Lecture one:

The Method of Coalgebra
– in some detail –

Jan Rutten

CWI Amsterdam & Radboud University Nijmegen

IPM, Tehran - 13 January 2016

Acknowledgements

My esteemed co-authors

- Marcello Bonsangue (Leiden, CWI)
- Helle Hansen (Delft)
- Alexandra Silva (City College London)
- Milad Niqui
- Clemens Kupke (Glasgow)
- Prakash Panangaden (McGill, Montreal)
- Filippo Bonchi (ENS, Lyon)
- Joost Winter (Warsaw), Jurriaan Rot (ENS, Lyon)
- and many others.

Samson Abramsky (Oxford)
Damien Pous (ENS, Lyon)

Acknowledgements

My esteemed co-authors

- Marcello Bonsangue (Leiden, CWI)
- Helle Hansen (Delft)
- Alexandra Silva (City College London)
- Milad Niqui
- Clemens Kupke (Glasgow)
- Prakash Panangaden (McGill, Montreal)
- Filippo Bonchi (ENS, Lyon)
- Joost Winter (Warsaw), Jurriaan Rot (ENS, Lyon)
- and many others.

Samson Abramsky (Oxford)
Damien Pous (ENS, Lyon)

Overview of todays lectures

Lecture one: The method of coalgebra

Lecture two: A coinductive calculus of streams

Lecture three: Automata and the algebra-coalgebra duality

Lecture four: Coalgebraic up-to techniques

Overview of Lecture one

1. Category theory (where coalgebra comes from)

2. Duality (where coalgebra comes from)

3. How coalgebra works (the method in slogans)

4. Duality: induction and coinduction

5. What coalgebra studies (its subject)

6. 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 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures.**
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures.**
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

Category Theory in 10 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. Categories as mathematical **contexts.**
5. Categories as mathematical **structures**.
6. Make definitions and constructions as **general** as possible.
7. **Functoriality!**
8. **Naturality!**
9. **Universality!**
10. **Adjoints** are everywhere.

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: $g \circ f$:

- **Axioms**:

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ id = f = id \circ f$$

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: $g \circ f$:

- **Axioms**:

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ id = f = id \circ f$$

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: $g \circ f$:

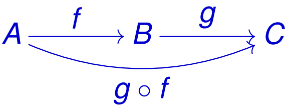
- **Axioms**:

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ id = f = id \circ f$$

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: $g \circ f$:



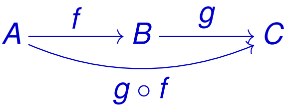
- **Axioms**:

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ id = f = id \circ f$$

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: $g \circ f$:



- **Axioms**:

$$h \circ (g \circ f) \;=\; (h \circ g) \circ f \qquad\qquad f \circ id \;=\; f = \; id \circ f$$

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

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

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

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

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

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

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

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

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

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \quad f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \quad f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \;\; f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \;\; f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \;\; g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \;\; f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \;\; f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \;\; g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \quad f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \;\; f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \;\; f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \;\; g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \quad f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

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

- **injective** if

$$\forall x, y \in X, \quad f(x) = f(y) \Rightarrow x = y$$

- **surjective** if

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

- **monic** if

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$

- **epic** if

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

Proposition

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

Arrows rather than elements

Defining the Cartesian **product** ...

- **with elements:**

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

where

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

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

Arrows rather than elements

Defining the Cartesian **product** ...

- **with elements:**

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

  where

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

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

Arrows rather than elements

Defining the Cartesian **product** . . .

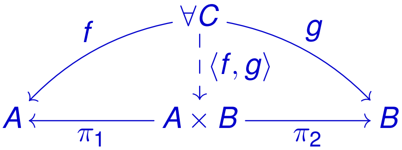
- **with elements:**

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

  where

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

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



2. Duality (where coalgebra comes from)

An additional slogan for categories: duality is omnipresent

- epi - mono
- product - sum
- initial object - final object
- algebra - coalgebra

2. Duality (where coalgebra comes from)

An additional slogan for categories: duality is omnipresent

- epi - mono
- product - sum
- initial object - final object
- algebra - coalgebra

Duality: monos and epis

- f is **monic**:

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$

- f is **epic**:

