1. The Method of Coalgebra

Jan Rutten

CWI Amsterdam & Radboud University Nijmegen

IMS, Singapore - 15 September 2016

Overview of Lecture one

1. Category theory (where coalgebra comes from)
2. Algebras and coalgebras
3. Induction and coinduction
4. The method of coalgebra
5. Discussion

1. Category theory
(where coalgebra comes from)

Why categories?

From Samson Abramsky's tutorial:

Categories, why and how?

(Dagstuhl, January 2015)

Why categories?

For **logicians**: gives a syntax-independent view of the fundamental structures of logic, opens up new kinds of models and interpretations.

For **philosophers**: a fresh approach to structuralist foundations of mathematics and science; an alternative to the traditional focus on set theory.

For **computer scientists**: gives a precise handle on abstraction, representation-independence, genericity and more. Gives the fundamental mathematical structures underpinning programming concepts.

Why categories?

For **mathematicians**: organizes your previous mathematical experience in a new and powerful way, reveals new connections and structure, allows you to “think bigger thoughts”.

For **physicists**: new ways of formulating physical theories in a structural form. Recent applications to Quantum Information and Computation.

For **economists and game theorists**: new tools, bringing complex phenomena into the scope of formalisation.

Category Theory in Slogans

1. Always ask: what are the **types**?
2. Think in terms of **arrows** rather than **elements.**
3. Ask what mathematical structures **do**, not what they **are.**
4. **Functoriality!**
5. **Universality!**
6. **Duality!**

   + several others.

All of the above are most relevant for coalgebra.

Categories: basic definitions

A category $\mathcal{C}$ consists of

- **Objects** $A, B, C, \ldots$
- **Morphisms/arrows**: for each pair of objects A, B, a set of morphisms $\mathcal{C}(A, B)$ with domain A and codomain B
- **Composition** of morphisms: $g \circ f$:

$$A \xrightarrow{f} B \xrightarrow{g} C \qquad A \xrightarrow{g \circ f} C$$

- **Identity** morphisms: $A \xrightarrow{1_A} A$
- **Axioms**:

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ 1_A = f = 1_B \circ f$$

Categories: examples

- Any kind of mathematical structure, together with structure preserving functions, forms a category. E.g.
  - **sets and functions**
  - groups and group homomorphisms
  - monoids and monoid homomorphisms
  - vector spaces over a field k, and linear maps
  - topological spaces and continuous functions
  - partially ordered sets and monotone functions
- Monoids are one-object categories
- **algebras**, and algebra homomorphisms
- **coalgebras**, and coalgebra homomorphisms

Always ask: what are the types?

$$A \xrightarrow{\quad f \quad} B \xrightarrow{\quad g \quad} C$$

For instance, for sets and functions:

Not:
let f be a function defined for any x by $f(x) = \ldots$

Rather:
let $f : X \to Y$ be a function defined for any $x \in X$ by $f(x) = \ldots$

Think in terms of arrows rather than elements

A function $f : X \to Y$ (between sets) is:

- **injective**: $\forall x, y \in X : \quad f(x) = f(y) \Rightarrow x = y$
- **surjective**: $\forall y \in Y \; \exists x \in X : \quad f(x) = y$

- **monic**: $\forall g, h : \quad f \circ g = f \circ h \Rightarrow g = h$

- **epic**: $\forall g, h : \quad g \circ f = h \circ f \Rightarrow g = h$

Think in terms of arrows rather than elements

A function $f : X \to Y$ (between sets) is:

- **injective**: $\forall x, y \in X : \quad f(x) = f(y) \Rightarrow x = y$
- **surjective**: $\forall y \in Y \; \exists x \in X : \quad f(x) = y$

- **monic**: $\forall g, h : \quad f \circ g = f \circ h \Rightarrow g = h$

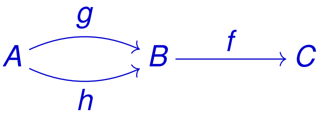
- **epic**: $\forall g, h : \quad g \circ f = h \circ f \Rightarrow g = h$

Think in terms of arrows rather than elements

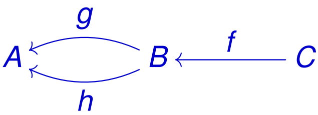
A function $f : X \to Y$ (between sets) is:

- **injective**: $\forall x, y \in X: \quad f(x) = f(y) \Rightarrow x = y$
- **surjective**: $\forall y \in Y \; \exists x \in X: \quad f(x) = y$

- **monic**: $\forall g, h: \quad f \circ g = f \circ h \Rightarrow g = h$



- **epic**: $\forall g, h: \quad g \circ f = h \circ f \Rightarrow g = h$



Think in terms of arrows rather than elements

Proposition

- m is injective iff m is monic.
- e is surjective iff e is epic.

Ask what mathematical structures **do**, not what they **are**

Defining the Cartesian **product** ...

- **with elements:**

$$A \times B = \{ \langle a, b \rangle \mid a \in A, b \in B \}$$

where

$$\langle a, b \rangle = \{\{a, b\}, b\}$$

This definition of the product is by no means canonical, does not seem to express any of its intrinsic properties, feels like coding.

Ask what mathematical structures **do**, not what they **are**

Defining the Cartesian **product** . . .

- **with elements:**

$$A \times B = \{\, \langle a, b \rangle \mid a \in A,\ b \in B \,\}$$

where

$$\langle a, b \rangle = \{\{a, b\},\ b\}$$

This definition of the product is by no means canonical, does not seem to express any of its intrinsic properties, feels like coding.

Ask what mathematical structures **do**, not what they **are**

Defining the Cartesian **product** ...

- **with elements:**

$$A \times B = \{ \langle a, b \rangle \mid a \in A,\ b \in B \}$$

where

$$\langle a, b \rangle = \{\{a, b\},\ b\}$$

This definition of the product is by no means canonical, does not seem to express any of its intrinsic properties, feels like coding.

Ask what mathematical structures **do**, not what they **are**

Defining the Cartesian **product** . . .

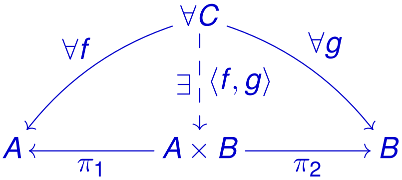
- **with arrows** (expressing a **universal** property):

This defines the **behaviour** of the product
by specifying its **interactions** with other objects.

Ask what mathematical structures **do**, not what they **are**

Defining the Cartesian **product** ...

