Coalgebra Dreams

Samson Abramsky

Department of Computer Science, University of Oxford

Why I like Coalgebra

Why I like Coalgebra

- The semantics dilemma: to follow or to lead?

Why I like Coalgebra

- The semantics dilemma: to follow or to lead?
- Robin Milner: Information Dynamics

Why I like Coalgebra

- The semantics dilemma: to follow or to lead?
- Robin Milner: Information Dynamics
- Tony Hoare:

> *The primary objective of this paper Is to give a simple mathematical model for communicating sequential processes. As the exposition unfolds, the examples begin to look like programs, and the notations begin to look like a programming language. Thus the design of a language seems to emerge naturally from its formal definition, in an intellectually pleasing fashion.*

Why I like Coalgebra

- The semantics dilemma: to follow or to lead?
- Robin Milner: Information Dynamics
- Tony Hoare:

  The primary objective of this paper Is to give a simple mathematical model for communicating sequential processes. As the exposition unfolds, the examples begin to look like programs, and the notations begin to look like a programming language. Thus the design of a language seems to emerge naturally from its formal definition, in an intellectually pleasing fashion.

- Peter Landin

  For some years I have aspired to syntax-free programming

  From "A program-machine symmetric automata theory"

Why I like Coalgebra

- The semantics dilemma: to follow or to lead?
- Robin Milner: Information Dynamics
- Tony Hoare:

  > *The primary objective of this paper Is to give a simple mathematical model for communicating sequential processes. As the exposition unfolds, the examples begin to look like programs, and the notations begin to look like a programming language. Thus the design of a language seems to emerge naturally from its formal definition, in an intellectually pleasing fashion.*

- Peter Landin

  > *For some years I have aspired to syntax-free programming*

  From "A program-machine symmetric automata theory"

- Coalgebra embodies a semantics-led approach.

Why I like Coalgebra

- The semantics dilemma: to follow or to lead?
- Robin Milner: Information Dynamics
- Tony Hoare:

  > *The primary objective of this paper Is to give a simple mathematical model for communicating sequential processes. As the exposition unfolds, the examples begin to look like programs, and the notations begin to look like a programming language. Thus the design of a language seems to emerge naturally from its formal definition, in an intellectually pleasing fashion.*

- Peter Landin

  > *For some years I have aspired to syntax-free programming*

  From "A program-machine symmetric automata theory"

- Coalgebra embodies a semantics-led approach.

  Where can this take us?

Coalgebra: out of the comfort zone

Coalgebra: out of the comfort zone

In fact, a lot of work in coalgebra does, very reasonably, follow established lines of thought.

Coalgebra: out of the comfort zone

In fact, a lot of work in coalgebra does, very reasonably, follow established lines of thought.

Examples:

- recovering a wide range of salient notions in automata theory
- reworking ideas in modal logic in general coalgebraic form.

Coalgebra: out of the comfort zone

In fact, a lot of work in coalgebra does, very reasonably, follow established lines of thought.

Examples:

- recovering a wide range of salient notions in automata theory
- reworking ideas in modal logic in general coalgebraic form.

Also much to be done in exploring the applications of coinduction.

Coalgebra: out of the comfort zone

In fact, a lot of work in coalgebra does, very reasonably, follow established lines of thought.

Examples:

- recovering a wide range of salient notions in automata theory
- reworking ideas in modal logic in general coalgebraic form.

Also much to be done in exploring the applications of coinduction.

But can we go beyond this?

Coalgebra: out of the comfort zone

In fact, a lot of work in coalgebra does, very reasonably, follow established lines of thought.

Examples:

- recovering a wide range of salient notions in automata theory
- reworking ideas in modal logic in general coalgebraic form.

Also much to be done in exploring the applications of coinduction.

But can we go beyond this?

I have tried to encourage novel areas of application of coalgebra, e.g.

- SA, Coalgebras, Chu Spaces and Representations of Physical Systems, LiCS 2010.
- SA and J. Zvesper, From Lawvere to Brandenburger-Keisler: interactive forms of diagonalization and self-reference, CMCS 2012 and JCSS 2015.
- SA and V. Winschel, Coalgebraic analysis of subgame-perfect equilibria in infinite games without discounting, MSCS 2017.

Coalgebra Dreams

Coalgebra Dreams

Here I want to address issues in Computer Science itself.

Coalgebra Dreams

Here I want to address issues in Computer Science itself.

The dream is to use structural methods to solve hard problems.

Coalgebra Dreams

Here I want to address issues in Computer Science itself.

The dream is to use structural methods to solve hard problems.

Can it be done? Only if we are bold enough to try!

Coalgebra Dreams

Here I want to address issues in Computer Science itself.

The dream is to use structural methods to solve hard problems.

Can it be done? Only if we are bold enough to try!

Let's not forget to dream!

Structure *vs* Power: The Great Divide

Structure:

- compositionality, semantics
- How we can master the complexity of computer systems and software?

Power:

- expressiveness, complexity
- How we can harness the power of computation and recognize its limits?

Structure *vs* Power: The Great Divide

Structure:

- compositionality, semantics
- How we can master the complexity of computer systems and software?

Power:

- expressiveness, complexity
- How we can harness the power of computation and recognize its limits?

A *shocking fact*: the current state of the art is almost *disjoint communities* of researchers studying Structure and Power respectively, with no common technical language or tools.

Structure *vs* Power: The Great Divide

Structure:

- compositionality, semantics
- How we can master the complexity of computer systems and software?

Power:

- expressiveness, complexity
- How we can harness the power of computation and recognize its limits?

A *shocking fact*: the current state of the art is almost *disjoint communities* of researchers studying Structure and Power respectively, with no common technical language or tools.

This is a major obstacle to fundamental progress in Computer Science.

Structure *vs* Power: The Great Divide

Structure:

- compositionality, semantics
- How we can master the complexity of computer systems and software?

Power:

- expressiveness, complexity
- How we can harness the power of computation and recognize its limits?

A *shocking fact*: the current state of the art is almost *disjoint communities* of researchers studying Structure and Power respectively, with no common technical language or tools.

This is a major obstacle to fundamental progress in Computer Science.

An analogy: the Grothendieck program in algebraic geometry. The (very abstract) tools developed there were ultimately critical for concrete results, e.g. Wiles/FLT.

Structure *vs* Power: The Great Divide

Structure:

- compositionality, semantics
- How we can master the complexity of computer systems and software?

Power:

- expressiveness, complexity
- How we can harness the power of computation and recognize its limits?

A *shocking fact*: the current state of the art is almost *disjoint communities* of researchers studying Structure and Power respectively, with no common technical language or tools.

This is a major obstacle to fundamental progress in Computer Science.

An analogy: the Grothendieck program in algebraic geometry. The (very abstract) tools developed there were ultimately critical for concrete results, e.g. Wiles/FLT.

Mazur quoting Lenstra:

> *twenty years ago he was firm in his conviction that he DID want to solve Diophantine equations, and that he DID NOT wish to represent functors – and now he is amused to discover himself representing functors in order to solve Diophantine equations!*

The topic for this talk

We shall discuss a novel approach to relating categorical semantics, which exemplifies "Structure", to finite model theory, which exemplifies "Power".

The topic for this talk

We shall discuss a novel approach to relating categorical semantics, which exemplifies "Structure", to finite model theory, which exemplifies "Power".

Based on:

- "The Pebbling Comonad in Finite Model Theory", SA, Anuj Dawar and Pengming Wang, LiCS 2017.
- "Relating Structure to Power: comonadic semantics for computational resources", SA and Nihil Shah, extended abstract in CMCS proceedings, to appear in CSL 2018.

and ongoing work with Nihil Shah and Tom Paine.

Model theory and deception

Model theory and deception

- In model theory, we see a structure, not "as it really is" (up to isomorphism) but only up to *definable properties*.

