are provable, then $A = B$.

Semantics of STLC terms in a ccc

Pairing:

$$M[\![\Gamma \vdash (s\,,t) : A \times B]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{\langle M[\![\Gamma \vdash s:A]\!], M[\![\Gamma \vdash t:B]\!]\rangle} M[\![A]\!] \times M[\![B]\!]$$

Projections:

$$M[\![\Gamma \vdash \mathtt{snd}\,t : B]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{M[\![\Gamma \vdash t:A \times B]\!]} M[\![A]\!] \times M[\![B]\!] \xrightarrow{\pi_2} M[\![B]\!]$$

(As for the case of $\mathtt{fst}$, if $\Gamma \vdash \mathtt{snd}\,t : B$, then $\Gamma \vdash t : A \times B$ already holds for a unique type A.)

Semantics of STLC terms in a ccc

Function abstraction:

$$M[\![\Gamma \vdash \lambda x : A.t : A \to B]\!] = \\ \texttt{cur}\, f : M[\![\Gamma]\!] \longrightarrow (M[\![A]\!] \to M[\![B]\!])$$

where

$$f = M[\![\Gamma, x : A \vdash t : B]\!] : M[\![\Gamma]\!] \times M[\![A]\!] \longrightarrow M[\![B]\!]$$

Semantics of STLC terms in a ccc

Function application:

$$M[\![\Gamma \vdash s\, t : B]\!] =$$

$$M[\![\Gamma]\!] \xrightarrow{\langle f,g\rangle} (M[\![A]\!] \to M[\![B]\!]) \times M[\![A]\!] \xrightarrow{\mathtt{app}} M[\![B]\!]$$

where

$A =$ unique type such that $\Gamma \vdash s : A \to B$ and $\Gamma \vdash t : A$ already holds (exists because $\Gamma \vdash s\, t : B$ holds)

$$f = M[\![\Gamma \vdash s : A \to B]\!] : M[\![\Gamma]\!] \longrightarrow (M[\![A]\!] \to M[\![B]\!])$$

$$g = M[\![\Gamma \vdash t : A]\!] : M[\![\Gamma]\!] \longrightarrow M[\![A]\!]$$

Example

Consider $\boxed{t \triangleq \lambda x : A.\, g(f\ x)}$ so that $\Gamma \vdash t : A \to C$ when $\Gamma \triangleq \diamond, f : A \to B, g : B \to C$.

Suppose $M[\![A]\!] = X$, $M[\![B]\!] = Y$ and $M[\![C]\!] = Z$ in $\mathbf{C}$. Then

$$\begin{aligned} M[\![\Gamma]\!] &= (1 \times Y^X) \times Z^Y \\ M[\![\Gamma, x : A]\!] &= ((1 \times Y^X) \times Z^Y) \times X \end{aligned}$$

$$\begin{aligned} M[\![\Gamma, x : A \vdash x : A]\!] &= \pi_2 \\ M[\![\Gamma, x : A \vdash g : B \to C]\!] &= \pi_2 \circ \pi_1 \\ M[\![\Gamma, x : A \vdash f : A \to B]\!] &= \pi_2 \circ \pi_1 \circ \pi_1 \\ M[\![\Gamma, x : A \vdash f\ x : B]\!] &= \mathtt{app} \circ \langle \pi_2 \circ \pi_1 \circ \pi_1\,, \pi_2 \rangle \\ M[\![\Gamma, x : A \vdash g\,(f\ x) : C]\!] &= \mathtt{app} \circ \langle \pi_2 \circ \pi_1\,, \mathtt{app} \circ \langle \pi_2 \circ \pi_1 \circ \pi_1\,, \pi_2 \rangle \rangle \\ M[\![\Gamma \vdash t : A \to C]\!] &= \mathtt{cur}(\mathtt{app} \circ \langle \pi_2 \circ \pi_1\,, \mathtt{app} \circ \langle \pi_2 \circ \pi_1 \circ \pi_1\,, \pi_2 \rangle \rangle) \end{aligned}$$

STLC equations

take the form $\boxed{\Gamma \vdash s = t : A}$ where $\Gamma \vdash s : A$ and $\Gamma \vdash t : A$ are provable.

Such an equation is satisfied by the semantics in a ccc if $M[\![\Gamma \vdash s : A]\!]$ and $M[\![\Gamma \vdash t : A]\!]$ are equal $\mathbf{C}$-morphisms $M[\![\Gamma]\!] \to M[\![A]\!]$.

Qu: which equations are always satisfied in any ccc?

STLC equations

take the form $\boxed{\Gamma \vdash s = t : A}$ where $\Gamma \vdash s : A$ and $\Gamma \vdash t : A$ are provable.

Such an equation is satisfied by the semantics in a ccc if $M[\![\Gamma \vdash s : A]\!]$ and $M[\![\Gamma \vdash t : A]\!]$ are equal $\mathbf{C}$-morphisms $M[\![\Gamma]\!] \to M[\![A]\!]$.

Qu: which equations are always satisfied in any ccc?

Ans: $(\alpha)\beta\eta$-equivalence — to define this, first have to define alpha-equivalence, substitution and its semantics.

Alpha equivalence of STLC terms

The names of λ-bound variables should not affect meaning.

E.g. $\lambda f : A \to B. \lambda x : A. f\ x$ should have the same meaning as $\lambda x : A \to B. \lambda y : A. x\ y$

Alpha equivalence of STLC terms

The names of λ-bound variables should not affect meaning.

E.g. $\lambda f : A \to B.\, \lambda x : A.\, f\, x$ should have the same meaning as $\lambda x : A \to B.\, \lambda y : A.\, x\, y$

This issue is best dealt with at the level of syntax rather than semantics: from now on we re-define "STLC term" to mean not an abstract syntax tree (generated as described before), but rather an equivalence class of such trees with respect to alpha-equivalence $\boxed{s =_\alpha t}$, defined as follows…

(Alternatively, one can use a "nameless" (de Bruijn) representation of terms.)

Alpha equivalence of STLC terms

$$\boxed{\frac{}{c^A =_\alpha c^A}} \quad \boxed{\frac{}{x =_\alpha x}} \quad \boxed{\frac{}{() =_\alpha ()}} \quad \boxed{\frac{s =_\alpha s' \qquad t =_\alpha t'}{(s\,,t) =_\alpha (s'\,,t')}} \quad \boxed{\frac{t =_\alpha t'}{\mathtt{fst}\,t =_\alpha \mathtt{fst}\,t'}}$$

$$\boxed{\frac{t =_\alpha t'}{\mathtt{snd}\,t =_\alpha \mathtt{snd}\,t'}} \quad \boxed{\frac{s =_\alpha s' \qquad t =_\alpha t'}{s\,t =_\alpha s'\,t'}}$$

$$\boxed{\frac{(y\ x) \cdot t =_\alpha (y\ x') \cdot t' \qquad y \text{ does not occur in } \{x, x', t, t'\}}{\lambda x : A.\, t =_\alpha \lambda x' : A.\, t'}}$$

Alpha equivalence of STLC terms

$$\frac{}{c^A =_\alpha c^A} \qquad \frac{}{x =_\alpha x} \qquad \frac{}{() =_\alpha ()} \qquad \frac{s =_\alpha s' \qquad t =_\alpha t'}{(s\,,t) =_\alpha (s'\,,t')} \qquad \frac{t =_\alpha t'}{\texttt{fst}\,t =_\alpha \texttt{fst}\,t'}$$

$$\frac{t =_\alpha t'}{\texttt{snd}\,t =_\alpha \texttt{snd}\,t'} \qquad \frac{s =_\alpha s' \qquad t =_\alpha t'}{s\,t =_\alpha s'\,t'}$$

$$\frac{(y\ x)\cdot t =_\alpha (y\ x')\cdot t' \qquad y \text{ does not occur in } \{x, x', t, t'\}}{\lambda x : A.\, t =_\alpha \lambda x' : A.\, t'}$$

result of replacing all occurrences of x with y in t

Alpha equivalence of STLC terms

$$\boxed{\frac{}{c^A =_\alpha c^A}} \quad \boxed{\frac{}{x =_\alpha x}} \quad \boxed{\frac{}{() =_\alpha ()}} \quad \boxed{\frac{s =_\alpha s' \qquad t =_\alpha t'}{(s\,,t) =_\alpha (s'\,,t')}} \quad \boxed{\frac{t =_\alpha t'}{\mathtt{fst}\,t =_\alpha \mathtt{fst}\,t'}}$$

$$\boxed{\frac{t =_\alpha t'}{\mathtt{snd}\,t =_\alpha \mathtt{snd}\,t'}} \quad \boxed{\frac{s =_\alpha s' \qquad t =_\alpha t'}{s\,t =_\alpha s'\,t'}}$$

$$\boxed{\frac{(y\ x)\cdot t =_\alpha (y\ x')\cdot t' \qquad y \text{ does not occur in } \{x, x', t, t'\}}{\lambda x : A.\, t =_\alpha \lambda x' : A.\, t'}}$$

E.g.

$\lambda x : A.\, x\, x =_\alpha \lambda y : A.\, y\, y \neq_\alpha \lambda x : A.\, x\, y$

$(\lambda y : A.\, y)\; x =_\alpha (\lambda x : A.\, x)\; x \neq_\alpha (\lambda x : A.\, x)\; y$

Substitution

$\boxed{t[s/x]}$ = result of replacing all free occurrences of variable x in term t (i.e. those not occurring within the scope of a $\lambda x : A.\_$ binder) by the term s, alpha-converting λ-bound variables in t to avoid them “capturing” any free variables of t.

E.g. $(\lambda y : A.\,(y\,,x))[y/x]$ is $\lambda z : A.\,(z\,,y)$ and is not $\lambda y : A.\,(y\,,y)$

Substitution

$\boxed{t[s/x]}$ = result of replacing all free occurrences of variable x in term t (i.e. those not occurring within the scope of a $\lambda x : A.\_$ binder) by the term s, alpha-converting λ-bound variables in t to avoid them "capturing" any free variables of t.

E.g. $(\lambda y : A.\,(y\,,x))[y/x]$ is $\lambda z : A.\,(z\,,y)$ and is not $\lambda y : A.\,(y\,,y)$

The relation $t[s/x] = t'$ can be inductively defined by the following rules…

Substitution

$$\boxed{\frac{}{c^A[s/x] = c^A}} \quad \boxed{\frac{}{x[s/x] = s}} \quad \boxed{\frac{y \neq x}{y[s/x] = y}} \quad \boxed{\frac{}{()[s/x] = ()}}$$

$$\boxed{\frac{t_1[s/x] = t_1' \qquad t_2[s/x] = t_2'}{(t_1\ ,\ t_2)[s/x] = (t_1'\ ,\ t_2')}} \quad \boxed{\frac{t[s/x] = t'}{(\texttt{fst}\ t)[s/x] = \texttt{fst}\ t'}}$$

$$\boxed{\frac{t[s/x] = t'}{(\texttt{snd}\ t)[s/x] = \texttt{snd}\ t'}} \quad \boxed{\frac{t_1[s/x] = t_1' \qquad t_2[s/x] = t_2'}{(t_1\ t_2)[s/x] = t_1' t_2'}}$$

$$\boxed{\frac{t[s/x] = t' \qquad y \neq x \text{ and } y \text{ does not occur in } s}{(\lambda y : A.\, t)[s/x] = \lambda y : A.\, t'}}$$

Semantics of substitution in a ccc

Substitution Lemma If $\Gamma \vdash s : A$ and $\Gamma, x : A \vdash t : B$ are provable, then so is $\Gamma \vdash t[s/x] : B$.

Substitution Theorem If $\Gamma \vdash s : A$ and $\Gamma, x : A \vdash t : B$ are provable, then in any ccc the following diagram commutes:

$$\begin{tikzcd}
M[\![\Gamma]\!] \arrow[r, "\langle \mathtt{id}, M[\![\Gamma \vdash s : A]\!]\rangle"] \arrow[dr, "M[\![\Gamma \vdash t[s/x] : B]\!]"'] & M[\![\Gamma]\!] \times M[\![A]\!] \arrow[d, "M[\![\Gamma, x : A \vdash t : B]\!]"] \\
 & M[\![B]\!]
\end{tikzcd}$$

Lecture 9

STLC equations

take the form $\boxed{\Gamma \vdash s = t : A}$ where $\Gamma \vdash s : A$ and $\Gamma \vdash t : A$ are provable.

Such an equation is satisfied by the semantics in a ccc if $M[\![\Gamma \vdash s : A]\!]$ and $M[\![\Gamma \vdash t : A]\!]$ are equal $\mathbf{C}$-morphisms $M[\![\Gamma]\!] \to M[\![A]\!]$.

Qu: which equations are always satisfied in any ccc?

Ans: $\beta\eta$-equivalence…

STLC $\beta\eta$-Equality

The relation $\boxed{\Gamma \vdash s =_{\beta\eta} t : A}$ (where Γ ranges over typing environments, s and t over terms and A over types) is inductively defined by the following rules:

STLC $\beta\eta$-Equality

The relation $\boxed{\Gamma \vdash s =_{\beta\eta} t : A}$ (where Γ ranges over typing environments, s and t over terms and A over types) is inductively defined by the following rules:

- β-conversions

$$\boxed{\frac{\Gamma, x : A \vdash t : B \qquad \Gamma \vdash s : A}{\Gamma \vdash (\lambda x : A.\, t)s =_{\beta\eta} t[s/x] : B}} \quad \boxed{\frac{\Gamma \vdash s : A \qquad \Gamma \vdash t : B}{\Gamma \vdash \mathtt{fst}(s\,,\,t) =_{\beta\eta} s : A}}$$

$$\boxed{\frac{\Gamma \vdash s : A \qquad \Gamma \vdash t : B}{\Gamma \vdash \mathtt{snd}(s\,,\,t) =_{\beta\eta} t : B}}$$

STLC $\beta\eta$-Equality

The relation $\boxed{\Gamma \vdash s =_{\beta\eta} t : A}$ (where Γ ranges over typing environments, s and t over terms and A over types) is inductively defined by the following rules:

- β-conversions
- η-conversions

$$\boxed{\dfrac{\Gamma \vdash t : A \to B \qquad x \text{ does not occur in } t}{\Gamma \vdash t =_{\beta\eta} (\lambda x : A.\, t\, x) : A \to B}}$$

$$\boxed{\dfrac{\Gamma \vdash t : A \times B}{\Gamma \vdash t =_{\beta\eta} (\mathtt{fst}\, t\,,\, \mathtt{snd}\, t) : A \times B}} \quad \boxed{\dfrac{\Gamma \vdash t : \mathtt{unit}}{\Gamma \vdash t =_{\beta\eta} () : \mathtt{unit}}}$$

STLC $\beta\eta$-Equality

The relation $\boxed{\Gamma \vdash s =_{\beta\eta} t : A}$ (where Γ ranges over typing environments, s and t over terms and A over types) is inductively defined by the following rules:

- β-conversions
- η-conversions
- congruence rules

$$\boxed{\frac{\Gamma, x : A \vdash t =_{\beta\eta} t' : B}{\Gamma \vdash \lambda x : A.\, t =_{\beta\eta} \lambda x : A.\, t' : A \to B}}$$

$$\boxed{\frac{\Gamma \vdash s =_{\beta\eta} s' : A \to B \qquad \Gamma \vdash t =_{\beta\eta} t' : A}{\Gamma \vdash s\,t =_{\beta\eta} s' t' : B}} \text{ etc}$$

STLC $\beta\eta$-Equality

The relation $\boxed{\Gamma \vdash s =_{\beta\eta} t : A}$ (where Γ ranges over typing environments, s and t over terms and A over types) is inductively defined by the following rules:

- β-conversions
- η-conversions
- congruence rules
- $=_{\beta\eta}$ is reflexive, symmetric and transitive

$$\boxed{\frac{\Gamma \vdash t : A}{\Gamma \vdash t =_{\beta\eta} t : A}} \quad \boxed{\frac{\Gamma \vdash s =_{\beta\eta} t : A}{\Gamma \vdash t =_{\beta\eta} s : A}}$$

$$\boxed{\frac{\Gamma \vdash r =_{\beta\eta} s : A \qquad \Gamma \vdash s =_{\beta\eta} t : A}{\Gamma \vdash r =_{\beta\eta} t : A}}$$

STLC $\beta\eta$-Equality

Soundness Theorem for semantics of STLC in a ccc.
If $\Gamma \vdash s =_{\beta\eta} t : A$ is provable, then in any ccc

$$M[\![\Gamma \vdash s : A]\!] = M[\![\Gamma \vdash t : A]\!]$$

are equal $\mathbf{C}$-morphisms $M[\![\Gamma]\!] \longrightarrow M[\![A]\!]$.

Proof is by induction on the structure of the proof of $\Gamma \vdash s =_{\beta\eta} t : A$.

Here we just check the case of β-conversion for functions.