- **with arrows** (expressing a **universal** property):

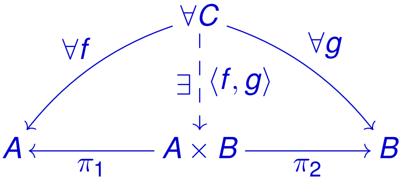


This defines the **behaviour** of the product
by specifying its **interactions** with other objects.

Ask what mathematical structures **do**, not what they **are**

Defining the Cartesian **product** ...

- **with arrows** (expressing a **universal** property):



This defines the **behaviour** of the product
by specifying its **interactions** with other objects.

Functoriality!

A *functor* $F : \mathcal{C} \to \mathcal{D}$ maps:

(i) each **object** A in $\mathcal{C}$ to an **object** $F(A)$ in $\mathcal{D}$

(ii) each **arrow** $f : A \to B$ in $\mathcal{C}$ to an **arrow** $F(f) : F(A) \to F(B)$ in $\mathcal{D}$

such that $F(g \circ f) = F(g) \circ F(f)$ and $F(id_A) = id_{F(A)}$

E.g., the *powerset functor* $\mathcal{P} : \mathrm{Set} \to \mathrm{Set}$ maps sets X to

$$\mathcal{P}(X) = \{V \mid V \subseteq X\}$$

and functions $f : X \to Y$ to

$$\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y) \qquad V \mapsto \{f(v) \mid v \in V\}$$

Functoriality!

A *functor* $F : \mathcal{C} \to \mathcal{D}$ maps:

(i) each **object** A in $\mathcal{C}$ to an **object** $F(A)$ in $\mathcal{D}$

(ii) each **arrow** $f : A \to B$ in $\mathcal{C}$ to
an **arrow** $F(f) : F(A) \to F(B)$ in $\mathcal{D}$

such that $F(g \circ f) = F(g) \circ F(f)$ and $F(id_A) = id_{F(A)}$

E.g., the *powerset functor* $\mathcal{P}$: $\mathsf{Set} \to \mathsf{Set}$ maps sets X to

$$\mathcal{P}(X) = \{V \mid V \subseteq X\}$$

and functions $f : X \to Y$ to

$$\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y) \qquad V \mapsto \{f(v) \mid v \in V\}$$

Functoriality!

Is just natural since all we have are objects and arrows.

Will be crucial for the definition of homomorphism between algebras and coalgebras.

Functoriality!

Is just natural since all we have are objects and arrows.

Will be crucial for the definition of homomorphism between algebras and coalgebras.

Universality!

Ideally, definitions are phrased in terms of *universal properties*, which are typically formulated as:

$$\forall \quad \dots \quad \exists! \quad \dots$$

E.g., an object A in a category $\mathcal{C}$ is **initial** if:
for any object B in $\mathcal{C}$ there exists a unique arrow from A to B:

$$\forall\, B \xleftarrow{\;\exists!\;} A$$

Similarly, an object A is **final** if:
for any object B in $\mathcal{C}$ there exists a unique arrow from B to A:

$$\forall\, B \xrightarrow{\;\exists!\;} A$$

Universality!

Ideally, definitions are phrased in terms of *universal properties*, which are typically formulated as:

$$\forall \quad \ldots \quad \exists! \quad \ldots$$

E.g., an object A in a category $\mathcal{C}$ is **initial** if:
for any object B in $\mathcal{C}$ there exists a unique arrow from A to B:

$$\forall\, B \xleftarrow{\;\exists!\;} A$$

Similarly, an object A is **final** if:
for any object B in $\mathcal{C}$ there exists a unique arrow from B to A:

$$\forall\, B \xrightarrow{\;\exists!\;} A$$

Duality!

Informally, duality refers to the elementary process of "reversing the arrows" in a diagram.

E.g., f is **monic**: $\forall g, h: \quad f \circ g = f \circ h \Rightarrow g = h$

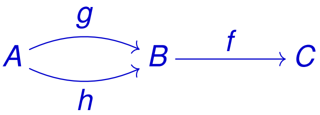
Reversing the arrows: $\forall g, h: \quad g \circ f = h \circ f \Rightarrow g = h$

That is, f is **epic**.

Duality!

Informally, duality refers to the elementary process of "reversing the arrows" in a diagram.

E.g., f is **monic**: $\forall g, h: \quad f \circ g = f \circ h \Rightarrow g = h$



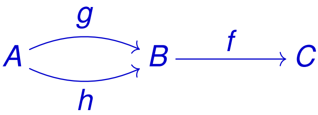
Reversing the arrows: $\forall g, h: \quad g \circ f = h \circ f \Rightarrow g = h$

That is, f is **epic**.

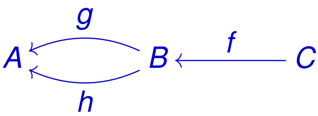
Duality!

Informally, duality refers to the elementary process of "reversing the arrows" in a diagram.

E.g., f is **monic**: $\forall g, h: \quad f \circ g = f \circ h \Rightarrow g = h$



Reversing the arrows: $\forall g, h: \quad g \circ f = h \circ f \Rightarrow g = h$

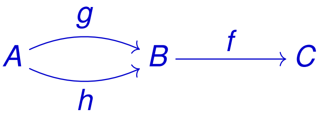


That is, f is **epic.**

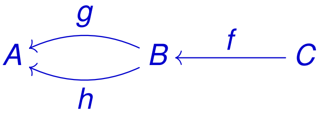
Duality!

Informally, duality refers to the elementary process of "reversing the arrows" in a diagram.

E.g., f is **monic**: $\forall g, h: \quad f \circ g = f \circ h \Rightarrow g = h$



Reversing the arrows: $\forall g, h: \quad g \circ f = h \circ f \Rightarrow g = h$



That is, f is **epic.**

Duality, formally

The *opposite* category $\mathcal{C}^{op}$ of a category $\mathcal{C}$ has:

- the same objects as $\mathcal{C}$
- precisely one arrow $f : B \to A$ for every arrow $f : A \to B$ in $\mathcal{C}$.

The principle of duality now says that we can dualize any statement about a category $\mathcal{C}$ by making the same statement about $\mathcal{C}^{op}$.

For instance, the notions of monic and epic are dual, since:

Proposition: f is monic in $\mathcal{C}$ iff f is epic in $\mathcal{C}^{op}$.

Duality, formally

The *opposite* category $\mathcal{C}^{op}$ of a category $\mathcal{C}$ has:

- the same objects as $\mathcal{C}$
- precisely one arrow $f : B \to A$ for every arrow $f : A \to B$ in $\mathcal{C}$.

