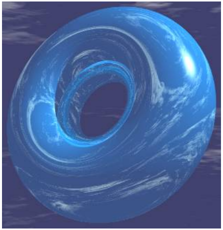
Conceptual Connections of Circularity and Category Theory

Larry Moss
Indiana University, Bloomington

ESSLLI 2012, Opole



set with algebraic operations	set with transitions and observations
algebra for a functor	coalgebra for a functor
initial algebra	final coalgebra
least fixed point	greatest fixed point
congruence relation	bisimulation equivalence rel'n
equational logic	modal logic
recursion: map out of an initial algebra	corecursion: map into a final coalgebra
Foundation Axiom	Anti-Foundation Axiom
iterative conception of set	coiterative conception of set
useful in syntax	useful in semantics
bottom-up	top-down

At this point, we have seen examples of circularly-defined sets such as

the set of streams

the set of infinite trees

One of the main goals of the course is to present a theory of how these "solution spaces" work.
The theory is based on the concept of a coalgebra for a functor and on similar notions from category theory.

Today's lecture includes an introduction to the main concepts which we'll need.

But it is not a systematic presentation of the subject.

$$
\begin{array}{rcl}
x & \approx & \langle 0, y \rangle \\
y & \approx & \langle 1, z \rangle \\
z & \approx & \langle 2, x \rangle
\end{array}
$$

I want to construe such a system as a function
from its set of variables.

So let $X = \{x, y, z\}$.

We regard the system as a function $e : X \to \{0, 1, 2\} \times N$.
(e stands for "equation".)

$$
\begin{array}{rcl}
e(x) & = & \langle 0, y \rangle \\
e(y) & = & \langle 1, z \rangle \\
e(z) & = & \langle 2, x \rangle
\end{array}
$$

$$\begin{array}{rcl} e(x) & = & \langle 0, y\rangle \\ e(y) & = & \langle 1, z\rangle \\ e(z) & = & \langle 2, x\rangle \end{array}$$

Let's write $\{0,1,2\}^\infty$ for the set of streams on $\{0,1,2\}$.

The solution to our system e is a function $e^\dagger : X \to \{0,1,2\}^\infty$.
Explicitly,

$$\begin{array}{rcl} e^\dagger(x) & = & (0,1,2,0,1,2,\ldots) \\ e^\dagger(y) & = & (1,2,0,1,2,0,\ldots) \\ e^\dagger(z) & = & (2,0,1,2,0,1\ldots) \end{array}$$

Now what we want to do is to talk
in an abstract way what the relation between
e and $e^\dagger$.

And what we say should hold for all systems.

$$
\begin{array}{rclcrcl}
e(x) & = & \langle 0, y\rangle & \qquad & e^\dagger(x) & = & (0,1,2,0,1,2,\ldots) \\
e(y) & = & \langle 1, z\rangle & & e^\dagger(y) & = & (1,2,0,1,2,0,\ldots) \\
e(z) & = & \langle 2, x\rangle & & e^\dagger(z) & = & (2,0,1,2,0,1\ldots)
\end{array}
$$

Here is what we want to say:

$$
\begin{tikzcd}
X \arrow[rr, "e"] \arrow[d, "e^\dagger"'] & & \{0,1,2\} \times X \arrow[d, "id_{\{0,1,2\}} \times e^\dagger"] \\
\{0,1,2\}^\infty \arrow[r, "\langle hd{,}tail\rangle"'] & \{0,1,2\} \times \{0,1,2\}^\infty &
\end{tikzcd}
$$

A commutative diagram is something that looks like

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\ C \arrow[r, "k"'] & D \end{tikzcd}$$

Most of the time these are going to be sets and functions.
(But we'll also need to be more general.)

And in that case, to say the diagram "commutes" means that if we start with an element $a \in A$ and "walk around" both ways, we get the same thing.

To say that the diagram

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, "h"'] & B \arrow[d, "g"] \\ C \arrow[r, "k"'] & D \end{tikzcd}$$

commutes means that for all $a \in A$, $g(f(a)) = k(h(a))$.
More abstractly, $g \cdot f = k \cdot h$.

I'm in the middle of explaining the diagram

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \{0,1,2\} \times X \arrow[d, "id_{\{0,1,2\}} \times e^\dagger"] \\ \{0,1,2\}^\infty \arrow[r, "\langle hd{,}tail \rangle"'] & \{0,1,2\} \times \{0,1,2\}^\infty \end{tikzcd}$$

The hard part is the function $id_{0,1,2} \times e^\dagger$.

For this, I will need some general notation on products.

If $f : C \to A$ and $g : C \to B$, then we get a new function

$$\langle f, g \rangle : C \to A \times B.$$

It is defined by

$$\langle f, g \rangle(c) \quad = \quad (f(c), g(c))$$

If $f : C \to A$ and $g : C \to B$, then we get a new function

$$\langle f, g\rangle : C \to A \times B.$$

It is defined by

$$\langle f, g\rangle(c) \quad = \quad (f(c), g(c))$$

The product set $A \times B$ itself comes with projections

$$A \xleftarrow{\pi_A} A \times B \xrightarrow{\pi_B} B$$

And then the diagram below commutes:

$$\begin{tikzcd} & C \arrow[dl, "f"'] \arrow[d, "\langle f,g\rangle"] \arrow[dr, "g"] & \\ A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B \end{tikzcd}$$

If $f : C \to A$ and $g : D \to B$, then we get a new function

$$f \times g : C \times D \to A \times B.$$

It is defined by

$$(f \times g)(c, d) \quad = \quad (f(c), g(d))$$

Note the difference between the notations $\langle f, g \rangle$ and $f \times g$.

They must be related, but how?