So suppose we have $\Gamma, x : A \vdash t : B$ and $\Gamma \vdash s : A$. We have to see that

$$M[\![\Gamma \vdash (\lambda x : A.\, t)s : B]\!] = M[\![\Gamma \vdash t[s/x] : B]\!]$$

Suppose
$$\begin{aligned}
M[\![\Gamma]\!] &= X\\
M[\![A]\!] &= Y\\
M[\![B]\!] &= Z\\
M[\![\Gamma, x : A \vdash t : B]\!] &= f : X \times Y \longrightarrow Z\\
M[\![\Gamma \vdash s : A]\!] &= g : X \longrightarrow Z
\end{aligned}$$

Then
$$M[\![\Gamma \vdash \lambda x : A.\, t : A \to B]\!] = \mathtt{cur}\, f : X \longrightarrow Z^{Y}$$

and hence

$$\begin{aligned}
&M[\![\Gamma \vdash (\lambda x : A.\, t)s : B]\!]\\
&= \mathtt{app} \circ \langle \mathtt{cur}\, f\,, g \rangle\\
&= \mathtt{app} \circ (\mathtt{cur}\, f \times \mathtt{id}_{Y}) \circ \langle \mathtt{id}_{X}\,, g \rangle && \text{since } (a \times b) \circ \langle c\,, d \rangle = \langle a \circ c\,, b \circ d \rangle\\
&= f \circ \langle \mathtt{id}_{X}\,, g \rangle && \text{by definition of } \mathtt{cur}\, f\\
&= M[\![\Gamma \vdash t[s/x] : B]\!] && \text{by the Substitution Theorem}
\end{aligned}$$

as required.

The internal language of a ccc, $\mathbf{C}$

- one ground type for each $\mathbf{C}$-object X
- for each $X \in \mathbf{C}$, one constant f^X for each $\mathbf{C}$-morphism $f : 1 \to X$ ("global element" of the object X)

The types and terms of STLC over this language usefully describe constructions on the objects and morphisms of $\mathbf{C}$ using its cartesian closed structure, but in an "element-theoretic" way.

For example…

Example

In any ccc $\mathbf{C}$, for any $X, Y, Z \in \mathbf{C}$ there is an isomorphism

$$\boxed{Z^{(X \times Y)} \cong (Z^Y)^X}$$

Example

In any ccc $\mathbf{C}$, for any $X, Y, Z \in \mathbf{C}$ there is an isomorphism

$$\boxed{Z^{(X \times Y)} \cong (Z^Y)^X}$$

which in the internal language of $\mathbf{C}$ is described by the terms

$$\diamond \vdash s : ((X \times Y) \to Z) \to (X \to (Y \to Z))$$
$$\diamond \vdash t : (X \to (Y \to Z)) \to ((X \times Y) \to Z)$$

where $\begin{cases} s & \triangleq \lambda f : (X \times Y) \to Z .\, \lambda x : X .\, \lambda y : Y .\, f(x\,,y) \\ t & \triangleq \lambda g : X \to (Y \to Z) .\, \lambda z : X \times Y .\, g\,(\mathtt{fst}\,z)\,(\mathtt{snd}\,z) \end{cases}$ and

which satisfy $\begin{cases} \diamond, f : (X \times Y) \to Z \vdash t(s\,f) =_{\beta\eta} f \\ \diamond, g : X \to (Y \to Z) \vdash s(t\,g) =_{\beta\eta} g \end{cases}$

Free cartesian closed categories

The Soundness Theorem has a converse—completeness.

In fact for a given set of ground types and typed constants there is a single ccc $\boxed{\mathbf{F}}$ (the free ccc for that language) with an interpretation function M so that $\Gamma \vdash s =_{\beta\eta} t : A$ is provable iff $M[\![\Gamma \vdash s : A]\!] = M[\![\Gamma \vdash t : A]\!]$ in $\mathbf{F}$.

Free cartesian closed categories

The Soundness Theorem has a converse—completeness.

In fact for a given set of ground types and typed constants there is a single ccc $\boxed{\mathbf{F}}$ (the free ccc for that language) with an interpretation function M so that $\Gamma \vdash s =_{\beta\eta} t : A$ is provable iff $M[\![\Gamma \vdash s : A]\!] = M[\![\Gamma \vdash t : A]\!]$ in $\mathbf{F}$.

- $\mathbf{F}$-objects are the STLC types over the given set of ground types
- $\mathbf{F}$-morphisms $A \longrightarrow B$ are equivalence classes of STLC terms t satisfying $\diamond \vdash t : A \to B$ (so t is a *closed* term—it has no free variables) with respect to the equivalence relation equating s and t if $\diamond \vdash s =_{\beta\eta} t : A \to B$ is provable.
- identity morphism on A is the equivalence class of $\diamond \vdash \lambda x : A.\, x : A \to A$.
- composition of a morphism $A \longrightarrow B$ represented by $\diamond \vdash s : A \to B$ and a morphism $B \longrightarrow C$ represented by $\diamond \vdash t : B \to C$ is represented by $\diamond \vdash \lambda x : A.\, t(s\,x) : A \to C$.

Curry-Howard correspondence

Logic		**Type Theory**
propositions	↔	types
proofs	↔	terms

E.g. IPL *versus* STLC.

Curry-Howard for IPL *vs* STLC

Proof of $\diamond, \varphi \Rightarrow \psi, \psi \Rightarrow \theta \vdash \varphi \Rightarrow \theta$ in IPL

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\mathrm{AX})}{\Phi \vdash \quad \psi \Rightarrow \theta}\,(\mathrm{WK}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\mathrm{AX})}{\cdots}\,(\mathrm{WK})}{\Phi \vdash \quad \varphi \Rightarrow \psi}\,(\mathrm{WK}) \qquad \dfrac{}{\Phi \vdash \quad \varphi}\,(\mathrm{AX})}{\Phi \vdash \quad \psi}\,(\Rightarrow\mathrm{E})}{\Phi \vdash \quad \theta}\,(\Rightarrow\mathrm{E})}{\diamond, \quad \varphi \Rightarrow \psi, \quad \psi \Rightarrow \theta \vdash \quad \varphi \Rightarrow \theta}\,(\Rightarrow\mathrm{I})$$

where $\Phi = \diamond, \quad \varphi \Rightarrow \psi, \quad \psi \Rightarrow \theta, \quad \varphi$

Curry-Howard for IPL *vs* STLC

and a corresponding STLC term

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\Phi \vdash z : \psi \Rightarrow \theta}\,(\textsc{wk}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\cdots}\,(\textsc{ax})}{\cdots}\,(\textsc{wk})}{\Phi \vdash y : \varphi \Rightarrow \psi}\,(\textsc{wk}) \qquad \dfrac{}{\Phi \vdash x : \varphi}\,(\textsc{ax})}{\Phi \vdash y\,x : \psi}\,(\Rightarrow\textsc{e})}{\Phi \vdash z(y\,x) : \theta}\,(\Rightarrow\textsc{e})}{\diamond, y : \varphi \Rightarrow \psi, z : \psi \Rightarrow \theta \vdash \lambda x : \varphi.\, z(y\,x) : \varphi \Rightarrow \theta}\,(\Rightarrow\textsc{i})$$

where $\Phi = \diamond, y : \varphi \Rightarrow \psi, z : \psi \Rightarrow \theta, x : \varphi$

Curry-Howard-Lawvere/Lambek correspondence

Logic		**Type Theory**		**Category Theory**
propositions	↔	types	↔	objects
proofs	↔	terms	↔	morphisms

E.g. IPL *versus* STLC *versus* CCCs

Curry-Howard-Lawvere/Lambek correspondence

Logic		**Type Theory**		**Category Theory**
propositions	↔	types	↔	objects
proofs	↔	terms	↔	morphisms

E.g. IPL *versus* STLC *versus* CCCs

These correspondences can be made into category-theoretic equivalences—we first need to define the notions of functor and natural transformation in order to define the notion of equivalence of categories.

Lecture 10

Functors

are the appropriate notion of morphism between categories

Given categories $\mathbf{C}$ and $\mathbf{D}$, a functor $\boxed{F : \mathbf{C} \longrightarrow \mathbf{D}}$ is specified by:

- a function $\mathtt{obj}\,\mathbf{C} \longrightarrow \mathtt{obj}\,\mathbf{D}$ whose value at X is written $\boxed{F\,X}$
- for all $X, Y \in \mathbf{C}$, a function $\mathbf{C}(X, Y) \longrightarrow \mathbf{D}(F\,X, F\,Y)$ whose value at $f : X \longrightarrow Y$ is written
  $\boxed{F\,f : F\,X \longrightarrow F\,Y}$
  and which is required to preserve composition and identity morphisms:
  $$\begin{aligned} F(g \circ f) &= F\,g \circ F\,f \\ F(\mathtt{id}_X) &= \mathtt{id}_{F\,X} \end{aligned}$$

Examples of functors

"Forgetful" functors from categories of set-with-structure back to **Set**.

E.g. $\boxed{U : \mathbf{Mon} \longrightarrow \mathbf{Set}}$

$$\begin{cases} U(M, \cdot, e) & = M \\ U((M_1, \cdot_1, e_1) \xrightarrow{f} (M_2, \cdot_2, e_2)) & = M_1 \xrightarrow{f} M_2 \end{cases}$$

Examples of functors

"Forgetful" functors from categories of set-with-structure back to **Set**.

E.g. $\boxed{U : \mathbf{Mon} \to \mathbf{Set}}$

$$\begin{cases} U(M, \cdot, e) & = M \\ U((M_1, \cdot_1, e_1) \xrightarrow{f} (M_2, \cdot_2, e_2)) & = M_1 \xrightarrow{f} M_2 \end{cases}$$

Similarly $U : \mathbf{Preord} \to \mathbf{Set}$.

Examples of functors

Free monoid functor $\boxed{F : \mathbf{Set} \longrightarrow \mathbf{Mon}}$

Given $\Sigma \in \mathbf{Set}$,

$$F\,\Sigma = (\mathtt{List}\,\Sigma, @, \mathtt{nil}), \text{ the free monoid on } \Sigma$$

Examples of functors

Free monoid functor $\boxed{F : \mathbf{Set} \longrightarrow \mathbf{Mon}}$

Given $\Sigma \in \mathbf{Set}$,

$$F\,\Sigma = (\mathtt{List}\,\Sigma, @, \mathtt{nil}), \text{ the free monoid on } \Sigma$$

Given a function $f : \Sigma_1 \longrightarrow \Sigma_2$, we get a function $F\,f : \mathtt{List}\,\Sigma_1 \longrightarrow \mathtt{List}\,\Sigma_2$ by mapping f over finite lists:

$$F\,f\,[a_1, \ldots, a_n] = [f\,a_1, \ldots, f\,a_n]$$

This gives a monoid morphism $F\,\Sigma_1 \to F\,\Sigma_2$; and mapping over lists preserves composition ($F(g \circ f) = F\,g \circ F\,f$) and identities ($F\,\mathtt{id}_\Sigma = \mathtt{id}_{F\,\Sigma}$). So we do get a functor from **Set** to **Mon**.

Examples of functors

If $\mathbf{C}$ is a category with binary products and $X \in \mathbf{C}$, then the function $(\_) \times X : \mathtt{obj}\,\mathbf{C} \to \mathtt{obj}\,\mathbf{C}$ extends to a functor $\boxed{(\_) \times X : \mathbf{C} \to \mathbf{C}}$ mapping morphisms $f : Y \to Y'$ to

$$f \times \mathtt{id}_X : Y \times X \to Y' \times X$$

$$\left(\text{recall that } f \times g \text{ is the unique morphism with } \begin{cases} \pi_1 \circ (f \times g) & = f \circ \pi_1 \\ \pi_2 \circ (f \times g) & = g \circ \pi_2 \end{cases}\right)$$

since it is the case that

$$\begin{cases} \mathtt{id}_X \times \mathtt{id}_Y & = \mathtt{id}_{X \times Y} \\ (f' \circ f) \times \mathtt{id}_X & = (f' \times \mathtt{id}_X) \circ (f \times \mathtt{id}_X) \end{cases}$$

(see Exercise Sheet 2, question 1c).

Examples of functors

If $\mathbf{C}$ is a cartesian closed category and $X \in \mathbf{C}$, then the function $(\_)^X : \mathtt{obj}\,\mathbf{C} \to \mathtt{obj}\,\mathbf{C}$ extends to a functor $\boxed{(\_)^X : \mathbf{C} \to \mathbf{C}}$ mapping morphisms $f : Y \to Y'$ to

$$f^X \triangleq \mathtt{cur}(f \circ \mathtt{app}) : Y^X \to Y'^X$$

since it is the case that $\begin{cases} (\mathtt{id}_Y)^X & = \mathtt{id}_{Y^X} \\ (g \circ f)^X & = g^X \circ f^X \end{cases}$

(see Exercise Sheet 3, question 4).

Contravariance

Given categories $\mathbf{C}$ and $\mathbf{D}$, a functor $F : \mathbf{C}^{\mathrm{op}} \longrightarrow \mathbf{D}$ is called a contravariant functor from $\mathbf{C}$ to $\mathbf{D}$.

Note that if $X \xrightarrow{f} Y \xrightarrow{g} Z$ in $\mathbf{C}$, then $X \xleftarrow{f} Y \xleftarrow{g} Z$ in $\mathbf{C}^{\mathrm{op}}$

so $F\,X \xleftarrow{F\,f} F\,Y \xleftarrow{F\,g} F\,Z$ in $\mathbf{D}$ and hence

$$\boxed{F(g \circ_{\mathbf{C}} f) = F\,f \circ_{\mathbf{D}} F\,g}$$

(contravariant functors reverse the order of composition)

A functor $\mathbf{C} \longrightarrow \mathbf{D}$ is sometimes called a covariant functor from $\mathbf{C}$ to $\mathbf{D}$.

Example of a contravariant functor

If $\mathbf{C}$ is a cartesian closed category and $X \in \mathbf{C}$, then the function $X^{(\_)} : \mathtt{obj}\,\mathbf{C} \to \mathtt{obj}\,\mathbf{C}$ extends to a functor $\boxed{X^{(\_)} : \mathbf{C}^{\mathrm{op}} \to \mathbf{C}}$ mapping morphisms $f : Y \to Y'$ to

$$X^f \triangleq \mathtt{cur}(\mathtt{app} \circ (\mathtt{id}_{X^{Y'}} \times f)) : X^{Y'} \to X^Y$$

since it is the case that $\begin{cases} X^{\mathtt{id}_Y} & = \mathtt{id}_{X^Y} \\ X^{g \circ f} & = X^f \circ X^g \end{cases}$

(see Exercise Sheet 3, question 5).

Note that since a functor $F : \mathbf{C} \to \mathbf{D}$ preserves domains, codomains, composition and identity morphisms

it sends commutative diagrams in $\mathbf{C}$ to commutative diagrams in $\mathbf{D}$

E.g.

$$\begin{tikzcd}
 & X \arrow[dl, "f"'] \arrow[dd, "h"] & & & F\,X \arrow[dl, "F\,f"'] \arrow[dd, "F\,h{=}F(g\circ f){=}F\,g\circ F\,f"] \\
Y \arrow[dr, "g"'] & & \overset{F}{\longmapsto} & F\,Y \arrow[dr, "F\,g"'] & \\
 & Z & & & F\,Z
\end{tikzcd}$$

Note that since a functor $F : \mathbf{C} \to \mathbf{D}$ preserves domains, codomains, composition and identity morphisms

it sends isomorphisms in $\mathbf{C}$ to isomorphisms in $\mathbf{D}$, because

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[dr, "\mathtt{id}_X"'] & Y \arrow[d, "g"] \arrow[dr, "\mathtt{id}_Y"] & & \overset{F}{\longmapsto} & F\,X \arrow[r, "F\,f"] \arrow[dr, "\mathtt{id}_{F\,X}"'] & F\,Y \arrow[d, "F\,g"] \arrow[dr, "\mathtt{id}_{F\,Y}"] & \\
& X \arrow[r, "f"'] & Y & & & F\,X \arrow[r, "F\,f"'] & F\,Y
\end{tikzcd}$$

so $\boxed{F(f^{-1}) = (F\,f)^{-1}}$

Composing functors

Given functors $F : \mathbf{C} \longrightarrow \mathbf{D}$ and $G : \mathbf{D} \longrightarrow \mathbf{E}$, we get a functor $\boxed{G \circ F : \mathbf{C} \longrightarrow \mathbf{E}}$ with

$$G \circ F \left(\begin{tikzcd} X \arrow[d, "f"] \\ Y \end{tikzcd} \right) = \begin{tikzcd} G(F\,X) \arrow[d, "G(F\,f)"] \\ G(F\,Y) \end{tikzcd}$$

(this preserves composition and identity morphisms, because F and G do)

Identity functor

on a category $\mathbf{C}$ is $\boxed{\mathtt{id}_{\mathbf{C}} : \mathbf{C} \to \mathbf{C}}$ where

$$\mathtt{id}_{\mathbf{C}}\left(\begin{tikzcd} X \arrow[d, "f"] \\ Y \end{tikzcd}\right) = \begin{tikzcd} X \arrow[d, "f"] \\ Y \end{tikzcd}$$

Functor composition and identity functors satisfy

associativity $\quad H \circ (G \circ F) = (H \circ G) \circ F$

unity $\quad \mathtt{id}_{\mathbf{D}} \circ F = F = F \circ \mathtt{id}_{\mathbf{C}}$

So we can get categories whose objects are categories and whose morphisms are functors

but we have to be a bit careful about size…

Size

One of the axioms of set theory is

set membership is a well-founded relation, that is, there is no infinite sequence of sets $X_0, X_1, X_2, \ldots$ with

$$\cdots \in X_{n+1} \in X_n \in \cdots \in X_2 \in X_1 \in X_0$$

So in particular there is no set X with $X \in X$.

