Coalgebra: Basic Concepts

Paul Blain Levy

University of Birmingham

1 Transition Systems

Integer Ltd. makes integer I/O machines, which have a button and a display. You press the button and it prints an integer. You press the button again and it prints another integer.

A machine has a variety of internal states. When you press the button, it's the current state that determines what integer gets printed, and what the new state will be (it could be a different state, or it could be the same state).

A machine is described by

- a set X (the set of states)
- a function $\zeta : X \longrightarrow \mathbb{Z} \times X$ (what happens when you press the button)
- the current state $x_0 \in X$

Exercise 1. Machine number 392 has $\mathbb{Z} \times \mathbb{Z}$ as set of states. The behaviour function is $\zeta : \langle n, n' \rangle \mapsto \langle n + n', \langle n' + 1, n - 2 \rangle \rangle$. The current state is $\langle 4, 6 \rangle$. What is printed when you press the button three times?

A rival company Integer And Boolean Inc. makes machines with three buttons and a display. If you press the red button or the green button it prints an integer, but if you press the bright pink button and enter an integer on the keyboard it prints a boolean. Such a machine is described by a pair (X, ζ), where

- a set X (the set of states)
- a function $\zeta_{\mathsf{red}} : X \longrightarrow \mathbb{Z} \times X$ (what happens when you press the red button)
- a function $\zeta_{\mathsf{green}} : X \longrightarrow \mathbb{Z} \times X$ (what happens when you press the green button)

- a function $\zeta_{\mathsf{brightpink}} : X \longrightarrow \mathbb{B} \times X$ (what happens when you press the bright pink button).
- the current state $x_0 \in X$

Exercise 2. Machine number 25 has $\mathbb{Z} \times \mathbb{Z}$ as set of states. The behaviour functions are

$$\begin{aligned} \zeta_{\mathsf{red}} &: \langle n, n' \rangle \mapsto \langle n, \langle n'+1, n-2 \rangle \rangle \\ \zeta_{\mathsf{green}} &: \langle n, n' \rangle \mapsto \langle n'+1, \langle n+n', 2n' \rangle \rangle \\ \zeta_{\mathsf{brightpink}} &: \langle n, n' \rangle \mapsto \langle n > n', \langle n', n' \rangle \rangle \end{aligned}$$

The current state is $\langle 3, 7 \rangle$. What is printed when you press the red button, then the green button, then the bright pink button, then the red button again?

Another company Interactive Integer make machines with a keyboard and a display. If you enter an integer, it prints another integer. Such a machine is described by

- a set X (the set of states)
- a function $\zeta : \mathbb{Z} \times X \longrightarrow \mathbb{Z} \times X$
- the current state $x_0 \in X$.

Exercise 3. Machine number 40 has $\mathbb{Z} \times \mathbb{Z}$ as set of states. The behaviour function is given by

$$\zeta : \langle m, \langle n, n' \rangle \rangle \mapsto \langle m+n, \langle 2m+n', n-1 \rangle \rangle$$

The current state is $\langle 4, 4 \rangle$. What is printed when you enter 5, then 3, then 5 again?

A somewhat unsuccessful company Unreliable Integer makes machines with a button and a display. If you press the button it might print an integer or it might print one of three error messages:

CRASH
BANG
WALLOP

Then the button jams shut and remains so forever. Such a machine is described by

- a set X (the set of states)

– a function $X \longrightarrow \mathbb{Z}\times X+E$, where E is the set of error messages,
– the current state $x_0 \in X$.

Exercise 4. Machine number 6 has $\mathbb{Z} \times \mathbb{Z}$ as set of states. The behaviour function is described by

$$\zeta : \langle n, n' \rangle \mapsto \begin{cases} \mathsf{inl}\ \langle n+3, \langle n', 7 \rangle \rangle & \text{if } n' > 4 \\ \mathsf{inr}\ \mathrm{BANG} & \text{otherwise} \end{cases}$$

The current state is $\langle 3, 2 \rangle$. What is printed if you press the button twice?

