Resources in Computation

Samson Abramsky

Department of Computer Science, UCL

Resources in Computation

- EPSRC-funded fellowship 2021–26
- Postdocs: Luca Reggio, Rafał Stefański
- Project partners:
  - Anuj Dawar (Cambridge)
  - Jouko Väänänen (Helsinki)
  - Thomas Ehrhard, Christine Tasson and Paul-André Melliès (IRIF, Paris)
  - David Pym (UCL)
  - Rui Soares Barbosa and Ernesto Galvão (IINL Braga)
  - Mikołaj Bojańczyk (Warsaw) and Bartek Klin (Oxford)
- Building on joint project with Anuj Dawar:
  Resources and Co-resources: a junction between categorical semantics and descriptive complexity, 1/10/2019–31/3/2023

Research Vision

Resources are pervasive throughout computation:

- if we think of algorithms and complexity, ideas of efficiency and complexity are based on consumption of resources such as space and time but also:
- resource logics, e.g. separation logic
- type theories, resource lambda calculi, closely related to differentiable calculi and the foundations of differentiable programming
- process semantics, game semantics etc.

Research Vision

Resources are pervasive throughout computation:

- if we think of algorithms and complexity, ideas of efficiency and complexity are based on consumption of resources such as space and time but also:
- resource logics, e.g. separation logic
- type theories, resource lambda calculi, closely related to differentiable calculi and the foundations of differentiable programming
- process semantics, game semantics etc.

Resources are studied in different-looking ways in these specialised areas.

Research Vision

Resources are pervasive throughout computation:

- if we think of algorithms and complexity, ideas of efficiency and complexity are based on consumption of resources such as space and time but also:
- resource logics, e.g. separation logic
- type theories, resource lambda calculi, closely related to differentiable calculi and the foundations of differentiable programming
- process semantics, game semantics etc.

Resources are studied in different-looking ways in these specialised areas.

Symptom of a wider phenomenon in CS, of sub-specialisms not communicating with each othjer

Research Vision

Resources are pervasive throughout computation:

- if we think of algorithms and complexity, ideas of efficiency and complexity are based on consumption of resources such as space and time but also:
- resource logics, e.g. separation logic
- type theories, resource lambda calculi, closely related to differentiable calculi and the foundations of differentiable programming
- process semantics, game semantics etc.

Resources are studied in different-looking ways in these specialised areas.

Symptom of a wider phenomenon in CS, of sub-specialisms not communicating with each othjer

Our vision is to bring these different strands together, and create a unified theory of resources in computation which will allow the flow of ideas and techniques between these different areas, leading to new kinds of results and insights.

Research Vision

Resources are pervasive throughout computation:

- if we think of algorithms and complexity, ideas of efficiency and complexity are based on consumption of resources such as space and time but also:
- resource logics, e.g. separation logic
- type theories, resource lambda calculi, closely related to differentiable calculi and the foundations of differentiable programming
- process semantics, game semantics etc.

Resources are studied in different-looking ways in these specialised areas.

Symptom of a wider phenomenon in CS, of sub-specialisms not communicating with each othjer

Our vision is to bring these different strands together, and create a unified theory of resources in computation which will allow the flow of ideas and techniques between these different areas, leading to new kinds of results and insights.

This is a bold but credible hypothesis, based on recent breakthrough results, and with clear ideas and directions for how to make progress.

Research Vision

Resources are pervasive throughout computation:

- if we think of algorithms and complexity, ideas of efficiency and complexity are based on consumption of resources such as space and time but also:
- resource logics, e.g. separation logic
- type theories, resource lambda calculi, closely related to differentiable calculi and the foundations of differentiable programming
- process semantics, game semantics etc.

Resources are studied in different-looking ways in these specialised areas.

Symptom of a wider phenomenon in CS, of sub-specialisms not communicating with each othjer

Our vision is to bring these different strands together, and create a unified theory of resources in computation which will allow the flow of ideas and techniques between these different areas, leading to new kinds of results and insights.

This is a bold but credible hypothesis, based on recent breakthrough results, and with clear ideas and directions for how to make progress.