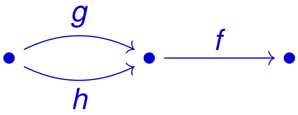
$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

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

Duality: monos and epis

- f is **monic**:

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$



- f is **epic**:

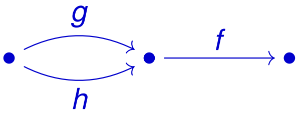
$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

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

Duality: monos and epis

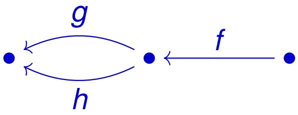
- f is **monic**:

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$



- f is **epic**:

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$

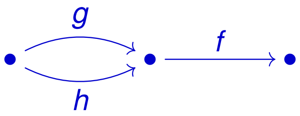


Proposition: f is monic in $\mathcal{C}$ iff f is epic in $\mathcal{C}^{op}$.

Duality: monos and epis

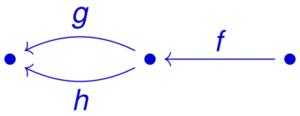
- f is **monic**:

$$\forall g, h, \quad f \circ g = f \circ h \Rightarrow g = h$$



- f is **epic**:

$$\forall g, h, \quad g \circ f = h \circ f \Rightarrow g = h$$



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

Duality: products and coproducts

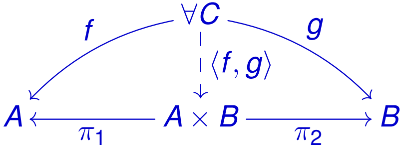
The **product** of A and B:

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:

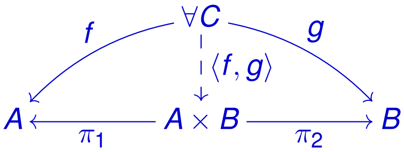


The **coproduct** of A and B:

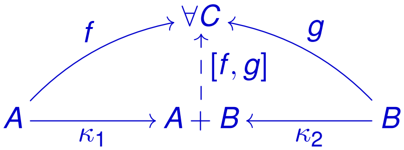
Proposition: O is product in $\mathcal{C}$ iff O is coproduct in $\mathcal{C}^{op}$.

Duality: products and coproducts

The **product** of A and B:



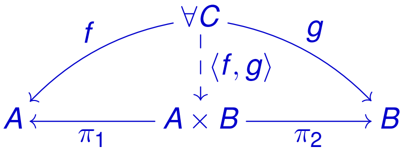
The **coproduct** of A and B:



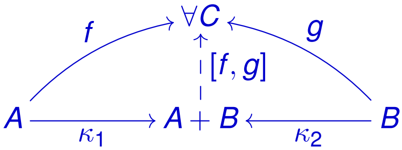
Proposition: O is product in $\mathcal{C}$ iff O is coproduct in $\mathcal{C}^{op}$.

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 ...

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

$$A \xrightarrow{!} B$$

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

$$B \xrightarrow{!} A$$

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

Proposition: Initial and final objects are unique up-to isomorphism.

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 $\mathcal{C}^{op}$.

Proposition: Initial and final objects are unique up-to isomorphism.

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 $\mathcal{C}^{op}$.

Proposition: Initial and final objects are unique up-to isomorphism.

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 $\mathcal{C}^{op}$.

Proposition: Initial and final objects are unique up-to isomorphism.

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 $\mathcal{C}^{op}$.

Proposition: Initial and final objects are unique up-to isomorphism.

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 $\mathcal{C}^{op}$.

Proposition: Initial and final objects are unique up-to isomorphism.

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,

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

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

Where coalgebra comes from

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

Classically, algebras are sets with operations.

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

Equivalently,

$$1 + \mathbb{N} \xrightarrow{[\mathsf{zero}, \mathsf{succ}]} \mathbb{N}$$

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

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,

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

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

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

where F is the type of the algebra.

Coalgebra, dually

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

where F is the type of the coalgebra.

Example: streams

Streams are our favourite example of a coalgebra:

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

where

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

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

3. How coalgebra works (its method in slogans)

- be precise about **types**
- ask what a system **does** rather than what it **is**
- **functoriality**
- interaction through **homomorphisms**
- aim for **universality**

Note that all these slogans are part of the categorical approach.