If $f : C \to A$ and $g : D \to B$, then we get a new function

$$f \times g : C \times D \to A \times B.$$

It is defined by

$$(f \times g)(c, d) \quad = \quad (f(c), g(d))$$

Note the difference between the notations $\langle f, g \rangle$ and $f \times g$.

They must be related, but how?

The $f \times g$ notation is a special case of pairing:

$$\begin{tikzcd} & C \times D \arrow[dl, "f \cdot \pi_C"'] \arrow[d, "\langle f \cdot \pi_C, g \cdot \pi_D \rangle"] \arrow[dr, "g \cdot \pi_D"] & \\ A & A \times B \arrow[l, "\pi_A"] \arrow[r, "\pi_B"'] & B \end{tikzcd}$$

So that

$$f \times g \quad = \quad \langle f \cdot \pi_C, g \cdot \pi_D \rangle.$$

Let's get back to the diagram

$$\begin{tikzcd}
X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \{0,1,2\} \times X \arrow[d, "id_{\{0,1,2\}} \times e^\dagger"] \\
\{0,1,2\}^\infty \arrow[r, "\langle hd, tail \rangle"'] & \{0,1,2\} \times \{0,1,2\}^\infty
\end{tikzcd}$$

Now we know about the function $id_{0,1,2} \times e^\dagger$.
$id_{0,1,2}$ is the identity function on $\{0,1,2\}$.
The definitions that we have seen tell us
that for $i \in \{0,1,2\}$ and $w \in X$,

$$(id_{0,1,2} \times e^\dagger)(i,w) \quad = \quad (i, e^\dagger(w)).$$

frame[shrink]
Let's get back to the diagram

$$\begin{tikzcd}
X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \{0,1,2\} \times X \arrow[d, "id_{\{0,1,2\}} \times e^\dagger"] \\
\{0,1,2\}^\infty \arrow[r, "\langle hd,tail \rangle"'] & \{0,1,2\} \times \{0,1,2\}^\infty
\end{tikzcd}$$

Recall that $X = \{x, y, z\}$ and that

$$\begin{array}{lcl}
e(x) & = & \langle 0, y\rangle \\
e(y) & = & \langle 1, z\rangle \\
e(z) & = & \langle 2, x\rangle
\end{array}
\qquad
\begin{array}{lcl}
e^\dagger(x) & = & (0,1,2,0,1,2,\ldots) \\
e^\dagger(y) & = & (1,2,0,1,2,0,\ldots) \\
e^\dagger(z) & = & (2,0,1,2,0,1\ldots)
\end{array}$$

We'll check that the diagram really does commute,

Let's start with y, for example, as a "random" element of X.

Across the top, we get $\langle 1, z\rangle$.
Then going down, we get $\langle 1, (2,0,1,2,0,1\ldots)\rangle$.

But starting again with y and going down, we get $(1, 2, 0, 1, 2, 0, \ldots)$.
And the head of this stream is 1; the tail is $(2, 0, 1, 2, \ldots)$.

So it really does commute!

In fact, we can verbalize what it means to say that our diagram

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \{0,1,2\} \times X \arrow[d, "id_{\{0,1,2\}} \times e^\dagger"] \\ \{0,1,2\}^\infty \arrow[r, "\langle hd{,}tail \rangle"'] & \{0,1,2\} \times \{0,1,2\}^\infty \end{tikzcd}$$

commutes.

For all $w \in X$, if $e(w) = \langle i, v \rangle$, then
$e^\dagger(w)$ is a stream whose head is i, and whose tail is
$e^\dagger(v)$.

On Tuedsay, we also saw decorations of graphs.

I want to change this point a little, and talk about systems of equations for sets.

(These are practically the same thing.)

Suppose we want sets x, y, and z so that

$$\begin{array}{lcl} x & = & \{y, z\} \\ y & = & \emptyset \\ z & = & \{x, z\} \end{array}$$

(This is impossible with ordinary sets, but possible with AFA.)

Again, we have

$$\begin{aligned} x &= \{y, z\} \\ y &= \emptyset \\ z &= \{x, z\} \end{aligned}$$

Let $X = \{x, y, z\}$.

We construe our system as a function $e : X \to \mathcal{P}(X)$.

Then what we want is a function $e^\dagger : X \to \text{Sets}$
with a certain property that we now want to spell out.

First, in words:

For all $w \in X$, $e^\dagger(w)$ is the set of
values $e^\dagger(v)$, where v ranges over $e(w)$.

For all $w \in X$, $e^\dagger(w)$ is the set of values $e^\dagger(v)$, where v ranges over $e(w)$.

$$\begin{tikzcd} X \arrow[rr, "e"] \arrow[dr, "e^\dagger"'] & & \mathcal{P}X \arrow[dl, "\mathcal{P}e^\dagger"] \\ & \text{Sets} & \end{tikzcd}$$

Here $\mathcal{P}e^\dagger(w) = \{e^\dagger(v) : v \in w\}$.

The point is that the commutativity of this diagram expresses what we want.

To make the diagram for sets look more like the one for streams,
we need to do a little more.

If $f : A \to B$, we get a new function $\mathcal{P}f : \mathcal{P}A \to \mathcal{P}B$ defined as follows. For a subset $s \subseteq A$,

$$\mathcal{P}f(s) \quad = \quad \{f(a) : a \in s\}.$$

(This is also written $f[s]$.)

Further, let's write V for the class of all sets,
and $\mathcal{P}V$ for the set of sub*sets* of V.

Now every set is a set of sets, and vice-versa.
So $\mathcal{P}V = V$.

To make the diagram for sets look more like the one for streams,
we need to do a little more.