Model theory and deception

- In model theory, we see a structure, not "as it really is" (up to isomorphism) but only up to *definable properties*.
- The crucial notion is equivalence of structures up to the equivalence $\equiv^{\mathcal{L}}$ induced by the logic $\mathcal{L}$:

$$\mathcal{A} \equiv^{\mathcal{L}} \mathcal{B} \quad \overset{\Delta}{\Longleftrightarrow} \quad \forall \varphi \in \mathcal{L}.\, \mathcal{A} \models \varphi \iff \mathcal{B} \models \varphi.$$

Model theory and deception

- In model theory, we see a structure, not "as it really is" (up to isomorphism) but only up to *definable properties*.
- The crucial notion is equivalence of structures up to the equivalence $\equiv^{\mathcal{L}}$ induced by the logic $\mathcal{L}$:

$$\mathcal{A} \equiv^{\mathcal{L}} \mathcal{B} \quad \overset{\Delta}{\Longleftrightarrow} \quad \forall \varphi \in \mathcal{L}.\, \mathcal{A} \models \varphi \iff \mathcal{B} \models \varphi.$$

- It is always true that if a class of structures $\mathcal{K}$ is definable in $\mathcal{L}$, then $\mathcal{K}$ must be saturated under $\equiv^{\mathcal{L}}$.

Model theory and deception

- In model theory, we see a structure, not "as it really is" (up to isomorphism) but only up to *definable properties*.
- The crucial notion is equivalence of structures up to the equivalence $\equiv^{\mathcal{L}}$ induced by the logic $\mathcal{L}$:

$$\mathcal{A} \equiv^{\mathcal{L}} \mathcal{B} \quad \overset{\Delta}{\Longleftrightarrow} \quad \forall \varphi \in \mathcal{L}.\, \mathcal{A} \models \varphi \iff \mathcal{B} \models \varphi.$$

- It is always true that if a class of structures $\mathcal{K}$ is definable in $\mathcal{L}$, then $\mathcal{K}$ must be saturated under $\equiv^{\mathcal{L}}$.
- In most cases of interest in FMT, the converse is true too.

Model theory and deception

- In model theory, we see a structure, not "as it really is" (up to isomorphism) but only up to *definable properties*.
- The crucial notion is equivalence of structures up to the equivalence $\equiv^{\mathcal{L}}$ induced by the logic $\mathcal{L}$:

$$\mathcal{A} \equiv^{\mathcal{L}} \mathcal{B} \quad \stackrel{\Delta}{\Longleftrightarrow} \quad \forall \varphi \in \mathcal{L}.\, \mathcal{A} \models \varphi \iff \mathcal{B} \models \varphi.$$

- It is always true that if a class of structures $\mathcal{K}$ is definable in $\mathcal{L}$, then $\mathcal{K}$ must be saturated under $\equiv^{\mathcal{L}}$.
- In most cases of interest in FMT, the converse is true too.
- In descriptive complexity, we seek to characterize a complexity class **C** (for decision problems) as those classes of structures $\mathcal{K}$ (e.g. graphs) definable in $\mathcal{L}$.

Syntax-independent characterizations of logical equivalence

Syntax-independent characterizations of logical equivalence

- A classic theme in Model theory: e.g. the Keisler-Shelah theorem.

Syntax-independent characterizations of logical equivalence

- A classic theme in Model theory: e.g. the Keisler-Shelah theorem.
- Especially important in finite model theory, where *model comparison games* such as Ehrenfeucht-Fraïssé games, pebble games and bisimulation games play a central role.

Syntax-independent characterizations of logical equivalence

- A classic theme in Model theory: e.g. the Keisler-Shelah theorem.
- Especially important in finite model theory, where *model comparison games* such as Ehrenfeucht-Fraïssé games, pebble games and bisimulation games play a central role.

The EF-game between $\mathcal{A}$ and $\mathcal{B}$. In the i'th round, Spoiler moves by choosing an element in A or B; Duplicator responds by choosing an element in the other structure. Duplicator wins after k rounds if the relation $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ is a partial isomorphism.

Syntax-independent characterizations of logical equivalence

- A classic theme in Model theory: e.g. the Keisler-Shelah theorem.
- Especially important in finite model theory, where *model comparison games* such as Ehrenfeucht-Fraïssé games, pebble games and bisimulation games play a central role.

The EF-game between $\mathcal{A}$ and $\mathcal{B}$. In the i'th round, Spoiler moves by choosing an element in A or B; Duplicator responds by choosing an element in the other structure. Duplicator wins after k rounds if the relation $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ is a partial isomorphism.

In the existential EF-game, Spoiler only plays in $\mathcal{A}$, and Duplicator responds in $\mathcal{B}$.

Syntax-independent characterizations of logical equivalence

- A classic theme in Model theory: e.g. the Keisler-Shelah theorem.
- Especially important in finite model theory, where *model comparison games* such as Ehrenfeucht-Fraïssé games, pebble games and bisimulation games play a central role.

The EF-game between $\mathcal{A}$ and $\mathcal{B}$. In the i'th round, Spoiler moves by choosing an element in A or B; Duplicator responds by choosing an element in the other structure. Duplicator wins after k rounds if the relation $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ is a partial isomorphism.

In the existential EF-game, Spoiler only plays in $\mathcal{A}$, and Duplicator responds in $\mathcal{B}$.

The Ehrenfeucht-Fraïssé theorem says that a winning strategy for Duplicator in the k-round EF game characterizes the equivalence $\equiv^{\mathcal{L}_k}$, where $\mathcal{L}_k$ is the fragment of first-order logic of formulas with quantifier rank $\leq k$.

Syntax-independent characterizations of logical equivalence

- A classic theme in Model theory: e.g. the Keisler-Shelah theorem.
- Especially important in finite model theory, where *model comparison games* such as Ehrenfeucht-Fraïssé games, pebble games and bisimulation games play a central role.

The EF-game between $\mathcal{A}$ and $\mathcal{B}$. In the i'th round, Spoiler moves by choosing an element in A or B; Duplicator responds by choosing an element in the other structure. Duplicator wins after k rounds if the relation $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ is a partial isomorphism.

In the existential EF-game, Spoiler only plays in $\mathcal{A}$, and Duplicator responds in $\mathcal{B}$.

The Ehrenfeucht-Fraïssé theorem says that a winning strategy for Duplicator in the k-round EF game characterizes the equivalence $\equiv^{\mathcal{L}_k}$, where $\mathcal{L}_k$ is the fragment of first-order logic of formulas with quantifier rank $\leq k$.

Similarly, there are k-pebble games, and bismulation games played to depth k.

Pebble Games

Pebble Games

Similar but subtly different to EF-games

Pebble Games

Similar but subtly different to EF-games

Spoiler moves by placing one from a fixed set of pebbles on an element of $\mathcal{A}$ or $\mathcal{B}$; Duplicator responds by placing their matching pebble on an element of the other structure.

Pebble Games

Similar but subtly different to EF-games

Spoiler moves by placing one from a fixed set of pebbles on an element of $\mathcal{A}$ or $\mathcal{B}$; Duplicator responds by placing their matching pebble on an element of the other structure.

Duplicator wins if after each round, the relation defined by the current positions of the pebbles is a partial isomorphism

Pebble Games

Similar but subtly different to EF-games

Spoiler moves by placing one from a fixed set of pebbles on an element of $\mathcal{A}$ or $\mathcal{B}$; Duplicator responds by placing their matching pebble on an element of the other structure.

Duplicator wins if after each round, the relation defined by the current positions of the pebbles is a partial isomorphism

Thus there is a "sliding window" on the structures, of fixed size. It is this size which bounds the resource, not the length of the play.

Pebble Games

Similar but subtly different to EF-games

Spoiler moves by placing one from a fixed set of pebbles on an element of $\mathcal{A}$ or $\mathcal{B}$; Duplicator responds by placing their matching pebble on an element of the other structure.

Duplicator wins if after each round, the relation defined by the current positions of the pebbles is a partial isomorphism

Thus there is a "sliding window" on the structures, of fixed size. It is this size which bounds the resource, not the length of the play.

Whereas the k-round EF game corresponds to bounding the quantifier rank, k-pebble games correspond to bounding the number of variables which can be used in a formula.

Pebble Games

Similar but subtly different to EF-games

Spoiler moves by placing one from a fixed set of pebbles on an element of $\mathcal{A}$ or $\mathcal{B}$; Duplicator responds by placing their matching pebble on an element of the other structure.