Another important aspect is bringing researchers and communities together, and building a new research community in this emerging area.

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, CSL 2018. Extended version in *Journal of Logic and Computation* 2021.

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, CSL 2018. Extended version in *Journal of Logic and Computation* 2021.

Current EPSRC project with Anuj Dawar (Cambridge) on:
Resources and Co-resources: a junction between categorical semantics and descriptive complexity.

Post-docs Dan Marsden, Luca Reggio (Marie-Curie Fellow), Tomáš Jakl
Ph.D. students Tom Paine, Nihil Shah, Adam Ó Conghaile.

People

People

Anuj Dawar

Dan Marsden

Luca Reggio

Tomáš Jakl

Nihil Shah

Tom Paine

Adam Ó Conghaile

The setting: relational structures

The setting: relational structures

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

The setting: 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: 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: 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: 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.
N.B. *Not* the usual setting in model theory.

The setting: 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.
N.B. *Not* the usual setting in model theory.

This will be the *extensional category*.

Adjunctions recalled

Given a ring R, the category of R-modules is denoted R–**Mod**. There is an evident forgetful functor $U : R\text{–}\mathbf{Mod} \to \mathbf{Set}$, and an adjunction

$$\begin{tikzcd} \mathbf{Set} \arrow[r, bend left, "R^{(\bullet)}"] \arrow[r, phantom, "\perp"] & R\text{–}\mathbf{Mod} \arrow[l, bend left, "U"] \end{tikzcd}$$

$R^{(X)}$ is the free module generated by X (formal finite R-linear combinations over X).

Adjunctions recalled

Given a ring R, the category of R-modules is denoted R–**Mod**. There is an evident forgetful functor $U : R\text{–}\mathbf{Mod} \to \mathbf{Set}$, and an adjunction

$$
\begin{tikzcd}
\mathbf{Set} \arrow[r, bend left, "R^{(\bullet)}"] \arrow[r, phantom, "\perp"] & R\text{–}\mathbf{Mod} \arrow[l, bend left, "U"]
\end{tikzcd}
$$

$R^{(X)}$ is the free module generated by X (formal finite R-linear combinations over X).

The universal mapping property:

$$
\begin{array}{ccc}
\mathbf{Set} & \qquad & R\text{–}\mathbf{Mod} \\
\begin{tikzcd}
X \arrow[r, "\eta"] \arrow[dr, "f"'] & UR^{(X)} \arrow[d, dashed, "U\hat{f}"] \\
& UM
\end{tikzcd}
& &
\begin{tikzcd}
R^{(X)} \arrow[d, dashed, "\hat{f}"] \\
M
\end{tikzcd}
\end{array}
$$

Monads and Comonads

Given an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "R"] \end{tikzcd}$$

there is an associated *monad RL* on $\mathcal{C}$, and *comonad LR* on $\mathcal{D}$.

Monads and Comonads

Given an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "R"] \end{tikzcd}$$

there is an associated *monad RL* on $\mathcal{C}$, and *comonad LR* on $\mathcal{D}$.

These notions are pervasive in mathematics:

- monads occur in topology, and finitary monads on **Set** subsume universal algebra.
- comonads feature in descent theory
- Galois correspondences: closures and coclosures.

Monads and Comonads

Given an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "R"] \end{tikzcd}$$

there is an associated *monad RL* on $\mathcal{C}$, and *comonad LR* on $\mathcal{D}$.

These notions are pervasive in mathematics:

- monads occur in topology, and finitary monads on **Set** subsume universal algebra.
- comonads feature in descent theory
- Galois correspondences: closures and coclosures.

In our context, we can think of a resource-indexed comonad C_k as a *modality*:

$$C_k A \to B$$

means we have a homomorphism which only needs to be checked against a limited part of the structure of A.

Monads and Comonads

Given an adjunction

$$\begin{tikzcd} \mathcal{C} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "R"] \end{tikzcd}$$

there is an associated *monad* RL on $\mathcal{C}$, and *comonad* LR on $\mathcal{D}$.

