Category theory and coalgebra, lecture 1, second part: first examples

Jurriaan Rot

February 3, 2025

In the first part of the lecture (see the slides), we've already talked a bit about the history and general aims of the theory of coalgebra, so we won't do this here. Instead, we'll see some first, basic examples of coalgebras, and the important notion of homomorphisms between them. The main point of this part is to see, through examples, that coalgebras provide a very general and flexible description of state-based systems with 'observable behaviour'; and that homomorphisms allow to compare and reduce coalgebras. We'll even get our first view of the famous *final coalgebras* which got the whole theory going in the first place!

We start with a first, rough, very informal approximation of a coalgebra: a set of states X together with a function

$$f \colon X \to F(X)$$

where

- we call X the *set of states*
- we call f the *transition function*, or *coalgebra structure*;
- F is, for now, informal: a "construction on sets (and functions)" which determines the *type* of coalgebra. Later, we will make this into something meaningful by taking F to be a *functor*, but it's not assumed you already know what this is right now.

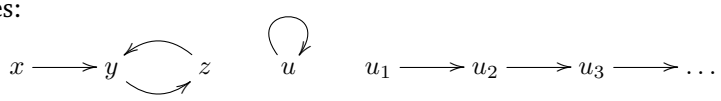
The actual definition of a coalgebra (and functors) is provided after introducing some of the basic notions of category theory.

1 Deterministic systems

Our first example is a "machine with a button": a machine which has no display, or other interface, only a button that can be pressed. It has some internal state, which we can't observe, and may change when we press the button. This exciting device is modelled as a pair (X, f) where f is a function

$$f \colon X \to X$$

We think of X as the set of internal states, and application of f as pressing the button and thereby changing the current state. We write $x \to_f y$ or simply $x \to y$ for $f(x) = y$. Then, we can make nice pictorial representations of such machines:



The "construction" F here is just the identity on sets: indeed, this is probably the most boring example of a coalgebra, and we'll soon leave it alone. But for now, it serves nicely to illustrate a few points. And there's something to say about it: Rutten [2] has a whole chapter about these systems!

For the moment, we call a coalgebra as above a *deterministic system*.

Example 1.1. In [1], Jacobs describes an example of such coalgebras $\mathsf{stat}\colon S \to S$, where stat ranges over the statements of some (imperative) programming language, and S models the set of program states, that is, assignments $\sigma\colon \mathsf{Var} \to \mathbb{Z}$ from variables to integers. For each statement stat, the coalgebra $\mathsf{stat}\colon S \to S$ describing how it changes the state. For instance, for $i \in \mathsf{Var}$, we could meaningfully define $[i := 5]\colon S \to S$ as $[i := 5](\sigma)(x) = \begin{cases} 5 & \text{if } x = i \\ \sigma(x) & \text{otherwise} \end{cases}$.

Homomorphisms are central in the theory of coalgebras, to compare and reduce systems. They are maps from one coalgebra to another which "preserve observations/behaviour" in a manner prescribed by the type F. Like coalgebras themselves, they will be properly defined later on.

In our example of deterministic systems, a homomorphism from $f\colon X \to X$ to $g\colon Y \to Y$ is a function $h\colon X \to Y$ such that $\forall x \in X. h(f(x)) = g(h(x))$, also written as $h \circ f = g \circ h$, also written by saying that the 'following diagram commutes' (this is terminology we'll see a lot!):

$$\begin{array}{ccc} X & \xrightarrow{h} & Y \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ X & \xrightarrow[h]{} & Y \end{array}$$

With the arrow notation of before, this means that for all $x, y \in X$:

$$x \to y \qquad \Rightarrow \qquad h(x) \to h(y)\,. \tag{1}$$

Between the three examples before, there are many homomorphisms:

1. the (unique!) map $h\colon \{x, y, z\} \to \{u\}$
2. $h\colon \{x, y, z\} \to \{u_1, u_2, u_3, \ldots\}$ – no homomorphism of this type!
3. $h\colon \{u_1, u_2, u_3, \ldots\} \to \{x, y, z\}$ with $h(u_1) = y$, $h(u_2) = z$, $h(u_3) = y$, ...
4. and many more!

The first example of a homomorphism is special. In fact, we can prove that for *any* coalgebra $f\colon X \to X$, the unique function $h\colon X \to \{u\}$ is a homomorphism, since the right-hand side of (1) is trivial: $h(x) \to h(y)$ always holds.

The deterministic system $\{u\}$ (with the trivial transition structure on it) is thus special: for every deterministic system (X, f), there *exists* a *unique* homomorphism from (X, f) to $\{u\}$. We therefore call it the *final deterministic system*; it is an example of a final coalgebra.

This final coalgebra, even if it's simple, tells us something meaningful. A homomorphism identifies those states that 'behave the same' to an external observer. In this case, there's nothing to observe (the type F is the identity) and indeed every two states are identified by the unique morphism to the final coalgebra: every two states behave the same! This is in line with our 'machine with a single button' example: whether the machine has one internal state or switches between many, the behaviour is always the same.

It's about time to add some observations, to get a more interesting type of coalgebra ...

2 Deterministic systems with termination

In our second example, our machine has new capabilities: it can break! We can again press a button, but it could very well happen that the machine blocks at some point, and the button can't be pressed anymore. No idea how many times we should press; possibly the machine will never break, and we can keep on pressing forever!