Duplicator wins if after each round, the relation defined by the current positions of the pebbles is a partial isomorphism

Thus there is a "sliding window" on the structures, of fixed size. It is this size which bounds the resource, not the length of the play.

Whereas the k-round EF game corresponds to bounding the quantifier rank, k-pebble games correspond to bounding the number of variables which can be used in a formula.

Just as for EF-games, there is an existential-positive version, in which Spoiler only plays in $\mathcal{A}$, and Duplicator responds in $\mathcal{B}$.

A new perspective

A new perspective

- We shall study these games, not as external artefacts, but as semantic constructions in their own right.

A new perspective

- We shall study these games, not as external artefacts, but as semantic constructions in their own right.
- For each type of game G, and value of the resource parameter k, we shall define a corresponding *comonad* $\mathbb{C}_k$ on $\mathcal{R}(\sigma)$.

A new perspective

- We shall study these games, not as external artefacts, but as semantic constructions in their own right.
- For each type of game G, and value of the resource parameter k, we shall define a corresponding *comonad* $\mathbb{C}_k$ on $\mathcal{R}(\sigma)$.
- The idea is that Duplicator strategies for the existential version of G-games from $\mathcal{A}$ to $\mathcal{B}$ will be recovered as coKleisli morphisms $\mathbb{C}_k\mathcal{A} \to \mathcal{B}$.

A new perspective

- We shall study these games, not as external artefacts, but as semantic constructions in their own right.
- For each type of game G, and value of the resource parameter k, we shall define a corresponding *comonad* $\mathbb{C}_k$ on $\mathcal{R}(\sigma)$.
- The idea is that Duplicator strategies for the existential version of G-games from $\mathcal{A}$ to $\mathcal{B}$ will be recovered as coKleisli morphisms $\mathbb{C}_k\mathcal{A} \to \mathcal{B}$.
- Thus the notion of local approximation built into the game is internalised into the category of σ-structures and homomorphisms.

A new perspective

- We shall study these games, not as external artefacts, but as semantic constructions in their own right.
- For each type of game G, and value of the resource parameter k, we shall define a corresponding *comonad* $\mathbb{C}_k$ on $\mathcal{R}(\sigma)$.
- The idea is that Duplicator strategies for the existential version of G-games from $\mathcal{A}$ to $\mathcal{B}$ will be recovered as coKleisli morphisms $\mathbb{C}_k\mathcal{A} \to \mathcal{B}$.
- Thus the notion of local approximation built into the game is internalised into the category of σ-structures and homomorphisms.
- This leads to comonadic and coalgebraic characterisations of a number of central concepts in Finite Model Theory and combinatorics.

The setting: homomorphisms of relational structures

The setting: homomorphisms of relational structures

A relational vocabulary σ is a family of relation symbols R, each of some arity $n > 0$.

The setting: homomorphisms of relational structures

A relational vocabulary σ is a family of relation symbols R, each of some arity $n > 0$.

A relational structure for σ is $\mathcal{A} = (A, \{R^A \mid R \in \sigma\})$, where $R^{\mathcal{A}} \subseteq A^n$.

The setting: homomorphisms of relational structures

A relational vocabulary σ is a family of relation symbols R, each of some arity $n > 0$.

A relational structure for σ is $\mathcal{A} = (A, \{R^A \mid R \in \sigma\})$, where $R^{\mathcal{A}} \subseteq A^n$.

A homomorphism of σ-structures $f : \mathcal{A} \to \mathcal{B}$ is a function $f : A \to B$ such that, for each relation $R \in \sigma$ of arity n and $(a_1, \ldots, a_n) \in A^n$:

$$(a_1, \ldots, a_n) \in R^{\mathcal{A}} \;\Rightarrow\; (f(a_1), \ldots, f(a_n))) \in R^{\mathcal{B}}.$$

The setting: homomorphisms of relational structures

A relational vocabulary σ is a family of relation symbols R, each of some arity $n > 0$.

A relational structure for σ is $\mathcal{A} = (A, \{R^A \mid R \in \sigma\}))$, where $R^{\mathcal{A}} \subseteq A^n$.

A homomorphism of σ-structures $f : \mathcal{A} \to \mathcal{B}$ is a function $f : A \to B$ such that, for each relation $R \in \sigma$ of arity n and $(a_1, \ldots, a_n) \in A^n$:

$$(a_1, \ldots, a_n) \in R^{\mathcal{A}} \;\Rightarrow\; (f(a_1), \ldots, f(a_n))) \in R^{\mathcal{B}}.$$

There notions are pervasive in

- logic (model theory),
- computer science (databases, constraint satisfaction, finite model theory)
- combinatorics (graphs and graph homomorphisms).

The setting: homomorphisms of relational structures

A relational vocabulary σ is a family of relation symbols R, each of some arity $n > 0$.

A relational structure for σ is $\mathcal{A} = (A, \{R^A \mid R \in \sigma\}))$, where $R^{\mathcal{A}} \subseteq A^n$.

A homomorphism of σ-structures $f : \mathcal{A} \to \mathcal{B}$ is a function $f : A \to B$ such that, for each relation $R \in \sigma$ of arity n and $(a_1, \ldots, a_n) \in A^n$:

$$(a_1, \ldots, a_n) \in R^{\mathcal{A}} \;\Rightarrow\; (f(a_1), \ldots, f(a_n))) \in R^{\mathcal{B}}.$$

There notions are pervasive in

- logic (model theory),
- computer science (databases, constraint satisfaction, finite model theory)
- combinatorics (graphs and graph homomorphisms).

Our setting will be $\mathcal{R}(\sigma)$, the category of relational structures and homomorphisms.

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{E}_k\mathcal{A}}$ to the set of pairs (s, t) such that

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{E}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$ (in prefix order)

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{E}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$ (in prefix order)
- $R^{\mathcal{A}}(\varepsilon_{\mathcal{A}}(s), \varepsilon_{\mathcal{A}}(t))$.

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{E}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$ (in prefix order)
- $R^{\mathcal{A}}(\varepsilon_{\mathcal{A}}(s), \varepsilon_{\mathcal{A}}(t))$.

Given a homomorphism $f : \mathbb{E}_k\mathcal{A} \to \mathcal{B}$, we define the coextension $f^* : A^{\leq k} \to B^{\leq k}$ by

$$f^*[a_1, \ldots, a_j] = [b_1, \ldots, b_j],$$

where $b_i = f[a_1, \ldots, a_i]$, $1 \leq i \leq j$.

The EF comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{E}_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{E}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$ (in prefix order)
- $R^{\mathcal{A}}(\varepsilon_{\mathcal{A}}(s), \varepsilon_{\mathcal{A}}(t))$.

Given a homomorphism $f : \mathbb{E}_k\mathcal{A} \to \mathcal{B}$, we define the coextension $f^* : A^{\leq k} \to B^{\leq k}$ by

$$f^*[a_1, \ldots, a_j] = [b_1, \ldots, b_j],$$

where $b_i = f[a_1, \ldots, a_i]$, $1 \leq i \leq j$.

This is easily verified to yield a comonad on $\mathcal{R}(\sigma)$.

CoKleisli maps are strategies

Intuitively, an element of $A^{\leq k}$ represents a play in $\mathcal{A}$ of length $\leq k$.

CoKleisli maps are strategies

Intuitively, an element of $A^{\leq k}$ represents a play in $\mathcal{A}$ of length $\leq k$.

A coKleisli morphism $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ represents a Duplicator strategy for the existential Ehrenfeucht-Fraïssé game with k rounds:

CoKleisli maps are strategies

Intuitively, an element of $A^{\leq k}$ represents a play in $\mathcal{A}$ of length $\leq k$.

A coKleisli morphism $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ represents a Duplicator strategy for the existential Ehrenfeucht-Fraïssé game with k rounds:

Spoiler plays only in $\mathcal{A}$, and $b_i = f[a_1, \ldots, a_i]$ represents Duplicator's response in $\mathcal{B}$ to the i'th move by Spoiler.