If $f : A \to B$, we get a new function $\mathcal{P}f : \mathcal{P}A \to \mathcal{P}B$ defined as follows. For a subset $s \subseteq A$,

$$\mathcal{P}f(s) \quad = \quad \{f(a) : a \in s\}.$$

(This is also written $f[s]$.)

Further, let's write V for the class of all sets,
and $\mathcal{P}V$ for the set of sub*sets* of V.

Now every set is a set of sets, and vice-versa.
So $\mathcal{P}V = V$.

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \mathcal{P}X \arrow[d, "\mathcal{P}e^\dagger"] \\ V \arrow[r, "id"'] & \mathcal{P}V \end{tikzcd}$$

To make the diagram for sets look more like the one for streams,
we need to do a little more.

If $f : A \to B$, we get a new function $\mathcal{P}f : \mathcal{P}A \to \mathcal{P}B$ defined as follows. For a subset $s \subseteq A$,

$$\mathcal{P}f(s) \quad = \quad \{f(a) : a \in s\}.$$

(This is also written $f[s]$.)

Further, let's write V for the class of all sets,
and $\mathcal{P}V$ for the set of sub*sets* of V.

Now every set is a set of sets, and vice-versa.
So $\mathcal{P}V = V$.

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \mathcal{P}X \arrow[d, "\mathcal{P}e^\dagger"] \\ V \arrow[r, "id"'] & \mathcal{P}V \end{tikzcd}$$

The reason for all these diagrams is that they enable us see the same kind of pattern coming up again and again.
We want an overall language to talk about it.

We have seen:

streams	$e : X \to A \times X$
sets	$e : X \to \mathcal{P}X$

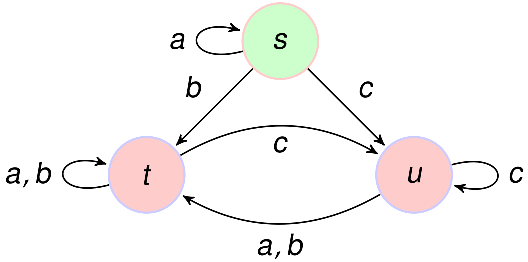
Let's think of $A \times X$ and $\mathcal{P}X$ as cooked versions of X.

So the kind of systems that we have seen are

functions from a raw object to a cooked version of it

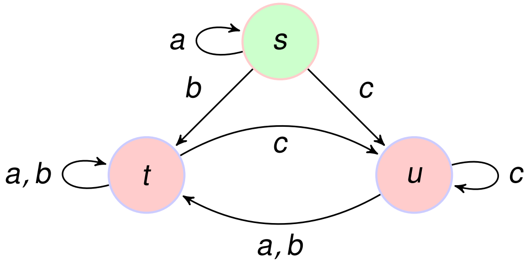
Later we'll call this a coalgebra.

Here is a deterministic automaton



The set of states is $S = \{s, t, u\}$.
We have one accepting state (in green).
The input alphabet is $A = \{a, b, c\}$.

We have a transition function $t : S \times A \to S$,
and also an output function $o : S \to 2$.
(Here $2 = \{0, 1\}$, and $o(s) = 1$ iff s is accepting.)



For all states s, the empty word ε is accepted at s if $acc(s) = 1$.

If w is a word and a an alphabet symbol, then

aw is accepted at s iff w is accepted at $t(s, a)$

So far a deterministic automaton on $\{a, b, c\}$ is

$$(S, s, acc),$$

where S is a set,

$$s : S \times A \to S,$$

and

$$acc : S \to 2.$$

So far a deterministic automaton on $\{a, b, c\}$ is

$$(S, s, acc),$$

where S is a set,

$$s : S \times A \to S,$$

and

$$acc : S \to 2.$$

We can curry s to get $\widehat{s} : S \to S^A$.

We also use pairing

$$\langle \widehat{s} \times acc \rangle : S \to 2 \times S^A.$$

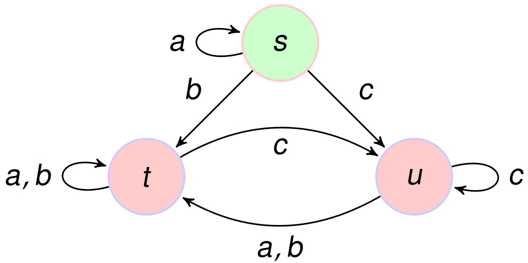
To match our earlier usage, we write e for $\langle \widehat{s} \times acc \rangle$.

We have another instance of

$$\text{raw} \to \text{cooked}$$

namely

$$e : S \to 2 \times S^A.$$



We have re-packaged the picture into a function

$$e : S \to 2 \times S^A$$

It is

$$\begin{array}{rcl} e(s) & = & (1, \{(a, s), (b, t), (c, u)\}) \\ e(t) & = & (0, \{(a, t), (b, t), (c, u)\}) \\ e(u) & = & (0, \{(a, t), (b, t), (c, u)\}) \end{array}$$

A^* is the set of finite words on A, including the empty word ϵ.

$\mathcal{L} = \mathcal{P}(A^*)$ is the set of languages X on A.

We want to think of language acceptance in the same way as we have seen for streams and sets.

$$
\begin{array}{c}
\text{syntax} \\
\big\downarrow \\
\text{semantics}
\end{array}
\qquad\qquad
\begin{tikzcd}
S \arrow[r, "e"] \arrow[d, "e^{\dagger}"'] & 2 \times S^A \arrow[d, "id_2 \times (e^{\dagger})^A"] \\
\text{languages} \arrow[r, "\ell"'] & 2 \times \text{languages}^A
\end{tikzcd}
$$

But this needs an explanation!