These notions are pervasive in mathematics:

- monads occur in topology, and finitary monads on **Set** subsume universal algebra.
- comonads feature in descent theory
- Galois correspondences: closures and coclosures.

In our context, we can think of a resource-indexed comonad C_k as a *modality*:

$$C_k A \to B$$

means we have a homomorphism which only needs to be checked against a limited part of the structure of A.

This is exactly what logical languages do!
They calibrate limited means for accessing structures.

The general scheme: resource-indexed adjunctions

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.
  Aside: analogies with game semantics for higher-order computation/logic.

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.
  Aside: analogies with game semantics for higher-order computation/logic.

- Such a tree cover will in general not have the full properties of the original structure, but be a “best approximation” in some resource-restricted setting.

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.
  Aside: analogies with game semantics for higher-order computation/logic.
- Such a tree cover will in general not have the full properties of the original structure, but be a "best approximation" in some resource-restricted setting.
- More precisely, this means that we have a (comonadic) adjunction, yielding the corresponding comonad.

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.
  Aside: analogies with game semantics for higher-order computation/logic.

- Such a tree cover will in general not have the full properties of the original structure, but be a "best approximation" in some resource-restricted setting.

- More precisely, this means that we have a (comonadic) adjunction, yielding the corresponding comonad.

- The objects of the category where the approximations live have an intrinsic tree structure, which can be captured axiomatically, as *arboreal categories*.

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.
  Aside: analogies with game semantics for higher-order computation/logic.
- Such a tree cover will in general not have the full properties of the original structure, but be a "best approximation" in some resource-restricted setting.
- More precisely, this means that we have a (comonadic) adjunction, yielding the corresponding comonad.
- The objects of the category where the approximations live have an intrinsic tree structure, which can be captured axiomatically, as *arboreal categories*.
- The tree encodes a process for generating (parts of) the relational structure, to which resource notions can be applied.

The general scheme: resource-indexed adjunctions

- In our approach, we build tree-structured covers of a given, purely extensional relational structure.
  Aside: analogies with game semantics for higher-order computation/logic.
- Such a tree cover will in general not have the full properties of the original structure, but be a "best approximation" in some resource-restricted setting.
- More precisely, this means that we have a (comonadic) adjunction, yielding the corresponding comonad.
- The objects of the category where the approximations live have an intrinsic tree structure, which can be captured axiomatically, as *arboreal categories*.
- The tree encodes a process for generating (parts of) the relational structure, to which resource notions can be applied.
- This allows us to apply resource notions to the objects of the extensional category via the adjunction.

First example

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

A forest morphism preserves roots and the covering relation.

First example

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

A *forest-ordered σ-structure* $(\mathcal{A}, \leq)$ is a σ-structure $\mathcal{A}$ with a forest order $\leq$ on A.

First example

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

A *forest-ordered σ-structure* $(\mathcal{A}, \leq)$ is a σ-structure $\mathcal{A}$ with a forest order $\leq$ on A.

This must satisfy condition (E): elements of A which are adjacent in the Gaifman graph of $\mathcal{A}$ must be comparable in the order.

First example

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

A *forest-ordered σ-structure* $(\mathcal{A}, \leq)$ is a σ-structure $\mathcal{A}$ with a forest order $\leq$ on A.

This must satisfy condition (E): elements of A which are adjacent in the Gaifman graph of $\mathcal{A}$ must be comparable in the order.

The *mininum height* of such a forest order on $\mathcal{A}$ is the *tree-depth* of $\mathcal{A}$ (Nešetřil and Ossona de Mendez).

First example

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

A *forest-ordered σ-structure* $(\mathcal{A}, \leq)$ is a σ-structure $\mathcal{A}$ with a forest order $\leq$ on A.

This must satisfy condition (E): elements of A which are adjacent in the Gaifman graph of $\mathcal{A}$ must be comparable in the order.

The *mininum height* of such a forest order on $\mathcal{A}$ is the *tree-depth* of $\mathcal{A}$ (Nešetřil and Ossona de Mendez).

Important combinatorial parameter, used extensively by Rossman in his Homomorphism Preservation Theorems.