So we cannot form the "set of all sets" or the "category of all categories".

Size

One of the axioms of set theory is

set membership is a well-founded relation, that is, there is no infinite sequence of sets $X_0, X_1, X_2, \ldots$ with

$$\cdots \in X_{n+1} \in X_n \in \cdots \in X_2 \in X_1 \in X_0$$

So in particular there is no set X with $X \in X$.

So we cannot form the “set of all sets” or the “category of all categories”.

But we do assume there are (lots of) big sets

$$\mathscr{U}_0 \in \mathscr{U}_1 \in \mathscr{U}_2 \in \cdots$$

where “big” means each $\mathscr{U}_n$ is a Grothendieck universe…

Grothendieck universes

A Grothendieck universe $\mathscr{U}$ is a set of sets satisfying

- $X \in Y \in \mathscr{U} \Rightarrow X \in \mathscr{U}$
- $X, Y \in \mathscr{U} \Rightarrow \{X, Y\} \in \mathscr{U}$
- $X \in \mathscr{U} \Rightarrow \mathscr{P}\,X \triangleq \{Y \mid Y \subseteq X\} \in \mathscr{U}$
- $X \in U \wedge F \in \mathscr{U}^X \Rightarrow$ $\{y \mid \exists x \in X,\ y \in F\,x\} \in \mathscr{U}$

  (hence also $X, Y \in \mathscr{U} \ \Rightarrow\ X \times Y \in \mathscr{U} \ \wedge\ Y^X \in \mathscr{U}$)

The above properties are satisfied by $\mathscr{U} = \emptyset$, but we will always assume

- $\mathbb{N} \in \mathscr{U}$

Size

We assume

there is an infinite sequence $\mathscr{U}_0 \in \mathscr{U}_1 \in \mathscr{U}_2 \in \cdots$ of bigger and bigger Grothendieck universes

and revise the previous definition of "the" category of sets and functions:

$\mathbf{Set}_n$ = category whose objects are all the sets in $\mathscr{U}_n$ and with $\mathbf{Set}_n(X, Y) = Y^X =$ all functions from X to Y.

Notation: $\boxed{\mathbf{Set} \triangleq \mathbf{Set}_0}$ — its objects are called small sets (and other sets we call large).

Size

Set is the category of small sets.

Definition. A category $\mathbf{C}$ is locally small if for all $X, Y \in \mathbf{C}$, the set of $\mathbf{C}$-morphisms $X \to Y$ is small, that is, $\mathbf{C}(X, Y) \in \mathbf{Set}$.

$\mathbf{C}$ is a small category if it is both locally small and $\mathtt{obj}\, \mathbf{C} \in \mathbf{Set}$.

E.g. **Set**, **Preord** and **Mon** are all locally small (but not small).

Given $P \in \mathbf{Preord}$, the category $\mathbf{C}_P$ it determines is small; similarly, the category $\mathbf{C}_M$ determined by $M \in \mathbf{Mon}$ is small.

The category of small categories, **Cat**

- objects are all small categories
- morphisms in $\mathbf{Cat}(\mathbf{C}, \mathbf{D})$ are all functors $\mathbf{C} \to \mathbf{D}$
- composition and identity morphisms as for functors

Cat is a locally small category

Lecture 11

The category of small categories

Recall definition of $\mathbf{Cat}$:

- objects are all small categories
- morphisms in $\mathbf{Cat}(\mathbf{C}, \mathbf{D})$ are all functors $\mathbf{C} \to \mathbf{D}$
- composition and identity morphisms as for functors

Cat has a terminal object

The category

$$\begin{tikzcd} 0 \arrow[loop right, "\mathtt{id}_0"] \end{tikzcd}$$

one object, one morphism

is terminal in **Cat**

$\mathbf{Cat}$ has binary products

Given small categories $\mathbf{C}, \mathbf{D} \in \mathbf{Cat}$, their product $\mathbf{C} \xleftarrow{\pi_1} \mathbf{C} \times \mathbf{D} \xrightarrow{\pi_2} \mathbf{D}$ is:

Cat has binary products

Given small categories $\mathbf{C}, \mathbf{D} \in \mathbf{Cat}$, their product $\mathbf{C} \xleftarrow{\pi_1} \mathbf{C} \times \mathbf{D} \xrightarrow{\pi_2} \mathbf{D}$ is:

- objects of $\mathbf{C} \times \mathbf{D}$ are pairs (X, Y) where $X \in \mathbf{C}$ and $Y \in \mathbf{D}$
- morphisms $(X, Y) \to (X', Y')$ in $\mathbf{C} \times \mathbf{D}$ are pairs (f, g) where $f \in \mathbf{C}(X, X')$ and $g \in \mathbf{D}(Y, Y')$
- composition and identity morphisms are given by those of $\mathbf{C}$ (in the first component) and $\mathbf{D}$ (in the second component)
- $$\begin{cases} \pi_1\left((X, Y) \xrightarrow{(f,g)} (X', Y')\right) = X \xrightarrow{f} X' \\ \pi_2\left((X, Y) \xrightarrow{(f,g)} (X', Y')\right) = Y \xrightarrow{g} Y' \end{cases}$$

Cat not only has finite products, it is also cartesian closed.

Exponentials in **Cat** are called functor categories.

To define them we need to consider natural transformations, which are the appropriate notion of morphism between functors.

Natural transformations

Motivating example: fix a set $S \in \mathbf{Set}$ and consider the two functors $F, G : \mathbf{Set} \longrightarrow \mathbf{Set}$ given by

$$F\left(X \xrightarrow{f} Y\right) = S \times X \xrightarrow{\mathtt{id}_S \times f} S \times Y$$

$$G\left(X \xrightarrow{f} Y\right) = X \times S \xrightarrow{f \times \mathtt{id}_S} Y \times S$$

Natural transformations

Motivating example: fix a set $S \in \mathbf{Set}$ and consider the two functors $F, G : \mathbf{Set} \to \mathbf{Set}$ given by

$$F\left(X \xrightarrow{f} Y\right) = S \times X \xrightarrow{\mathtt{id}_S \times f} S \times Y$$

$$G\left(X \xrightarrow{f} Y\right) = X \times S \xrightarrow{f \times \mathtt{id}_S} Y \times S$$

For each $X \in \mathbf{Set}$ there is an isomorphism (bijection) $\theta_X : F\,X \cong G\,X$ in **Set** given by $\langle \pi_2\,, \pi_1 \rangle : S \times X \to X \times S$.

These isomorphisms do not depend on the particular nature of each set X (they are "polymorphic in X"). One way to make this precise is…

…if we change from X to Y along a function $f : X \to Y$, then we get a commutative diagram in **Set**:

$$\begin{tikzcd}
S \times X \arrow[r, "\langle \pi_2, \pi_1 \rangle"] \arrow[d, "\mathtt{id} \times f"'] & X \times S \arrow[d, "f \times \mathtt{id}"] \\
S \times Y \arrow[r, "\langle \pi_2, \pi_1 \rangle"'] & Y \times S
\end{tikzcd}$$

The square commutes because for all $s \in S$ and $x \in X$

$$\begin{aligned}
\langle \pi_2 \, , \, \pi_1 \rangle((\mathtt{id} \times f)(s, x)) &= \langle \pi_2 \, , \, \pi_1 \rangle(s, f\, x) \\
&= (f\, x, s) \\
&= (f \times \mathtt{id})(x, s) \\
&= (f \times \mathtt{id})(\langle \pi_2 \, , \, \pi_1 \rangle(s, x))
\end{aligned}$$

…if we change from X to Y along a function $f : X \to Y$, then we get a commutative diagram in **Set**:

$$\begin{tikzcd}
F\,X \arrow[r, "\theta_X"] \arrow[d, "F\,f"'] & G\,X \arrow[d, "G\,f"] \\
F\,Y \arrow[r, "\theta_Y"'] & G\,Y
\end{tikzcd}$$

We say that the family $(\theta_X \mid X \in \mathbf{Set})$ is natural in X.

Natural transformations

Definition. Given categories and functors $F, G : \mathbf{C} \longrightarrow \mathbf{D}$, a natural transformation $\boxed{\theta : F \longrightarrow G}$ is a family of $\mathbf{D}$-morphisms $\theta_X \in \mathbf{D}(F\,X, G\,X)$, one for each $X \in \mathbf{C}$, such that for all $\mathbf{C}$-morphisms $f : X \longrightarrow Y$, the diagram

$$\begin{tikzcd}
F\,X \arrow[r, "\theta_X"] \arrow[d, "F\,f"'] & G\,X \arrow[d, "G\,f"] \\
F\,Y \arrow[r, "\theta_Y"'] & G\,Y
\end{tikzcd}$$

commutes in $\mathbf{D}$, that is, $\theta_Y \circ F\,f = G\,f \circ \theta_X$.

Example

Recall forgetful (U) and free (F) functors:

$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend right, "F"'] & \mathbf{Mon} \arrow[l, bend right, "U"'] \end{tikzcd}$$

There is a natural transformation $\eta : \mathtt{id}_{\mathbf{Set}} \longrightarrow U \circ F$, where for each $\Sigma \in \mathbf{Set}$

$$\eta_\Sigma : \Sigma \longrightarrow U(F\,\Sigma) = \mathtt{List}\,\Sigma$$

$$a \in \Sigma \longmapsto [a] \in \mathtt{List}\,\Sigma \text{ (one-element list)}$$

(Easy to see that
$$\begin{tikzcd} \Sigma \arrow[r, "\eta_\Sigma"] \arrow[d, "f"'] & U(F\,\Sigma) \arrow[d, "U(F\,f)"] \\ \Sigma' \arrow[r, "\eta_{\Sigma'}"'] & U(F\,\Sigma') \end{tikzcd}$$
commutes.)

Example

The covariant powerset functor $\mathscr{P} : \mathbf{Set} \to \mathbf{Set}$ is

$$\mathscr{P}\,X \triangleq \{S \mid S \subseteq X\}$$

$$\mathscr{P}\left(X \xrightarrow{f} Y\right) \triangleq \mathscr{P}\,X \xrightarrow{\mathscr{P}\,f} \mathscr{P}\,Y$$

$$S \mapsto \mathscr{P}\,f\,S \triangleq \{f\,x \mid x \in S\}$$

Example

The covariant powerset functor $\mathscr{P} : \mathbf{Set} \to \mathbf{Set}$ is

$$\mathscr{P}\,X \triangleq \{S \mid S \subseteq X\}$$

$$\mathscr{P}\left(X \xrightarrow{f} Y\right) \triangleq \mathscr{P}\,X \xrightarrow{\mathscr{P}\,f} \mathscr{P}\,Y$$

$$S \mapsto \mathscr{P}\,f\,S \triangleq \{f\,x \mid x \in S\}$$

There is a natural transformation $\cup : \mathscr{P} \circ \mathscr{P} \to \mathscr{P}$ whose component at $X \in \mathbf{Set}$ sends $\mathcal{S} \in \mathscr{P}(\mathscr{P}\,X)$ to

$$\cup_X \mathcal{S} \triangleq \{x \in X \mid \exists S \in \mathcal{S},\ x \in S\} \in \mathscr{P}\,X$$

(check that $\cup_X$ is natural in X)

Non-example

The classic example of an "un-natural transformation" (the one that caused Eilenburg and MacLane to invent the concept of naturality) is the linear isomorphism between a finite dimensional real vectorspace V and its dual V^* (= vectorspace of linear functions $V \to \mathbb{R}$).

Both V and V^* have the same finite dimension, so are isomorphic by choosing bases; but there is no choice of basis for each V that makes the family of isomorphisms natural in V.

For a similar, more elementary non-example, see Ex. Sh. 5, question 4.

Composing natural transformations

Given functors $F, G, H : \mathbf{C} \to \mathbf{D}$ and natural transformations $\theta : F \to G$ and $\varphi : G \to H$,

we get $\boxed{\varphi \circ \theta} : F \to H$ with

$$(\varphi \circ \theta)_X = \left(F\,X \xrightarrow{\theta_X} G\,X \xrightarrow{\varphi_X} H\,X \right)$$

Composing natural transformations

Given functors $F, G, H : \mathbf{C} \to \mathbf{D}$ and natural transformations $\theta : F \to G$ and $\varphi : G \to H$,

we get $\boxed{\varphi \circ \theta} : F \to H$ with

$$(\varphi \circ \theta)_X = \left(F\,X \xrightarrow{\theta_X} G\,X \xrightarrow{\varphi_X} H\,X \right)$$

Check naturality:

$$\begin{aligned}
H\,f \circ (\varphi \circ \theta)_X &\triangleq H\,f \circ \varphi_X \circ \theta_X \\
&= \varphi_Y \circ G\,f \circ \theta_X && \text{naturality of } \varphi \\
&= \varphi_Y \circ \theta_Y \circ F\,f && \text{naturality of } \theta \\
&\triangleq (\varphi \circ \theta)_Y \circ F\,f
\end{aligned}$$

Identity natural transformation

Given a functor $F : \mathbf{C} \longrightarrow \mathbf{D}$, we get a natural transformation $\boxed{\mathtt{id}_F : F \longrightarrow F}$ with

$$(\mathtt{id}_F)_X = F\,X \xrightarrow{\mathtt{id}_{F\,X}} F\,X$$

Identity natural transformation

Given a functor $F : \mathbf{C} \longrightarrow \mathbf{D}$, we get a natural transformation $\boxed{\mathtt{id}_F : F \longrightarrow F}$ with

$$(\mathtt{id}_F)_X = F\,X \xrightarrow{\mathtt{id}_{F\,X}} F\,X$$

Check naturality:

$$F\,f \circ (\mathtt{id}_F)_X \triangleq F\,f \circ \mathtt{id}_{F\,X} = F\,f = \mathtt{id}_{F\,Y} \circ F\,f \triangleq (\mathtt{id}_F)_Y \circ F\,f$$

Functor categories

It is easy to see that composition and identities for natural transformations satisfy

$$(\psi \circ \varphi) \circ \theta = \psi \circ (\varphi \circ \theta)$$
$$\mathtt{id}_G \circ \theta = \theta \circ \mathtt{id}_F$$

so that we get a category:

Definition. Given categories $\mathbf{C}$ and $\mathbf{D}$, the functor category $\boxed{\mathbf{D}^{\mathbf{C}}}$ has

- objects are all functors $\mathbf{C} \longrightarrow \mathbf{D}$
- given $F, G : \mathbf{C} \longrightarrow \mathbf{D}$, morphism from F to G in $\mathbf{D}^{\mathbf{C}}$ are the natural transformations $F \longrightarrow G$
- composition and identity morphisms as above

If $\mathscr{U}$ is a Grothendieck universe, then for each $X \in \mathscr{U}$ and $F \in \mathscr{U}^X$ we have that their dependent product and dependent function sets

$$\textstyle\sum_{x\in X} F\,x \triangleq \{(x,y) \mid x \in X \wedge y \in F\,x\}$$
$$\textstyle\prod_{x\in X} F\,x \triangleq \{f \subseteq \displaystyle\sum_{x\in X} F\,x \mid f \text{ is single-valued and total}\}$$

are also in $\mathscr{U}$; and as a special case (of $\prod$, when F is a constant function with value Y) we also have that $X, Y \in \mathscr{U}$ implies $Y^X \in \mathscr{U}$.

If $\mathscr{U}$ is a Grothendieck universe, then for each $X \in \mathscr{U}$ and $F \in \mathscr{U}^X$ we have that their dependent product and dependent function sets

$$\textstyle\sum_{x\in X} F\,x \triangleq \{(x, y) \mid x \in X \wedge y \in F\,x\}$$
$$\textstyle\prod_{x\in X} F\,x \triangleq \{f \subseteq \displaystyle\sum_{x\in X} F\,x \mid f \text{ is single-valued and total}\}$$

are also in $\mathscr{U}$; and as a special case (of $\prod$, when F is a constant function with value Y) we also have that $X, Y \in \mathscr{U}$ implies $Y^X \in \mathscr{U}$. Hence

If $\mathbf{C}$ and $\mathbf{D}$ are small categories, then so is $\mathbf{D}^{\mathbf{C}}$.

because

$$\textstyle\mathtt{obj}(\mathbf{D}^{\mathbf{C}}) \subseteq \sum_{F\in(\mathtt{obj}\,D)^{\mathtt{obj}\,\mathbf{C}}} \prod_{X,Y\in\mathtt{obj}\,\mathbf{C}} \mathbf{D}(F\,X, F\,Y)^{\mathbf{C}(X,Y)}$$
$$\textstyle\mathbf{D}^{\mathbf{C}}(F, G) \subseteq \prod_{X\in\mathtt{obj}\,\mathbf{C}} \mathbf{D}(F\,X, G\,X)$$

If $\mathscr{U}$ is a Grothendieck universe, then for each $X \in \mathscr{U}$ and $F \in \mathscr{U}^X$ we have that their dependent product and dependent function sets

$$\begin{aligned}
&\textstyle\sum_{x\in X} F\,x \triangleq \{(x,y) \mid x \in X \wedge y \in F\,x\}\\
&\textstyle\prod_{x\in X} F\,x \triangleq \{f \subseteq \displaystyle\sum_{x\in X} F\,x \mid f \text{ is single-valued and total}\}
\end{aligned}$$

are also in $\mathscr{U}$; and as a special case (of $\prod$, when F is a constant function with value Y) we also have that $X, Y \in \mathscr{U}$ implies $Y^X \in \mathscr{U}$. Hence

If $\mathbf{C}$ and $\mathbf{D}$ are small categories, then so is $\mathbf{D}^{\mathbf{C}}$.

because

$$\begin{aligned}
&\mathtt{obj}(\mathbf{D}^{\mathbf{C}}) \subseteq \textstyle\sum_{F\in(\mathtt{obj}\,D)^{\mathtt{obj}\,\mathbf{C}}} \prod_{X,Y\in\mathtt{obj}\,\mathbf{C}} \mathbf{D}(F\,X, F\,Y)^{\mathbf{C}(X,Y)}\\
&\mathbf{D}^{\mathbf{C}}(F,G) \subseteq \textstyle\prod_{X\in\mathtt{obj}\,\mathbf{C}} \mathbf{D}(F\,X, G\,X)
\end{aligned}$$

Aim to show that functor category $\mathbf{D}^{\mathbf{C}}$ is the exponential of $\mathbf{C}$ and $\mathbf{D}$ in $\mathbf{Cat}$…

$\mathbf{Cat}$ is cartesian closed

Theorem. There is an application functor

$$\mathtt{app} : \mathbf{D}^{\mathbf{C}} \times \mathbf{C} \to \mathbf{D}$$

that makes $\mathbf{D}^{\mathbf{C}}$ the exponential for $\mathbf{C}$ and $\mathbf{D}$ in $\mathbf{Cat}$.