We want to think of language acceptance in the same way as we have seen for streams and sets.

$$
\begin{tikzcd}
\text{syntax} \arrow[d] & & S \arrow[r, "e"] \arrow[d, "e^\dagger"'] & 2 \times S^A \arrow[d, "id_2 \times (e^\dagger)^A"] \\
\text{semantics} & & \text{languages} \arrow[r, "\ell"'] & 2 \times \text{languages}^A
\end{tikzcd}
$$

But this needs an explanation!
Let's write $\mathcal{L}$ for the set of all languages on A.
(This is just $\mathcal{P}(A^*)$.)
We make $\mathcal{L}$ into an automaton (!) in our cooked sense by

$$\ell : \mathcal{L} \to 2 \times \mathcal{L}^A.$$

where

$$\ell(X) = (1 \text{ iff } \epsilon \in X, a \mapsto \{w : aw \in X\})$$

To explain the map $(e^\dagger)^A$,
here is a general definition.

If $f : X \to Y$, then $f^A : X^A \to Y^A$ is given by

$$g : A \to X \quad \mapsto \quad f \cdot g.$$

Taking $f : X \to Y$ to $f^A : X^A \to Y^A$

To explain the map $(e^\dagger)^A$,
here is a general definition.

If $f : X \to Y$, then $f^A : X^A \to Y^A$ is given by

$$g : A \to X \quad \mapsto \quad f \cdot g.$$

Now we understand

$$\begin{array}{c} \text{syntax} \\ \text{language acceptance} \big\downarrow \\ \text{semantics} \end{array}$$

$$\begin{tikzcd} S \arrow[r, "e"] \arrow[d, "e^\dagger"'] & 2 \times S^A \arrow[d, "id_2 \times (e^\dagger)^A"] \\ \mathcal{L} \arrow[r, "\ell"'] & 2 \times \mathcal{L}^A \end{tikzcd}$$

$$\begin{tikzcd} \text{syntax} \arrow[d, "\text{language acceptance}"'] \\ \text{semantics} \end{tikzcd} \qquad \begin{tikzcd} S \arrow[r, "e"] \arrow[d, "e^\dagger"'] & 2 \times S^A \arrow[d, "id_2 \times (e^\dagger)^A"] \\ \mathcal{L} \arrow[r, "\ell"'] & 2 \times \mathcal{L}^A \end{tikzcd}$$

For all states s, language accepted at s has two features:

- it contains the empty word iff s is an accepting state; that is, if $\pi_2(e(s)) = 1$.
- for all words w and all a, it contains aw iff w is in the language accepted at $\pi_{S^A}(e(s))(a)$.

A category C consists of

1. objects c, d, $\ldots$
   The collection of objects might be a proper class.
2. For each two objects c and d, a collection of morphisms f, g, $\ldots$.
   with emphdomain c and codomain d.
   We write $f : c \to d$ to say that f is such a morphism.
3. identity morphisms id_a for all objects.
4. a composition operation:
   if $f : a \to b$ and $g : b \to c$, then $g \cdot f : a \to c$.

subject to the requirements that

- Composition is associative.
- If $f : a \to b$, then $id_b \cdot f = f = f \cdot id_a$.

The objects of Set are sets (all of them).

A morphism from X to Y is a function from X to Y.

The identity morphism id_a for a set a is the identity function on a.

The composition operation of morphisms
is the one we know from sets.

The objects of Pos are posets $(P, \leq)$.
(That is, $\leq$ is reflexive, transitive, and anti-symmetric.)

A morphism from P to Q is a monotone function f from X to Y.
(If $p \leq p'$ in P, then $f(p) \leq f(p')$ in Q.

The identity morphism id_a for a poset P is the identity function on the underlying set P.

The composition operation of morphisms
is the one we know from sets.

Third example: every poset is itself a category

Let $(P, \leq)$ be a poset.

We consider P to be a poset by taking its elements as the objects.
The morphisms $f : p \to q$ are just the pairs (p, q) with $p \leq q$.

Unlike sets, between any two objects there is either 0 or 1 morphisms.

The morphism id_p is (p, p).

$(q, r) \cdot (p, q) = (p, r)$.

MS is the category of metric spaces (X, d),
with $d : X \times X \to [0, 1]$ satisfying the metric properties:

- $d(x, x) = 0$
- If $d(x, y) = 0$, then $x = y$.
- $d(x, y) = d(y, x)$.
- $d(x, z) \leq d(x, y) + d(y, z)$.

A morphisms from (X, d) to (Y, d') is a *non-expanding function* $f : X \to Y$.
This means that

$$d'(f(x), f(y)) \quad \leq \quad d(x, y)$$

The category CMS is the same, but we use *complete* metric spaces.

Objects are $(X, \top, \bot)$, where X is a set and $\top$ and $\bot$ are elements of X.
We require $\bot \neq \top$.

A morphism $f : (X, \top, \bot) \to (Y, \top, \bot)$ is a function $f : X \to Y$ such that $f(\top) = \top$ and $f(\bot) = \bot$.

The rest of the structure is as in Set, or any other concrete category.

Let C be a category.

An initial object is an object c such that
for all d, there is unique morphism $f : c \to d$.

An final object is an object c such that
for all d, there is unique morphism $f : d \to c$.

Let C be a category.

An initial object is an object c such that
for all d, there is unique morphism $f : c \to d$.

An final object is an object c such that
for all d, there is unique morphism $f : d \to c$.

In Set, $\emptyset$ is initial, and every singleton $\{x\}$ is final.

(Recall that the empty function is a function from $\emptyset$ to any set.
Also, there is no function from $\{x\}$ to $\emptyset$.
Finally, if Y is non-empty, there is a unique function $f : Y \to \{x\}$.