The principle of duality now says that we can dualize any statement about a category $\mathcal{C}$ by making the same statement about $\mathcal{C}^{op}$.

For instance, the notions of monic and epic are dual, since:

Proposition: f is monic in $\mathcal{C}$ iff f is epic in $\mathcal{C}^{op}$.

Duality, formally

The *opposite* category $\mathcal{C}^{op}$ of a category $\mathcal{C}$ has:

- the same objects as $\mathcal{C}$
- precisely one arrow $f : B \to A$ for every arrow $f : A \to B$ in $\mathcal{C}$.

The principle of duality now says that we can dualize any statement about a category $\mathcal{C}$ by making the same statement about $\mathcal{C}^{op}$.

For instance, the notions of monic and epic are dual, since:

Proposition: f is monic in $\mathcal{C}$ iff f is epic in $\mathcal{C}^{op}$.

Duality: products and coproducts

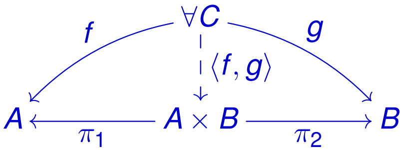
The **product** of A and B:

The **coproduct** of A and B:

Proposition: O is product in $\mathcal{C}$ iff O is coproduct in $\mathcal{C}^{op}$.

Duality: products and coproducts

The **product** of A and B:

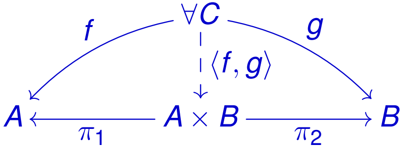


The **coproduct** of A and B:

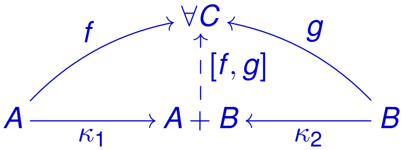
Proposition: O is product in $\mathcal{C}$ iff O is coproduct in $\mathcal{C}^{op}$.

Duality: products and coproducts

The **product** of A and B:



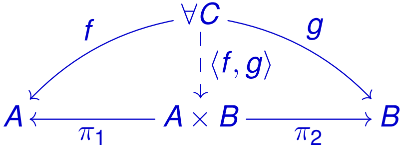
The **coproduct** of A and B:



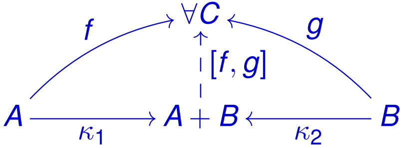
Proposition: O is product in $\mathcal{C}$ iff O is coproduct in $\mathcal{C}^{op}$.

Duality: products and coproducts

The **product** of A and B:



The **coproduct** of A and B:



Proposition: O is product in $\mathcal{C}$ iff O is coproduct in $\mathcal{C}^{op}$.

Duality: initial and final objects

An object A in a category $\mathcal{C}$ is $\ldots$

- **initial** if for any object B there exists a unique arrow

$$A \overset{!}{\dashrightarrow} B$$

- **final** if for any object B there exists a unique arrow

$$B \overset{!}{\dashrightarrow} A$$

Proposition: A is initial in $\mathcal{C}$ iff A is final in $\mathcal{C}^{op}$.

Duality: initial and final objects

An object A in a category $\mathcal{C}$ is ...

- **initial** if for any object B there exists a unique arrow

$$A \overset{!}{\dashrightarrow} B$$

- **final** if for any object B there exists a unique arrow

$$B \overset{!}{\dashrightarrow} A$$

Proposition: A is initial in $\mathcal{C}$ iff A is final in $\mathcal{C}^{op}$.

2. Algebras and Coalgebras

Where coalgebra comes from

By **duality.** From **algebra**!

Classically, algebras are sets with operations.

Ex. $(\mathbb{N}, 0, \mathsf{succ})$, with $0 \in \mathbb{N}$ and $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$.

Equivalently,

$$\begin{array}{c} 1 + \mathbb{N} \\ {\scriptstyle [\mathsf{zero}, \mathsf{succ}]} \big\downarrow \\ \mathbb{N} \end{array}$$

where $1 = \{*\}$ and $\mathsf{zero}(*) = 0$.

Where coalgebra comes from

By **duality**. From **algebra**!

Classically, algebras are sets with operations.

Ex. $(\mathbb{N}, 0, \text{succ})$, with $0 \in \mathbb{N}$ and $\text{succ} : \mathbb{N} \to \mathbb{N}$.

Equivalently,

$$\begin{array}{c} 1 + \mathbb{N} \\ {\scriptstyle [\text{zero}, \text{succ}]} \downarrow \\ \mathbb{N} \end{array}$$

where $1 = \{*\}$ and $\text{zero}(*) = 0$.

Where coalgebra comes from

By **duality**. From **algebra**!

Classically, algebras are sets with operations.

Ex. $(\mathbb{N}, 0, \mathsf{succ})$, with $0 \in \mathbb{N}$ and $\mathsf{succ} : \mathbb{N} \to \mathbb{N}$.

Equivalently,

$$\begin{array}{c} 1 + \mathbb{N} \\ {\scriptstyle [\mathsf{zero}, \mathsf{succ}]} \big\downarrow \\ \mathbb{N} \end{array}$$

where $1 = \{*\}$ and $\mathsf{zero}(*) = 0$.

Algebra

Classically, algebras are sets with operations.

Ex.

$$\begin{array}{c} \mathit{Prog} \times \mathit{Prog} \\ \alpha \big\downarrow \\ \mathit{Prog} \end{array}$$

with $\alpha(P_1, P_2) = P_1; P_2$.

Algebra, categorically

For a functor $F : \mathcal{C} \to \mathcal{C}$, an F**-algebra** is a pair (A, α) with

$$\begin{array}{c} F(X) \\ \alpha \downarrow \\ X \end{array}$$

We call F the **type** and α the **structure map** of (A, α).

The structure map α tells us how the elements of A are **constructed** from other elements in A.

E.g., $a^*; b$ is constructed from the expressions a^* and b by applying the operation of concatenation.

Algebra, categorically

For a functor $F : \mathcal{C} \to \mathcal{C}$, an **$F$-algebra** is a pair (A, α) with

$$\begin{array}{c} F(X) \\ \Big\downarrow{\scriptstyle \alpha} \\ X \end{array}$$

We call F the **type** and α the **structure map** of (A, α).

The structure map α tells us how the elements of A are **constructed** from other elements in A.

E.g., $a^*; b$ is constructed from the expressions a^* and b by applying the operation of concatenation.