A more popular company is Probabilistic Integer. If you press the button it consults some random data to decide what integer to print. The machine is described by

– a set X (the set of states)
– a function $\zeta : X \times (\mathbb{Z}\times X) \longrightarrow [0,1]$, where $\sum_{\langle n,y \rangle \in \mathbb{Z}\times X} \zeta(x, \langle n, y \rangle) = 1$ for each $x \in X$.
– the current state $x_0 \in X$.

A newcomer to the market is Nondeterministic Integer who make machines with a button and a display. If you press the button it prints an integer. But the behaviour doesn't just depend on the internal state, it also depends on a monkey hidden inside the machine. The machine is described by

– a set X (the set of states)
– a relation $r : X \longrightarrow\!\!\!\!\!\!\!+\;\; X$
– the current state $x_0 \in X$.

Exercise 5. Machine number 24 has set of states $\mathbb{Z} \times \mathbb{Z}$. The behaviour relation is described by

$$\langle n, n' \rangle \ r \ \langle m, \langle p, p' \rangle \rangle \overset{\text{def}}{\Leftrightarrow} m > n \text{ and } p = p' + n$$

The current state $\langle 2, 5 \rangle$ is. Describe one possible output if you press the button three times.

2 Coalgebras

These descriptions have more in common than appears at first sight.
A machine consists of a set X together with a function

- $X \longrightarrow \mathbb{Z} \times X$ (Integer Ltd.)
- $X \longrightarrow (\mathbb{Z} \times X) \times (\mathbb{Z} \times X) \times (\mathbb{B} \times X)$ (Integer And Boolean Inc.)
- $X \longrightarrow (\mathbb{Z} \times X)^{\mathbb{Z}}$ (Interactive Integer)
- $X \longrightarrow \mathbb{Z} \times X + E$ (Unreliable Integer)
- $X \longrightarrow DX$ (Probabilistic Integer), where DX is the set of discrete probability distributions on X.
- $X \longrightarrow \mathcal{P}^{>0}X$ (Nondeterministic Integer)

and a current state $x_0 \in X$.

Definition 1. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories. A* functor $F : \mathcal{C} \longrightarrow \mathcal{D}$ *associates*

- *to each $\mathcal{C}$-object X, a $\mathcal{D}$-object FX*
- *to each $\mathcal{C}$-morphism $X \xrightarrow{f} Y$, a $\mathcal{D}$-morhpism $FX \xrightarrow{Ff} FY$*

in such a way that

- *for every object X we have $F\mathsf{id}_X = \mathsf{id}_{FX}$*
- *for any morphisms $X \xrightarrow{f} Y \xrightarrow{g} Z$ we have $F(f;g) = Ff;Fg$.*

A *endofunctor* on a category $\mathcal{C}$ is a functor $F : \mathcal{C} \longrightarrow \mathcal{C}$.
For example, there's an endofunctor on **Set** that maps

- a set X to the set $\mathbb{Z} \times X$
- a function $X \xrightarrow{f} Y$ is mapped to the function $\mathbb{Z} \times X \xrightarrow{\mathbb{Z} \times f} \mathbb{Z} \times Y$ that sends $\langle n, x\rangle$ to $\langle n, f(x)\rangle$.

Typically we write a functor by saying only what it does to objects, but this is sloppy.

Definition 2. *Let $\mathcal{C}$ be a category and let F be an endofunctor on $\mathcal{C}$. A $\mathcal{C}$-*coalgebra *consists of*

- *a $\mathcal{C}$-object X*
- *a $\mathcal{C}$-morphism $\zeta : X \longrightarrow FX$.*

We call X the carrier *of the coalgebra and ζ the* behaviour *or* structure *of the coalgebra.*

For example, a machine made by Integer Ltd. is a $X \mapsto \mathbb{Z} \times X$ coalgebra. Only one thing is missing: a coalgebra does not have a current state. If F is an endofunctor on **Set**, we say that a *pointed* F-coalgebra is an F-coalgebra (X, ζ) together with a state $x_0 \in X$. In general a *pointed* set is a set X together with an element $x_0 \in X$.