Given $(F, X) \in \mathbf{D}^{\mathbf{C}} \times \mathbf{C}$, we define

$$\mathtt{app}(F, X) \triangleq F\,X$$

and given $(\theta, f) : (F, X) \to (G, Y)$ in $\mathbf{D}^{\mathbf{C}} \times \mathbf{C}$, we define

$$\begin{aligned}\mathtt{app}\left((F, X) \xrightarrow{(\theta, f)} (G, Y)\right) &\triangleq F\,X \xrightarrow{F\,f} F\,Y \xrightarrow{\theta_Y} G\,Y \\ &= F\,X \xrightarrow{\theta_X} G\,X \xrightarrow{G\,f} G\,Y\end{aligned}$$

Check: $\begin{cases}\mathtt{app}(\mathtt{id}_F, \mathtt{id}_X) & = \mathtt{id}_{F\,X} \\ \mathtt{app}(\varphi \circ \theta, g \circ f) & = \mathtt{app}(\varphi, g) \circ \mathtt{app}(\theta, f)\end{cases}$

Cat is cartesian closed

Theorem. There is an application functor

$$\mathtt{app} : \mathbf{D}^{\mathbf{C}} \times \mathbf{C} \to \mathbf{D}$$

that makes $\mathbf{D}^{\mathbf{C}}$ the exponential for $\mathbf{C}$ and $\mathbf{D}$ in **Cat**.

Definition of currying: given functor $F : \mathbf{E} \times \mathbf{C} \to \mathbf{D}$, we get a functor $\mathtt{cur}\, F : \mathbf{E} \to \mathbf{D}^{\mathbf{C}}$ as follows. For each $Z \in \mathbf{E}$, $\mathtt{cur}\, F\, Z \in \mathbf{D}^{\mathbf{C}}$ is the functor

$$\mathtt{cur}\, F\, Z \left(\begin{tikzcd} X \arrow[d, "f"] \\ X' \end{tikzcd} \right) \triangleq \begin{tikzcd} F(Z, X) \arrow[d, "F(\mathtt{id}_Z, f)"] \\ F(Z, X') \end{tikzcd}$$

For each $g : Z \to Z'$ in $\mathbf{E}$, $\mathtt{cur}\, F\, g : \mathtt{cur}\, F\, Z \to \mathtt{cur}\, F\, Z'$ is the natural transformation whose component at each $X \in \mathbf{C}$ is

$$(\mathtt{cur}\, F\, g)_X \triangleq F(g, \mathtt{id}_X) : F(Z, X) \to F(Z', X)$$

(Check that this is natural in X; and that $\mathtt{cur}\, F$ preserves composition and identities in $\mathbf{E}$.)

Cat is cartesian closed

Theorem. There is an application functor

$$\mathtt{app} : \mathbf{D}^{\mathbf{C}} \times \mathbf{C} \longrightarrow \mathbf{D}$$

that makes $\mathbf{D}^{\mathbf{C}}$ the exponential for $\mathbf{C}$ and $\mathbf{D}$ in **Cat**.

Have to check that $\mathtt{cur}\, F$ is the unique functor $G : \mathbf{E} \longrightarrow \mathbf{D}^{\mathbf{C}}$ that makes

$$\begin{tikzcd}
\mathbf{E} \times \mathbf{C} \arrow[rr, "F"] \arrow[d, "G \times \mathtt{id}_{\mathbf{C}}"'] & & \mathbf{D} \\
\mathbf{D}^{\mathbf{C}} \times \mathbf{C} \arrow[urr, "\mathtt{app}"'] & &
\end{tikzcd}$$

commute in **Cat** (exercise).

Lecture 12

Adjoint functors

The concepts of “category”, “functor” and “natural transformation” were invented by Eilenberg and MacLane in order to formalise “adjoint situations”.

They appear everywhere in mathematics, logic and (hence) computer science.

Examples of adjoint situations that we have already seen…

Free monoids

$$\begin{array}{c} \Sigma \longrightarrow U(M, \cdot, e) \quad \text{morphisms in } \textbf{Set} \\ \hline\hline F\,\Sigma \longrightarrow (M, \cdot, e) \quad \text{morphisms in } \textbf{Mon} \end{array}$$

bijection

$$\textbf{Set}(\Sigma, U(M, \cdot, e)) \cong \textbf{Mon}(F\,\Sigma, (M, \cdot, e))$$

$$f \mapsto \hat{f}$$

$$g \circ \eta_\Sigma \;\leftarrow\!\!\dashv\; g$$

(where $\eta_\Sigma : \Sigma \longrightarrow F\,\Sigma = \texttt{List}\,\Sigma$ is $a \mapsto [a]$)

The bijection is “natural in Σ and $(M, \cdot, e)$” (to be explained)

Binary product in a category $\mathbf{C}$

$$\begin{array}{c} (Z,Z) \longrightarrow (X,Y) \quad \text{morphisms in } \mathbf{C}\times\mathbf{C} \\ \hline\hline Z \longrightarrow X \times Y \quad \text{morphisms in } \mathbf{C} \end{array}$$

bijection

$$(\mathbf{C}\times\mathbf{C})((Z,Z),(X,Y)) \cong \mathbf{C}(Z, X\times Y)$$

$$(f,g) \mapsto \langle f\,,g\rangle$$

$$(\pi_1 \circ h, \pi_2 \circ h) \leftarrow\!\shortmid h$$

This bijection is "natural in X, Y, Z" (to be explained)

Exponentials in a category $\mathbf{C}$ with binary products

$$\begin{array}{c} Z \times X \longrightarrow Y \ \text{morphisms in } \mathbf{C} \\ \hline\hline Z \longrightarrow Y^X \ \text{morphisms in } \mathbf{C} \end{array}$$

bijection

$$\mathbf{C}(Z \times X, Y) \cong \mathbf{C}(Z, Y^X)$$

$$f \mapsto \mathtt{cur}\, f$$

$$\mathtt{app} \circ (g \times \mathtt{id}_X) \leftarrow g$$

The bijection is "natural in X, Y, Z" (to be explained)

Adjunction

Definition. An adjunction between two categories $\mathbf{C}$ and $\mathbf{D}$ is specified by:

- functors $$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$$
- for each $X \in \mathbf{C}$ and $Y \in \mathbf{D}$ a bijection $\theta_{X,Y} : \mathbf{D}(F\,X, Y) \cong \mathbf{C}(X, G\,Y)$ which is natural in X and Y.

for all $\begin{cases} u : X' \longrightarrow X \text{ in } \mathbf{C} \\ v : Y \longrightarrow Y' \text{ in } \mathbf{D} \end{cases}$ and all $g : F\,X \longrightarrow Y$ in $\mathbf{D}$

$$X' \xrightarrow{u} X \xrightarrow{\theta_{X,Y}(g)} G\,Y \xrightarrow{G\,v} G\,Y' = \theta_{X',Y'}\left(F\,X' \xrightarrow{F\,u} F\,X \xrightarrow{g} Y \xrightarrow{v} Y'\right)$$

Adjunction

Definition. An adjunction between two categories $\mathbf{C}$ and $\mathbf{D}$ is specified by:

- functors $$\begin{tikzcd}\mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"]\end{tikzcd}$$
- for each $X \in \mathbf{C}$ and $Y \in \mathbf{D}$ a bijection
  $\theta_{X,Y} : \mathbf{D}(F\,X, Y) \cong \mathbf{C}(X, G\,Y)$
  which is natural in X and Y.

what has this to do with the concept of natural transformation between functors?

Hom functors

If $\mathbf{C}$ is a locally small category, then we get a functor

$$\boxed{\mathtt{Hom}_{\mathbf{C}} : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \longrightarrow \mathbf{Set}}$$

with $\mathtt{Hom}_{C}(X, Y) \triangleq \mathbf{C}(X, Y)$ and

$$\mathtt{Hom}_{\mathbf{C}}\left((X, Y) \xrightarrow{(f, g)} (X', Y') \right) \triangleq \mathbf{C}(X, Y) \xrightarrow{\mathtt{Hom}_{C}(f, g)} \mathbf{C}(X', Y')$$

$$\mathtt{Hom}_{C}(f, g)\, h \triangleq g \circ h \circ f$$

Hom functors

If **C** is a locally small category, then we get a functor

$$\boxed{\mathtt{Hom}_{\mathbf{C}} : \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \longrightarrow \mathbf{Set}}$$

with $\mathtt{Hom}_C(X, Y) \triangleq \mathbf{C}(X, Y)$ and

$$\mathtt{Hom}_{\mathbf{C}}\left((X, Y) \xrightarrow{(f, g)} (X', Y')\right) \triangleq \mathbf{C}(X, Y) \xrightarrow{\mathtt{Hom}_C(f,g)} \mathbf{C}(X', Y')$$

$$\mathtt{Hom}_C(f, g)\, h \triangleq g \circ h \circ f$$

If $(f, g) : (X, Y) \to (X', Y')$ in $\mathbf{C}^{\mathrm{op}} \times \mathbf{C}$ and $h : X \to Y$ in **C**,
then in **C** we have $f : X' \to X$, $g : Y \to Y'$ and so $g \circ h \circ f : X' \to Y'$

Natural isomorphisms

Given functors $F, G : \mathbf{C} \to \mathbf{D}$, a natural isomorphism $\theta : F \cong G$ is simply an isomorphism between F and G in the functor category $\mathbf{D}^{\mathbf{C}}$.

Natural isomorphisms

Given functors $F, G : \mathbf{C} \to \mathbf{D}$, a natural isomorphism $\theta : F \cong G$ is simply an isomorphism between F and G in the functor category $\mathbf{D}^{\mathbf{C}}$.

Lemma. If $\theta : F \to G$ is a natural transformation and for each $X \in \mathbf{C}$, $\theta_X : F\,X \to G\,X$ is an isomorphism in $\mathbf{D}$, then the family of morphisms $(\theta_X^{-1} : G\,X \to F\,X \mid X \in \mathbf{C})$ gives a natural transformation $\theta^{-1} : G \to F$ which is inverse to θ in $\mathbf{D}^{\mathbf{C}}$ and hence θ is a natural isomorphism. □

An adjunction between locally small categories $\mathbf{C}$ and $\mathbf{D}$ is simply a triple (F, G, θ) where

- $$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$$
- θ is a natural isomorphism between the functors

$$\begin{tikzcd}
 & \mathbf{D}^{\mathtt{op}} \times \mathbf{D} \arrow[dr, "\mathtt{Hom}_{\mathbf{D}}"] & \\
\mathbf{C}^{\mathtt{op}} \times \mathbf{D} \arrow[ur, "F^{\mathtt{op}} \times \mathtt{id}_{\mathbf{D}}"] \arrow[dr, "\mathtt{id}_{\mathbf{C}^{\mathtt{op}}} \times G"'] & \text{and} & \mathbf{Set} \\
 & \mathbf{C}^{\mathtt{op}} \times \mathbf{C} \arrow[ur, "\mathtt{Hom}_{\mathbf{C}}"'] &
\end{tikzcd}$$

Terminology:

Given $$\begin{tikzcd}\mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"]\end{tikzcd}$$

is there is some natural isomorphism
$\theta : \mathtt{Hom}_{\mathbf{D}} \circ (F^{\mathtt{op}} \times \mathtt{id}_{\mathbf{D}}) \cong \mathtt{Hom}_{\mathbf{C}} \circ (\mathtt{id}_{\mathbf{C}^{\mathtt{op}}} \times G)$

one says

F is a left adjoint for G

G is a right adjoint for F

and writes

$$\boxed{F \dashv G}$$

Notation associated with an adjunction(F, G, θ)

Given $\begin{cases} g : F\,X \longrightarrow Y \\ f : X \longrightarrow G\,Y \end{cases}$

we write $\begin{cases} \overline{g} & \triangleq \theta_{X,Y}(g) : X \longrightarrow G\,Y \\ \overline{f} & \triangleq \theta^{-1}_{X,Y}(f) : F\,X \longrightarrow Y \end{cases}$

Thus $\overline{\overline{g}} = g$, $\overline{\overline{f}} = f$ and naturality of $\theta_{X,Y}$ in X and Y means that

$$\overline{v \circ g \circ F\,u} = G\,v \circ \overline{g} \circ u$$

Notation associated with an adjunction (F, G, θ)

The existence of θ is sometimes indicated by writing

$$\begin{array}{c} F\,X \xrightarrow{g} Y \\ \hline\hline X \xrightarrow{\bar{g}} G\,Y \end{array}$$

Using this notation, one can split the naturality condition for θ into two:

$$\begin{array}{c} F\,X' \xrightarrow{F\,u} F\,X \xrightarrow{g} Y \\ \hline\hline X' \xrightarrow{u} X \xrightarrow{\bar{g}} G\,Y \end{array} \qquad \begin{array}{c} F\,X \xrightarrow{g} Y \xrightarrow{v} Y' \\ \hline\hline X \xrightarrow{\bar{g}} G\,Y \xrightarrow{G\,v} G\,Y' \end{array}$$

Lecture 13

Recall:

Given categories and functors $\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$,

an adjunction $\boxed{F \dashv G}$ is specified by functions

$$\theta_{X,Y}\downarrow \begin{array}{c} F\,X \xrightarrow{g} Y \\ \hline\hline X \xrightarrow{\overline{g}} G\,Y \end{array} \qquad \uparrow\theta^{-1}_{X,Y} \begin{array}{c} F\,X \xrightarrow{\overline{f}} Y \\ \hline\hline X \xrightarrow{f} G\,Y \end{array}$$

(for each $X \in \mathbf{C}$ and $Y \in \mathbf{D}$) satisfying $\overline{\overline{f}} = f$, $\overline{\overline{g}} = g$ and

$$\begin{array}{c} F\,X' \xrightarrow{F\,u} F\,X \xrightarrow{g} Y \\ \hline\hline X' \xrightarrow{u} X \xrightarrow{\overline{g}} G\,Y \end{array} \qquad \begin{array}{c} F\,X \xrightarrow{g} Y \xrightarrow{v} Y' \\ \hline\hline X \xrightarrow{\overline{g}} G\,Y \xrightarrow{G\,v} G\,Y' \end{array}$$

Theorem. A category $\mathbf{C}$ has binary products iff the diagonal functor $\Delta = \langle \mathtt{id}_{\mathbf{C}}, \mathtt{id}_{C} \rangle : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ has a right adjoint.

Theorem. A category $\mathbf{C}$ with binary products also has all exponentials of pairs of objects iff for all $X \in \mathbf{C}$, the functor $(\_) \times X : \mathbf{C} \to \mathbf{C}$ has a right adjoint.

Common situation: we are given a functor $F : \mathbf{C} \to \mathbf{D}$ and want to know whether it has a right adjoint $G : \mathbf{D} \to \mathbf{C}$ (and dually for left adjoints).

Q: what is the least info we need to specify the existence of a right adjoint?

Theorem. A category $\mathbf{C}$ has binary products iff the diagonal functor $\Delta = \langle \mathtt{id}_{\mathbf{C}}\,, \mathtt{id}_{C} \rangle : \mathbf{C} \to \mathbf{C} \times \mathbf{C}$ has a right adjoint.

Theorem. A category $\mathbf{C}$ with binary products also has all exponentials of pairs of objects iff for all $X \in \mathbf{C}$, the functor $(\_) \times X : \mathbf{C} \to \mathbf{C}$ has a right adjoint.

Common situation: we are given a functor $F : \mathbf{C} \to \mathbf{D}$ and want to know whether it has a right adjoint $G : \mathbf{D} \to \mathbf{C}$ (and dually for left adjoints).

Q: what is the least info we need to specify the existence of a right adjoint?

Both the above theorems are instances of the following theorem, which is a very useful characterisation of when a functor has a right adjoint (or dually, a left adjoint).