Note that there is more than one final object, but they are
all "isomorphic" in a sense that we'll explicate.

Let C be a category.

An initial object is an object c such that
for all d, there is unique morphism $f : c \to d$.

An final object is an object c such that
for all d, there is unique morphism $f : d \to c$.

In Pos, the empty poset is initial, and the one-point poset $\{x\}$ is final.

The same basically works for MS.

Let C be a category.

An initial object is an object c such that
for all d, there is unique morphism $f : c \to d$.

An final object is an object c such that
for all d, there is unique morphism $f : d \to c$.

In a poset P, an initial object would be a minimal element,
and a final object would be a maximal element.

(These may or may not exist.)

Let C be a category.

An initial object is an object c such that
for all d, there is unique morphism $f : c \to d$.

An final object is an object c such that
for all d, there is unique morphism $f : d \to c$.

In BiP, the initial object is any object
based on a two-element set: $(\{\top, \bot\}, \top, \bot)$.

This is also a final object.

We often write 0 for an initial object and 1 for a final one.

So in BiP, $0 = 1$.

Let C and D be categories.
A functor from C to D consists of

- An object mapping $a \mapsto Fa$, taking objects of C to objects of D.
- A morphism mapping $f \mapsto Ff$, taking morphisms of C to morphisms of D.

such that

- If $f : a \to b$, then $Ff : Fa \to Fb$.
- $Fid_a = id_{Fa}$.
- $F(f \cdot g) = Ff \cdot Fg$.

A functor from C to itself is an endofunctor.
(In our terminology, an endofunctor is a "recipe for cooking".

On Set, we have seen some endofunctors already:

- For any set A, $FX = A \times X$.
  If $f : X \to Y$, then $Ff : A \times X \to A \times Y$ is
  $$Ff(a,x) \quad = \quad (a, fx).$$
  Aother way to say this:
  $$Ff(w) \quad = \quad \langle \mathrm{fst}(w), f(\mathrm{snd}(w) \rangle.$$
  for all $w \in A \times X$.
- The power set functor $\mathcal{P}$.
  Here $\mathcal{P}(X)$ is the set of subsets of X.
  If $f : X \to Y$, then $\mathcal{P}f : \mathcal{P}(X) \to \mathcal{P}(Y)$ works by taking direct images.
- $F(X) = 2 \times X^A$, where $2 = \{0, 1\}$, and A is a fixed set.
  If $f : X \to Y$, then $Ff : 2 \times X^A \to 2 \times Y^A$ is given by
  $$Ff(i, g) \quad = \quad (i, h \mapsto h \cdot g).$$

- upclosed : Pos $\to$ Pos
  taking a poset P to the set of upward closed subsets, under $\subseteq$.

  If $f : (P, \leq) to (Q, \leq)$, you might like to think about how Ff should work.

- U : Pos $\to$ Set taking a poset to its underlying set.

- On a particular poset P, a functor $F : P \to P$ is the same thing as a monotone function $F : P \to P$.

  In fact the monotonicity property of an endofunction corresponds to the functoriality property $F(f \cdot g) = Ff \cdot Fg$

Let $(X, \top, \bot)$ be a bipointed set.
We define $X \oplus X$ to be

- two separate copies of X which I'll write as X_1 and X_2.
- The $\bot$ of $X \oplus X$ is the $\bot$ of X_1.
- The $\top$ of $X \oplus X$ is the $\top$ of X_2.
- The $\top$ of X_1 is identified with the $\bot$ of X_2.
  (This is called the midpoint of $X \oplus X$.)

We get a functor $F : \mathrm{BiP} \to \mathrm{BiP}$ by

$$FX \quad = \quad X \oplus X$$

If $f : X \to Y$ is a BiP morphism, then $Ff : X \oplus X \to Y \oplus Y$ works in the obvious way, preserving the midpoint.

Let d be an object of D.
We get $F : C \to D$, the constant functor d by:
$Fc = d$,
$Ff = id_d$.

The composition of functors is again a functor.

Let $F : C \to C$ be an endofunctor.
An algebra for F is a pair (A, a), where $a : FA \to A$ in C.
The leading example is when F is a signature functor, say F_Σ.
Then an F-algebra is a set A together with interpretations of the symbols in Σ.
A morphism from (A, a) to (B, b) is $h : A \to B$ such that

$$\begin{tikzcd} FA \arrow[r, "a"] \arrow[d, "Fh"'] & A \arrow[d, "h"] \\ FB \arrow[r, "b"'] & B \end{tikzcd}$$

commutes.

So now we have a category of algebras for an endofunctor.

Let $F : C \to C$ be an endofunctor.
A coalgebra for F is a pair (A, a), where $a : A \to FA$ in C.
We have already seen many examples!

A morphism from (A, a) to (B, b) is $h : A \to B$ such that

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "h"'] & FA \arrow[d, "Fh"] \\ B \arrow[r, "b"'] & FB \end{tikzcd}$$

commutes.

So now we have a category of coalgebras for an endofunctor.

Let $(A, a : FA \to A)$ and $(B, b : FB \to B)$ be algebras.
A morphism in the algebra category of F is $f : A \to B$ in the same underlying category so that

$$\begin{tikzcd} FA \arrow[r, "a"] \arrow[d, "Ff"'] & A \arrow[d, "f"] \\ FB \arrow[r, "b"'] & B \end{tikzcd}$$

commutes.

Let $(A, a : A \to FA)$ and $(B, b : B \to FB)$ be coalgebras.
A morphism in the coalgebra category of F is $f : A \to B$ in the same underlying category so that

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "f"'] & FA \arrow[d, "Ff"] \\ B \arrow[r, "b"'] & FB \end{tikzcd}$$

commutes.