CoKleisli maps are strategies

Intuitively, an element of $A^{\leq k}$ represents a play in $\mathcal{A}$ of length $\leq k$.

A coKleisli morphism $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ represents a Duplicator strategy for the existential Ehrenfeucht-Fraïssé game with k rounds:

Spoiler plays only in $\mathcal{A}$, and $b_i = f[a_1, \ldots, a_i]$ represents Duplicator's response in $\mathcal{B}$ to the i'th move by Spoiler.

The winning condition for Duplicator in this game is that, after k rounds have been played, the induced relation $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ is a partial homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

CoKleisli maps are strategies

Intuitively, an element of $A^{\leq k}$ represents a play in $\mathcal{A}$ of length $\leq k$.

A coKleisli morphism $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ represents a Duplicator strategy for the existential Ehrenfeucht-Fraïssé game with k rounds:

Spoiler plays only in $\mathcal{A}$, and $b_i = f[a_1, \ldots, a_i]$ represents Duplicator's response in $\mathcal{B}$ to the i'th move by Spoiler.

The winning condition for Duplicator in this game is that, after k rounds have been played, the induced relation $\{(a_i, b_i) \mid 1 \leq i \leq k\}$ is a partial homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Theorem

The following are equivalent:

1. *There is a homomorphism $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$.*
2. *Duplicator has a winning strategy for the existential Ehrenfeucht-Fraïssé game with k rounds, played from $\mathcal{A}$ to $\mathcal{B}$.*
3. *For every existential positive sentence φ with quantifier rank $\leq k$, $\mathcal{A} \models \varphi \;\Rightarrow\; \mathcal{B} \models \varphi$.*

The pebbling comonad

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{P}_k\mathcal{A}}$ to the set of pairs (s, t) such that

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{P}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{P}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$
- If $s \sqsubseteq t$, then the pebble index of the last move in s does not appear in the suffix of s in t; and symmetrically if $t \sqsubseteq s$.

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{P}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$
- If $s \sqsubseteq t$, then the pebble index of the last move in s does not appear in the suffix of s in t; and symmetrically if $t \sqsubseteq s$.
- $R^{\mathcal{A}}(\varepsilon_{\mathcal{A}}(s), \varepsilon_{\mathcal{A}}(t))$.

The pebbling comonad

Given a structure $\mathcal{A}$, the universe of $\mathbb{P}_k\mathcal{A}$ is $(\mathbf{k} \times A)^+$, the set of finite non-empty sequences of moves (p, a). Note this will be infinite even if $\mathcal{A}$ is finite.
We showed that this is essential!

The counit map $\varepsilon_{\mathcal{A}} : \mathbb{E}_k\mathcal{A} \to \mathcal{A}$ sends a sequence $[(p_1, a_1), \ldots, (p_n, a_n)]$ to a_n.

How do we lift the relations on $\mathcal{A}$ to $\mathbb{E}_k\mathcal{A}$?

Given e.g. a binary relation R, we define $R^{\mathbb{P}_k\mathcal{A}}$ to the set of pairs (s, t) such that

- $s \sqsubseteq t$ or $t \sqsubseteq s$
- If $s \sqsubseteq t$, then the pebble index of the last move in s does not appear in the suffix of s in t; and symmetrically if $t \sqsubseteq s$.
- $R^{\mathcal{A}}(\varepsilon_{\mathcal{A}}(s), \varepsilon_{\mathcal{A}}(t))$.

Given a homomorphism $f : \mathbb{P}_k\mathcal{A} \to \mathcal{B}$, we define the coextension $f^* : \mathbb{P}_k\mathcal{A} \to \mathbb{P}_k\mathcal{B}$ by

$$f^*[(p_1, a_1), \ldots, (p_j, a_j)] = [(p_1, b_1), \ldots, (p_j, b_j)],$$

where $b_i = f[(p_1, a_1), \ldots, (p_i, a_i)]$, $1 \leq i \leq j$.

The modal comonad

The modal comonad

The flexibility of the comonadic approach is illustrated by showing that it also covers the well-known construction of unfolding a Kripke structure into a tree ("unravelling").

The modal comonad

The flexibility of the comonadic approach is illustrated by showing that it also covers the well-known construction of unfolding a Kripke structure into a tree ("unravelling").

For the modal case, we assume that the relational vocabulary σ contains only symbols of arity at most 2.

The modal comonad

The flexibility of the comonadic approach is illustrated by showing that it also covers the well-known construction of unfolding a Kripke structure into a tree ("unravelling").

For the modal case, we assume that the relational vocabulary σ contains only symbols of arity at most 2.

We can thus regard a σ-structure as a Kripke structure for a multi-modal logic. If there are no unary symbols, such structures are exactly the labelled transition systems.

The modal comonad

The flexibility of the comonadic approach is illustrated by showing that it also covers the well-known construction of unfolding a Kripke structure into a tree ("unravelling").

For the modal case, we assume that the relational vocabulary σ contains only symbols of arity at most 2.

We can thus regard a σ-structure as a Kripke structure for a multi-modal logic. If there are no unary symbols, such structures are exactly the labelled transition systems.

Modal logic localizes its notion of satisfaction in a structure to a world.
We reflect this by using the category of *pointed relational structures* $(\mathcal{A}, a)$.

The modal comonad

The flexibility of the comonadic approach is illustrated by showing that it also covers the well-known construction of unfolding a Kripke structure into a tree ("unravelling").

For the modal case, we assume that the relational vocabulary σ contains only symbols of arity at most 2.

We can thus regard a σ-structure as a Kripke structure for a multi-modal logic. If there are no unary symbols, such structures are exactly the labelled transition systems.

Modal logic localizes its notion of satisfaction in a structure to a world.
We reflect this by using the category of *pointed relational structures* $(\mathcal{A}, a)$.

For $k > 0$ we define a comonad $\mathbb{M}_k$, where $\mathbb{M}_k(\mathcal{A}, a)$ corresponds to unravelling the structure $\mathcal{A}$, starting from a, to depth k.

The modal comonad

The flexibility of the comonadic approach is illustrated by showing that it also covers the well-known construction of unfolding a Kripke structure into a tree ("unravelling").

For the modal case, we assume that the relational vocabulary σ contains only symbols of arity at most 2.

We can thus regard a σ-structure as a Kripke structure for a multi-modal logic. If there are no unary symbols, such structures are exactly the labelled transition systems.

Modal logic localizes its notion of satisfaction in a structure to a world.
We reflect this by using the category of *pointed relational structures* $(\mathcal{A}, a)$.

For $k > 0$ we define a comonad $\mathbb{M}_k$, where $\mathbb{M}_k(\mathcal{A}, a)$ corresponds to unravelling the structure $\mathcal{A}$, starting from a, to depth k.

The universe of $\mathbb{M}_k(\mathcal{A}, a)$ comprises $[a]$, which is the distinguished element, together with all sequences of the form $[a_0, \alpha_1, a_1, \ldots, \alpha_j, a_j]$, where $a = a_0$, $1 \leq j \leq k$, and $R^{\mathcal{A}}_{\alpha_i}(a_i, a_{i+1})$, $0 \leq i < j$.

Logical equivalences

Logical equivalences

For each of our three types of game, there are corresponding fragments $\mathcal{L}_k$ of first-order logic:

Logical equivalences

For each of our three types of game, there are corresponding fragments $\mathcal{L}_k$ of first-order logic:

- For Ehrenfeucht-Fraïssé games, $\mathcal{L}_k$ is the fragment of quantifier-rank $\leq k$.

Logical equivalences

For each of our three types of game, there are corresponding fragments $\mathcal{L}_k$ of first-order logic:

- For Ehrenfeucht-Fraïssé games, $\mathcal{L}_k$ is the fragment of quantifier-rank $\leq k$.
- For pebble games, $\mathcal{L}_k$ is the k-variable fragment.

Logical equivalences

For each of our three types of game, there are corresponding fragments $\mathcal{L}_k$ of first-order logic:

- For Ehrenfeucht-Fraïssé games, $\mathcal{L}_k$ is the fragment of quantifier-rank $\leq k$.
- For pebble games, $\mathcal{L}_k$ is the k-variable fragment.
- For bismulation games over relational vocabularies with symbols of arity at most 2, $\mathcal{L}_k$ is the modal fragment with modal depth $\leq k$.