Characterisation of right adjoints

Theorem. A functor $F : \mathbf{C} \longrightarrow \mathbf{D}$ has a right adjoint iff for all $\mathbf{D}$-objects $Y \in \mathbf{D}$, there is a $\mathbf{C}$-object $G\,Y \in \mathbf{C}$ and a $\mathbf{D}$-morphism $\varepsilon_Y : F(G\,Y) \longrightarrow Y$ with the following "universal property":

$$\text{(UP)}\quad \boxed{\begin{array}{l}\text{for all } X \in \mathbf{C} \text{ and } g \in \mathbf{D}(F\,X, Y)\\ \text{there is a unique } \overline{g} \in \mathbf{C}(X, G\,Y)\\ \text{satisfying } \varepsilon_Y \circ F(\overline{g}) = g\end{array}}$$

Characterisation of right adjoints

Theorem. A functor $F : \mathbf{C} \to \mathbf{D}$ has a right adjoint iff for all $\mathbf{D}$-objects $Y \in \mathbf{D}$, there is a $\mathbf{C}$-object $G\,Y \in \mathbf{C}$ and a $\mathbf{D}$-morphism $\varepsilon_Y : F(G\,Y) \to Y$ with the following "universal property":

(UP)

for all $X \in \mathbf{C}$ and $g \in \mathbf{D}(F\,X, Y)$
there is a unique $\overline{g} \in \mathbf{C}(X, G\,Y)$
satisfying $\varepsilon_Y \circ F(\overline{g}) = g$

$$\begin{tikzcd} & Y \\ \forall \quad F\,X \arrow[ur, "g"'] & \end{tikzcd}$$

Characterisation of right adjoints

Theorem. A functor $F : \mathbf{C} \longrightarrow \mathbf{D}$ has a right adjoint iff for all $\mathbf{D}$-objects $Y \in \mathbf{D}$, there is a $\mathbf{C}$-object $G\,Y \in \mathbf{C}$ and a $\mathbf{D}$-morphism $\varepsilon_Y : F(G\,Y) \longrightarrow Y$ with the following "universal property":

$$\text{(UP)}\quad \boxed{\begin{array}{l} \text{for all } X \in \mathbf{C} \text{ and } g \in \mathbf{D}(F\,X, Y) \\ \text{there is a unique } \overline{g} \in \mathbf{C}(X, G\,Y) \\ \text{satisfying } \varepsilon_Y \circ F(\overline{g}) = g \end{array}}$$

$$\forall \quad \begin{tikzcd} & Y \\ F\,X \arrow[ur, "g"'] & \end{tikzcd} \qquad \exists! \quad \begin{tikzcd} G\,Y \\ X \arrow[u, dashed, "\overline{g}"] \end{tikzcd} \qquad \text{with}$$

Characterisation of right adjoints

Theorem. A functor $F : \mathbf{C} \longrightarrow \mathbf{D}$ has a right adjoint iff for all $\mathbf{D}$-objects $Y \in \mathbf{D}$, there is a $\mathbf{C}$-object $G\,Y \in \mathbf{C}$ and a $\mathbf{D}$-morphism $\varepsilon_Y : F(G\,Y) \longrightarrow Y$ with the following "universal property":

$$\text{(UP)}\ \boxed{\begin{array}{l}\text{for all } X \in \mathbf{C} \text{ and } g \in \mathbf{D}(F\,X, Y)\\ \text{there is a unique } \overline{g} \in \mathbf{C}(X, G\,Y)\\ \text{satisfying } \varepsilon_Y \circ F(\overline{g}) = g\end{array}}$$

$\forall$

$$\begin{tikzcd} & Y \\ F\,X \arrow[ur, "g"'] & \end{tikzcd}$$

$\exists!$

$$\begin{tikzcd} G\,Y \\ X \arrow[u, dashed, "\overline{g}"] \end{tikzcd}$$

with

$$\begin{tikzcd} F(G\,Y) \arrow[r, "\varepsilon_Y"] & Y \\ F\,X \arrow[u, dashed, "F\,\overline{g}"] \arrow[ur, "g"'] & \end{tikzcd}$$

Proof of the Theorem—“only if” part:

Given an adjunction (F, G, θ), for each $Y \in \mathbf{D}$ we produce $\varepsilon_Y : F(G\,Y) \to Y$ in $\mathbf{D}$ satisfying (UP).

Proof of the Theorem—“only if” part:

Given an adjunction (F, G, θ), for each $Y \in \mathbf{D}$ we produce $\varepsilon_Y : F(G\,Y) \to Y$ in $\mathbf{D}$ satisfying (UP).

We are given $\theta_{X,Y} : \mathbf{D}(F\,X, Y) \cong \mathbf{C}(X, G\,Y)$, natural in X and Y. Define

$$\boxed{\varepsilon_Y \triangleq \theta^{-1}_{G\,Y,Y}(\mathtt{id}_{G\,Y}) : F(G\,Y) \longrightarrow Y}$$

In other words $\varepsilon_Y = \overline{\mathtt{id}_{G\,Y}}$.

Proof of the Theorem—“only if” part:

Given an adjunction (F, G, θ), for each $Y \in \mathbf{D}$ we produce $\varepsilon_Y : F(G\,Y) \to Y$ in $\mathbf{D}$ satisfying (UP).

We are given $\theta_{X,Y} : \mathbf{D}(F\,X, Y) \cong \mathbf{C}(X, G\,Y)$, natural in X and Y. Define

$$\boxed{\varepsilon_Y \triangleq \theta^{-1}_{G\,Y,Y}(\mathtt{id}_{G\,Y}) : F(G\,Y) \to Y}$$

In other words $\varepsilon_Y = \overline{\mathtt{id}_{G\,Y}}$.

Given any $\begin{cases} g : F\,X \to Y & \text{in } \mathbf{D} \\ f : X \to G\,Y & \text{in } \mathbf{C} \end{cases}$, by naturality of θ we have

$$\frac{F\,X \xrightarrow{g} Y}{X \xrightarrow{\overline{g}} G\,Y} \text{ and } \frac{\varepsilon_Y \circ F\,f : F\,X \xrightarrow{F\,f} F(G\,Y) \xrightarrow{\overline{\mathtt{id}_{G\,Y}}} Y}{f : X \xrightarrow{f} G\,Y \xrightarrow{\mathtt{id}_{G\,Y}} G\,Y}$$

Hence $g = \varepsilon_Y \circ F\,\overline{g}$ and $g = \varepsilon_Y \circ F\,f \;\Rightarrow\; \overline{g} = f$.

Thus we do indeed have (UP).

Proof of the Theorem—“if” part:

We are given $F : \mathbf{C} \to \mathbf{D}$ and for each $Y \in \mathbf{D}$ a $\mathbf{C}$-object $G\,Y$ and $\mathbf{C}$-morphism $\varepsilon_Y : F(G\,Y) \to Y$ satisfying (UP). We have to

1. *extend* $Y \mapsto G\,Y$ *to a functor* $G : \mathbf{D} \to \mathbf{C}$
2. *construct a natural isomorphism* $\theta : \mathtt{Hom}_{\mathbf{D}} \circ (F^{\mathrm{op}} \times \mathtt{id}_{\mathbf{D}}) \cong \mathtt{Hom}_{\mathbf{C}} \circ (\mathtt{id}_{\mathbf{C}^{\mathrm{op}}} \times G)$

Proof of the Theorem—“if” part:

We are given $F : \mathbf{C} \to \mathbf{D}$ and for each $Y \in \mathbf{D}$ a $\mathbf{C}$-object $G\,Y$ and $\mathbf{C}$-morphism $\varepsilon_Y : F(G\,Y) \to Y$ satisfying (UP). We have to

1. *extend* $Y \mapsto G\,Y$ *to a functor* $G : \mathbf{D} \to \mathbf{C}$

For each $\mathbf{D}$-morphism $g : Y' \to Y$ we get $F(G\,Y') \xrightarrow{\varepsilon_{Y'}} Y' \xrightarrow{g} Y$ and can apply (UP) to get

$$G\,g \triangleq \overline{g \circ \varepsilon_{Y'}} : G\,Y' \to G\,Y$$

The uniqueness part of (UP) implies

$$G\,\mathtt{id} = \mathtt{id} \quad \text{and} \quad G(g' \circ g) = G\,g' \circ G\,g$$

so that we get a functor $G : \mathbf{D} \to \mathbf{C}$. □

Proof of the Theorem—“if” part:

We are given $F : \mathbf{C} \longrightarrow \mathbf{D}$ and for each $Y \in \mathbf{D}$ a $\mathbf{C}$-object $G\,Y$ and $\mathbf{C}$-morphism $\varepsilon_Y : F(G\,Y) \longrightarrow Y$ satisfying (UP). We have to

2. *construct a natural isomorphism* $\theta : \mathtt{Hom}_{\mathbf{D}} \circ (F^{\mathrm{op}} \times \mathtt{id}_{\mathbf{D}}) \cong \mathtt{Hom}_{\mathbf{C}} \circ (\mathtt{id}_{\mathbf{C}^{\mathrm{op}}} \times G)$

Since for all $g : F\,X \longrightarrow Y$ there is a unique $f : X \longrightarrow G\,Y$ with $g = \varepsilon_Y \circ F\,f$,

$$f \mapsto \overline{f} \triangleq \varepsilon_Y \circ F\,f$$

determines a bijection $\mathbf{C}(X, G\,Y) \cong \mathbf{C}(F\,X, Y)$; and it is natural in X & Y because

$$\begin{aligned}
\overline{G\,v \circ f \circ u} &\triangleq \varepsilon_{Y'} \circ F(G\,v \circ f \circ u) \\
&= (\varepsilon_{Y'} \circ F(G\,v)) \circ F\,f \circ F\,u && \text{since } F \text{ is a functor} \\
&= (v \circ \varepsilon_Y) \circ F\,f \circ F\,u && \text{by definition of } G\,v \\
&= v \circ \overline{f} \circ F\,u && \text{by definition of } \overline{f}
\end{aligned}$$

So we can take θ to be the inverse of this natural isomorphism. □

Dual of the Theorem:

$G : \mathbf{C} \leftarrow \mathbf{D}$ has a left adjoint iff for all $X \in \mathbf{C}$ there are $F\,X \in \mathbf{D}$ and $\eta_X \in \mathbf{C}(X, G(F\,X))$ with the universal property:

for all $Y \in \mathbf{D}$ and $\underline{f} \in \mathbf{C}(X, G\,Y)$
there is a unique $\overline{f} \in \mathbf{D}(F\,X, Y)$
satisfying $G\,\overline{f} \circ \eta_X = \underline{f}$

Dual of the Theorem:

$G : \mathbf{C} \leftarrow \mathbf{D}$ has a left adjoint iff for all $X \in \mathbf{C}$ there are $F\,X \in \mathbf{D}$ and $\eta_X \in \mathbf{C}(X, G(F\,X))$ with the universal property:

$$\boxed{\begin{array}{l}\text{for all } Y \in \mathbf{D} \text{ and } f \in \mathbf{C}(X, G\,Y)\\ \text{there is a unique } \overline{f} \in \mathbf{D}(F\,X, Y)\\ \text{satisfying } G\,\overline{f} \circ \eta_X = f\end{array}}$$

E.g. we can conclude that the forgetful functor $U : \mathbf{Mon} \to \mathbf{Set}$ has a left adjoint $F : \mathbf{Set} \to \mathbf{Mon}$, because of the universal property of

$$F\,\Sigma \triangleq (\mathtt{List}\,\Sigma, @, \mathtt{nil}) \quad \text{and} \quad \eta_\Sigma : \Sigma \to \mathtt{List}\,\Sigma$$

noted in Lecture 3.

Why are adjoint functors important/useful?

Their universal property (UP) usually embodies some useful mathematical construction

(e.g. “freely generated structures are left adjoints for forgetting-stucture”)

and pins it down uniquely up to isomorphism.

Lecture 14

Dependent Types

A brief look at some category theory for modelling type theories with dependent types.

Will restrict attention to the case of **Set**, rather than in full generality.

Further reading:
M. Hofmann, *Syntax and Semantics of Dependent Types.* In: A.M. Pitts and P. Dybjer (eds), *Semantics and Logics of Computation* (CUP, 1997).

Simple types

$$\diamond, x_1 : T_1, \ldots, x_n : T_n \vdash t(x_1, \ldots, x_n) : T$$

Dependent types

$$\diamond, x_1 : T_1, \ldots, x_n : T_n \vdash t(x_1, \ldots, x_n) : T(x_1, \ldots, x_n)$$

and more generally

$$\diamond,\ x_1 : T_1,\ x_2 : T_2(x_1),\ x_3 : T_3(x_1, x_2),\ \ldots \vdash \\ t(x_1, x_2, x_3, \ldots) : T(x_1, x_2, x_3, \ldots)$$

If type expressions denote sets, then

a type $T_1(x)$ dependent upon $x : T$

should denote

an indexed family of sets $(E\,i \mid i \in I)$
(where I is the set denoted by type T)

i.e. $E : I \to \mathbf{Set}$ is a set-valued function on a set I.

For each $I \in \mathbf{Set}$, let $\boxed{\mathbf{Set}^I}$ be the category with

- $\mathtt{obj}(\mathbf{Set}^I) \triangleq (\mathtt{obj}\,\mathbf{Set})^I$, so objects are I-indexed families of sets, $X = (X_i \mid i \in I)$
- morphisms $f : X \longrightarrow Y$ in $\mathbf{Set}^I$ are I-indexed families of functions $f = (f_i \in \mathbf{Set}(X_i, Y_i) \mid i \in I)$
- composition: $(g \circ f) \triangleq (g_i \circ f_i \mid i \in I)$

  (i.e. use composition of functions in $\mathbf{Set}$ at each index $i \in I$)
- identity: $\mathtt{id}_X \triangleq (\mathtt{id}_{X_i} \mid i \in I)$

  (i.e. use identity functions in $\mathbf{Set}$ at each index $i \in I$)

For each $p : I \longrightarrow J$ in $\mathbf{Set}$, let $\boxed{p^* : \mathbf{Set}^J \longrightarrow \mathbf{Set}^I}$ be the functor defined by:

$$p^* \left(\begin{array}{c} Y_j \\ \Big\downarrow{\scriptstyle f_j} \\ Y'_j \end{array} \;\middle|\; j \in J \right) \triangleq \left(\begin{array}{c} Y_{p\,i} \\ \Big\downarrow{\scriptstyle f_{p\,i}} \\ Y'_{p\,i} \end{array} \;\middle|\; i \in I \right)$$

i.e. p^* takes J-indexed families of sets/functions to I-indexed ones by precomposing with p

Dependent products

of families of sets

For $I, J \in \mathbf{Set}$, consider the functor $\pi_1^* : \mathbf{Set}^I \to \mathbf{Set}^{I \times J}$ induced by precomposition with the first projection function $\pi_1 : I \times J \to I$.

Theorem. π_1^* has a left adjoint $\Sigma : \mathbf{Set}^{I \times J} \to \mathbf{Set}^I$.

Proof. We apply the Theorem from Lecture 13: for each $E \in \mathbf{Set}^{I \times J}$ we define $\Sigma\, E \in \mathbf{Set}^I$ and $\eta_E : E \to \pi_1^*(\Sigma\, E)$ in $\mathbf{Set}^{I \times J}$ with the required universal property…

Theorem. π_1^* has a left adjoint $\Sigma : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^I$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Sigma\, E \in \mathbf{Set}^I$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Sigma\, E)_i \triangleq \textstyle\sum_{j\in J} E_{(i,j)} = \{(j,e) \mid j \in J \ \wedge \ e \in E_{(i,j)}\}}$$

Theorem. π_1^* has a left adjoint $\Sigma : \mathbf{Set}^{I \times J} \longrightarrow \mathbf{Set}^I$.

For each $E \in \mathbf{Set}^{I \times J}$, define $\Sigma\, E \in \mathbf{Set}^I$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Sigma\, E)_i \triangleq \textstyle\sum_{j \in J} E_{(i,j)} = \{(j, e) \mid j \in J \ \wedge \ e \in E_{(i,j)}\}}$$

and define $\eta_E : E \longrightarrow \pi_1^*(\Sigma\, E)$ in $\mathbf{Set}^{I \times J}$ to be the function mapping each $(i, j) \in I \times J$ to the function $(\eta_E)_{(i.j)} : E_{(i,j)} \longrightarrow (\Sigma\, E)_i$ given by $\boxed{e \mapsto (j, e)}$.

Universal property–

Theorem. π_1^* has a left adjoint $\Sigma : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^I$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Sigma\, E \in \mathbf{Set}^I$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Sigma\, E)_i \triangleq \textstyle\sum_{j\in J} E_{(i,j)} = \{(j,e) \mid j \in J \ \wedge \ e \in E_{(i,j)}\}}$$

and define $\eta_E : E \longrightarrow \pi_1^*(\Sigma\, E)$ in $\mathbf{Set}^{I\times J}$ to be the function mapping each $(i,j) \in I \times J$ to the function $(\eta_E)_{(i.j)} : E_{(i,j)} \longrightarrow (\Sigma\, E)_i$ given by $\boxed{e \mapsto (j,e)}$.