This we model as a coalgebra of the form

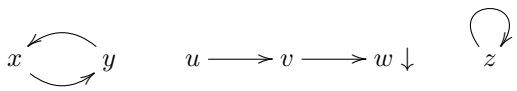
$$f\colon X \to X_\perp$$

where

$$X_\perp = X \cup \{\perp\}$$

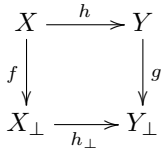
and we assume $\perp \notin X$. For such a coalgebra, we write $x \to y$ if $f(x) = y$ for some $y \in X$, and we write $x \downarrow$ if $f(x) = \perp$. We'll call them *deterministic systems with termination*.

Here's some examples:



(2)

A *homomorphism* from $f\colon X \to X_\perp$ to $g\colon Y \to Y_\perp$ is a map $h\colon X \to Y$ such that the following diagram commutes:



(3)

But what does $h_\perp$ mean? Well, it's another function $h_\perp \colon X_\perp \to Y_\perp$, defined as

$$h_\perp(x) = \begin{cases} \perp & \text{if } x = \perp \\ h(x) & \text{otherwise} \end{cases}$$

Notice that such a map $h_\perp$ can be defined for any h; it is a construction on functions. Later on, we'll see that this is part of the informally defined 'type' F from the beginning of this section: the mapping of a set X to $X_\perp$ and from a function h to $h_\perp$ are both encoded in F.

Anyway, let's express h being a morphism, that is, commutativity of diagram (3) in terms of our arrow notation: for all x, we require:

- if $x \downarrow$ then $h(x) \downarrow$, and
- if $x \to y$ then $h(x) \to h(y)$.

This is quite similar to the case of deterministic systems, but here we require that x and $h(x)$ "terminate at the same time".

In the examples above (2), the only homomorphism (other than identity functions) is $h \colon \{x, y\} \to \{z\}$ given by $h(x) = h(y) = z$. And, for two coalgebras

$$u \to v \to w \downarrow \qquad \text{and } a \to b \to c \to d \to e \downarrow$$

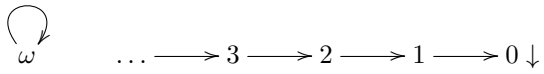
the only morphism from $\{u, v, w\}$ to $\{a, b, c, d, e\}$ maps u to c, v to d, and w to e.

In general, h is a homomorphism if for all x, either:

- x and $h(x)$ both terminate (that is, they reach $\perp$) after the same number of steps (applications of the coalgebra structures), or
- neither x nor $h(x)$ terminate.

This is all possible behaviour of our type of coalgebras: every state either never terminates, or it terminates after a certain number of steps.

Consider:



This is a coalgebra over the set $\mathbb{N} \cup \{\omega\}$ of 'extended natural numbers'. Written as a function: $g \colon \mathbb{N} \cup \{\omega\} \to (\mathbb{N} \cup \{\omega\})_\perp$,

$$g(x) = \begin{cases} \omega & \text{if } x = \omega \\ x - 1 & \text{if } x \in \mathbb{N} \text{ and } x > 0 \\ \perp & \text{if } x = 0 \end{cases}$$

For every deterministic system with termination (X, f), there is a unique homomorphism to $(\mathbb{N} \cup \{\omega\}, g)$, so the latter coalgebra is *final*. This reflects the idea that for this type of coalgebra, an external observer can observe exactly one thing: after how many steps the system terminates; in terms of our machine analogy, the machine is determined by how many button presses it takes to break the machine!

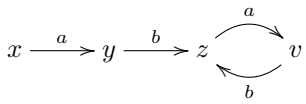
2.1 More (interesting) examples

So far, we focus on a few very basic examples to illustrate the notions of coalgebra, homomorphism and final coalgebra. In the next lectures we'll often focus on other examples, one prominent instance being coalgebras of the form:

$$f \colon X \to A \times X$$

for some fixed set of 'outputs' A. If we take for instance $A = \mathbb{N}$, then this reflects a machine which has two buttons and a display. Pressing one of the buttons shows a number; pressing the other changes shows nothing but changes the internal state (so possibly we'll see another number if we press the first one again). That's about enough for the machines analogy ... in the next lecture we'll study these systems properly.

For now, here's an example:



where x, y, z, v are states and $a, b \in A$ are output elements. What is the observable behaviour here?

Another super-important example is coalgebras of the form

$$f \colon X \to \{0,1\} \times X^A$$

where X^A is the set of functions from A to X. What is the behaviour of such a coalgebra? What do we observe in a state x?

We'll conclude with a nice list of examples.

$f \colon X \to X$	deterministic system
$f \colon X \to X_\bot$	det. system with termination
$f \colon X \to X_\bot \cup (E \times X)$	– with termination and exceptions
$f \colon X \to A \times X$	stream system
$f \colon X \to (A \times X)_\bot$	– with termination
$f \colon X \to \{0,1\} \times X^A$	deterministic automaton
$f \colon X \to \mathcal{P}(X)$	transition system
$f \colon X \to \{0,1\} \times (\mathcal{P}(X))^A$	non-deterministic automaton
$f \colon X \to \mathcal{P}(A \times X)$	labelled transition system
...	

where $\mathcal{P}(X)$ is the set of subsets of X (the powerset).

Concluding, we've seen basic, first examples of coalgebras as state-based systems; homomorphisms to compare and reduce systems; and a first glimpse of the generality of coalgebras. In the next lecture, we'll however focus on a specific instance: we'll see coalgebraic techniques to define infinite sequences, and prove properties about them.

References

[1] B. Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2016.

[2] J. Rutten. The method of coalgebra: exercises in coinduction. CWI Technical report, available at: `https://ir.cwi.nl/pub/28550/`, 2019.