Logical equivalences

For each of our three types of game, there are corresponding fragments $\mathcal{L}_k$ of first-order logic:

- For Ehrenfeucht-Fraïssé games, $\mathcal{L}_k$ is the fragment of quantifier-rank $\leq k$.
- For pebble games, $\mathcal{L}_k$ is the k-variable fragment.
- For bismulation games over relational vocabularies with symbols of arity at most 2, $\mathcal{L}_k$ is the modal fragment with modal depth $\leq k$.

In each case, we write

- $\exists\mathcal{L}_k$ for the existential positive fragment of $\mathcal{L}_k$
- $\mathcal{L}_k(\#)$ for the extension of $\mathcal{L}_k$ with counting quantifiers $\exists_{\leq n}$, $\exists_{\geq n}$

Characterization

Characterization

We can generically define two equivalences based on our indexed comonads $\mathbb{E}_k$:

- $\mathcal{A} \rightleftarrows_k^{\mathbb{E}} \mathcal{B}$ iff there are coKleisli morphisms $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ and $\mathbb{E}_k\mathcal{B} \to \mathcal{A}$. Note that there need be no relationship between these morphisms.
- $\mathcal{A} \cong_k^{\mathbb{E}} \mathcal{B}$ iff $\mathcal{A}$ and $\mathcal{B}$ are isomorphic in the coKleisli category $\mathsf{Kl}(\mathbb{E}_k)$.

Characterization

We can generically define two equivalences based on our indexed comonads $\mathbb{E}_k$:

- $\mathcal{A} \rightleftarrows_k^{\mathbb{E}} \mathcal{B}$ iff there are coKleisli morphisms $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ and $\mathbb{E}_k\mathcal{B} \to \mathcal{A}$. Note that there need be no relationship between these morphisms.

- $\mathcal{A} \cong_k^{\mathbb{E}} \mathcal{B}$ iff $\mathcal{A}$ and $\mathcal{B}$ are isomorphic in the coKleisli category $\mathsf{Kl}(\mathbb{E}_k)$.

To complete the picture, we need to show how to define a back-and-forth equivalence $\leftrightarrow_k$ which characterizes $\equiv^{\mathcal{L}_k}$ *purely in terms of coKleisli morphisms*.

Characterization

We can generically define two equivalences based on our indexed comonads $\mathbb{E}_k$:

- $\mathcal{A} \rightleftarrows_k^{\mathbb{E}} \mathcal{B}$ iff there are coKleisli morphisms $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ and $\mathbb{E}_k\mathcal{B} \to \mathcal{A}$. Note that there need be no relationship between these morphisms.

- $\mathcal{A} \cong_k^{\mathbb{E}} \mathcal{B}$ iff $\mathcal{A}$ and $\mathcal{B}$ are isomorphic in the coKleisli category $\mathsf{KI}(\mathbb{E}_k)$.

To complete the picture, we need to show how to define a back-and-forth equivalence $\leftrightarrow_k$ which characterizes $\equiv^{\mathcal{L}_k}$ *purely in terms of coKleisli morphisms*.

Our solution to this, while not completely generic, is general enough to apply to all our game comonads – so we subsume EF equivalence, bisimulation equivalence and pebble game equivalence as instances of a single construction.

Characterization

We can generically define two equivalences based on our indexed comonads $\mathbb{E}_k$:

- $\mathcal{A} \rightleftarrows_k^{\mathbb{E}} \mathcal{B}$ iff there are coKleisli morphisms $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ and $\mathbb{E}_k\mathcal{B} \to \mathcal{A}$. Note that there need be no relationship between these morphisms.

- $\mathcal{A} \cong_k^{\mathbb{E}} \mathcal{B}$ iff $\mathcal{A}$ and $\mathcal{B}$ are isomorphic in the coKleisli category $\mathsf{KI}(\mathbb{E}_k)$.

To complete the picture, we need to show how to define a back-and-forth equivalence $\leftrightarrow_k$ which characterizes $\equiv^{\mathcal{L}_k}$ *purely in terms of coKleisli morphisms*.

Our solution to this, while not completely generic, is general enough to apply to all our game comonads – so we subsume EF equivalence, bisimulation equivalence and pebble game equivalence as instances of a single construction.

An interesting feature is that it can be described in terms of approximations and fixpoints. We use total coKleisli morphisms to approximate partial isomorphisms.

Characterization

We can generically define two equivalences based on our indexed comonads $\mathbb{E}_k$:

- $\mathcal{A} \rightleftarrows_k^{\mathbb{E}} \mathcal{B}$ iff there are coKleisli morphisms $\mathbb{E}_k\mathcal{A} \to \mathcal{B}$ and $\mathbb{E}_k\mathcal{B} \to \mathcal{A}$. Note that there need be no relationship between these morphisms.

- $\mathcal{A} \cong_k^{\mathbb{E}} \mathcal{B}$ iff $\mathcal{A}$ and $\mathcal{B}$ are isomorphic in the coKleisli category $\mathsf{KI}(\mathbb{E}_k)$.

To complete the picture, we need to show how to define a back-and-forth equivalence $\leftrightarrow_k$ which characterizes $\equiv^{\mathcal{L}_k}$ *purely in terms of coKleisli morphisms*.

Our solution to this, while not completely generic, is general enough to apply to all our game comonads – so we subsume EF equivalence, bisimulation equivalence and pebble game equivalence as instances of a single construction.

An interesting feature is that it can be described in terms of approximations and fixpoints. We use total coKleisli morphisms to approximate partial isomorphisms.

For details, see the CSL paper.

Characterizing logical equivalences

Characterizing logical equivalences

Theorem

For structures $\mathcal{A}$ and $\mathcal{B}$:

(1) $\mathcal{A} \equiv^{\exists \mathcal{L}_k} \mathcal{B} \iff \mathcal{A} \rightleftarrows_k \mathcal{B}.$

(2) $\mathcal{A} \equiv^{\mathcal{L}_k} \mathcal{B} \iff \mathcal{A} \leftrightarrow_k \mathcal{B}.$

(3) $\mathcal{A} \equiv^{\mathcal{L}_k(\#)} \mathcal{B} \iff \mathcal{A} \cong_{\mathsf{Kl}(\mathbb{C}_k)} \mathcal{B}.$

Characterizing logical equivalences

Theorem

For structures $\mathcal{A}$ and $\mathcal{B}$:

(1) $\mathcal{A} \equiv^{\exists \mathcal{L}_k} \mathcal{B} \iff \mathcal{A} \rightleftarrows_k \mathcal{B}.$

(2) $\mathcal{A} \equiv^{\mathcal{L}_k} \mathcal{B} \iff \mathcal{A} \leftrightarrow_k \mathcal{B}.$

(3) $\mathcal{A} \equiv^{\mathcal{L}_k(\#)} \mathcal{B} \iff \mathcal{A} \cong_{\mathsf{Kl}(\mathbb{C}_k)} \mathcal{B}.$

Note that this is really a family of three theorems, one for each type of game G.

Characterizing logical equivalences

Theorem

For structures $\mathcal{A}$ and $\mathcal{B}$:

$$
\begin{array}{lllll}
(1) & \mathcal{A} \equiv^{\exists \mathcal{L}_k} \mathcal{B} & \iff & \mathcal{A} \rightleftarrows_k \mathcal{B}. \\
(2) & \mathcal{A} \equiv^{\mathcal{L}_k} \mathcal{B} & \iff & \mathcal{A} \leftrightarrow_k \mathcal{B}. \\
(3) & \mathcal{A} \equiv^{\mathcal{L}_k(\#)} \mathcal{B} & \iff & \mathcal{A} \cong_{\mathsf{Kl}(\mathbb{C}_k)} \mathcal{B}.
\end{array}
$$

Note that this is really a family of three theorems, one for each type of game G.

Thus in each case, we capture the salient logical equivalences in syntax-free, categorical form.