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

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

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

- Given a structure $\mathcal{A}$, the universe of $G_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.
- This is forest-ordered by the prefix order.

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

- Given a structure $\mathcal{A}$, the universe of $G_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.
- This is forest-ordered by the prefix order.
- The counit map $\varepsilon_{\mathcal{A}} : A^{\leq k} \to A$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

- Given a structure $\mathcal{A}$, the universe of $G_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.
- This is forest-ordered by the prefix order.
- The counit map $\varepsilon_{\mathcal{A}} : A^{\leq k} \to A$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.
- How do we lift the relations on $\mathcal{A}$ to $G_k\mathcal{A}$?

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

- Given a structure $\mathcal{A}$, the universe of $G_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.
- This is forest-ordered by the prefix order.
- The counit map $\varepsilon_{\mathcal{A}} : A^{\leq k} \to A$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.
- How do we lift the relations on $\mathcal{A}$ to $G_k\mathcal{A}$?
- Given e.g. a binary relation R, we define $R^{G_k\mathcal{A}}$ to the set of pairs (s, t) such that

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

- Given a structure $\mathcal{A}$, the universe of $G_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.
- This is forest-ordered by the prefix order.
- The counit map $\varepsilon_{\mathcal{A}} : A^{\leq k} \to A$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.
- How do we lift the relations on $\mathcal{A}$ to $G_k\mathcal{A}$?
- Given e.g. a binary relation R, we define $R^{G_k\mathcal{A}}$ to the set of pairs (s, t) such that
  - $s \sqsubseteq t$ or $t \sqsubseteq s$ (in prefix order)

Resource cover and adjunction

There is a category $\mathcal{R}^E(\sigma)$, and an evident forgetful functor $U : \mathcal{R}^E(\sigma) \to \mathcal{R}(\sigma)$.

For each $k > 0$, if we restrict to forest orders of height $\leq k$, we get a sub-category $\mathcal{R}^E_k(\sigma)$, and a functor $U_k : \mathcal{R}^E_k(\sigma) \to \mathcal{R}(\sigma)$.

This functor has a right adjoint G_k, giving rise to a comonad $\mathbb{E}_k = U_k G_k$ on $\mathcal{R}(\sigma)$.

- Given a structure $\mathcal{A}$, the universe of $G_k\mathcal{A}$ is $A^{\leq k}$, the non-empty sequences of length $\leq k$.
- This is forest-ordered by the prefix order.
- The counit map $\varepsilon_{\mathcal{A}} : A^{\leq k} \to A$ sends a sequence $[a_1, \ldots, a_n]$ to a_n.
- How do we lift the relations on $\mathcal{A}$ to $G_k\mathcal{A}$?
- Given e.g. a binary relation R, we define $R^{G_k\mathcal{A}}$ 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 couniversal property

Arboreal category

$$\begin{tikzcd} G_k\mathcal{A} \\ \mathcal{F} \arrow[u, dashed, "\hat{f}"] \end{tikzcd}$$

Extensional category

$$\begin{tikzcd} U_k G_k\mathcal{A} \arrow[r, "\varepsilon_{\mathcal{A}}"] & \mathcal{A} \\ U_k\mathcal{F} \arrow[u, dashed, "U_k\hat{f}"] \arrow[ur, "f"'] & \end{tikzcd}$$

The couniversal property

Arboreal category

$$
\begin{tikzcd}
G_k \mathcal{A} \\
\mathcal{F} \arrow[u, dashed, "\hat{f}"]
\end{tikzcd}
$$

Extensional category

$$
\begin{tikzcd}
U_k G_k \mathcal{A} \arrow[r, "\varepsilon_{\mathcal{A}}"] & \mathcal{A} \\
U_k \mathcal{F} \arrow[u, dashed, "U_k \hat{f}"] \arrow[ur, "f"'] &
\end{tikzcd}
$$

Moreover, the adjunction is *comonadic*, meaning that the category of coalgebras for $\mathbb{E}_k$ is exactly $\mathcal{R}_k^E(\sigma)$.