Universal property–*existence part*: given any $X \in \mathbf{Set}^I$ and $f : E \longrightarrow \pi_1^*(X)$ in $\mathbf{Set}^{I\times J}$, we have

$$\begin{tikzcd} E \arrow[r, "\eta_E"] \arrow[dr, "f"'] & \pi_1^*(\Sigma\, E) \arrow[d, dashed, "\pi_1^*(\overline{f})"] & \Sigma\, E \arrow[d, dashed, "\overline{f}"] \\ & \pi_1^*(X) & X \end{tikzcd}$$

where for all $i \in I$, $j \in J$ and $e \in E_{(i,j)}$ $\boxed{\overline{f}_i(j,e) \triangleq f_{(i,j)}(e)}$

Theorem. π_1^* has a left adjoint $\Sigma : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^{I}$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Sigma\, E \in \mathbf{Set}^{I}$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Sigma\, E)_i \triangleq \textstyle\sum_{j\in J} E_{(i,j)} = \{(j,e) \mid j \in J \ \wedge \ e \in E_{(i,j)}\}}$$

and define $\eta_E : E \longrightarrow \pi_1^*(\Sigma\, E)$ in $\mathbf{Set}^{I\times J}$ to be the function mapping each $(i,j) \in I \times J$ to the function $(\eta_E)_{(i.j)} : E_{(i,j)} \longrightarrow (\Sigma\, E)_i$ given by $\boxed{e \mapsto (j,e)}$.

Universal property–*uniqueness part*: given $g : \Sigma\, E \longrightarrow X$ in $\mathbf{Set}^{I}$ making

$$\begin{tikzcd}
E \arrow[r, "\eta_E"] \arrow[dr, "f"'] & \pi_1^*(\Sigma\, E) \arrow[d, dashed, "\pi_1^*(g)"] \\
& \pi_1^*(X)
\end{tikzcd}$$

commute in $\mathbf{Set}^{I\times J}$,

then for all $i \in I$, and $(j,e) \in (\Sigma\, E)_i$ we have

$$\overline{f}_i(j,e) \triangleq f_{(i,j)}(e) = (\pi_1^* g \circ \eta_E)_{(i,j)}\, e = (\pi_1^* g)_{(i,j)}((\eta_E)_{(i,j)}\, e) \triangleq g_i(j,e)$$

so $g = \overline{f}$. □

Dependent functions

of families of sets

We have seen that the left adjoint to $\pi_1^* : \mathbf{Set}^I \to \mathbf{Set}^{I\times J}$ is given by dependent products of sets.

Dually, dependent function sets give:

Theorem. π_1^* has a right adjoint $\Pi : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^I$.

Proof. We apply the Theorem from Lecture 13: for each $E \in \mathbf{Set}^{I\times J}$ we define $\Pi\, E \in \mathbf{Set}^I$ and $\varepsilon_E : \pi_1^*(\Pi\, E) \to E$ in $\mathbf{Set}^{I\times J}$ with the required universal property…

Theorem. π_1^* has a right adjoint $\Pi : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^I$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Pi\, E \in \mathbf{Set}^I$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Pi\, E)_i \triangleq \prod_{j\in J} E_{(i,j)} = \{f \subseteq (\Sigma\, E)_i \mid f \text{ is single-value and total}\}}$$

where $f \subseteq (\Sigma\, E)_i$ is

single-valued if $\forall j \in J, \forall e, e' \in E_{(i,j)},\ (j, e) \in f \wedge (j, e') \in f \Rightarrow e = e'$

total if $\forall j \in J, \exists e \in E_{(i,j)}\ (j, e) \in f$

Thus each $f \in (\Pi\, E)_i$ is a dependently typed function mapping elements $j \in J$ to elements of $E_{(i,j)}$ (result set depends on the argument j).

Theorem. π_1^* has a right adjoint $\Pi : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^I$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Pi\, E \in \mathbf{Set}^I$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Pi\, E)_i \triangleq \prod_{j\in J} E_{(i,j)} = \{f \subseteq (\Sigma\, E)_i \mid f \text{ is single-value and total}\}}$$

and define $\varepsilon_E : \pi_1^*(\Pi\, E) \longrightarrow E$ in $\mathbf{Set}^{I\times J}$ to be the function mapping each $(i, j) \in I \times J$ to the function $(\varepsilon_E)_{(i,j)} : (\Pi\, E)_i \longrightarrow E_{(i,j)}$ given by $\boxed{f \mapsto f\ j}$ = unique $e \in E_{(i,j)}$ such that $(j, e) \in f$.

Universal property–

Theorem. π_1^* has a right adjoint $\Pi : \mathbf{Set}^{I\times J} \to \mathbf{Set}^I$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Pi\, E \in \mathbf{Set}^I$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Pi\, E)_i \triangleq \textstyle\prod_{j\in J} E_{(i,j)} = \{f \subseteq (\Sigma\, E)_i \mid f \text{ is single-value and total}\}}$$

and define $\varepsilon_E : \pi_1^*(\Pi\, E) \to E$ in $\mathbf{Set}^{I\times J}$ to be the function mapping each $(i,j) \in I \times J$ to the function $(\varepsilon_E)_{(i,j)} : (\Pi\, E)_i \to E_{(i,j)}$ given by $\boxed{f \mapsto f\ j}$ = unique $e \in E_{(i,j)}$ such that $(j,e) \in f$.

Universal property–*existence part*: given any $X \in \mathbf{Set}^I$ and $f : \pi_1^*(X) \to E$ in $\mathbf{Set}^{I\times J}$, we have

$$\begin{tikzcd}
\Pi\, E & \pi_1^*(\Pi\, E) \arrow[r, "\varepsilon_E"] & E \\
X \arrow[u, dashed, "\overline{f}"] & \pi_1^*(X) \arrow[u, dashed, "\pi_1^*(\overline{f})"] \arrow[ur, "f"'] &
\end{tikzcd}$$

where for all $i \in I$ and $x \in X_i$ $\boxed{\overline{f}_i\, x \triangleq \{(j, f_{(i,j)}\, x) \mid j \in J\}}$

Theorem. π_1^* has a right adjoint $\Pi : \mathbf{Set}^{I\times J} \longrightarrow \mathbf{Set}^{I}$.

For each $E \in \mathbf{Set}^{I\times J}$, define $\Pi\, E \in \mathbf{Set}^{I}$ to be the function mapping each $i \in I$ to the set

$$\boxed{(\Pi\, E)_i \triangleq \textstyle\prod_{j\in J} E_{(i,j)} = \{f \subseteq (\Sigma\, E)_i \mid f \text{ is single-value and total}\}}$$

and define $\varepsilon_E : \pi_1^*(\Pi\, E) \longrightarrow E$ in $\mathbf{Set}^{I\times J}$ to be the function mapping each $(i,j) \in I \times J$ to the function $(\varepsilon_E)_{(i,j)} : (\Pi\, E)_i \longrightarrow E_{(i,j)}$ given by $\boxed{f \mapsto f\ j}$ = unique $e \in E_{(i,j)}$ such that $(j,e) \in f$.

Universal property–*uniqueness part*: given $g : X \longrightarrow \Pi\, E$ in $\mathbf{Set}^{I}$ making

$$\begin{tikzcd}
\pi_1^*(\Pi\, E) \arrow[r, "\varepsilon_E"] & E \\
\pi_1^*(X) \arrow[u, "\pi_1^*(g)"] \arrow[ur, "f"'] &
\end{tikzcd}$$

commute in $\mathbf{Set}^{I\times J}$,

then for all $i \in I$, $j \in J$ and $x \in X_i$ we have

$$\overline{f}_i\, x\, j \triangleq f_{(i,j)}\, x = (\varepsilon_E \circ \pi_1^* g)_{(i,j)}\, x = (\varepsilon_E)_{(i,j)}(g_i\, x) \triangleq g_i\, x\, j$$

so $g = \overline{f}$. □

Isomorphism of categories

Two categories $\mathbf{C}$ and $\mathbf{D}$ are isomorphic if they are isomorphic objects in the category of all categories of some given size, that is, if there are functors

$$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$$ with $\mathtt{id}_{\mathbf{C}} = G \circ F$ and $F \circ G = \mathtt{id}_{\mathbf{D}}$.

In which case, as usual, we write $\boxed{\mathbf{C} \cong \mathbf{D}}$.

Equivalence of categories

Two categories $\mathbf{C}$ and $\mathbf{D}$ are equivalent if there are functors
$$\begin{tikzcd}\mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"]\end{tikzcd}$$
and natural isomorphisms $\eta : \mathtt{id}_{\mathbf{C}} \cong G \circ F$ and $\varepsilon : F \circ G \cong \mathtt{id}_{\mathbf{D}}$.
In which case, one writes $\boxed{\mathbf{C} \simeq \mathbf{D}}$.

Equivalence of categories

Two categories $\mathbf{C}$ and $\mathbf{D}$ are equivalent if there are functors

$$\begin{tikzcd}\mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"]\end{tikzcd}$$

and natural isomorphisms $\eta : \mathtt{id}_{\mathbf{C}} \cong G \circ F$ and $\varepsilon : F \circ G \cong \mathtt{id}_{\mathbf{D}}$.
In which case, one writes $\boxed{\mathbf{C} \simeq \mathbf{D}}$.

Some deep results in mathematics take the form of equivalences of categories. E.g.

Stone duality: $\begin{pmatrix}\text{category of}\\ \text{Boolean algebras}\end{pmatrix}^{\mathrm{op}} \simeq \begin{pmatrix}\text{category of compact}\\ \text{totally disconnected}\\ \text{Hausdorff spaces}\end{pmatrix}$

Gelfand duality: $\begin{pmatrix}\text{category of}\\ \text{abelian } C^* \text{ algebras}\end{pmatrix}^{\mathrm{op}} \simeq \begin{pmatrix}\text{category of compact}\\ \text{Hausdorff spaces}\end{pmatrix}$

Example: $\mathbf{Set}^I \simeq \mathbf{Set}/I$

$\mathbf{Set}/I$ is a slice category:

- objects are pairs (E, p) where $E \in \mathtt{obj}\,\mathbf{Set}$ and $p \in \mathbf{Set}(E, I)$
- morphisms $g : (E, p) \longrightarrow (E', p')$ are $f \in \mathbf{Set}(E, E')$ satisfying $p' \circ f = p$ in $\mathbf{Set}$
- composition and identities – as for $\mathbf{Set}$

Example: $\mathbf{Set}^I \simeq \mathbf{Set}/I$

There are functors $F : \mathbf{Set}^I \longrightarrow \mathbf{Set}/I$ and $G : \mathbf{Set}/I \longrightarrow \mathbf{Set}^I$, given on objects and morphisms by:

$$\begin{aligned} F\,X &\triangleq (\{(i,x) \mid i \in I \wedge x \in X_i\}, \mathtt{fst}) \\ F\,f\,(i,x) &\triangleq (i, f_i\,x) \end{aligned}$$

$$\begin{aligned} G(E,p) &\triangleq (\{e \in E \mid p\,e = i\} \mid i \in I) \\ (G\,f)_i\,e &\triangleq f\,e \end{aligned}$$

Example: $\mathbf{Set}^I \simeq \mathbf{Set}/I$

There are functors $F : \mathbf{Set}^I \longrightarrow \mathbf{Set}/I$ and $G : \mathbf{Set}/I \longrightarrow \mathbf{Set}^I$, given on objects and morphisms by:

$$F\,X \triangleq (\{(i,x) \mid i \in I \wedge x \in X_i\}, \mathtt{fst})$$

$$F\,f\,(i,x) \triangleq (i, f_i\,x)$$

$$G(E,p) \triangleq (\{e \in E \mid p\,e = i\} \mid i \in I)$$

$$(G\,f)_i\,e \triangleq f\,e$$

There are natural isomorphisms

$$\eta : \mathtt{id}_{\mathbf{Set}^I} \cong G \circ F \text{ and } \varepsilon : F \circ G \cong \mathtt{id}_{\mathbf{Set}/I}$$

defined by…[exercise]

FACT Given $p : I \to J$ in **Set**, the composition

$$\mathbf{Set}/J \simeq \mathbf{Set}^J \xrightarrow{p^*} \mathbf{Set}^I \simeq \mathbf{Set}/I$$

is the functor "pullback along p".

One can generalize from **Set** to any category $\mathbf{C}$ with pullbacks and model Σ/Π types by left/right adjoints to pullback functors – see locally cartesian closed categories in the literature.

Lecture 15

Presheaf categories

Let $\mathbf{C}$ be a small category. The functor category $\boxed{\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}}$ is called the category of presheaves on $\mathbf{C}$.

- objects are contravariant functors from $\mathbf{C}$ to $\mathbf{Set}$
- morphisms are natural transformations

Much used in the semantics of various dependently-typed languages and logics.

Given a category $\mathbf{C}$ with a terminal object $\mathtt{1}$

A global element of an object $X \in \mathtt{obj}\,\mathbf{C}$ is by definition a morphism $\mathtt{1} \to X$ in $\mathbf{C}$

E.g. in **Set** …

E.g. in **Mon** …

Given a category $\mathbf{C}$ with a terminal object $\mathtt{1}$

A global element of an object $X \in \mathtt{obj}\,\mathbf{C}$ is by definition a morphism $\mathtt{1} \to X$ in $\mathbf{C}$

We say $\mathbf{C}$ is well-pointed if for all $f, g : X \to Y$ in $\mathbf{C}$ we have:

$$\left(\forall 1 \xrightarrow{x} X,\ f \circ x = g \circ x\right) \implies f = g$$

(**Set** is, **Mon** isn't.)

Idea:

replace global elements of X, $\mathtt{1} \xrightarrow{x} X$

by arbitrary morphisms $Y \xrightarrow{x} X$ (for any $Y \in \mathtt{obj}\, \mathbf{C}$)

Idea:

replace global elements of X, $\mathtt{1} \xrightarrow{x} X$

by arbitrary morphisms $Y \xrightarrow{x} X$ (for any $Y \in \mathtt{obj}\,\mathbf{C}$)

Some people use the notation $\boxed{x \in_Y X}$ and say "x is a generalised element of X at stage Y"

Have to take into account "change of stage":

$$x \in_Y X \ \wedge \ Z \xrightarrow{f} Y \ \Rightarrow \ x \circ f \in_Z X$$

(cf. Kripke's "possible world" semantics of intuitionistic and modal logics)

Yoneda functor

$$\boxed{\mathrm{y} : \mathbf{C} \to \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}}$$

(where $\mathbf{C}$ is a small category)

is the Curried version of the hom functor

$$\mathbf{C} \times \mathbf{C}^{\mathrm{op}} \cong \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \xrightarrow{\mathtt{Hom}_{\mathbf{C}}} \mathbf{Set}$$

Yoneda functor

$$\boxed{\mathrm{y} : \mathbf{C} \longrightarrow \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}}$$

(where $\mathbf{C}$ is a small category)

is the Curried version of the hom functor

$$\mathbf{C} \times \mathbf{C}^{\mathrm{op}} \cong \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \xrightarrow{\mathtt{Hom}_{\mathbf{C}}} \mathbf{Set}$$

- For each $\mathbf{C}$-object X, the object $\mathrm{y}X \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is the functor $\mathbf{C}(\_, X) : \mathbf{C}^{\mathrm{op}} \longrightarrow \mathbf{Set}$ given by

$$\begin{tikzcd}
Z \arrow[dd, "f"] & \mapsto & \mathbf{C}(Z,X) & g \circ f \\
 & \mapsto & & \\
Y & \mapsto & \mathbf{C}(Y,X) \arrow[uu] & g \arrow[uu, mapsto]
\end{tikzcd}$$

Yoneda functor

$$\boxed{\mathrm{y} : \mathbf{C} \to \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}}$$

(where $\mathbf{C}$ is a small category)

is the Curried version of the hom functor

$$\mathbf{C} \times \mathbf{C}^{\mathrm{op}} \cong \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \xrightarrow{\mathtt{Hom}_{\mathbf{C}}} \mathbf{Set}$$

- For each $\mathbf{C}$-object X, the object $\mathrm{y}X \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is the functor $\mathbf{C}(\_, X) : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}$ given by

$$\begin{tikzcd}
Z \arrow[dd, "f"] & \mapsto & \mathbf{C}(Z,X) & g \circ f \\
 & \mapsto & & \\
Y & \mapsto & \mathbf{C}(Y,X) \arrow[uu] & g \arrow[uu, mapsto, "f^*"']
\end{tikzcd}$$

this function is often written as f^*

Yoneda functor

$$\boxed{\mathrm{y} : \mathbf{C} \longrightarrow \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}}$$

(where $\mathbf{C}$ is a small category)

is the Curried version of the hom functor

$$\mathbf{C} \times \mathbf{C}^{\mathrm{op}} \cong \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \xrightarrow{\mathtt{Hom}_{\mathbf{C}}} \mathbf{Set}$$

- For each $\mathbf{C}$-morphism $Y \xrightarrow{f} X$, the morphism $\mathrm{y}Y \xrightarrow{\mathrm{y}f} \mathrm{y}X$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is the natural transformation whose component at any given $Z \in \mathbf{C}^{\mathrm{op}}$ is the function

$$\begin{tikzcd}
\mathrm{y}Y(Z) \arrow[r, "(\mathrm{y}f)_Z"] \arrow[d, phantom, "\shortparallel"] & \mathrm{y}X(Z) \arrow[d, phantom, "\shortparallel"] \\
\mathbf{C}(Z, Y) & \mathbf{C}(Z, X) \\
g \arrow[r, mapsto] & f \circ g
\end{tikzcd}$$

Yoneda functor

$$\boxed{\mathrm{y} : \mathbf{C} \longrightarrow \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}}$$

(where $\mathbf{C}$ is a small category)

is the Curried version of the hom functor

$$\mathbf{C} \times \mathbf{C}^{\mathrm{op}} \cong \mathbf{C}^{\mathrm{op}} \times \mathbf{C} \xrightarrow{\mathtt{Hom}_{\mathbf{C}}} \mathbf{Set}$$

- For each $\mathbf{C}$-morphism $Y \xrightarrow{f} X$, the morphism $\mathrm{y}Y \xrightarrow{\mathrm{y}f} \mathrm{y}X$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is the natural transformation whose component at any given $Z \in \mathbf{C}^{\mathrm{op}}$ is the function

this function is often written as f_*

$$\begin{tikzcd}
\mathrm{y}Y(Z) \arrow[r, "(\mathrm{y}f)_Z"] \arrow[d, phantom, "\shortparallel"] & \mathrm{y}X(Z) \arrow[d, phantom, "\shortparallel"] \\
\mathbf{C}(Z,Y) & \mathbf{C}(Z,X) \\
g \arrow[r, mapsto, "f_*"] & f \circ g
\end{tikzcd}$$

The Yoneda Lemma

For each small category $\mathbf{C}$, each object $X \in \mathbf{C}$ and each presheaf $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, there is a bijection of sets

$$\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$$

which is natural in both X and F.