Coalgebras and combinatorial parameters

Coalgebras and combinatorial parameters

A beautiful feature of these comonads is that they let us capture crucial combinatorial parameters of structures using the indexed comonadic structure.

Coalgebras and combinatorial parameters

A beautiful feature of these comonads is that they let us capture crucial combinatorial parameters of structures using the indexed comonadic structure.

Conceptually, we can think of the morphisms $f : \mathbb{C}_k\mathcal{A} \to \mathcal{B}$ in the co-Kleisli category for $\mathbb{C}_k$ as those which only have to respect the k-local structure of $\mathcal{A}$.

Coalgebras and combinatorial parameters

A beautiful feature of these comonads is that they let us capture crucial combinatorial parameters of structures using the indexed comonadic structure.

Conceptually, we can think of the morphisms $f : \mathbb{C}_k\mathcal{A} \to \mathcal{B}$ in the co-Kleisli category for $\mathbb{C}_k$ as those which only have to respect the k-local structure of $\mathcal{A}$.

The lower the value of k, the less information available to Spoiler, and the easier it is for Duplicator to have a winning strategy. Equivalently, the easier it is to have a morphism from $\mathcal{A}$ to $\mathcal{B}$ in the co-Kleisli category.

Coalgebras and combinatorial parameters

A beautiful feature of these comonads is that they let us capture crucial combinatorial parameters of structures using the indexed comonadic structure.

Conceptually, we can think of the morphisms $f : \mathbb{C}_k\mathcal{A} \to \mathcal{B}$ in the co-Kleisli category for $\mathbb{C}_k$ as those which only have to respect the k-local structure of $\mathcal{A}$.

The lower the value of k, the less information available to Spoiler, and the easier it is for Duplicator to have a winning strategy. Equivalently, the easier it is to have a morphism from $\mathcal{A}$ to $\mathcal{B}$ in the co-Kleisli category.

What about morphisms $\mathcal{A} \to \mathbb{C}_k\mathcal{B}$?

Coalgebras and combinatorial parameters

A beautiful feature of these comonads is that they let us capture crucial combinatorial parameters of structures using the indexed comonadic structure.

Conceptually, we can think of the morphisms $f : \mathbb{C}_k\mathcal{A} \to \mathcal{B}$ in the co-Kleisli category for $\mathbb{C}_k$ as those which only have to respect the k-local structure of $\mathcal{A}$.

The lower the value of k, the less information available to Spoiler, and the easier it is for Duplicator to have a winning strategy. Equivalently, the easier it is to have a morphism from $\mathcal{A}$ to $\mathcal{B}$ in the co-Kleisli category.

What about morphisms $\mathcal{A} \to \mathbb{C}_k\mathcal{B}$?

Restricting the access to $\mathcal{B}$ makes it *harder* for Duplicator to win the homomorphism game.

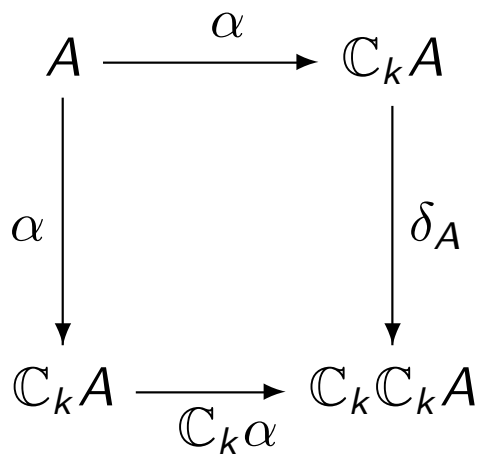
Coalgebras: a novel perspective

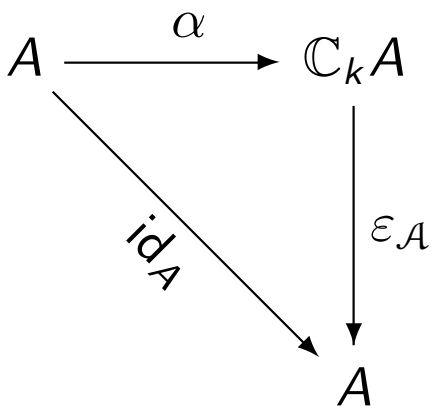
Another fundamental aspect of comonads is that they have an associated notion of *coalgebra*.

Coalgebras: a novel perspective

Another fundamental aspect of comonads is that they have an associated notion of *coalgebra*.

A coalgebra for a comonad (G, ε, δ) is a morphism $\alpha : A \to GA$ such that the following diagrams commute:

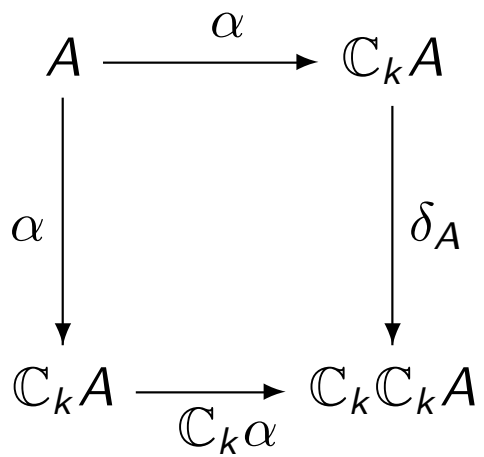


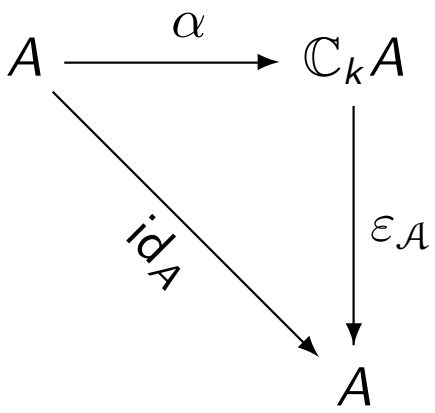


Coalgebras: a novel perspective

Another fundamental aspect of comonads is that they have an associated notion of *coalgebra*.

A coalgebra for a comonad (G, ε, δ) is a morphism $\alpha : A \to GA$ such that the following diagrams commute:



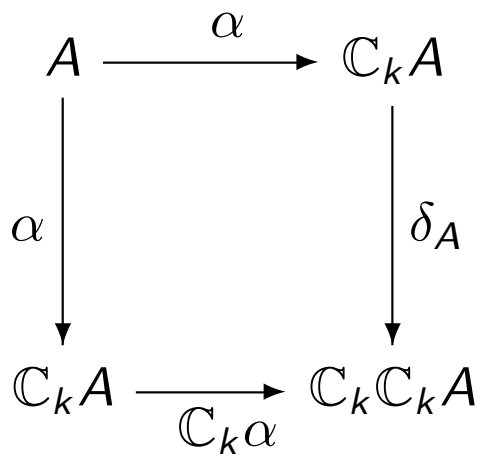


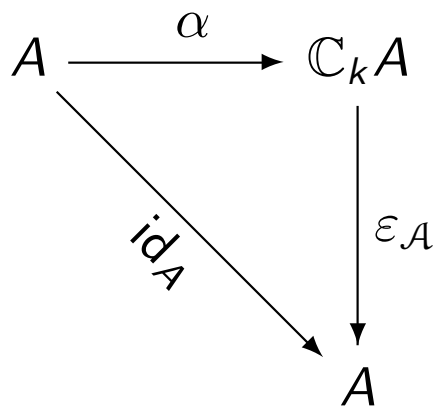
We should only expect a coalgebra structure to exist when the k-local information on A is sufficient to determine the structure of A.

Coalgebras: a novel perspective

Another fundamental aspect of comonads is that they have an associated notion of *coalgebra*.

A coalgebra for a comonad (G, ε, δ) is a morphism $\alpha : A \to GA$ such that the following diagrams commute:





We should only expect a coalgebra structure to exist when the k-local information on A is sufficient to determine the structure of A.

Our use of indexed comonads $\mathbb{C}_k$ opens up a new kind of question for coalgebras. Given a structure $\mathcal{A}$, we can ask: what is the least value of k such that a $\mathbb{C}_k$-coalgebra exists on $\mathcal{A}$? We call this the *coalgebra number* of $\mathcal{A}$.