An initial algebra is an initial object of the algebra category.
A final coalgebra is a final object of the coalgebra category.

initial algebra

$$\begin{tikzcd} FA \arrow[r, "a"] \arrow[d, "Ff"'] & A \arrow[d, "f"] \\ FB \arrow[r, "b"'] & B \end{tikzcd}$$

$$\begin{tikzcd} A \arrow[r, "a"] \arrow[d, "f"'] & FA \arrow[d, "Ff"] \\ B \arrow[r, "b"'] & FB \end{tikzcd}$$

final coalgebra

(One could also consider final algebras and initial coalgebras, but they turn out to be much less interesting.)

Recall that an endofunctor on a poset $(P, \leq)$
is a monotone function $f : P \to P$.
An algebra for f is some p such that $f(p) \leq p$.

A coalgebra for f is some p such that $p \leq f(p)$.

An initial algebra for f is some p such that

- $f(p) \leq p$.
- If $f(q) \leq q$, then $p \leq q$.

An final algebra for f is some p such that

- $p \leq f(p)$.
- If $q \leq f(q) \leq q$, then $q \leq p$.

It's good to check that these correspond to
least fixed points and greatest fixed points,
respectively.

Let $F(X) = 1 + X$ be the disjoint union
of a singleton set $1 = \{*\}$ and X.

F is a functor in the following way:
If $f : X \to Y$, then $Ff(*) = *$, and for $x \in X$, $Ff(x) = f(x)$.

An initial algebra on Set: the natural numbers

Let $F(X) = 1 + X$ be the disjoint union
of a singleton set $1 = \{*\}$ and X.

F is a functor in the following way:
If $f : X \to Y$, then $Ff(*) = *$, and for $x \in X$, $Ff(x) = f(x)$.

Let N be the set of natural numbers.

We have an algebra structure $t : F(N) \to N$ given by

$$\begin{array}{rcl} t(*) & = & 0 \\ t(n) & = & n+1 \end{array}$$

Let $F(X) = 1 + X$ be the disjoint union of a singleton set $1 = \{*\}$ and X.

F is a functor in the following way:
If $f : X \to Y$, then $Ff(*) = *$, and for $x \in X$, $Ff(x) = f(x)$.

RECURSION PRINCIPLE FOR N

For all sets X, all $x \in X$, all $f : X \to X$,
there is a unique $\varphi : N \to X$ so that

$$\begin{array}{lcl} \varphi(0) & = & x \\ \varphi(n+1) & = & f(\varphi(x)) \end{array}$$

Let $F(X) = 1 + X$ be the disjoint union of a singleton set $1 = \{*\}$ and X.

F is a functor in the following way:
If $f : X \to Y$, then $Ff(*) = *$, and for $x \in X$, $Ff(x) = f(x)$.

Recursion Principle for N

For all sets X, all $x \in X$, all $f : X \to X$, there is a unique $\varphi : N \to X$ so that

$$\begin{tikzcd} 1+N \arrow[r, "t"] \arrow[d, "1+\varphi"'] & N \arrow[d, "\varphi"] \\ 1+X \arrow[r, "f"'] & X \end{tikzcd}$$

commutes.

Initiality of N as an algebra for $1 + X$

For all sets A, all $a \in A$, and all $f : A \to A$,
there is a unique $\varphi : N \to A$ so that
$\varphi(0) = a$, and $\varphi(n+1) = f(\varphi(n))$ for all n.

The two principles are equivalent in set theory without Infinity.

Let $FX = N \times X$.
Stream systems are coalgebras of F, maps of the form $e : X \to FX$.
Even more, the solution $e^\dagger : X \to N^\infty$ would be a coalgebra morphism:

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & FX \arrow[d, "Fe^\dagger"] \\ N^\infty \arrow[r, "id"'] & FN^\infty \end{tikzcd}$$

The point is that for $x \in X$,

$$\begin{aligned} Fe^\dagger(e(x)) &= Fe^\dagger\langle \mathrm{fst}(e(x)), \mathrm{snd}(e(x))\rangle \\ &= \langle \mathrm{fst}(e(x)), e^\dagger(\mathrm{snd}(e(x)))\rangle \end{aligned}$$

We have seen this formulation before.

For any set X, $\mathcal{P}X$ is the set of subsets of X.
$\mathcal{P}$ extends to an endofunctor, taking

$$f : X \to Y$$

to

$$\mathcal{P}f : \mathcal{P}X \to \mathcal{P}Y$$

given by direct images: for $a \subseteq X$, $\mathcal{P}f(a) = f[a] = \{f(x) : x \in a\}$.
We similarly have functors such as the finite power set functor $\mathcal{P}_{fin}$.

By the way, a transitive set is a set $X \subseteq \mathcal{P}X$.
This is the same thing as a subcoalgebra $X \to V$

$$\begin{tikzcd} X \arrow[r, "i"] \arrow[d, "inclusion"'] & pow(X) \arrow[d, "\mathcal{P}(inclusion)"] \\ V \arrow[r, "="'] & \mathcal{P}(V) \end{tikzcd}$$

Let $FX = \mathcal{P}X$.
Set systems are coalgebras of F, maps of the form
$e : X \to \mathcal{P}X$.
Even more, the solution $e^\dagger : X \to V$ is practically a coalgebra morphism:

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "e^\dagger"'] & \mathcal{P}X \arrow[d, "\mathcal{P}e^\dagger"] \\ V \arrow[r, "id"'] & \mathcal{P}V \end{tikzcd}$$

The point is that for $x \in X$,

$$\begin{aligned} \mathcal{P}e^\dagger(e(x)) &= e^\dagger[e(x)] \\ &= \{e^\dagger(a) : a \in e(x)\} \end{aligned}$$

So $e^\dagger$ satisfies the same equation as a decoration from before.

set with algebraic operations	set with transitions and observations
algebra for a functor	coalgebra for a functor
initial algebra	final coalgebra
useful in syntax	useful in semantics

Lemma (Lambek's Lemma)

Let C be a category, let $F : C \to C$ be a functor, and let (a, f) be an initial algebra for F. Then f is an isomorphism: there is a morphism $g : Fa \to a$ such that $g \cdot f = id_a$ and $f \cdot g = id_{Fa}$.