What about the other machines? Each of these is given as a (pointed) coalgebra for a suitable endofunctor on **Set**.

- If F, G, H are endofunctors on **Set** then so is $X \mapsto FX \times GX \times HX$, with $X \xrightarrow{f} Y$ mapping to

$$FX \times GX \times HX \xrightarrow{Ff \times Gf \times Hf} FY \times GY \times HY$$

  that sends (a, b, c) to $((Ff)a, (Gf)b, (Hf)c)$, and so is $X \mapsto FX + GX + HX$.
- $X \mapsto X^{\mathbb{Z}}$ is an endofunctor, with $X \xrightarrow{f} Y$ mapping to

$$X^{\mathbb{Z}} \xrightarrow{f^{\mathbb{Z}}} Y^{\mathbb{Z}}$$

  that sends $(a_i)_{i \in I}$ to $(f(a_i))_{i \in \mathbb{Z}}$.
- $X \mapsto X + E$ is an endofunctor, with $X \xrightarrow{f} Y$ mapping to

$$X + E \xrightarrow{f+E} Y + E$$

  that sends $\mathsf{inl}\ x$ to $\mathsf{inl}\ f(x)$ and $\mathsf{inr}\ e$ to $\mathsf{inr}\ e$.
- The endofunctor D maps X to the set of discrete distributions on X is an endofunctor. A *discrete distribution* is a function $d : X \longrightarrow [0, 1]$ such that $\sum_{x \in x} d(x) = 1$. The function $X \xrightarrow{f} Y$ is mapped to $DX \xrightarrow{Df} DY$ that sends d to $y \mapsto \sum_{x \in f^{-1}(y)} d(x)$.
- The endofunctor $\mathcal{P}$ maps X to the set of subsets of X. A function $X \xrightarrow{f} Y$ is mapped to $\mathcal{P}X \xrightarrow{\mathcal{P}f} \mathcal{P}Y$ that sends U to $\{f(x) \mid x \in U\}$.

3 Subfunctors

Let F be an endofunctor on **Set**. A *subfunctor* G of F associates to each set X a subset GX of FX, in such a way that for any function $X \xrightarrow{f} Y$ and element $a \in GX$, we have $(Ff)a \in GY$. This enables us to define $GX \xrightarrow{Gf} GY$ to be Ff, so G is also an endofunctor on **Set**. If we have an F-coalgebra (X, ζ) we can ask: is it a G-coalgebra? In other words, is $\zeta(x) \in GX$ for all $x \in X$?

For example, $D_{\mathsf{fin}}X$ is the set of *finite distributions* on X, i.e. those $d \in DX$ such that the set $\{x \in X \mid d(x) > 0\}$ is finite. A D_{fin}-coalgebra is a special kind of probabilistic transition system.

Exercise 6. Which of these are subfunctors of $\mathcal{P}$?

- X maps to the set of nonempty subsets of X (Hint: yes)
- X maps to the set of finite subsets of X (Hint: yes)
- X maps to the set of subsets of X of size at most 3
- X maps to the set of finite subsets of even size
- X maps to the set of countable subsets of X.

(If you know about cardinals:) Give all the subfunctors of $\mathcal{P}$.

Thus we have *lively* transition systems and *finitely branching* transition systems.

4 The Category of Coalgebras

Of course we want to make coalgebras into a category.

Definition 3. *Let $\mathcal{C}$ be a category and let F be an endofunctor on $\mathcal{C}$. Let (X, ζ) and (Y, ξ) be F-coalgebras. A* F-coalgebra morphism *From (X, ζ) to (Y, ϕ) is a morphism $X \xrightarrow{f} Y$ such that*

$$\begin{tikzcd} X \arrow[r, "f"] \arrow[d, "\zeta"'] & Y \arrow[d, "\xi"] \\ FX \arrow[r, "Ff"'] & FY \end{tikzcd}$$

Now we get a category $\mathbf{Coalg}(F)$ whose objects are F-coalgebras and whose morphisms are F-coalgebra morphisms. Composition and identities are the same as in $\mathcal{C}$.