Algebra, categorically

For a functor $F : \mathcal{C} \to \mathcal{C}$, an F**-algebra** is a pair (A, α) with

$$\begin{array}{c} F(X) \\ \downarrow{\scriptstyle \alpha} \\ X \end{array}$$

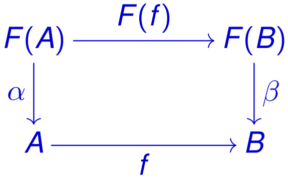
We call F the **type** and α the **structure map** of (A, α).

The structure map α tells us how the elements of A are **constructed** from other elements in A.

E.g., $a^*; b$ is constructed from the expressions a^* and b by applying the operation of concatenation.

Algebra homomorphisms

A *homomorphism of F-algebras* is an arrow $f : A \to B$ such that



Note: **functoriality**!

Homomorphisms are for algebras what functions are for sets: they allow us to express how algebras interact with other algebras.

Algebra homomorphisms

A *homomorphism of F-algebras* is an arrow $f : A \to B$ such that

$$\begin{array}{ccc} F(A) & \xrightarrow{F(f)} & F(B) \\ \alpha \downarrow & & \downarrow \beta \\ A & \xrightarrow[f]{} & B \end{array}$$

Note: **functoriality**!

Homomorphisms are for algebras what functions are for sets: they allow us to express how algebras interact with other algebras.

Algebra homomorphisms

A *homomorphism of F-algebras* is an arrow $f : A \to B$ such that

$$\begin{array}{ccc} F(A) & \xrightarrow{F(f)} & F(B) \\ \alpha \big\downarrow & & \big\downarrow \beta \\ A & \xrightarrow[f]{} & B \end{array}$$

Note: **functoriality**!

Homomorphisms are for algebras what functions are for sets: they allow us to express how algebras interact with other algebras.

Coalgebra, dually

For a functor $F : \mathcal{C} \to \mathcal{C}$, an F-**coalgebra** is a pair (A, α) with

$$\begin{array}{c} X \\ \alpha \big\downarrow \\ F(X) \end{array}$$

We call F the **type** and α the **structure map** of (A, α).

Our favourite example: streams

$$\begin{array}{c} \mathbb{N}^\omega \\ \Big\downarrow \langle \text{head}, \text{tail} \rangle \\ \mathbb{N} \times \mathbb{N}^\omega \end{array}$$

where

$$\begin{aligned} \text{head}(\sigma) &= \sigma(0) \\ \text{tail}(\sigma) &= (\sigma(1), \sigma(2), \sigma(3), \ldots) \end{aligned}$$

for any stream $\sigma = (\sigma(0), \sigma(1), \sigma(2), \ldots) \in \mathbb{N}^\omega$.

Here the structure map tells us how streams are **decomposed** into a natural number and a stream.

Our favourite example: streams

$$\begin{array}{c} \mathbb{N}^\omega \\ \big\downarrow \langle \mathsf{head}, \mathsf{tail} \rangle \\ \mathbb{N} \times \mathbb{N}^\omega \end{array}$$

where

$$\begin{aligned} \mathsf{head}(\sigma) &= \sigma(0) \\ \mathsf{tail}(\sigma) &= (\sigma(1), \sigma(2), \sigma(3), \ldots) \end{aligned}$$

for any stream $\sigma = (\sigma(0), \sigma(1), \sigma(2), \ldots) \in \mathbb{N}^\omega$.

Here the structure map tells us how streams are **decomposed** into a natural number and a stream.

Coalgebra homomorphisms

A *homomorphism of F-coalgebras* is an arrow $f : A \to B$ with

$$\begin{array}{ccc} A & \xrightarrow{\quad f \quad} & B \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ F(A) & \xrightarrow[\quad F(f) \quad]{} & F(B) \end{array}$$

Note: **functoriality**!

Homomorphisms are for coalgebras what functions are for sets: they allow us to express the interaction between coalgebras.

Coalgebra homomorphisms

A *homomorphism of F-coalgebras* is an arrow $f : A \to B$ with

$$\begin{array}{ccc} A & \xrightarrow{\quad f \quad} & B \\ {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle \beta} \\ F(A) & \xrightarrow[\quad F(f) \quad]{} & F(B) \end{array}$$

Note: **functoriality**!

Homomorphisms are for coalgebras what functions are for sets: they allow us to express the interaction between coalgebras.

Coalgebra homomorphisms

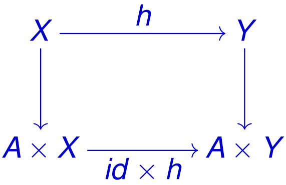
A *homomorphism of F-coalgebras* is an arrow $f : A \to B$ with

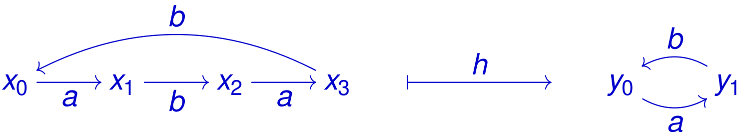
$$\begin{array}{ccc} A & \xrightarrow{\quad f \quad} & B \\ {\scriptstyle \alpha}\downarrow & & \downarrow{\scriptstyle \beta} \\ F(A) & \xrightarrow[\quad F(f) \quad]{} & F(B) \end{array}$$

Note: **functoriality**!

Homomorphisms are for coalgebras what functions are for sets: they allow us to express the interaction between coalgebras.

Example of a homomorphism





The homomorphism h identifies behaviourally equivalent states.

3. Induction and coinduction

- initial algebra - final coalgebra
- congruence - bisimulation
- induction - coinduction
- least fixed point - greatest fixed point

Initial algebra

The natural numbers are an example of an **initial algebra**:

$$\begin{array}{ccc}
1 + \mathbb{N} & \dashrightarrow & 1 + S \\
{\scriptstyle [\mathsf{zero}, \mathsf{succ}]}\big\downarrow & & \big\downarrow {\scriptstyle \forall\,\beta} \\
\mathbb{N} & \underset{\exists !\, h}{\dashrightarrow} & S
\end{array}$$

Inductive **definitions** are based on the (unique) **existence** of h.

Inductive **proofs** are based on the **uniqueness** of h.

Initial algebra

The natural numbers are an example of an **initial algebra**:

$$
\begin{array}{ccc}
1 + \mathbb{N} & \dashrightarrow & 1 + S \\
{\scriptstyle [\mathsf{zero}, \mathsf{succ}]}\downarrow & & \downarrow{\scriptstyle \forall\ \beta} \\
\mathbb{N} & \underset{\exists!\, h}{\dashrightarrow} & S
\end{array}
$$

Inductive **definitions** are based on the (unique) **existence** of h.