Consequences

Remember that the right adjoint is uniquely determined by the forgetful functor, and the comonad by the adjunction.

Consequences

Remember that the right adjoint is uniquely determined by the forgetful functor, and the comonad by the adjunction.

As we shall now see, this structure gives us directly:

- The Ehrenfeucht-Fraïssé game
- The quantifier-rank indexed fragments of FOL
- Equivalences of structures induced by:
  - the full fragment of q.r. $\leq k$
  - the existential positive part of the fragment
  - the extension of the fragment with counting quantifiers
- We also recover the important tree-depth combinatorial parameter from the coalgebras of the comonad

Consequences

Remember that the right adjoint is uniquely determined by the forgetful functor, and the comonad by the adjunction.

As we shall now see, this structure gives us directly:

- The Ehrenfeucht-Fraïssé game
- The quantifier-rank indexed fragments of FOL
- Equivalences of structures induced by:
  - the full fragment of q.r. $\leq k$
  - the existential positive part of the fragment
  - the extension of the fragment with counting quantifiers
- We also recover the important tree-depth combinatorial parameter from the coalgebras of the comonad

Moreover, this template can be used to give similar analyses of a wealth of other logical and combinatorial notions.

Consequences

Remember that the right adjoint is uniquely determined by the forgetful functor, and the comonad by the adjunction.

As we shall now see, this structure gives us directly:

- The Ehrenfeucht-Fraïssé game
- The quantifier-rank indexed fragments of FOL
- Equivalences of structures induced by:
  - the full fragment of q.r. $\leq k$
  - the existential positive part of the fragment
  - the extension of the fragment with counting quantifiers
- We also recover the important tree-depth combinatorial parameter from the coalgebras of the comonad

Moreover, this template can be used to give similar analyses of a wealth of other logical and combinatorial notions.

This general pattern has been axiomatised in *Arboreal Categories and Resources*, SA and Luca Reggio (ICALP 2021, available at `arXiv:2102.08109`).

Model comparison games

Especially important in finite model theory, where *model comparison games* such as Ehrenfeucht-Fraïssé games, pebble games and bisimulation games play a central role.

Model comparison games

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.

Model comparison games

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 $\mathcal{B}$.

Model comparison games

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 $\leq k$.

Model comparison games

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.

Model comparison games

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.

Aside: other kinds of logic games, e.g. evaluation games, proof games? Cf. Jouko's talk.

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 $\leq k$, $\mathcal{A} \models \varphi \;\Rightarrow\; \mathcal{B} \models \varphi$.*

Open pathwise embeddings and back-and-forth equivalences

How do we capture back-and-forth equivalences, and hence the whole logic rather than just the existential positive part?

Open pathwise embeddings and back-and-forth equivalences

How do we capture back-and-forth equivalences, and hence the whole logic rather than just the existential positive part?

The key idea is to work in the arboreal category $\mathcal{R}_k^E(\sigma)$, where we have enough process structure to make game and bisimulation notions meaningful.

Open pathwise embeddings and back-and-forth equivalences

How do we capture back-and-forth equivalences, and hence the whole logic rather than just the existential positive part?

The key idea is to work in the arboreal category $\mathcal{R}_k^E(\sigma)$, where we have enough process structure to make game and bisimulation notions meaningful.

The key notions are

- *paths*, *i.e.* objects of $\mathcal{R}_k^E(\sigma)$ in which the order is linear (so the forest is a single branch), and
- *path embeddings*, *i.e.* forest morphisms with paths as domains which are embeddings of relational structures.

Open pathwise embeddings and back-and-forth equivalences

How do we capture back-and-forth equivalences, and hence the whole logic rather than just the existential positive part?

The key idea is to work in the arboreal category $\mathcal{R}_k^E(\sigma)$, where we have enough process structure to make game and bisimulation notions meaningful.

The key notions are

- *paths*, *i.e.* objects of $\mathcal{R}_k^E(\sigma)$ in which the order is linear (so the forest is a single branch), and
- *path embeddings*, *i.e.* forest morphisms with paths as domains which are embeddings of relational structures.