5 Active and Passive States

In the examples above, the states of the system are *passive*, waiting for input from outside. We could also consider a set of *active* states, that are executing a program and will then output. For example, a machine made by Interactive Input could be described as

- a set Y of active states
- a function $\xi : Y \longrightarrow \mathbb{Z} \times (Y^{\mathbb{Z}})$
- a current state $y_0 \in Y$.

Or it could be described as

- a set X of passive states
- a set Y of active states
- a function $\zeta : X \longrightarrow Y^{\mathbb{Z}}$
- a function $\xi : Y \longrightarrow \mathbb{Z} \times X$.

Each of these (leaving aside the current state) is a coalgebra. In the last case we use an endofunctor on $\mathbf{Set}^2$ that maps (X, Y) to $(Y^{\mathbb{Z}}, \mathbb{Z} \times X)$.

6 Traces and Final Coalgebras

If you buy a machine from Integer Ltd., i.e. a pointed $\mathbb{Z} \times -$ coalgebra (X, ζ, x), its full behaviour over time is described by an infinite sequence of integers. This is called the *infinite trace* of (X, ζ, x). Two machines with the same infinite trace are *trace equivalent*. They are equivalent for all practical purposes. Admittedly they have different states, but those states are internal so you cannot observe them.

So why bother with states at all? An employee at Integer Ltd. called Lazy Liszt makes a machine in which the set of states is $\mathbb{Z}^{\omega}$, the set of infinite sequences of integers. The behaviour function ζ maps a sequence s to $\langle s_0, s' \rangle$, where $s' : n \mapsto s(n+1)$. Thus the infinite trace of a state s is actually s.

After making this machine, Lazy Liszt is amazed to see it glowing, then he hears the chanting of an angelic choir. For the coalgebra he has made is final.

Definition 4. *Let $\mathcal{C}$ be a category. An object X is* final *(or* terminal*) if for every object Y there is a unique morphism from X to Y.*

A category can have more than one final object, but they are all isomorphic. More precisely, final objects are *unique up to unique isomorphism.*

A *final F-coalgebra* is a final object in the category $\mathbf{Coalg}(F)$. If we take any $\mathbb{Z} \times -$ coalgebra, there is a morphism to Lazy Liszt's coalgebra that maps each state to its infinite trace. **Exercise** Show that this is a coalgebra morphism, and that it's unique.

If you buy a machine from Integer and Boolean Inc., the full behaviour is defined by an infinite tree rather than an infinite list. To be more precise, consider *finite traces* such as the following:

I pressed the red button.
The machine printed 17.
I pressed the bright pink button.
The machine printed TRUE.
I pressed the red button.
The machine printed 42.

A finite trace is a sequence $a_0, b_0, a_1, b_1, \ldots, a_{n-1}, b_{n-1}$ where each a_i is a button and b_i is an appropriate response (integer if a_i is the red button or green button, boolean if a_i is the bright pink button).

Now an infinite tree is a set U of finite traces with the following properties:

- the *empty trace* $\varepsilon \in U$
- if s and t are traces and s is a *prefix* of t and $t \in U$ then $s \in U$.
- if $s \in U$ and a is a button then there is a unique appropriate response b to a such that $s + (a, b) \in U$.

Now if U is an infinite tree, then for each button a

- let b_a be the response such that $(a, b_a) \in U$
- let U_a be the set of all traces t such that $(a, b_a) + t \in U$.

The set of infinite trees, with the function ζ mapping U at a to (b_a, U_a), forms a coalgebra for

$$X \mapsto (\mathbb{Z} \times X) \times (\mathbb{Z} \times X) \times (\mathbb{B} \times X)$$

In fact it is a final coalgebra.