Coalgebra numbers

Theorem

- *For the pebbling comonad, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-width of $\mathcal{A}$.*
- *For the Ehrenfeucht-Fraïssé comonad, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-depth of $\mathcal{A}$.*
- *For the modal comonad, the coalgebra number of $(\mathcal{A}, a)$ corresponds precisely to the synchronization tree depth of a in $\mathcal{A}$.*

Coalgebra numbers

Theorem

- *For the pebbling comonad, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-width of $\mathcal{A}$.*
- *For the Ehrenfeucht-Fraïssé comonad, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-depth of $\mathcal{A}$.*
- *For the modal comonad, the coalgebra number of $(\mathcal{A}, a)$ corresponds precisely to the synchronization tree depth of a in $\mathcal{A}$.*

The main idea behind these results is that coalgebras on $\mathcal{A}$ are in bijective correspondence with decompositions of $\mathcal{A}$ of the appropriate form.

Coalgebra numbers

Theorem

- *For the pebbling comonad, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-width of $\mathcal{A}$.*
- *For the Ehrenfeucht-Fraïssé comonad, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-depth of $\mathcal{A}$.*
- *For the modal comonad, the coalgebra number of $(\mathcal{A}, a)$ corresponds precisely to the synchronization tree depth of a in $\mathcal{A}$.*

The main idea behind these results is that coalgebras on $\mathcal{A}$ are in bijective correspondence with decompositions of $\mathcal{A}$ of the appropriate form.

We thus obtain categorical characterizations of these key combinatorial parameters.

Tree depth and the Ehrenfeucht-Fraïssé comonad

A *forest* is a poset $(F, \leq)$ such that, for all $x \in F$, the set of predecessors is a finite chain.

Tree depth and the Ehrenfeucht-Fraïssé comonad

A *forest* is a poset $(F, \leq)$ such that, for all $x \in F$, the set of predecessors is a finite chain.

A forest cover for G is a forest $(F, \leq)$ such that $V \subseteq F$, and if $v \frown v'$, then $v \uparrow v'$.

Tree depth and the Ehrenfeucht-Fraïssé comonad

A *forest* is a poset $(F, \leq)$ such that, for all $x \in F$, the set of predecessors is a finite chain.

A forest cover for G is a forest $(F, \leq)$ such that $V \subseteq F$, and if $v \frown v'$, then $v \uparrow v'$.

The tree-depth $\mathrm{td}(G)$ is defined to be $\min_F \mathrm{ht}(F)$, where F ranges over forest covers of G.

Tree depth and the Ehrenfeucht-Fraïssé comonad

A *forest* is a poset $(F, \leq)$ such that, for all $x \in F$, the set of predecessors is a finite chain.

A forest cover for G is a forest $(F, \leq)$ such that $V \subseteq F$, and if $v \frown v'$, then $v \uparrow v'$.

The tree-depth $\mathrm{td}(G)$ is defined to be $\min_F \mathrm{ht}(F)$, where F ranges over forest covers of G.

Given a σ-structure $\mathcal{A}$, the Gaifman graph $\mathcal{G}(\mathcal{A})$ is $(A, \frown)$, where $a \frown a'$ iff for some relation $R \in \sigma$, for some $(a_1, \ldots, a_n) \in R^{\mathcal{A}}$, $a = a_i$, $a' = a_j$, $i \neq j$. The tree-depth of $\mathcal{A}$ is $\mathrm{td}(\mathcal{G}(\mathcal{A}))$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

- The first coalgebra equation says that $\alpha(a_i) = [a_1, \ldots, a_i]$, $1 \leq i \leq j$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

- The first coalgebra equation says that $\alpha(a_i) = [a_1, \ldots, a_i]$, $1 \leq i \leq j$.
- The second says that $a_j = a$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

- The first coalgebra equation says that $\alpha(a_i) = [a_1, \ldots, a_i]$, $1 \leq i \leq j$.
- The second says that $a_j = a$.

Thus α is an injective map whose image is a prefix-closed subset of $A^{\leq k}$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

- The first coalgebra equation says that $\alpha(a_i) = [a_1, \ldots, a_i]$, $1 \leq i \leq j$.
- The second says that $a_j = a$.

Thus α is an injective map whose image is a prefix-closed subset of $A^{\leq k}$.

Defining $a \leq a'$ iff $\alpha(a) \sqsubseteq \alpha(a')$ yields a forest order on A, of height $\leq k$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

- The first coalgebra equation says that $\alpha(a_i) = [a_1, \ldots, a_i]$, $1 \leq i \leq j$.
- The second says that $a_j = a$.

Thus α is an injective map whose image is a prefix-closed subset of $A^{\leq k}$.

Defining $a \leq a'$ iff $\alpha(a) \sqsubseteq \alpha(a')$ yields a forest order on A, of height $\leq k$.

If $a \frown a'$ in $\mathcal{G}(\mathcal{A})$, the fact that α is a homomorphism implies $a \uparrow a'$.

Theorem

Let $\mathcal{A}$ be a finite σ-structure, and $k > 0$. There is a bijective correspondence between

1. *$\mathbb{E}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$.*
2. *Forest covers of $\mathcal{G}(\mathcal{A})$ of height $< k$.*

Proof outline

Suppose that $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ is a coalgebra. For $a \in A$, let $\alpha(a) = [a_1, \ldots, a_j]$.

- The first coalgebra equation says that $\alpha(a_i) = [a_1, \ldots, a_i]$, $1 \leq i \leq j$.
- The second says that $a_j = a$.

Thus α is an injective map whose image is a prefix-closed subset of $A^{\leq k}$.

Defining $a \leq a'$ iff $\alpha(a) \sqsubseteq \alpha(a')$ yields a forest order on A, of height $\leq k$.

If $a \frown a'$ in $\mathcal{G}(\mathcal{A})$, the fact that α is a homomorphism implies $a \uparrow a'$.

Thus $(A, \leq)$ is a forest cover of $\mathcal{A}$, of height $\leq k$.

Tree width

Tree width

A tree $(T, \leq)$ is a forest with a least element (the root).

Tree width

A tree $(T, \leq)$ is a forest with a least element (the root).

The unique path from x to x' is the set $\mathrm{path}(x, x') := [x \wedge x', x] \cup [x \wedge x', x']$, where we use interval notation: $[y, y'] := \{z \in T \mid y \leq z \leq y'\}$.

Tree width

A tree $(T, \leq)$ is a forest with a least element (the root).

The unique path from x to x' is the set $\mathrm{path}(x, x') := [x \wedge x', x] \cup [x \wedge x', x']$, where we use interval notation: $[y, y'] := \{z \in T \mid y \leq z \leq y'\}$.

A tree-decomposition of a graph $G = (V, \frown)$ is a tree $(T, \leq)$ together with a labelling function $\lambda : T \to \mathcal{P}(V)$ satisfying the following conditions:

- (TD1) for all $v \in V$, for some $x \in T$, $v \in \lambda(x)$;
- (TD2) if $v \frown v'$, then for some $x \in T$, $\{v, v'\} \subseteq \lambda(x)$;
- (TD3) if $v \in \lambda(x) \cap \lambda(x')$, then for all $y \in \mathrm{path}(x, x')$, $v \in \lambda(y)$.

Tree width

A tree $(T, \leq)$ is a forest with a least element (the root).

The unique path from x to x' is the set $\mathrm{path}(x, x') := [x \wedge x', x] \cup [x \wedge x', x']$, where we use interval notation: $[y, y'] := \{z \in T \mid y \leq z \leq y'\}$.

A tree-decomposition of a graph $G = (V, \frown)$ is a tree $(T, \leq)$ together with a labelling function $\lambda : T \to \mathcal{P}(V)$ satisfying the following conditions:

- (TD1) for all $v \in V$, for some $x \in T$, $v \in \lambda(x)$;
- (TD2) if $v \frown v'$, then for some $x \in T$, $\{v, v'\} \subseteq \lambda(x)$;
- (TD3) if $v \in \lambda(x) \cap \lambda(x')$, then for all $y \in \mathrm{path}(x, x')$, $v \in \lambda(y)$.