These are special cases of notions which are axiomatised in the arboreal categories setting in great generality.

Pathwise embeddings and open maps

A morphism $f\colon X \to Y$ in $\mathcal{R}_k^E(\sigma)$ is a *pathwise embedding* if, for all path embeddings $m\colon P \rightarrowtail X$, the composite $f \circ m$ is a path embedding.

Pathwise embeddings and open maps

A morphism $f\colon X \to Y$ in $\mathcal{R}_k^E(\sigma)$ is a *pathwise embedding* if, for all path embeddings $m\colon P \rightarrowtail X$, the composite $f \circ m$ is a path embedding.

To capture the "back" or p-morphism condition, we introduce a notion of *open map* (Joyal-Nielsen-Winskel) that, combined with the concept of pathwise embedding, will allow us to define an appropriate notion of bisimulation.

Pathwise embeddings and open maps

A morphism $f\colon X \to Y$ in $\mathcal{R}_k^E(\sigma)$ is a *pathwise embedding* if, for all path embeddings $m\colon P \rightarrowtail X$, the composite $f \circ m$ is a path embedding.

To capture the "back" or p-morphism condition, we introduce a notion of *open map* (Joyal-Nielsen-Winskel) that, combined with the concept of pathwise embedding, will allow us to define an appropriate notion of bisimulation.

A morphism $f\colon X \to Y$ in $\mathcal{R}_k^E(\sigma)$ is said to be *open* if it satisfies the following path-lifting property: Given any commutative square

$$\begin{tikzcd} P \arrow[r, tail] \arrow[d, tail] & Q \arrow[d, tail] \arrow[dl, dashed] \\ X \arrow[r, "f"] & Y \end{tikzcd}$$

with P, Q paths, there exists a diagonal filler $Q \to X$ (*i.e.* an arrow $Q \to X$ making the two triangles commute).

Bisimulations

A *bisimulation* between objects X, Y of $\mathcal{R}_k^E(\sigma)$ is a span of open pathwise embeddings

$$
\begin{tikzcd}
 & R \arrow[dl] \arrow[dr] & \\
X & & Y
\end{tikzcd}
$$

If such a bisimulation exists, we say that X and Y are *bisimilar*.

Theorem

$G_k\mathcal{A}$ and $G_k\mathcal{B}$ are bisimilar in $\mathcal{R}_k^E(\sigma)$ iff Duplicator has a winning strategy in the k-round Ehrenfeucht-Fraïssé game between $\mathcal{A}$ and $\mathcal{B}$.

Bisimulations

A *bisimulation* between objects X, Y of $\mathcal{R}_k^E(\sigma)$ is a span of open pathwise embeddings

$$\begin{tikzcd} & R \arrow[dl] \arrow[dr] & \\ X & & Y \end{tikzcd}$$

If such a bisimulation exists, we say that X and Y are *bisimilar*.

Theorem

$G_k\mathcal{A}$ and $G_k\mathcal{B}$ are bisimilar in $\mathcal{R}_k^E(\sigma)$ iff Duplicator has a winning strategy in the k-round Ehrenfeucht-Fraïssé game between $\mathcal{A}$ and $\mathcal{B}$.

Note that we use the *resource category* $\mathcal{R}_k^E(\sigma)$ to study logical properties of objects of the *extensional category* $\mathcal{R}(\sigma)$.

Connection to logic

Fragments of first-order logic:

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

Connection to logic

Fragments of first-order logic:

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

We define three resource-indexed equivalences on structures $\mathcal{A}$, $\mathcal{B}$ of $\mathcal{R}(\sigma)$ using the resource categories $\mathcal{R}_k^E(\sigma)$:

- $\mathcal{A} \rightleftarrows_k \mathcal{B}$ iff there are morphisms $G_k\mathcal{A} \to G_k\mathcal{B}$ and $G_k\mathcal{B} \to G_k\mathcal{A}$.
  Note that there need be no relationship between these morphisms.
- $\mathcal{A} \leftrightarrow_k \mathcal{B}$ iff $G_k\mathcal{A}$ and $G_k\mathcal{B}$ are bisimilar in $\mathcal{R}_k^E(\sigma)$.
- $\mathcal{A} \cong_k \mathcal{B}$ iff $G_k\mathcal{A}$ and $G_k\mathcal{B}$ are isomorphic in $\mathcal{R}_k^E(\sigma)$.