Inductive **proofs** are based on the **uniqueness** of h.

Initial algebra

The natural numbers are an example of an **initial algebra**:

$$
\begin{array}{ccc}
1 + \mathbb{N} & \dashrightarrow & 1 + S \\
{\scriptstyle [\mathsf{zero}, \mathsf{succ}]} \downarrow & & \downarrow {\scriptstyle \forall\, \beta} \\
\mathbb{N} & \underset{\exists !\, h}{\dashrightarrow} & S
\end{array}
$$

Inductive **definitions** are based on the (unique) **existence** of h.

Inductive **proofs** are based on the **uniqueness** of h.

Final coalgebra

Streams are an example of a **final coalgebra**:

$$\begin{array}{ccc} S & \overset{\exists!\, h}{\dashrightarrow} & \mathbb{N}^\omega \\ {\scriptstyle\forall}\downarrow{\scriptstyle\beta} & & \downarrow{\scriptstyle\langle \text{head}, \text{tail} \rangle} \\ \mathbb{N} \times S & \dashrightarrow & \mathbb{N} \times \mathbb{N}^\omega \end{array}$$

(Note: instead of $\mathbb{N}$, we could have taken **any** set.)

Coinductive **definitions** are based on the **existence** of h.

Coinductive **proofs** are based on the **uniqueness** of h.

Final coalgebra

Streams are an example of a **final coalgebra**:

$$\begin{array}{ccc} S & \xrightarrow{\exists!\, h} & \mathbb{N}^\omega \\ {\scriptstyle \forall}\big\downarrow{\scriptstyle \beta} & & \big\downarrow{\scriptstyle \langle \mathsf{head}, \mathsf{tail} \rangle} \\ \mathbb{N} \times S & \dashrightarrow & \mathbb{N} \times \mathbb{N}^\omega \end{array}$$

(Note: instead of $\mathbb{N}$, we could have taken **any** set.)

Coinductive **definitions** are based on the **existence** of h.

Coinductive **proofs** are based on the **uniqueness** of h.

Final coalgebra

Streams are an example of a **final coalgebra**:

$$\begin{array}{ccc} S & \overset{\exists!\, h}{\dashrightarrow} & \mathbb{N}^{\omega} \\ {\scriptstyle \forall}\downarrow{\scriptstyle \beta} & & \downarrow{\scriptstyle \langle \mathrm{head}, \mathrm{tail} \rangle} \\ \mathbb{N} \times S & \dashrightarrow & \mathbb{N} \times \mathbb{N}^{\omega} \end{array}$$

(Note: instead of $\mathbb{N}$, we could have taken **any** set.)

Coinductive **definitions** are based on the **existence** of h.

Coinductive **proofs** are based on the **uniqueness** of h.

Algebra and induction

Induction = definition and proof principle for algebras.

Ex. mathematical induction: for all $P \subseteq \mathbb{N}$,

$$(P(0) \text{ and } (\forall n : P(n) \Rightarrow P(\text{succ}(n)))) \Rightarrow \forall n : P(n)$$

(Other examples: transfinite, well-founded, tree, structural, etc.)

We show that induction is a property of **initial algebras.**

Algebra and induction

Induction = definition and proof principle for algebras.

Ex. mathematical induction: for all $P \subseteq \mathbb{N}$,

$$(\ P(0)\ \text{and}\ (\forall n:\ P(n) \Rightarrow P(\text{succ}(n)))\ )\ \Rightarrow\ \forall n:\ P(n)$$

(Other examples: transfinite, well-founded, tree, structural, etc.)

We show that induction is a property of **initial algebras.**

Algebra and induction

Induction = definition and proof principle for algebras.

Ex. mathematical induction: for all $P \subseteq \mathbb{N}$,

$$(\ P(0)\ \text{ and }\ (\forall n :\ P(n) \Rightarrow P(\mathrm{succ}(n)))\ )\ \Rightarrow\ \forall n :\ P(n)$$

(Other examples: transfinite, well-founded, tree, structural, etc.)

We show that induction is a property of **initial algebras.**

Algebra and induction

Induction = definition and proof principle for algebras.

Ex. mathematical induction: for all $P \subseteq \mathbb{N}$,

$$(P(0) \text{ and } (\forall n : P(n) \Rightarrow P(\text{succ}(n)))) \Rightarrow \forall n : P(n)$$

(Other examples: transfinite, well-founded, tree, structural, etc.)

We show that induction is a property of **initial algebras.**

Algebras and congruences (ex. natural numbers)

We call $R \subseteq \mathbb{N} \times \mathbb{N}$ a **congruence** if

$$(i) \qquad (0,0) \in R \ \text{ and}$$
$$(ii) \qquad (n,m) \in R \Rightarrow (\mathrm{succ}(n), \mathrm{succ}(m)) \in R$$

(Note: R is **not** required to be an equivalence relation.)

Equivalently, $R \subseteq \mathbb{N} \times \mathbb{N}$ is a **congruence** if

$$\begin{array}{ccccc}
1+\mathbb{N} & \longleftarrow & 1+R & \longrightarrow & 1+\mathbb{N} \\
{\scriptstyle [\mathrm{zero},\mathrm{succ}]}\downarrow & & \downarrow{\scriptstyle \exists\, \gamma} & & \downarrow{\scriptstyle [\mathrm{zero},\mathrm{succ}]} \\
\mathbb{N} & \underset{\pi_1}{\longleftarrow} & R & \underset{\pi_2}{\longrightarrow} & \mathbb{N}
\end{array}$$

for some function $\gamma : 1 + R \to R$.

Algebras and congruences (ex. natural numbers)

We call $R \subseteq \mathbb{N} \times \mathbb{N}$ a **congruence** if

$$\begin{array}{ll} (i) & (0,0) \in R \ \text{ and} \\ (ii) & (n,m) \in R \ \Rightarrow \ (\mathrm{succ}(n), \mathrm{succ}(m)) \in R \end{array}$$

(Note: R is **not** required to be an equivalence relation.)

Equivalently, $R \subseteq \mathbb{N} \times \mathbb{N}$ is a **congruence** if

$$\begin{array}{ccccc} 1 + \mathbb{N} & \longleftarrow & 1 + R & \longrightarrow & 1 + \mathbb{N} \\ {\scriptstyle [\mathrm{zero}, \mathrm{succ}]} \downarrow & & \exists \, \downarrow \gamma & & \downarrow {\scriptstyle [\mathrm{zero}, \mathrm{succ}]} \\ \mathbb{N} & \underset{\pi_1}{\longleftarrow} & R & \underset{\pi_2}{\longrightarrow} & \mathbb{N} \end{array}$$

for some function $\gamma : 1 + R \to R$.