The Yoneda Lemma

For each small category $\mathbf{C}$, each object $X \in \mathbf{C}$ and each presheaf $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, there is a bijection of sets

$$\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$$

which is natural in both X and F.

the value of
$F : \mathbf{C}^{\mathrm{op}} \longrightarrow \mathbf{Set}$
at X

the set of natural transformations from
the functor $\mathrm{y}X : \mathbf{C}^{\mathrm{op}} \longrightarrow \mathbf{Set}$
to the functor $F : \mathbf{C}^{\mathrm{op}} \longrightarrow \mathbf{Set}$

The Yoneda Lemma

For each small category $\mathbf{C}$, each object $X \in \mathbf{C}$ and each presheaf $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, there is a bijection of sets

$$\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$$

which is natural in both X and F.

Definition of the function $\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \to F(X)$:

for each $\theta : \mathrm{y}X \to F$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ we have the function $\mathbf{C}(X, X) = \mathrm{y}X(X) \xrightarrow{\theta_X} F(X)$ and define

$$\boxed{\eta_{X,F}(\theta) \triangleq \theta_X(\mathtt{id}_X)}$$

The Yoneda Lemma

For each small category $\mathbf{C}$, each object $X \in \mathbf{C}$ and each presheaf $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, there is a bijection of sets

$$\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$$

which is natural in both X and F.

Definition of the function $\eta_{X,F}^{-1} : F(X) \to \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F)$:

for each $x \in F(X)$, $Y \in \mathbf{C}$ and $f \in \mathrm{y}X(Y) = \mathbf{C}(Y, X)$,

we get a $F(X) \xrightarrow{F(f)} F(Y)$ in $\mathbf{Set}$ and hence $F(f)(x) \in F(Y)$;

The Yoneda Lemma

For each small category $\mathbf{C}$, each object $X \in \mathbf{C}$ and each presheaf $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, there is a bijection of sets

$$\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$$

which is natural in both X and F.

Definition of the function $\eta_{X,F}^{-1} : F(X) \to \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F)$:

for each $x \in F(X)$, $Y \in \mathbf{C}$ and $f \in \mathrm{y}X(Y) = \mathbf{C}(Y, X)$,

we get a $F(X) \xrightarrow{F(f)} F(Y)$ in $\mathbf{Set}$ and hence $F(f)(x) \in F(Y)$;

Define $\left(\eta_{X,F}^{-1}(x)\right)_Y : \mathrm{y}X(Y) \to F(Y)$ by

$$\boxed{\left(\eta_{X,F}^{-1}(x)\right)_Y (f) \triangleq F(f)(x)}$$

check this gives a natural transformation $\eta_{X,F}^{-1}(x) : \mathrm{y}X \to F$

Proof of $\boxed{\eta_{X,F} \circ \eta^{-1}_{X,F} = \mathtt{id}_{F(X)}}$

For any $x \in F(X)$ we have

$$\begin{aligned}
\eta_{X,F}\left(\eta^{-1}_{X,F}(x)\right) &\triangleq \left(\eta^{-1}_{X,F}(x)\right)_X(\mathtt{id}_X) && \text{by definition of } \eta_{X,F} \\
&\triangleq F(\mathtt{id}_X)(x) && \text{by definition of } \eta^{-1}_{X,F} \\
&= \mathtt{id}_{F(X)}(x) && \text{since } F \text{ is a functor} \\
&= x
\end{aligned}$$

Proof of $\boxed{\eta^{-1}_{X,F} \circ \eta_{X,F} = \mathtt{id}_{\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X,F)}}$

For any $\mathrm{y}X \xrightarrow{\theta} F$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ and $Y \xrightarrow{f} X$ in $\mathbf{C}$, we have

$$\begin{aligned}
\left(\eta^{-1}_{X,F}\left(\eta_{X,F}(\theta)\right)\right)_Y f &\triangleq \left(\eta^{-1}_{X,F}\left(\theta_X(\mathtt{id}_X)\right)\right)_Y f && \text{by definition of } \eta_{X,F} \\
&\triangleq F(f)(\theta_X(\mathtt{id}_X)) && \text{by definition of } \eta^{-1}_{X,F} \\
&= \theta_Y(f^*(id_X)) && \text{by naturality of } \theta \\
&\triangleq \theta_Y(\mathtt{id}_X \circ f) && \text{by definition of } f^* \\
&= \theta_Y(f)
\end{aligned}$$

naturality of θ

$$\begin{tikzcd}
\mathrm{y}X(Y) \arrow[r, "\theta_Y"] & F(Y) \\
\mathrm{y}X(X) \arrow[u, "f^*"'] \arrow[r, "\theta_X"'] & F(X) \arrow[u, "F(f)"']
\end{tikzcd}$$

Proof of $\boxed{\eta^{-1}_{X,F} \circ \eta_{X,F} = \mathtt{id}_{\mathbf{Set}^{\mathrm{C}^{\mathrm{op}}}(\mathrm{y}X,F)}}$

For any $\mathrm{y}X \xrightarrow{\theta} F$ in $\mathbf{Set}^{\mathrm{C}^{\mathrm{op}}}$ and $Y \xrightarrow{f} X$ in $\mathbf{C}$, we have

$$\begin{aligned}
\left(\eta^{-1}_{X,F}\left(\eta_{X,F}(\theta)\right)\right)_Y f &\triangleq \left(\eta^{-1}_{X,F}\left(\theta_X(\mathtt{id}_X)\right)\right)_Y f && \text{by definition of } \eta_{X,F} \\
&\triangleq F(f)(\theta_X(\mathtt{id}_X)) && \text{by definition of } \eta^{-1}_{X,F} \\
&= \theta_Y(f^*(id_X)) && \text{by naturality of } \theta \\
&\triangleq \theta_Y(\mathtt{id}_X \circ f) && \text{by definition of } f^* \\
&= \theta_Y(f)
\end{aligned}$$

so $\forall \theta, Y,\ \left(\eta^{-1}_{X,F}\left(\eta_{X,F}(\theta)\right)\right)_Y = \theta_Y$

so $\forall \theta,\ \eta^{-1}_{X,F}\left(\eta_{X,F}(\theta)\right) = \theta$

so $\eta^{-1}_{X,F} \circ \eta_{X,F} = \mathtt{id}$.

The Yoneda Lemma

For each small category $\mathbf{C}$, each object $X \in \mathbf{C}$ and each presheaf $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, there is a bijection of sets

$$\eta_{X,F} : \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$$

which is natural in both X and F.

Proof that $\eta_{X,F}$ is natural in F:

Given $F \xrightarrow{\varphi} G$ in $\mathbf{Set}^{\mathrm{C}^{\mathrm{op}}}$, have to show that

$$\begin{tikzcd}
\mathbf{Set}^{\mathrm{C}^{\mathrm{op}}}(\mathrm{y}X, F) \arrow[r, "\eta_{X,F}"] \arrow[d, "\varphi_*"'] & F(X) \arrow[d, "\varphi_X"] \\
\mathbf{Set}^{\mathrm{C}^{\mathrm{op}}}(\mathrm{y}X, G) \arrow[r, "\eta_{X,G}"'] & G(X)
\end{tikzcd}$$

commutes in $\mathbf{Set}$. For all $\mathrm{y}X \xrightarrow{\theta} F$ we have

$$\begin{aligned}
\varphi_X\left(\eta_{X,F}(\theta)\right) &\triangleq \varphi_X\left(\theta_X(\mathtt{id}_X)\right) \\
&\triangleq (\varphi \circ \theta)_X(\mathtt{id}_X) \\
&\triangleq \eta_{X,G}(\varphi \circ \theta) \\
&\triangleq \eta_{X,G}(\varphi_*(\theta))
\end{aligned}$$

Proof that $\eta_{X,F}$ is natural in X:

Given $Y \xrightarrow{f} X$ in $\mathbf{C}$, have to show that

$$\begin{tikzcd}
\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \arrow[r, "\eta_{X,F}"] \arrow[d, "(\mathrm{y}f)^*"'] & F(X) \arrow[d, "F(f)"] \\
\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}Y, F) \arrow[r, "\eta_{Y,F}"'] & F(Y)
\end{tikzcd}$$

commutes in $\mathbf{Set}$. For all $\mathrm{y}X \xrightarrow{\theta} F$ we have

$$\begin{aligned}
F(f)((\eta_{X,F}(\theta)) &\triangleq F(f)(\theta_X(\mathtt{id}_X)) \\
&= \theta_Y(f^*(\mathtt{id}_X)) && \text{by naturality of } \theta \\
&= \theta_Y(f) \\
&= \theta_Y(f_*(\mathtt{id}_Y)) \\
&\triangleq (\theta \circ \mathrm{y}f)_Y(\mathtt{id}_Y) \\
&\triangleq \eta_{Y,F}(\theta \circ \mathrm{y}f) \\
&\triangleq \eta_{Y,F}((\mathrm{y}f)^*(\theta))
\end{aligned}$$

Corollary of the Yoneda Lemma:

the functor $\mathbf{y} : \mathbf{C} \longrightarrow \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is full and faithful.

In general, a functor $F : \mathbf{C} \longrightarrow \mathbf{D}$ is

- faithful if for all $X, Y \in \mathbf{C}$ the function

$$\begin{array}{ccc} \mathbf{C}(X, Y) & \longrightarrow & \mathbf{D}(F(X), F(Y)) \\ f & \longmapsto & F(f) \end{array}$$

  is injective:

$$\forall f, f' \in \mathbf{C}(X, Y),\ F(f) = F(f') \Rightarrow f = f'$$

- full if the above functions are all surjective:

$$\forall g \in \mathbf{D}(F(X), F(Y)), \exists f \in \mathbf{C}(X, Y),\ F(f) = g$$

Corollary of the Yoneda Lemma:

the functor $\mathrm{y} : \mathbf{C} \longrightarrow \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is full and faithful.

Proof. From the proof of the Yoneda Lemma, for each $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ we have a bijection

$$F(X) \xrightarrow{(\eta_{X,F})^{-1}} \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F)$$

By definition of $(\eta_{X,F})^{-1}$, when $F = \mathrm{y}Y$ the above function is equal to

$$\begin{array}{rcl} \mathrm{y}Y(X) = \mathbf{C}(X,Y) & \longrightarrow & \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, \mathrm{y}Y) \\ f & \mapsto & f_* = \mathrm{y}f \end{array}$$

So, being a bijection, $f \mapsto \mathrm{y}f$ is both injective and surjective; so y is both faithful and full. □

Recall (for a small category $\mathbf{C}$):

Yoneda functor $\mathrm{y} : \mathbf{C} \to \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$

Yoneda Lemma: there is a bijection $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, F) \cong F(X)$ which is natural both in $F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ and $X \in \mathbf{C}$.

An application of the Yoneda Lemma:

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

Proof sketch.

Terminal object in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is the functor $\mathtt{1} : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}$ given by

$$\begin{cases} \mathtt{1}(X) \triangleq \{0\} & \text{terminal object in } \mathbf{Set} \\ \mathtt{1}(f) \triangleq \mathtt{id}_{\{0\}} \end{cases}$$

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

Proof sketch.

Product of $F, G \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ is the functor $F \times G : \mathbf{C}^{\mathrm{op}} \to \mathbf{Set}$ given by

$$\begin{cases} (F \times G)(X) \triangleq F(X) \times G(X) & \text{cartesian product of sets} \\ (F \times G)(f) \triangleq F(f) \times G(f) \end{cases}$$

with projection morphisms $F \xleftarrow{\pi_1} F \times G \xrightarrow{\pi_2} G$ given by the natural transformations whose components at $X \in \mathbf{C}$ are the projection functions $F(X) \xleftarrow{\pi_1} F(X) \times G(X) \xrightarrow{\pi_2} G(X)$.

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

Proof sketch.

We can work out what the value of the exponential $G^F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ at $X \in \mathbf{C}$ has to be using the Yoneda Lemma:

$$G^F(X) \underset{\text{Yoneda Lemma}}{\cong} \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, G^F) \underset{\text{universal property of the exponential}}{\cong} \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G)$$

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

Proof sketch.

We can work out what the value of the exponential $G^F \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ at $X \in \mathbf{C}$ has to be using the Yoneda Lemma:

$$G^F(X) \cong \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X, G^F) \cong \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G)$$

We take the set $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G)$ to be the definition of the value of G^F at X…

Exponential objects in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$:

$$\boxed{G^F(X) \triangleq \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G)}$$

Given $Y \xrightarrow{f} X$ in $\mathbf{C}$, we have $\mathrm{y}Y \xrightarrow{\mathrm{y}f} \mathrm{y}X$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ and hence

$$\begin{array}{rcl} G^F(Y) \triangleq \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}Y \times F, G) & \to & \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G) \triangleq G^F(X) \\ \theta & \mapsto & \theta \circ (\mathrm{y}f \times \mathtt{id}_F) \end{array}$$

We define

$$\boxed{G^F(f) \triangleq (\mathrm{y}f \times \mathtt{id}_F)^*}$$

Have to check that these definitions make G^F ino a functor $\mathbf{C}^{\mathrm{op}} \to \mathbf{Set}$.

Application morphisms in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$:

Given $F, G \in \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, the application morphism

$$\mathtt{app} : G^F \times F \longrightarrow G$$

is the natural transformation whose component at $X \in \mathbf{C}$ is the function

$$(G^F \times F)(X) \triangleq G^F(X) \times F(X) \triangleq \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G) \times F(X) \xrightarrow{\mathtt{app}_X} G(X)$$

defined by

$$\boxed{\mathtt{app}_X(\theta, x) \triangleq \theta_X(\mathtt{id}_X, x)}$$

Have to check that this is natural in X.

Currying operation in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$:

$$\left(H \times F \xrightarrow{\theta} G\right) \mapsto \left(H \xrightarrow{\mathtt{cur}\,\theta} G^F\right)$$

Given $H \times F \xrightarrow{\theta} G$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$, the component of $\mathtt{cur}\,\theta$ at $X \in \mathbf{C}$

$$H(X) \xrightarrow{(\mathtt{cur}\,\theta)_X} G^F(X) \triangleq \mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}(\mathrm{y}X \times F, G)$$

is the function mapping each $z \in H(X)$ to the natural transformation $\mathrm{y}X \times F \to G$ whose component at $Y \in \mathbf{C}$ is the function

$$(\mathrm{y}X \times F)(Y) \triangleq \mathbf{C}(Y, X) \times F(Y) \to G(Y)$$

defined by

$$\boxed{((\mathtt{cur}\,\theta)_X(z))_Y(f, y) \triangleq \theta_Y(H(f)(z), y)}$$

Currying operation in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$:

$$\left(H \times F \xrightarrow{\theta} G\right) \mapsto \left(H \xrightarrow{\mathtt{cur}\,\theta} G^F\right)$$

$$\boxed{((\mathtt{cur}\,\theta)_X(z))_Y(f, y) \triangleq \theta_Y(H(f)(z), y)}$$

Have to check that this is natural in Y,

then that $(\mathtt{cur}\,\theta)_X$ is natural in X,

then that $\mathtt{cur}\,\theta$ is the unique morphism $H \xrightarrow{\varphi} G^F$ in $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ satisfying $\mathtt{app} \circ (\varphi \times \mathtt{id}_F) = \theta$.

Theorem. For each small category $\mathbf{C}$, the category $\mathbf{Set}^{\mathbf{C}^{\mathrm{op}}}$ of presheaves is cartesian closed.

So we can interpret simply typed lambda calculus in any presheaf category.

More than that, presheaf categories (usefully) model dependently-typed languages.

Lecture 16

Monads

Used in Haskell to abstract generic aspects of computation (return a value, sequencing) and to encapsulate effectful code.

Concept imported into functional programming from category theory, first for its denotational semantics by Moggi and then for its practice by Wadler.

Monads

Used in Haskell to abstract generic aspects of computation (return a value, sequencing) and to encapsulate effectful code.

Concept imported into functional programming from category theory, first for its denotational semantics by Moggi and then for its practice by Wadler.

Here, a quick overview of:

- Moggi's computational λ-calculus and its categorical semantics using (strong) monads
- monads and adjunctions

Computational Lambda Calculus (CLC)

CLC extends STLC with new types, terms and equations...