The same statement holds for final coalgebras of F.

Note first that (Fa, Ff) is an algebra for F. The square below commutes:

$$\begin{tikzcd} FFa \arrow[r, "Ff"] \arrow[d, "Ff"'] & Fa \arrow[d, "f"] \\ Fa \arrow[r, "f"'] & a \end{tikzcd}$$

By initiality, there is a morphism $g : a \to Fa$ so that the square on the top commutes:

$$\begin{tikzcd} Fa \arrow[r, "f"] \arrow[d, "Fg"'] & a \arrow[d, "g"] \\ FFa \arrow[r, "Ff"] \arrow[d, "Ff"'] & Fa \arrow[d, "f"] \\ Fa \arrow[r, "f"'] & a \end{tikzcd}$$

The bottom is obvious, the outside of the figure thus commutes.

By initiality, there is a morphism $g : a \to Fa$ so that the square on the top commutes:

$$\begin{tikzcd}
Fa \arrow[r, "f"] \arrow[d, "Fg"'] \arrow[dd, bend right, "F(f\cdot g)"'] & a \arrow[d, "g"] \arrow[dd, bend left, "f\cdot g"] \\
FFa \arrow[r, "Ff"] \arrow[d, "Ff"'] & Fa \arrow[d, "f"] \\
Fa \arrow[r, "f"'] & a
\end{tikzcd}$$

By initiality, we see that $f \cdot g = id_a$.
And then from that top square again,

$$g \cdot f = Ff \cdot Fg = F(f \cdot g) = Fid_a = id_{Fa}.$$

This completes the proof.

There are no initial algebras or final coalgebras of $\mathcal{P}$

An isomorphism in Set is a bijection.
And there are no maps from any set onto its power set (Cantor's Theorem).
Together with Lambek's Lemma, we see that $\mathcal{P}$ on Set has no initial algebra and no final coalgebra.
To get around this, one either

1. moves from Set to the category Class.
2. moves from $\mathcal{P}$ to $\mathcal{P}_\kappa$, the functor giving the subsets of a set of size $< \kappa$.

We'll generally go the second route, and in particular consider $\mathcal{P}_{\mathit{fin}}$.
It turns out that (H_κ, id) is an initial algebra of $\mathcal{P}_\kappa$.
We'll see the final coalgebra later.

For $\kappa = \omega$, we have the finite power set functor, and the final coalgebra can be described using

Let C be a category.
A coproduct of objects x and y, is an object $x + y$ with morphisms $\mathrm{inl} : x \to x + y$ and $\mathrm{inr} : y \to x + y$ meeting the following condition:
if $f : x \to z$ and $g : y \to z$, then there is a unique $[f, g] : x + y \to z$ such that $[f, g] \cdot \mathrm{inl} = f$ and $[f, g] \cdot \mathrm{inr} = g$.
Technically, the coproduct of x and y is the triple $(x + y, \mathrm{inl}, \mathrm{inr})$.
Usually there's no need to add the names of the objects to the coproduct maps inl and inr.
But if we would need to, we could write

$$\mathrm{inl}_{a,a+b} : a \to a + b,$$

and similarly for $\mathrm{inr}_{b,a+b}$.

If $f : a \to b$ and $g : c \to d$, then we have

$$f + g : a + c \to b + d$$

given by

$$f + g = [\mathrm{inl} \cdot f, \mathrm{inr} \cdot g].$$

In pictures,

$$\begin{tikzcd}
a \arrow[r, "f"] & c \arrow[dr, "\mathrm{inl}"] & \\
a+b \arrow[rr, "{[\mathrm{inl}\cdot f, \mathrm{inr}\cdot g]}"] & & c+d \\
b \arrow[r, "g"'] & d \arrow[ur, "\mathrm{inr}"'] &
\end{tikzcd}$$

C has coproducts if every two objects have a coproduct.
A category with coproducts is a tuple $(C, +)$, where C is a category and $+$ is a coproduct operation on C, giving for each x and y the triple $(x + y, \mathrm{inl}_{x,x+y}, \mathrm{inr}_{y,x+y})$.
Set has coproducts in the following way: let

$$x + y \quad = \quad (x \times \{0\}) \cup (y \times \{1\}),$$

let $\mathrm{inl}(a) = (a, 0)$ for $a \in x$, and let $\mathrm{inr}(b) = (b, 1)$ for $b \in y$.

Let C be a category and $(D, +)$ be a category with coproducts.
If $F : C \to D$ and $G : C \to D$, we define

$$F + G : C \to D$$

by $(F + G)a = Fa + Ga$,
and if $f : a \to b$, then

$$(F + G)f : Fa + Ga \to Fb + Gb$$

is given by

$$(F + G)f \quad = \quad Ff + Gf$$

A discrete measure on a set A is a function $\mu : A \to [0, 1]$ such that

1. μ has *finite support*: $\{a \in A \mid \mu(a) > 0\}$ is finite.
2. $\sum_{a \in A} \mu(a) = 1$.