Initial algebras and congruences

Theorem: induction proof principle
Every congruence $R \subseteq \mathbb{N} \times \mathbb{N}$ contains the **diagonal**:

$$\Delta \subseteq R$$

where $\Delta = \{(n, n) \mid n \in \mathbb{N}\}$.

Proof: Because $(\mathbb{N}, [\text{zero}, \text{succ}])$ is an initial algebra,

we have $\pi_1 \circ ! = id = \pi_2 \circ !$, which implies $!(n) = (n, n)$, all $n \in \mathbb{N}$.

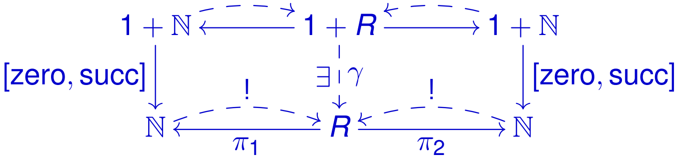
Initial algebras and congruences

Theorem: induction proof principle
Every congruence $R \subseteq \mathbb{N} \times \mathbb{N}$ contains the **diagonal**:

$$\Delta \subseteq R$$

where $\Delta = \{(n, n) \mid n \in \mathbb{N}\}$.

Proof: Because $(\mathbb{N}, [\text{zero}, \text{succ}])$ is an initial algebra,



we have $\pi_1 \circ ! = id = \pi_2 \circ !$, which implies $!(n) = (n, n)$, all $n \in \mathbb{N}$.

Initial algebras and induction

Theorem: The following are equivalent:

1. For every congruence relation $R \subseteq \mathbb{N} \times \mathbb{N}$,

$$\Delta \subseteq R$$

2. For every predicate $P \subseteq \mathbb{N}$,

$$(\ P(0)\ \text{and}\ (\forall n :\ P(n) \Rightarrow P(\text{succ}(n)))\ )\ \Rightarrow\ \forall n :\ P(n)$$

Proof: Exercise. □

In other words: two equivalent formulations of **induction**!

Initial algebras and induction

Theorem: The following are equivalent:

1. For every congruence relation $R \subseteq \mathbb{N} \times \mathbb{N}$,

$$\Delta \subseteq R$$

2. For every predicate $P \subseteq \mathbb{N}$,

$$(\, P(0) \;\text{ and }\; (\forall n : \; P(n) \Rightarrow P(\text{succ}(n)))\,) \;\Rightarrow\; \forall n : \; P(n)$$

Proof: Exercise. $\square$

In other words: two equivalent formulations of **induction**!

Initial algebras and induction

Theorem: The following are equivalent:

1. For every congruence relation $R \subseteq \mathbb{N} \times \mathbb{N}$,

$$\Delta \subseteq R$$

2. For every predicate $P \subseteq \mathbb{N}$,

$$(\ P(0)\ \text{and}\ (\forall n :\ P(n) \Rightarrow P(\mathrm{succ}(n)))\ )\ \Rightarrow\ \forall n :\ P(n)$$

Proof: Exercise. □

In other words: two equivalent formulations of **induction**!

Coalgebra and coinduction

Coinduction = definition and proof principle for coalgebras.

Coinduction is **dual** to induction, in a very precise way.

Categorically, coinduction is a property of **final coalgebras**.

Algorithmically, coinduction generalises Robin Milner's

bisimulation proof method.

Coalgebra and coinduction

Coinduction = definition and proof principle for coalgebras.

Coinduction is **dual** to induction, in a very precise way.

Categorically, coinduction is a property of **final coalgebras**.

Algorithmically, coinduction generalises Robin Milner's

bisimulation proof method.

Coalgebra and coinduction

Coinduction = definition and proof principle for coalgebras.

Coinduction is **dual** to induction, in a very precise way.

Categorically, coinduction is a property of **final coalgebras.**

Algorithmically, coinduction generalises Robin Milner's

bisimulation proof method.

Coalgebra and coinduction

Coinduction = definition and proof principle for coalgebras.

Coinduction is **dual** to induction, in a very precise way.

Categorically, coinduction is a property of **final coalgebras.**

Algorithmically, coinduction generalises Robin Milner's

bisimulation proof method.

Coalgebras and bisimulations (ex. streams)

We call $R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ a **bisimulation** if, for all $(\sigma, \tau) \in R$,

$$
\begin{array}{ll}
(i) & \text{head}(\sigma) = \text{head}(\tau) \ \text{ and} \\
(ii) & (\text{tail}(\sigma), \text{tail}(\tau)) \in R
\end{array}
$$

Equivalently, $R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ is a **bisimulation** if

$$
\begin{array}{ccccc}
\mathbb{N}^\omega & \xleftarrow{\ \pi_1\ } & R & \xrightarrow{\ \pi_2\ } & \mathbb{N}^\omega \\
\langle \text{head}, \text{tail} \rangle \downarrow & & \exists \downarrow \gamma & & \downarrow \langle \text{head}, \text{tail} \rangle \\
\mathbb{N} \times \mathbb{N}^\omega & \longleftarrow & \mathbb{N} \times R & \longrightarrow & \mathbb{N} \times \mathbb{N}^\omega
\end{array}
$$

for some function $\gamma : R \to \mathbb{N} \times R$.

Coalgebras and bisimulations (ex. streams)

We call $R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ a **bisimulation** if, for all $(\sigma, \tau) \in R$,

$$\begin{array}{ll} (i) & \text{head}(\sigma) = \text{head}(\tau) \;\text{ and} \\ (ii) & (\text{tail}(\sigma), \text{tail}(\tau)) \in R \end{array}$$

Equivalently, $R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ is a **bisimulation** if

$$\begin{array}{ccccc} \mathbb{N}^\omega & \xleftarrow{\;\pi_1\;} & R & \xrightarrow{\;\pi_2\;} & \mathbb{N}^\omega \\ {\scriptstyle \langle \text{head}, \text{tail} \rangle}\downarrow & & \downarrow{\scriptstyle \exists\, \gamma} & & \downarrow{\scriptstyle \langle \text{head}, \text{tail} \rangle} \\ \mathbb{N} \times \mathbb{N}^\omega & \longleftarrow & \mathbb{N} \times R & \longrightarrow & \mathbb{N} \times \mathbb{N}^\omega \end{array}$$

for some function $\gamma : R \to \mathbb{N} \times R$.