The width of a tree decomposition is given by $\max_{x \in T} |\lambda(x)| - 1$.

Tree width

A tree $(T, \leq)$ is a forest with a least element (the root).

The unique path from x to x' is the set $\mathrm{path}(x, x') := [x \wedge x', x] \cup [x \wedge x', x']$, where we use interval notation: $[y, y'] := \{z \in T \mid y \leq z \leq y'\}$.

A tree-decomposition of a graph $G = (V, \frown)$ is a tree $(T, \leq)$ together with a labelling function $\lambda : T \to \mathcal{P}(V)$ satisfying the following conditions:

- (TD1) for all $v \in V$, for some $x \in T$, $v \in \lambda(x)$;
- (TD2) if $v \frown v'$, then for some $x \in T$, $\{v, v'\} \subseteq \lambda(x)$;
- (TD3) if $v \in \lambda(x) \cap \lambda(x')$, then for all $y \in \mathrm{path}(x, x')$, $v \in \lambda(y)$.

The width of a tree decomposition is given by $\max_{x \in T} |\lambda(x)| - 1$.

We define the tree-width $\mathrm{tw}(G)$ of a graph G as $\min_T \mathrm{width}(T)$, where T ranges over tree decompositions of G.

Tree width

A tree $(T, \leq)$ is a forest with a least element (the root).

The unique path from x to x' is the set $\mathrm{path}(x, x') := [x \wedge x', x] \cup [x \wedge x', x']$, where we use interval notation: $[y, y'] := \{z \in T \mid y \leq z \leq y'\}$.

A tree-decomposition of a graph $G = (V, \frown)$ is a tree $(T, \leq)$ together with a labelling function $\lambda : T \to \mathcal{P}(V)$ satisfying the following conditions:

- (TD1) for all $v \in V$, for some $x \in T$, $v \in \lambda(x)$;
- (TD2) if $v \frown v'$, then for some $x \in T$, $\{v, v'\} \subseteq \lambda(x)$;
- (TD3) if $v \in \lambda(x) \cap \lambda(x')$, then for all $y \in \mathrm{path}(x, x')$, $v \in \lambda(y)$.

The width of a tree decomposition is given by $\max_{x \in T} |\lambda(x)| - 1$.

We define the tree-width $\mathrm{tw}(G)$ of a graph G as $\min_T \mathrm{width}(T)$, where T ranges over tree decompositions of G.

This parameter plays a fundamental role in combinatorics, algorithms and parameterized complexity.

Tree-width and pebbling

Tree-width and pebbling

We shall now give an alternative formulation of tree-width which will provide a useful bridge to the coalgebraic characterization.

Tree-width and pebbling

We shall now give an alternative formulation of tree-width which will provide a useful bridge to the coalgebraic characterization.

It is also interesting in its own right: it clarifies the relationship between tree-width and tree-depth, and shows how pebbling arises naturally in connection with tree-width.

Tree-width and pebbling

We shall now give an alternative formulation of tree-width which will provide a useful bridge to the coalgebraic characterization.

It is also interesting in its own right: it clarifies the relationship between tree-width and tree-depth, and shows how pebbling arises naturally in connection with tree-width.

A k-pebble forest cover for a graph $G = (V, \frown)$ is a forest cover $(V, \leq)$ together with a pebbling function $p : V \to \mathbf{k}$ such that, if $v \frown v'$ with $v \leq v'$, then for all $w \in (v, v']$, $p(v) \neq p(w)$.

Tree-width and pebbling

We shall now give an alternative formulation of tree-width which will provide a useful bridge to the coalgebraic characterization.

It is also interesting in its own right: it clarifies the relationship between tree-width and tree-depth, and shows how pebbling arises naturally in connection with tree-width.

A k-pebble forest cover for a graph $G = (V, \frown)$ is a forest cover $(V, \leq)$ together with a pebbling function $p : V \to \mathbf{k}$ such that, if $v \frown v'$ with $v \leq v'$, then for all $w \in (v, v']$, $p(v) \neq p(w)$.

Theorem

Let G be a finite graph. The following are equivalent:

1. *G has a tree decomposition of width $< k$.*
2. *G has a k-pebble forest cover.*

Treewidth as coalgebra number

Treewidth as coalgebra number

Theorem

Let $\mathcal{A}$ be a finite σ-structure. There is a bijective correspondence between:

1. *$\mathbb{P}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{P}_k\mathcal{A}$*
2. *k-pebble forest covers of $\mathcal{G}(\mathcal{A})$.*

Treewidth as coalgebra number

Theorem

Let $\mathcal{A}$ be a finite σ-structure. There is a bijective correspondence between:

1. *$\mathbb{P}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{P}_k\mathcal{A}$*
2. *k-pebble forest covers of $\mathcal{G}(\mathcal{A})$.*

We write $\kappa^{\mathbb{P}}(\mathcal{A})$ for the coalgebra number of $\mathcal{A}$ with respect to the the pebbling comonad.

Treewidth as coalgebra number

Theorem

Let $\mathcal{A}$ be a finite σ-structure. There is a bijective correspondence between:

1. *$\mathbb{P}_k$-coalgebras $\alpha : \mathcal{A} \to \mathbb{P}_k\mathcal{A}$*
2. *k-pebble forest covers of $\mathcal{G}(\mathcal{A})$.*

We write $\kappa^{\mathbb{P}}(\mathcal{A})$ for the coalgebra number of $\mathcal{A}$ with respect to the the pebbling comonad.

Theorem

For all finite structures $\mathcal{A}$: $\mathsf{tw}(\mathcal{A}) = \kappa^{\mathbb{P}}(\mathcal{A}) - 1$.

Further Horizons I

Currently investigating Rossman's Homomorphism Preservation Theorem with Tom Paine.

Further Horizons I

Currently investigating Rossman's Homomorphism Preservation Theorem with Tom Paine.

A major result in finite model theory.

Further Horizons I

Currently investigating Rossman's Homomorphism Preservation Theorem with Tom Paine.

A major result in finite model theory.

- We have identified a functorial construction underlying the Equirank HPT.

Further Horizons I

Currently investigating Rossman's Homomorphism Preservation Theorem with Tom Paine.

A major result in finite model theory.

- We have identified a functorial construction underlying the Equirank HPT.
- It uses colimits to glue existential positive witnesses, in order to reduce elementary equivalence (at given quantifier rank) to existential positive equivalence. (A sort of quantifier elimination).

Further Horizons I

Currently investigating Rossman's Homomorphism Preservation Theorem with Tom Paine.

A major result in finite model theory.

- We have identified a functorial construction underlying the Equirank HPT.
- It uses colimits to glue existential positive witnesses, in order to reduce elementary equivalence (at given quantifier rank) to existential positive equivalence. (A sort of quantifier elimination).
- Should be generalizable, e.g. to finite variable/pebbling case.

Further Horizons II

Major algorithmic metatheorems:

- Courcelle's theorem
- decision procedures for guarded fragments

Further Horizons II

Major algorithmic metatheorems:

- Courcelle's theorem
- decision procedures for guarded fragments

Can we combine coalgebras witnessing treewidth/tree model property with coalgebraic descriptions of automata, to obtain structural - and generalizable - proofs of these results?

Further Horizons II

Major algorithmic metatheorems:

- Courcelle's theorem
- decision procedures for guarded fragments

Can we combine coalgebras witnessing treewidth/tree model property with coalgebraic descriptions of automata, to obtain structural - and generalizable - proofs of these results?

Can we embed our coalgebraic semantics in a type theory, allowing algorithms to be extracted by Curry-Howard?

Further Horizons II

Major algorithmic metatheorems:

- Courcelle's theorem
- decision procedures for guarded fragments

Can we combine coalgebras witnessing treewidth/tree model property with coalgebraic descriptions of automata, to obtain structural - and generalizable - proofs of these results?

Can we embed our coalgebraic semantics in a type theory, allowing algorithms to be extracted by Curry-Howard?

The wider issue: can we get Structure and Power to work with each other to address genuinely deep questions?

Envoi

Envoi

Let's not forget to dream!