$\mathcal{D}(A)$ is the set of discrete measures on A.
We make $\mathcal{D}$ into a functor by setting,
for $f : A \to B$, $\mathcal{D}f(\mu)(b) = \mu(f^{-1}(b))$;
this is $\sum\{\mu(a) : f(a) = b\}$.
(As usual, we extend discrete measures on A to functions on $\mathcal{P}(A)$ by summing.)

In a category C, an object c is called initial
if for every object a, there is a unique $! : c \to a$.
c is terminal, or final
if for every object a, there is a unique $! : a \to c$.
In Set, the final objects are exactly the singletons.

In a category C, an object c is called initial
if for every object a, there is a unique $! : c \to a$.
In Set, $\emptyset$ is initial.
For every a, the empty function is the unique function from $\emptyset$ to a.
c is terminal, or final
if for every object a, there is a unique $! : a \to c$.
In Set, the final objects are exactly the singletons.

In a category C, an object c is called initial
if for every object a, there is a unique $! : c \to a$.

---

c is terminal, or final
if for every object a, there is a unique $! : a \to c$.
In Set, the final objects are exactly the singletons.

Examples of coalgebras for different functors

Let $F : C \to D$ be a functor between two categories,
and let $G : C \to D$ also be a functor between the same two.
Then a natural transformation from F to G is a family
η of morphisms of D indexed by objects of C, in particular each
η_x is a morphism in D from Fx to Gx.
The requirement on η is that for each morphism in C of the
form $f : x \to y$, the square below commutes:

$$\begin{tikzcd} x \arrow[d, "f"] \\ y \end{tikzcd} \qquad \begin{tikzcd} Fx \arrow[r, "\eta_x"] \arrow[d, "Ff"'] & Gx \arrow[d, "Gf"] \\ Fy \arrow[r, "\eta_y"'] & Gy \end{tikzcd}$$

In symbols, $Gf \cdot \eta_x = \eta_y \cdot Ff$.
One writes $\eta : F \to G$.
For each object x of C, η_x is called the component of η at x.

Suppose first that $\eta : F \to G$, and let $H : B \to C$ be another functor.
Then $F \cdot H : B \to D$ and $G \cdot H : B \to D$.
We get a natural transformation called ηH from $F \cdot H$ to $G \cdot H$ by

$$\eta_H b \quad = \quad \eta_{Hb}.$$

Now let $H : D \to E$. So now $H \cdot F$ and $H \cdot G$ are functors from C to E.
We get a natural transformation from $H \cdot F$ to $H \cdot G$, this time called $H\eta$, by

$$(H\eta)_x \quad = \quad H\eta_x.$$

That is, we apply the functor H to the morphism η_x. The verification of naturality is a little different: we apply H throughout.

Suppose first that $\eta : F \to G$, and let $H : B \to C$ be another functor.
Then $F \cdot H : B \to D$ and $G \cdot H : B \to D$.
We get a natural transformation called ηH from $F \cdot H$ to $G \cdot H$ by

$$\eta_H b \quad = \quad \eta_{Hb}.$$

To check that this is indeed natural, let $f : x \to y$ be a morphism in B. Then for each x in B
$(\eta H)_x : (F \cdot H)x \to G \cdot H)x$. And we have the diagram

$$\begin{tikzcd}
(F \cdot H)x \arrow[r, "(\eta H)_x"] \arrow[d, "(F \cdot H)f"'] & (G \cdot H)x \arrow[d, "(G \cdot H)f"] \\
(F \cdot H)y \arrow[r, "(\eta H)_y"'] & (G \cdot H)y
\end{tikzcd}$$

This is literally the same as

$$F(Hx) \xrightarrow{\eta_{Hx}} G(Hx)$$

If $\eta : F \to G$ and $\mu : F \to H$, then we get a natural transformation $\mu \cdot \eta : F \to H$ by $(\mu \cdot \eta)_x = \mu_x \cdot \eta_x$. The verification of naturality is easy.
Finally, suppose that $F, G : D \to E$ and $H, K : C \to D$, and let $\eta : F \to G$ and $\mu : H \to K$. We get a natural transformation $\mu * \eta : F \cdot H \to K \cdot G$ by

$$\begin{tikzcd}
F \cdot H \arrow[r, "F\mu"] \arrow[d, "\eta H"'] \arrow[dr, "\mu * \eta"] & F \cdot K \arrow[d, "\eta K"] \\
G \cdot H \arrow[r, "G\mu"'] & G \cdot K
\end{tikzcd}$$

That is, we claim that the outside of the figure commutes, and then we define $\mu * \eta$ to be the composite in either direction; this will be a natural transformation by the three constructions which we have already seen. But for each object x of C the square above is a naturality square for η, applied to the morphsim $\mu_x : Hx \to Kx$.

Definition

A functor $F : C \to D$ preserves pullbacks if the image of every pullback square is a pullback square.

Lemma

Concerning preservation of pullbacks:

1. *Constant functors preserve pullbacks.*
2. *If F and G preserve pullbacks, so do $F + G$, $F \times G$, and $F \cdot G$.*
3. *$\mathcal{P}$, $\mathcal{P}_{\mathrm{fin}}$, and $\mathcal{D}$ do not preserve pullbacks.*

LEMMA

Every functor F on Set *preserves all surjective maps.*

PROOF.

Suppose $g : X \to Y$ is surjective.
Let $h : Y \to X$ be such that $g \cdot h = id_Y$.
Then $Fg \cdot Fh = id_{FY}$, and so Fg must be surjective. $\square$

PROPOSITION

If $FX = \emptyset$ for some $X \neq \emptyset$, then F is the constant functor $\emptyset$.

PROOF.

Let Y be any set. Then there is $f : Y \to X$; f could be a constant, for example. And now $Ff : FY \to \emptyset$. So FY must be empty also, since there are no maps from a non-empty set to $\emptyset$. $\square$