Connection to logic

Fragments of first-order logic:

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

We define three resource-indexed equivalences on structures $\mathcal{A}$, $\mathcal{B}$ of $\mathcal{R}(\sigma)$ using the resource categories $\mathcal{R}_k^E(\sigma)$:

- $\mathcal{A} \rightleftarrows_k \mathcal{B}$ iff there are morphisms $G_k\mathcal{A} \to G_k\mathcal{B}$ and $G_k\mathcal{B} \to G_k\mathcal{A}$. Note that there need be no relationship between these morphisms.
- $\mathcal{A} \leftrightarrow_k \mathcal{B}$ iff $G_k\mathcal{A}$ and $G_k\mathcal{B}$ are bisimilar in $\mathcal{R}_k^E(\sigma)$.
- $\mathcal{A} \cong_k \mathcal{B}$ iff $G_k\mathcal{A}$ and $G_k\mathcal{B}$ are isomorphic in $\mathcal{R}_k^E(\sigma)$.

Theorem

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

$$
\begin{array}{llcl}
(1) & \mathcal{A} \equiv^{\exists\mathcal{L}_k} \mathcal{B} & \Longleftrightarrow & \mathcal{A} \rightleftarrows_k \mathcal{B}. \\
(2) & \mathcal{A} \equiv^{\mathcal{L}_k} \mathcal{B} & \Longleftrightarrow & \mathcal{A} \leftrightarrow_k \mathcal{B}. \\
(3) & \mathcal{A} \equiv^{\mathcal{L}_k(\#)} \mathcal{B} & \Longleftrightarrow & \mathcal{A} \cong_k \mathcal{B}.
\end{array}
$$

Coalgebra number and tree-depth

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

$$
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[dr, "\mathsf{id}_A"'] & GA \arrow[d, "\varepsilon_A"] \\
 & A
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & GA \arrow[d, "\delta_A"] \\
GA \arrow[r, "G\alpha"] & G^2 A
\end{tikzcd}
$$

Coalgebra number and tree-depth

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

$$
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[dr, "\mathrm{id}_A"'] & GA \arrow[d, "\varepsilon_A"] \\
& A
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & GA \arrow[d, "\delta_A"] \\
GA \arrow[r, "G\alpha"] & G^2A
\end{tikzcd}
$$

We should only expect a coalgebra structure $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ to exist when the k-local information on $\mathcal{A}$ is sufficient to determine the structure of $\mathcal{A}$.

Coalgebra number and tree-depth

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

$$
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[dr, "\mathsf{id}_A"'] & GA \arrow[d, "\varepsilon_A"] \\
& A
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & GA \arrow[d, "\delta_A"] \\
GA \arrow[r, "G\alpha"] & G^2 A
\end{tikzcd}
$$

We should only expect a coalgebra structure $\alpha : \mathcal{A} \to \mathbb{E}_k \mathcal{A}$ to exist when the k-local information on $\mathcal{A}$ is sufficient to determine the structure of $\mathcal{A}$.

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 number and tree-depth

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

$$\begin{tikzcd} A \arrow[r, "\alpha"] \arrow[dr, "\mathrm{id}_A"'] & GA \arrow[d, "\varepsilon_A"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & GA \arrow[d, "\delta_A"] \\ GA \arrow[r, "G\alpha"] & G^2 A \end{tikzcd}$$

We should only expect a coalgebra structure $\alpha : \mathcal{A} \to \mathbb{E}_k \mathcal{A}$ to exist when the k-local information on $\mathcal{A}$ is sufficient to determine the structure of $\mathcal{A}$.

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 $\mathcal{A}$.

Theorem

For the Ehrenfeucht-Fraïssé comonad $\mathbb{E}_k$, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-depth of $\mathcal{A}$.