3. How coalgebra works (its method in slogans)

- be precise about **types**
- ask what a system **does** rather than what it **is**
- **functoriality**
- interaction through **homomorphisms**
- aim for **universality**

Note that all these slogans are part of the categorical approach.

Starting point: the system's type

A coalgebra of type F is a pair (X, α) with

$$\alpha : X \to F(X)$$

For instance, *non-deterministic transition systems*:

$$X \xrightarrow{\alpha} \mathcal{P}_f(A \times X) \qquad X \xrightarrow{\alpha} \mathcal{P}_f(X)^A \qquad X \xrightarrow{\alpha} \mathcal{P}(X)^A \qquad X \xrightarrow{\alpha} 2 \times \mathcal{P}(X)^A$$

Formally, the type F of a coalgebra/system is a **functor**.

The importance of knowing the system's type

The type F of a coalgebra/system

$$\alpha : X \to F(X)$$

determines

- a canonical notion of system equivalence: **bisimulation**
- a canonical notion of **minimization**
- a canonical interpretation: **final coalgebra** semantics
- (a canonical **logic**)

Doing versus being

Doing > Being

Behaviour > Construction

Systems as black boxes (with internal states)

Behavioural specification

Doing versus being

Doing $>$ Being

Behaviour $>$ Construction

Systems as black boxes (with internal states)

Behavioural specification

Doing versus being

Doing $>$ Being

Behaviour $>$ Construction

Systems as black boxes (with internal states)

Behavioural specification

Doing versus being

Doing $>$ Being

Behaviour $>$ Construction

Systems as black boxes (with internal states)

Behavioural specification

Doing versus being

Doing $>$ Being

Behaviour $>$ Construction

Systems as black boxes (with internal states)

Behavioural specification

Example: the shuffle product of streams

Being:

$$(\sigma \otimes \tau)(n) = \sum_{k=0}^{n} \binom{n}{k} \cdot \sigma(k) \cdot \tau(n-k)$$

Doing:

$$\sigma \otimes \tau \xrightarrow{\sigma(0) \cdot \tau(0)} (\sigma' \otimes \tau) + (\sigma \otimes \tau')$$

Example: the shuffle product of streams

Being:

$$(\sigma \otimes \tau)(n) = \sum_{k=0}^{n} \binom{n}{k} \cdot \sigma(k) \cdot \tau(n-k)$$

Doing:

$$\sigma \otimes \tau \xrightarrow{\sigma(0) \cdot \tau(0)} (\sigma' \otimes \tau) + (\sigma \otimes \tau')$$

Example: the shuffle product of streams

Being:

$$(\sigma \otimes \tau)(n) = \sum_{k=0}^{n} \binom{n}{k} \cdot \sigma(k) \cdot \tau(n-k)$$

Doing:

$$\sigma \otimes \tau \xrightarrow{\sigma(0) \cdot \tau(0)} (\sigma' \otimes \tau) + (\sigma \otimes \tau')$$

Example: the Hamming numbers

Being:

The increasing stream h of all natural numbers that are divisible by only 2, 3, or 5:

$$h = (1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, \dots)$$

$$h(n) = ?$$

Doing:

$$h \xrightarrow{1} (2 \cdot h) \parallel (3 \cdot h) \parallel (5 \cdot h)$$

Example: the Hamming numbers

Being:

The increasing stream h of all natural numbers that are divisible by only 2, 3, or 5:

$$h = (1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, \ldots)$$

$$h(n) = ?$$

Doing:

$$h \xrightarrow{1} (2 \cdot h) \parallel (3 \cdot h) \parallel (5 \cdot h)$$

Example: the Hamming numbers

Being:

The increasing stream h of all natural numbers that are divisible by only 2, 3, or 5:

$$h = (1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, \ldots)$$

$$h(n) = ?$$

Doing:

$$h \xrightarrow{1} (2 \cdot h) \parallel (3 \cdot h) \parallel (5 \cdot h)$$

Example: the Hamming numbers

Being:

The increasing stream h of all natural numbers that are divisible by only 2, 3, or 5:

$$h = (1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, \dots)$$

$$h(n) = ?$$

Doing:

$$h \xrightarrow{1} (2 \cdot h) \parallel (3 \cdot h) \parallel (5 \cdot h)$$

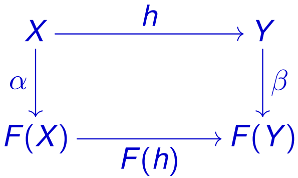
Homomorphisms

$$\begin{array}{ccc} X & \xrightarrow{\;h\;} & Y \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\beta} \\ F(X) & \xrightarrow[F(h)]{} & F(Y) \end{array}$$

... are for systems/coalgebras what **functions** are for sets.