Types: $A, B, \ldots ::=$ STLC types, plus

$\mathtt{T}(A)$ type of "computations" of values of type A

Terms: $s, t, \ldots ::=$ STLC terms, plus

$\mathtt{return}\, t$ trivial computation

$\mathtt{do}\{x \leftarrow s; t\}$ sequenced computation (binds free x in t)

As for STLC, we identify CLC syntax trees up to α-equivalence, where $=_\alpha$ is extended by the rules

$$\frac{t =_\alpha t'}{\mathtt{return}\, t =_\alpha \mathtt{return}\, t'} \quad \text{and} \quad \frac{s =_\alpha s' \qquad (y\ x) \cdot t =_\alpha (y\ x') \cdot t' \qquad y \text{ does not occur in } \{s, s', x, x', t, t'\}}{\mathtt{do}\{x \leftarrow s; t\} =_\alpha \mathtt{do}\{x' \leftarrow s'; t'\}}$$

Computational Lambda Calculus (CLC)

CLC extends STLC with new types, terms and equations...

Types: $A, B, \ldots ::=$ STLC types, plus

$\mathtt{T}(A)$ type of "computations" of values of type A

Terms: $s, t, \ldots ::=$ STLC terms, plus

$\mathtt{return}\, t$ trivial computation

$\mathtt{do}\{x \leftarrow s; t\}$ sequenced computation (binds free x in t)

Typing rules:

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash \mathtt{return}\, t : \mathtt{T}(A)}\ (\textsc{val}) \qquad \frac{\Gamma \vdash s : \mathtt{T}(A) \qquad \Gamma, x : A \vdash t : \mathtt{T}(B)}{\Gamma \vdash \mathtt{do}\{x \leftarrow s; t\} : \mathtt{T}(B)}\ (\textsc{seq})$$

Equations...

CLC equations

Extend STLC $\beta\eta$-equality ($\Gamma \vdash s =_{\beta\eta} t : A$) to a relation $\Gamma \vdash s = t : A$ by adding the following rules:

$$\boxed{\dfrac{\Gamma \vdash s : A \qquad \Gamma, x : A \vdash t : \mathtt{T}(B)}{\Gamma \vdash \mathtt{do}\{x \leftarrow \mathtt{return}\, s; t\} = t[s/x] : \mathtt{T}(B)}}$$

$$\boxed{\dfrac{\Gamma \vdash t : \mathtt{T}(A)}{\Gamma \vdash t = \mathtt{do}\{x \leftarrow t; \mathtt{return}\, x\} : \mathtt{T}(A)}}$$

$$\boxed{\dfrac{\Gamma \vdash s : \mathtt{T}(A) \qquad \Gamma, x : A \vdash t : \mathtt{T}(B) \qquad \Gamma, y : B \vdash u : \mathtt{T}(C)}{\Gamma \vdash \mathtt{do}\{y \leftarrow \mathtt{do}\{x \leftarrow s; t\}; u\} = \mathtt{do}\{x \leftarrow s; \mathtt{do}\{y \leftarrow t; u\}\}}}$$

CLC equations

Extend STLC $\beta\eta$-equality ($\Gamma \vdash s =_{\beta\eta} t : A$) to a relation $\Gamma \vdash s = t : A$ by adding the following rules:

$$\boxed{\frac{\Gamma \vdash s : A \qquad \Gamma, x : A \vdash t : \mathtt{T}(B)}{\Gamma \vdash \mathtt{do}\{x \leftarrow \mathtt{return}\, s; t\} = t[s/x] : \mathtt{T}(B)}}$$

$$\boxed{\frac{\Gamma \vdash t : \mathtt{T}(A)}{\Gamma \vdash t = \mathtt{do}\{x \leftarrow t; \mathtt{return}\, x\} : \mathtt{T}(A)}}$$

$$\boxed{\frac{\Gamma \vdash s : \mathtt{T}(A) \qquad \Gamma, x : A \vdash t : \mathtt{T}(B) \qquad \Gamma, y : B \vdash u : \mathtt{T}(C)}{\Gamma \vdash \mathtt{do}\{y \leftarrow \mathtt{do}\{x \leftarrow s; t\}; u\} = \mathtt{do}\{x \leftarrow s; \mathtt{do}\{y \leftarrow t; u\}\}}}$$

(To describe a particular notion of computation (I/O, mutable state, exceptions, concurrent processes, ...) one can consider extensions of vanilla CLC, e.g. with extra ground types, constants and equations.)

Parameterised Kleisli triple

is the following extra structure on a category $\mathbf{C}$ with binary products:

- a function mapping each $X \in \texttt{obj}\,\mathbf{C}$ to an object $T(X) \in \texttt{obj}\,\mathbf{C}$
- for each $X \in \texttt{obj}\,\mathbf{C}$, a $\mathbf{C}$-morphism $X \xrightarrow{\eta_X} T(X)$
- for each $\mathbf{C}$-morphism $X \times Y \xrightarrow{f} T(Z)$ a $\mathbf{C}$-morphism $X \times T(Y) \xrightarrow{f^*} T(Z)$

satisfying…

Parameterised Kleisli triple[cont.]

- if $X \xrightarrow{f} X'$ and $X' \times Y \xrightarrow{g} T(Z)$, then

$$(g \circ (f \times \mathtt{id}_Y))^* = g^* \circ (f \times \mathtt{id}_{T(Y)})$$

- if $X \times Y \xrightarrow{f} T(Z)$, then

$$f^* \circ (\mathtt{id}_X \times \eta_Y) = f$$

- if $X \times Y \xrightarrow{f} T(Z)$ and $X \times Z \xrightarrow{g} T(W)$, then

$$(g^* \circ \langle \pi_1 , f \rangle)^* = g^* \circ \langle \pi_1 , f^* \rangle$$

Examples in **Set**

State: fix a set S (of “states”) and define

$T(X) \triangleq (X \times S)^S$

$\eta_X \, x \, s \triangleq (x, s)$

$f^*(x, t) \, s \triangleq f(x, y) \, s'$ where $t \, s = (y, s')$

Examples in **Set**

State: fix a set S (of “states”) and define

$$T(X) \triangleq (X \times S)^S$$

computations are functions $S \to X \times S$ taking states to values in X paired with a next state

$$\eta_X\, x\, s \triangleq (x, s)$$

$$f^*(x, t)\, s \triangleq f(x, y)\, s' \text{ where } t\, s = (y, s')$$

$f^*(x, \_)$ first “runs” $t \in T(Y)$ in state s to get (y, s'), then runs $f(x, y) \in T(Z)$ in the new state s'

Examples in **Set**

Error:

$$T(X) \triangleq X + 1 = \{(0, x) \mid x \in X\} \cup \{(1, 0)\}$$

$$\eta_X\, x \triangleq (0, x)$$

$$f^*(x, t) \triangleq \begin{cases} f(x, y) & \text{if } t = (0, y) \\ (1, 0) & \text{if } t = (1, 0) \end{cases}$$

Examples in **Set**

Error:

$$T(X) \triangleq X + 1 = \{(0, x) \mid x \in X\} \cup \{(1, 0)\}$$

computations are either copies $(0, x)$ of values in $x \in X$ or an error $(1, 0)$

$$\eta_X\, x \triangleq (0, x)$$

$$f^*(x, t) \triangleq \begin{cases} f(x, y) & \text{if } t = (0, y) \\ (1, 0) & \text{if } t = (1, 0) \end{cases}$$

if $t \in T(Y)$ is the error, then $f^*(x, \_)$ propagates it, otherwise it acts like f

Examples in **Set**

Continuations: fix a set R (of “results”) and define

$T(X) \triangleq R^{(R^X)}$

$\eta_X\, x \triangleq \lambda c \in R^X .\, c\, x$

$f^*(x, r) \triangleq \lambda c \in R^Z .\, r(\lambda y \in Y .\, f(x, y)\, c)$

Examples in **Set**

Continuations: fix a set R (of “results”) and define

$$T(X) \triangleq R^{(R^X)}$$

computations are functions $r : R^X \to R$ mapping continuations $c \in R^X$ of the computation to results $r\,c \in R$

$$\eta_X\, x \triangleq \lambda c \in R^X .\, c\, x$$

$$f^*(x, r) \triangleq \lambda c \in R^Z .\, r(\lambda y \in Y .\, f(x, y)\, c)$$

f^* maps a computation $r \in R^{(R^Y)}$ to the function taking a continuation $c \in R^Z$ to the result of applying r to the continuation $\lambda y \in Y .\, f(x, y)\, c$ in R^Y

Semantics of CLC

Given a ccc $\mathbf{C}$ equipped with a parameterised Kleisli triple $(T, \eta, (\_)^*)$, we can extend the semantics of STLC to one for CLC.

Computation types: $[\![\mathtt{T}(A)]\!] = T([\![A]\!])$

Trivial computations:

$$[\![\Gamma \vdash \mathtt{return}\, t : \mathtt{T}(A)]\!] = [\![\Gamma]\!] \xrightarrow{[\![\Gamma \vdash t : A]\!]} [\![A]\!] \xrightarrow{\eta_{[\![A]\!]}} T([\![A]\!])$$

Sequencing: $[\![\Gamma \vdash \mathtt{do}\{x \leftarrow s; t\} : \mathtt{T}(B)]\!] = f^* \circ \langle \mathtt{id}_{[\![\Gamma]\!]}, g \rangle$

where $\begin{cases} f & = [\![\Gamma]\!] \times [\![A]\!] \xrightarrow{[\![\Gamma, x:A \vdash t:\mathtt{T}(B)]\!]} T([\![B]\!]) \\ g & = [\![\Gamma]\!] \xrightarrow{[\![\Gamma \vdash s:\mathtt{T}(A)]\!]} T([\![A]\!]) \end{cases}$

(and where A is uniquely determined from the proof of $\Gamma \vdash \mathtt{do}\{x \leftarrow s; t\} : \mathtt{T}(B)$)

Semantics of CLC

Given a ccc $\mathbf{C}$ equipped with a parameterised Kleisli triple $(T, \eta, (\_)^*)$, we can extend the semantics of STLC to one for CLC.

As for STLC versus cccs,

- the semantics of CLC in cc+Kleisli categories is equationally sound and complete
- one can use CLC as an internal language for describing constructs in cc+Kleisli categories
- there is a correspondence between equational theories in CLC and cc+Kleisli categories

Monads

A monad on a category $\mathbf{C}$ is given by a functor $T : \mathbf{C} \to \mathbf{C}$ and natural transformations $\eta : \mathtt{id} \to T$ and $\mu : T \circ T \to T$ satisfying

$$\begin{tikzcd}
T \arrow[r, "T\eta"] \arrow[dr, "\mathtt{id}_T"'] & T \circ T \arrow[d, "\mu"] & T \arrow[l, "\eta_T"'] \arrow[dl, "\mathtt{id}_T"] \\
& T &
\end{tikzcd}
\qquad
\begin{tikzcd}
T \circ T \circ T \arrow[r, "\mu_T"] \arrow[d, "T\mu"'] & T \circ T \arrow[d, "\mu"] \\
T \circ T \arrow[r, "\mu"'] & T
\end{tikzcd}$$

Monads

A monad on a category $\mathbf{C}$ is given by a functor $T : \mathbf{C} \to \mathbf{C}$ and natural transformations $\eta : \mathtt{id} \to T$ and $\mu : T \circ T \to T$ satisfying

$$\begin{tikzcd}
T \arrow[r, "T\eta"] \arrow[dr, "\mathtt{id}_T"'] & T \circ T \arrow[d, "\mu"] & T \arrow[l, "\eta_T"'] \arrow[dl, "\mathtt{id}_T"] \\
 & T &
\end{tikzcd}
\qquad
\begin{tikzcd}
T \circ T \circ T \arrow[r, "\mu_T"] \arrow[d, "T\mu"'] & T \circ T \arrow[d, "\mu"] \\
T \circ T \arrow[r, "\mu"'] & T
\end{tikzcd}$$

If $\mathbf{C}$ has binary products, then the monad is strong if there is a family of $\mathbf{C}$-morphisms $(X \times T(Y) \xrightarrow{s_{X,Y}} T(X \times Y) \mid X, Y \in \mathtt{obj}\,\mathbf{C})$ satisfying a number (7, in fact) of commutative diagrams (details omitted, see Moggi).

Monads

A monad on a category $\mathbf{C}$ is given by a functor $T : \mathbf{C} \to \mathbf{C}$ and natural transformations $\eta : \mathtt{id} \to T$ and $\mu : T \circ T \to T$ satisfying

$$\begin{tikzcd}
T \arrow[r, "T\eta"] \arrow[dr, "\mathtt{id}_T"'] & T \circ T \arrow[d, "\mu"] & T \arrow[l, "\eta_T"'] \arrow[dl, "\mathtt{id}_T"] \\
 & T &
\end{tikzcd}
\qquad
\begin{tikzcd}
T \circ T \circ T \arrow[r, "\mu_T"] \arrow[d, "T\mu"'] & T \circ T \arrow[d, "\mu"] \\
T \circ T \arrow[r, "\mu"'] & T
\end{tikzcd}$$

If $\mathbf{C}$ has binary products, then the monad is strong if there is a family of $\mathbf{C}$-morphisms $(X \times T(Y) \xrightarrow{s_{X,Y}} T(X \times Y) \mid X, Y \in \mathtt{obj}\,\mathbf{C})$ satisfying a number (7, in fact) of commutative diagrams (details omitted, see Moggi).

FACT: for a given category with binary products, "parameterised Kleisli triple" and "strong monad" are equivalent notions – each gives rise to the other in a bijective fashion.

Monads and adjunctions

- Given an adjunction $$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$$ $F \dashv G$

  we get a monad $(G \circ F, \eta, \mu)$ on $\mathbf{C}$

  where $\begin{cases} \eta_X & = \overline{\mathtt{id}_{F\,X}} \\ \mu_X & = G(\overline{\mathtt{id}_{G(F\,X)}}) \end{cases}$

  E.g. for $$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "F"] & \mathbf{Mon} \arrow[l, bend left, "U"] \end{tikzcd}$$ where U is the forgetful functor, $T = U \circ F$ is the list monad on $\mathbf{Set}$ ($T(X) = \mathtt{List}\,X$, η given by singleton lists, μ by flattening lists of lists). It's a strong monad (all monads of $\mathbf{Set}$ have a strength), but in general the monad associated with an adjunction may not be strong.

Monads and adjunctions

- Given an adjunction $\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$ $\underline{F \dashv G}$

  we get a monad $(G \circ F, \eta, \mu)$ on $\mathbf{C}$
- Given a monad (T, η, μ) on $\mathbf{C}$ we get an adjunction

  $$\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{C}^T \arrow[l, bend left, "G"] \end{tikzcd} \qquad \underline{F \dashv G}$$

Monads and adjunctions

- Given an adjunct

  we get a monad (

- Given a monad (

$$\begin{tikzcd}
\mathbf{C} \arrow[r, bend left, "F"] & \mathbf{C}^T \arrow[l, bend left, "G"]
\end{tikzcd}$$

$\mathbf{C}^T$ is the category of Eilenberg-Moore algebras for the monad T, which has objects (A, α) with $\alpha : T(A) \longrightarrow A$ satisfying

$$\begin{tikzcd}
A \arrow[r, "\eta_A"] \arrow[dr, "\mathtt{id}_A"'] & T\,A \arrow[d, "\alpha"] \\
& A
\end{tikzcd}
\qquad
\begin{tikzcd}
T(T\,A) \arrow[r, "\mu_A"] \arrow[d, "T\,\alpha"'] & T\,A \arrow[d, "\alpha"] \\
T\,A \arrow[r, "\alpha"'] & A
\end{tikzcd}$$

and morphisms $f(A, \alpha) \longrightarrow (B, \beta)$ with $f : A \longrightarrow B$ satisfying

$$\begin{tikzcd}
T\,A \arrow[r, "T\,f"] \arrow[d, "\alpha"'] & T\,B \arrow[d, "\beta"] \\
A \arrow[r, "f"'] & B
\end{tikzcd}$$

Monads and adjunctions

- Given an adjunction $\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$ $F \dashv G$

  we get a monad $(G \circ F, \eta, \mu)$ on $\mathbf{C}$

- Given a monad (T, η, μ) on $\mathbf{C}$ we get an adjunction

  $\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{C}^T \arrow[l, bend left, "G"] \end{tikzcd}$ $F \dashv G$

- Starting from $\begin{tikzcd} \mathbf{C} \arrow[r, bend left, "F"] & \mathbf{D} \arrow[l, bend left, "G"] \end{tikzcd}$ $F \dashv G$ and forming the monad

  $T = G \circ F$, there's an obvious functor $K : \mathbf{D} \to \mathbf{C}^T$.

  Monadicity Theorems impose conditions on $G : \mathbf{D} \to \mathbf{C}$ which ensure that K is an equivalence of categories. E.g. **Mon** is equivalent to the category of Eilenberg-Moore algebras for the list monad on **Set** (and similarly for any algebraic theory).

Some current themes involving category theory in computer science

- semantics of effects & co-effects in programming languages

  (monads and comonads)

- homotopy type theory

  (higher-dimensional category theory)

- structural aspects of networks, quantum computation/protocols, …

  (string diagrams for monoidal categories)

Next term: *Advanced Topics in Category Theory* (ACS module L118).