Final coalgebras and bisimulations

Theorem: coinduction proof principle
Every bisimulation $R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ **is contained** in the diagonal:

$$R \subseteq \Delta$$

where $\Delta = \{(\sigma, \sigma) \mid \sigma \in \mathbb{N}^\omega\}$.

Proof: Because $(\mathbb{N}^\omega, \langle \text{head}, \text{tail} \rangle)$ is a final coalgebra,

$$\begin{array}{ccccc}
\mathbb{N}^\omega & \xleftarrow{\ \pi_1\ } & R & \xrightarrow{\ \pi_2\ } & \mathbb{N}^\omega \\
{\scriptstyle \langle \text{head}, \text{tail} \rangle} \downarrow & & \downarrow {\scriptstyle \exists ! \gamma} & & \downarrow {\scriptstyle \langle \text{head}, \text{tail} \rangle} \\
\mathbb{N} \times \mathbb{N}^\omega & \longleftarrow & \mathbb{N} \times R & \longrightarrow & \mathbb{N} \times \mathbb{N}^\omega
\end{array}$$

we have $\pi_1 = \pi_2$, which implies $\sigma = \tau$, for all $(\sigma, \tau) \in \mathbb{N}^\omega$.

Final coalgebras and bisimulations

Theorem: coinduction proof principle
Every bisimulation $R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ **is contained** in the diagonal:

$$R \subseteq \Delta$$

where $\Delta = \{(\sigma, \sigma) \mid \sigma \in \mathbb{N}^\omega\}$.

Proof: Because $(\mathbb{N}^\omega, \langle \text{head}, \text{tail} \rangle)$ is a final coalgebra,

$$\begin{array}{ccccc}
\mathbb{N}^\omega & \xleftarrow{\quad \pi_1 \quad} & R & \xrightarrow{\quad \pi_2 \quad} & \mathbb{N}^\omega \\
{\scriptstyle \langle \text{head}, \text{tail} \rangle} \downarrow & & \exists \downarrow \gamma & & \downarrow {\scriptstyle \langle \text{head}, \text{tail} \rangle} \\
\mathbb{N} \times \mathbb{N}^\omega & \longleftarrow & \mathbb{N} \times R & \longrightarrow & \mathbb{N} \times \mathbb{N}^\omega
\end{array}$$

we have $\pi_1 = \pi_2$, which implies $\sigma = \tau$, for all $(\sigma, \tau) \in \mathbb{N}^\omega$.

Congruences and bisimulations: dual?

$R \subseteq \mathbb{N} \times \mathbb{N}$ is a **congruence** if

$$\begin{array}{ccccc}
1+\mathbb{N} & \longleftarrow & 1+R & \longrightarrow & 1+\mathbb{N} \\
{\scriptstyle [\mathrm{zero},\mathrm{succ}]}\downarrow & & \exists\, \downarrow \gamma & & \downarrow{\scriptstyle [\mathrm{zero},\mathrm{succ}]} \\
\mathbb{N} & \underset{\pi_1}{\longleftarrow} & R & \underset{\pi_2}{\longrightarrow} & \mathbb{N}
\end{array}$$

$R \subseteq \mathbb{N}^\omega \times \mathbb{N}^\omega$ is a **bisimulation** if

$$\begin{array}{ccccc}
\mathbb{N}^\omega & \overset{\pi_1}{\longleftarrow} & R & \overset{\pi_2}{\longrightarrow} & \mathbb{N}^\omega \\
{\scriptstyle \langle \mathrm{head},\mathrm{tail}\rangle}\downarrow & & \exists\, \downarrow \gamma & & \downarrow{\scriptstyle \langle \mathrm{head},\mathrm{tail}\rangle} \\
\mathbb{N}\times\mathbb{N}^\omega & \longleftarrow & \mathbb{N}\times R & \longrightarrow & \mathbb{N}\times\mathbb{N}^\omega
\end{array}$$

Congruences and bisimulations

$R \subseteq S \times T$ is an F-**congruence** if

$$
\begin{array}{ccccc}
F(S) & \longleftarrow & F(R) & \longrightarrow & F(T) \\
\alpha \big\downarrow & & \exists \, \gamma \, \big\downarrow & & \big\downarrow \beta \\
S & \underset{\pi_1}{\longleftarrow} & R & \underset{\pi_2}{\longrightarrow} & T
\end{array}
$$

$R \subseteq S \times T$ is an F-**bisimulation** if

$$
\begin{array}{ccccc}
S & \overset{\pi_1}{\longleftarrow} & R & \overset{\pi_2}{\longrightarrow} & T \\
\alpha \big\downarrow & & \exists \, \gamma \, \big\downarrow & & \big\downarrow \beta \\
F(S) & \longleftarrow & F(R) & \longrightarrow & F(T)
\end{array}
$$

Induction and coinduction

For every **congruence** relation $R \subseteq \mathbb{N} \times \mathbb{N}$,

$$\Delta \subseteq R$$

For every **bisimulation** relation $R \subseteq \mathbb{N}^{\omega} \times \mathbb{N}^{\omega}$,

$$R \subseteq \Delta$$

Induction and coinduction

For every **congruence** relation R on an **initial algebra**:

$$\Delta \subseteq R$$

For every **bisimulation** relation R on a **final coalgebra**:

$$R \subseteq \Delta$$

An aside: fixed points

Let $(P, \leq)$ be a preorder and $f : P \to P$ a monotone map.

Classically, least fixed point **induction** is:

$$\forall p \in P : \quad f(p) \leq p \;\Rightarrow\; \mu f \leq p$$

Classically, greatest fixed point **coinduction** is:

$$\forall p \in P : \quad p \leq f(p) \;\Rightarrow\; p \leq \nu f$$

An aside: fixed points

Let $(P, \leq)$ be a preorder and $f : P \to P$ a monotone map.

Classically, least fixed point **induction** is:

$$\forall p \in P : \quad f(p) \leq p \;\Rightarrow\; \mu f \leq p$$

Classically, greatest fixed point **coinduction** is:

$$\forall p \in P : \quad p \leq f(p) \;\Rightarrow\; p \leq \nu f$$

An aside: fixed points

Let $(P, \leq)$ be a preorder and $f : P \to P$ a monotone map.