... are **behaviour preserving** functions.

Functoriality



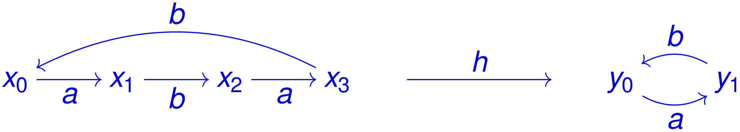
Note that for the definition of homomorphism,

the type F needs to be a **functor**:

F acts on sets: $F(X)$, $F(Y)$ **and** *on functions: $F(h)$*

Example of a homomorphism

$$
\begin{array}{ccc}
X & \xrightarrow{\quad h \quad} & Y \\
\downarrow & & \downarrow \\
O \times X & \xrightarrow[id \times h]{} & O \times Y
\end{array}
$$



Minimization through (canonical) homomorphism.

Universality

Always aim at universal/canonical formulations.

For instance: final coalgebras

In final coalgebras: Being $=$ Doing

$\Rightarrow$ coinduction (to be discussed shortly)

$\Rightarrow$ semantics

Universality

Always aim at universal/canonical formulations.

For instance: final coalgebras

In final coalgebras: Being $=$ Doing

$\Rightarrow$ coinduction (to be discussed shortly)

$\Rightarrow$ semantics

Universality

Always aim at universal/canonical formulations.

For instance: final coalgebras

In final coalgebras: Being $=$ Doing

$\Rightarrow$ coinduction (to be discussed shortly)

$\Rightarrow$ semantics

Universality

Always aim at universal/canonical formulations.

For instance: final coalgebras

In final coalgebras: Being $=$ Doing

$\Rightarrow$ coinduction (to be discussed shortly)

$\Rightarrow$ semantics

Universality

Always aim at universal/canonical formulations.

For instance: final coalgebras

In final coalgebras: Being $=$ Doing

$\Rightarrow$ coinduction (to be discussed shortly)

$\Rightarrow$ semantics

Universality

Always aim at universal/canonical formulations.

For instance: final coalgebras

In final coalgebras: Being $=$ Doing

$\Rightarrow$ coinduction (to be discussed shortly)

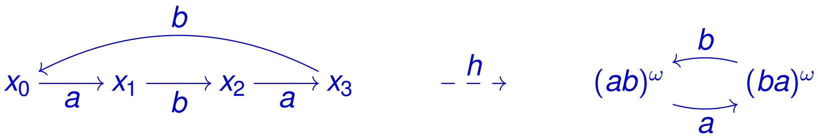
$\Rightarrow$ semantics

Semantics by finality: streams

The **final** homomorphism into the set of streams:

$$\begin{array}{ccc} X & \overset{\exists!\, h}{\dashrightarrow} & O^{\omega} \\ \downarrow & & \downarrow \\ O \times X & \xrightarrow[id \times h]{} & O \times O^{\omega} \end{array}$$

maps any system X to its **minimization**: e.g.,



$$x_0, x_2 \overset{h}{\longmapsto} (ab)^{\omega} \qquad x_1, x_3 \overset{h}{\longmapsto} (ba)^{\omega}$$

4. Duality: 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**:

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

Note: any **two** homomorphisms from $\mathbb{N}$ to S are **equal**.

Note: $id : \mathbb{N} \to \mathbb{N}$ is a homomorphism.

Note: $[\mathsf{zero}, \mathsf{succ}] : 1 + \mathbb{N} \cong \mathbb{N}$.

Initial algebra

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

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

Note: any **two** homomorphisms from $\mathbb{N}$ to S are **equal.**

Note: $id : \mathbb{N} \to \mathbb{N}$ is a homomorphism.

Note: $[\mathsf{zero}, \mathsf{succ}] : 1 + \mathbb{N} \cong \mathbb{N}$.

Initial algebra

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

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

Note: any **two** homomorphisms from $\mathbb{N}$ to S are **equal.**

Note: $id : \mathbb{N} \to \mathbb{N}$ is a homomorphism.

Note: $[\mathrm{zero}, \mathrm{succ}] : 1 + \mathbb{N} \cong \mathbb{N}$.

Initial algebra

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

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

Note: any **two** homomorphisms from $\mathbb{N}$ to S are **equal.**

Note: $id : \mathbb{N} \to \mathbb{N}$ is a homomorphism.

Note: $[\mathrm{zero}, \mathrm{succ}] : 1 + \mathbb{N} \cong \mathbb{N}$.

Final coalgebra

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

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

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

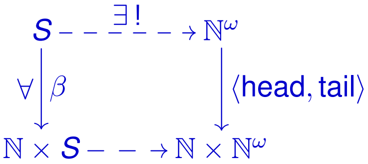
Note: any **two** homomorphisms from S to $\mathbb{N}^\omega$ are **equal**.

Note: $id : \mathbb{N}^\omega \to \mathbb{N}^\omega$ is a homomorphism.

Note: $\langle \mathsf{head}, \mathsf{tail} \rangle : \mathbb{N}^\omega \cong \mathbb{N} \times \mathbb{N}^\omega$.

Final coalgebra

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



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

Note: any **two** homomorphisms from S to $\mathbb{N}^\omega$ are **equal.**

Note: $\mathit{id} : \mathbb{N}^\omega \to \mathbb{N}^\omega$ is a homomorphism.

Note: $\langle \mathit{head}, \mathit{tail} \rangle : \mathbb{N}^\omega \cong \mathbb{N} \times \mathbb{N}^\omega$.

Final coalgebra

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

$$\begin{array}{ccc} S & \overset{\exists !}{- - - \to} & \mathbb{N}^\omega \\ {\scriptstyle \forall}\big\downarrow{\scriptstyle \beta} & & \big\downarrow{\scriptstyle \langle \mathit{head}, \mathit{tail} \rangle} \\ \mathbb{N} \times S & - - \to & \mathbb{N} \times \mathbb{N}^\omega \end{array}$$

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

Note: any **two** homomorphisms from S to $\mathbb{N}^\omega$ are **equal.**

Note: $id : \mathbb{N}^\omega \to \mathbb{N}^\omega$ is a homomorphism.

Note: $\langle \mathit{head}, \mathit{tail} \rangle : \mathbb{N}^\omega \cong \mathbb{N} \times \mathbb{N}^\omega$.

Final coalgebra

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

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

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

Note: any **two** homomorphisms from S to $\mathbb{N}^\omega$ are **equal.**

Note: $id : \mathbb{N}^\omega \to \mathbb{N}^\omega$ is a homomorphism.

Note: $\langle \text{head}, \text{tail} \rangle : \mathbb{N}^\omega \cong \mathbb{N} \times \mathbb{N}^\omega$.

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 $P \subseteq \mathbb{N}$,

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

(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.**

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

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

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

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

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

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

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

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 $n \in \mathbb{N}$.

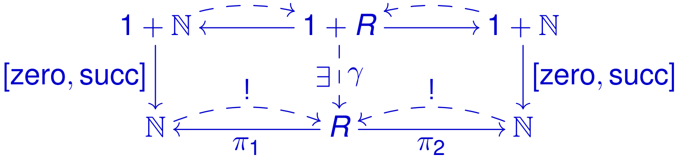
Initial algebras and congruences

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 $(\mathbb{N}, [\mathrm{zero}, \mathrm{succ}])$ 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 $P \subseteq \mathbb{N}$,

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

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

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

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

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

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

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,

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

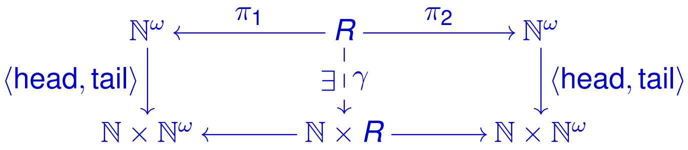
Final coalgebras and bisimulations

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 $(\mathbb{N}^\omega, \langle \mathrm{head}, \mathrm{tail} \rangle)$ is a final coalgebra,



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

Final coalgebras and coinduction

The following are equivalent:

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

$$R \subseteq \Delta$$

2. ??