Classically, least fixed point **induction** is:

$$\forall p \in P : \quad f(p) \leq p \;\Rightarrow\; \mu f \leq p$$

Classically, greatest fixed point **coinduction** is:

$$\forall p \in P : \quad p \leq f(p) \;\Rightarrow\; p \leq \nu f$$

An aside: fixed points

Any preorder $(P, \leq)$ is a category, with arrows:

$$p \to q \;\equiv\; p \leq q$$

Any monotone map is a functor:

$$p \to q \quad \mapsto \quad f(p) \to f(q)$$

Lfp **induction** and gfp **coinduction** become:

$$\begin{array}{ccc} f(\mu f) & \dashrightarrow & f(p) \\ \downarrow & & \downarrow \\ \mu f & \dashrightarrow & p \end{array} \qquad \begin{array}{ccc} p & \dashrightarrow & \nu f \\ \downarrow & & \downarrow \\ f(p) & \dashrightarrow & f(\nu f) \end{array}$$

An aside: fixed points

Any preorder $(P, \leq)$ is a category, with arrows:

$$p \to q \;\equiv\; p \leq q$$

Any monotone map is a functor:

$$p \to q \quad \mapsto \quad f(p) \to f(q)$$

Lfp **induction** and gfp **coinduction** become:

$$\begin{array}{ccc} f(\mu f) & \dashrightarrow & f(p) \\ \downarrow & & \downarrow \\ \mu f & \dashrightarrow & p \end{array} \qquad\qquad \begin{array}{ccc} p & \dashrightarrow & \nu f \\ \downarrow & & \downarrow \\ f(p) & \dashrightarrow & f(\nu f) \end{array}$$

An aside: fixed points

Any preorder $(P, \leq)$ is a category, with arrows:

$$p \to q \;\equiv\; p \leq q$$

Any monotone map is a functor:

$$p \to q \quad \mapsto \quad f(p) \to f(q)$$

Lfp **induction** and gfp **coinduction** become:

$$\begin{array}{ccc} f(\mu f) & \dashrightarrow & f(p) \\ \downarrow & & \downarrow \\ \mu f & \dashrightarrow & p \end{array} \qquad\qquad \begin{array}{ccc} p & \dashrightarrow & \nu f \\ \downarrow & & \downarrow \\ f(p) & \dashrightarrow & f(\nu f) \end{array}$$

Fixed point (co)induction = initiality and finality

$$
\begin{array}{ccc}
f(\mu f) & \dashrightarrow & f(p) \\
\downarrow & & \downarrow \\
\mu f & \dashrightarrow & p
\end{array}
\qquad\qquad
\begin{array}{ccc}
p & \dashrightarrow & \nu f \\
\downarrow & & \downarrow \\
f(p) & \dashrightarrow & f(\nu f)
\end{array}
$$

$$
\begin{array}{ccc}
F(A) & \dashrightarrow & F(S) \\
\downarrow & & \downarrow \\
A & \underset{\exists!}{\dashrightarrow} & S
\end{array}
\qquad\qquad
\begin{array}{ccc}
S & \overset{\exists!}{\dashrightarrow} & Z \\
\downarrow & & \downarrow \\
F(S) & \dashrightarrow & F(Z)
\end{array}
$$

Fixed point (co)induction = initiality and finality

$$
\begin{array}{ccc}
f(\mu f) & \dashrightarrow & f(p) \\
\downarrow & & \downarrow \\
\mu f & \dashrightarrow & p
\end{array}
\qquad\qquad
\begin{array}{ccc}
p & \dashrightarrow & \nu f \\
\downarrow & & \downarrow \\
f(p) & \dashrightarrow & f(\nu f)
\end{array}
$$

$$
\begin{array}{ccc}
F(A) & \dashrightarrow & F(S) \\
\downarrow & & \downarrow \\
A & \underset{\exists !}{\dashrightarrow} & S
\end{array}
\qquad\qquad
\begin{array}{ccc}
S & \overset{\exists !}{\dashrightarrow} & Z \\
\downarrow & & \downarrow \\
F(S) & \dashrightarrow & F(Z)
\end{array}
$$

4. The method of coalgebra

Summarizing the coalgebraic method

- The study of any class of coalgebras begins with the definition of its **type**, that is, a **functor** $F : \mathcal{C} \to \mathcal{C}$. Often, $\mathcal{C} = \mathsf{Set}$.

- The approach, which is essentially categorical, will be to describe what coalgebras **do**, rather than to specify what they **are**.

- The basis of the **local behaviour** of each coalgebra (S, α) is its structure map $\alpha : S \to F(S)$, which defines its local dynamics and outputs.

Summarizing the coalgebraic method

- The **global behaviour** of a coalgebra (S, α) is then given by its interaction with other coalgebras, that is, by **homomorphisms** between (S, α) and other coalgebras.

- The unique homomorphism into the **final** F-coalgebra assigns to every state a **canonical representation** of its global behaviour.

- Homomorphisms are structure preserving functions. Similarly, **bisimulations** are structure preserving relations. They are used in the formulation of the **coinduction** proof principle.

Examples of coalgebraic types

$$
\begin{array}{c} S \\ \downarrow{\scriptstyle \alpha} \\ A \end{array}
\qquad\qquad
\begin{array}{c} S \\ \downarrow{\scriptstyle \beta} \\ S \end{array}
\qquad\qquad
\begin{array}{c} S \\ \downarrow{\scriptstyle \gamma} \\ S^A \end{array}
$$

$$
\begin{array}{c} S \\ \downarrow{\scriptstyle \alpha} \\ S \end{array}
\qquad\qquad
\begin{array}{c} S \\ \downarrow{\scriptstyle \beta} \\ A \times S \end{array}
\qquad\qquad
\begin{array}{c} S \\ \downarrow{\scriptstyle \gamma} \\ 2 \times S^A \end{array}
\qquad\qquad
\begin{array}{c} S \\ \downarrow{\scriptstyle \delta} \\ A + S \end{array}
$$

Where coalgebra is used

- logic, set theory
- automata
- control theory
- combinatorics
- data types
- dynamical systems
- games

Where coalgebra is used

- economy
- ecology
- Kabbalah
  - Physarum Polycephalum Syllogistic L-Systems and Judaic Roots of Unconventional Computing, by A. Schumann. Studies in Logic, Grammar and Rhetoric, 2016.
  - Abstract:
    We show that in Kabbalah, the esoteric teaching of Judaism, there were developed ideas of unconventional automata in which ...
- and more ...

5. Discussion

- Relatively new way of thinking – give it time
- Extensive example: streams (Lecture two)
- Recent developments: obtaining the best of two worlds by combining algebra and coalgebra
  - Cf. CALCO
  - bisimulation up-to (cf. PhD thesis Jurriaan Rot)
  - Cf. Hacking nondeterminism with induction and coinduction Bonchi and Pous, Comm. ACM Vol. 58(2), 2015