In other words: no obvious equivalent formulation of **coinduction**!

Final coalgebras and coinduction

The following are equivalent:

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

$$R \subseteq \Delta$$

2. ??

In other words: no obvious equivalent formulation of **coinduction**!

Final coalgebras and coinduction

The following are equivalent:

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

$$R \subseteq \Delta$$

2. ??

In other words: no obvious equivalent formulation of **coinduction**!

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: dual?

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

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

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

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

Induction and coinduction: dual?

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: dual?

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:

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

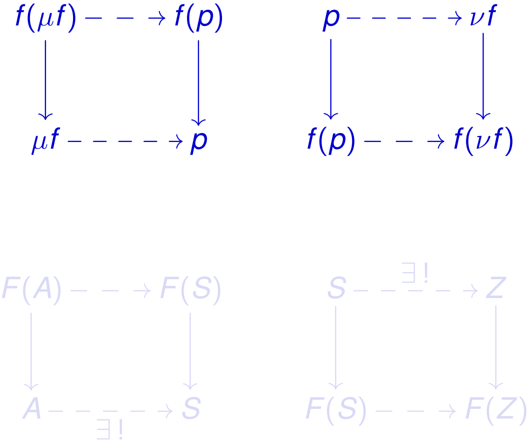
Any monotone map is a functor:

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

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



Fixed point (co)induction = initiality and finality

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

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

5. What coalgebra studies

- the behaviour of – often infinite, circular – **systems**

  (their equivalence, minimization, synthesis)
- rather: the **universal principles** underlying this behaviour
- these days applied in many different scientific disciplines

5. What coalgebra studies

- the behaviour of – often infinite, circular – **systems**

  (their equivalence, minimization, synthesis)
- rather: the **universal principles** underlying this behaviour
- these days applied in many different scientific disciplines

5. What coalgebra studies

- the behaviour of – often infinite, circular – **systems**

  (their equivalence, minimization, synthesis)
- rather: the **universal principles** underlying this behaviour
- these days applied in many different scientific disciplines

5. What coalgebra studies

- the behaviour of – often infinite, circular – **systems**

  (their equivalence, minimization, synthesis)
- rather: the **universal principles** underlying this behaviour
- these days applied in many different scientific disciplines

Example: dynamical systems

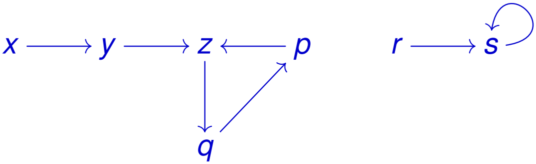
A dynamical system is:

set of states X and a transition function $t : X \to X$

Notation for transitions:

$$x \to y \quad \equiv \quad t(x) = y$$

Examples:

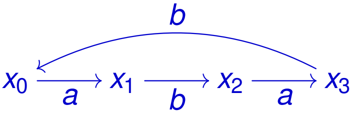


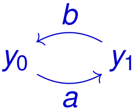
Example: systems with output

A system with output:

$$\langle o, t \rangle : X \to O \times X$$

Notation: $x \xrightarrow{a} y \quad \equiv \quad o(x) = a$ and $t(x) = y$.





Example: infinite data types

For instance, streams of natural numbers:

$$\mathbb{N}^\omega = \{\sigma \mid \sigma : \mathbb{N} \to \mathbb{N}\}$$

The behaviour of streams:

$$(\sigma(0), \sigma(1), \sigma(2), \ldots) \xrightarrow{\sigma(0)} (\sigma(1), \sigma(2), \sigma(3), \ldots)$$

where we call

$\sigma(0)$: the *initial value* (= head)

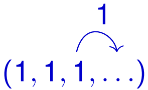
$\sigma' = (\sigma(1), \sigma(2), \sigma(3), \ldots)$: the *derivative* (= tail)

Example: streams

$$(1, 1, 1, \ldots) \xrightarrow{1} (1, 1, 1, \ldots) \xrightarrow{1} (1, 1, 1, \ldots) \xrightarrow{1} \cdots$$

Example: streams

$$(1, 1, 1, \ldots) \xrightarrow{1} (1, 1, 1, \ldots) \xrightarrow{1} (1, 1, 1, \ldots) \xrightarrow{1} \cdots$$



Example: streams

$$(1,2,3,\ldots) \xrightarrow{1} (2,3,4,\ldots) \xrightarrow{2} (3,4,5,\ldots) \xrightarrow{3} \cdots$$

$$(1,2,3,\ldots) \xrightarrow{1} (2,3,4,\ldots) = (1,2,3,\ldots) + (1,1,1,\ldots)$$

$$(1,2,3,\ldots) \xrightarrow{\quad 1 \quad} (1,1,1,\ldots)$$

Example: streams

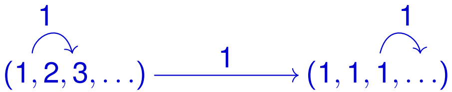
$$(1,2,3,\ldots) \xrightarrow{1} (2,3,4,\ldots) \xrightarrow{2} (3,4,5,\ldots) \xrightarrow{3} \cdots$$

$$(1,2,3,\ldots) \xrightarrow{1} (2,3,4,\ldots) = (1,2,3,\ldots) + (1,1,1,\ldots)$$

Example: streams

$$(1,2,3,\dots) \xrightarrow{\;1\;} (2,3,4,\dots) \xrightarrow{\;2\;} (3,4,5,\dots) \xrightarrow{\;3\;} \cdots$$

$$(1,2,3,\dots) \xrightarrow{\;1\;} (2,3,4,\dots) = (1,2,3,\dots) + (1,1,1,\dots)$$



Example: non-well-founded sets

Historically important: Peter Aczel's book.

$$x = \{x\} \qquad y = \{y\}$$

$$x = \{y\} \qquad y = \{z\} \qquad z = \{x, y\}$$

Example: non-well-founded sets

Historically important: Peter Aczel's book.

$$x = \{x\} \qquad y = \{y\}$$

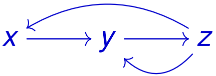
$$x = \{y\} \qquad y = \{z\} \qquad z = \{x, y\}$$

Example: non-well-founded sets

Historically important: Peter Aczel's book.

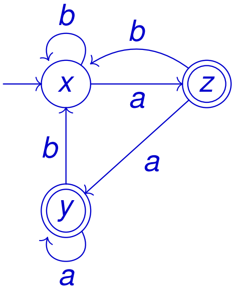
$$x = \{x\} \qquad y = \{y\}$$

$$x = \{y\} \qquad y = \{z\} \qquad z = \{x, y\}$$



Example: automata

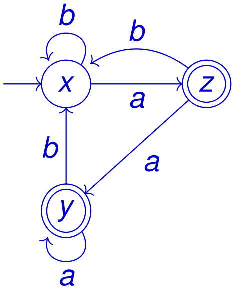
A deterministic automaton



- initial state: x • final states: y and z
- $L(x) = \{a, b\}^* a$

Example: automata

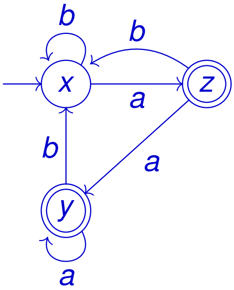
A deterministic automaton



- initial state: x
- final states: y and z
- $L(x) = \{a, b\}^* a$

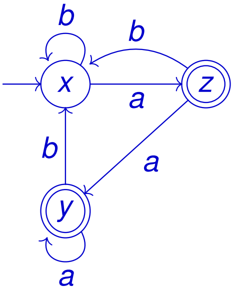
Example: automata

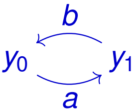
A deterministic automaton

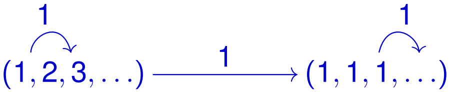


- initial state: x
- final states: y and z
- $L(x) \; = \; \{a, b\}^* \, a$

All these examples: **circular** behaviour







Where coalgebra is used

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

6. Discussion

- New way of thinking – give it time
- Extensive example: streams (Lecture two)
- Algebra and coalgebra (Lecture three and four)
  - bisimulation up-to
  - cf. CALCO
- Algorithms, tools (Lecture four)
  - Cf. Hacking nondeterminism with induction and coinduction
    Bonchi and Pous, Comm. ACM Vol. 58(2), 2015