Coalgebra number and tree-depth

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

$$
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[dr, "\mathrm{id}_A"'] & GA \arrow[d, "\varepsilon_A"] \\
& A
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[r, "\alpha"] \arrow[d, "\alpha"'] & GA \arrow[d, "\delta_A"] \\
GA \arrow[r, "G\alpha"] & G^2 A
\end{tikzcd}
$$

We should only expect a coalgebra structure $\alpha : \mathcal{A} \to \mathbb{E}_k\mathcal{A}$ to exist when the k-local information on $\mathcal{A}$ is sufficient to determine the structure of $\mathcal{A}$.

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 $\mathcal{A}$.

Theorem

For the Ehrenfeucht-Fraïssé comonad $\mathbb{E}_k$, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-depth of $\mathcal{A}$.

This follows from the comonadicity of the adjunction.

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 $\mathcal{B}$.

Same same . . .

We can now run exactly the same script as for the Ehrenfeucht-Fraïssé case:

- There is a category of k-pebbled forest-ordered structures, and a resource-indexed adjunction with relational structures
- We can define paths, pathwise embeddings, open maps, bisimilarity in $\mathcal{R}^P_k(\sigma)$ in exactly the same fashion as we did for $\mathcal{R}^E_k(\sigma)$.
- Hence we can define bisimulations between object of the extensional category $\mathcal{R}(\sigma)$ using the resource category $\mathcal{R}^P_k(\sigma)$.
- We can define the equivalence relations $\mathcal{A} \rightleftarrows_k \mathcal{B}$, $\mathcal{A} \leftrightarrow_k \mathcal{B}$, $\mathcal{A} \cong_k \mathcal{B}$ with respect to $\mathcal{R}^P_k(\sigma)$.

Same same . . .

We can now run exactly the same script as for the Ehrenfeucht-Fraïssé case:

- There is a category of k-pebbled forest-ordered structures, and a resource-indexed adjunction with relational structures
- We can define paths, pathwise embeddings, open maps, bisimilarity in $\mathcal{R}^P_k(\sigma)$ in exactly the same fashion as we did for $\mathcal{R}^E_k(\sigma)$.
- Hence we can define bisimulations between object of the extensional category $\mathcal{R}(\sigma)$ using the resource category $\mathcal{R}^P_k(\sigma)$.
- We can define the equivalence relations $\mathcal{A} \rightleftarrows_k \mathcal{B}$, $\mathcal{A} \leftrightarrow_k \mathcal{B}$, $\mathcal{A} \cong_k \mathcal{B}$ with respect to $\mathcal{R}^P_k(\sigma)$.

We now take $\mathcal{L}_k$ to be the k-variable fragment of first-order logic. $\exists\mathcal{L}_k$ is the existential-positive part of this fragment, $\mathcal{L}_k(\#)$ its extension with counting quantifiers.

Same same . . .

We can now run exactly the same script as for the Ehrenfeucht-Fraïssé case:

- There is a category of k-pebbled forest-ordered structures, and a resource-indexed adjunction with relational structures
- We can define paths, pathwise embeddings, open maps, bisimilarity in $\mathcal{R}^P_k(\sigma)$ in exactly the same fashion as we did for $\mathcal{R}^E_k(\sigma)$.
- Hence we can define bisimulations between object of the extensional category $\mathcal{R}(\sigma)$ using the resource category $\mathcal{R}^P_k(\sigma)$.
- We can define the equivalence relations $\mathcal{A} \rightleftarrows_k \mathcal{B}$, $\mathcal{A} \leftrightarrow_k \mathcal{B}$, $\mathcal{A} \cong_k \mathcal{B}$ with respect to $\mathcal{R}^P_k(\sigma)$.

We now take $\mathcal{L}_k$ to be the k-variable fragment of first-order logic. $\exists\mathcal{L}_k$ is the existential-positive part of this fragment, $\mathcal{L}_k(\#)$ its extension with counting quantifiers.

With this notation, we get verbatim the same result as before, giving comonadic characterizations of logical equivalences.

Coalgebra number and tree-width

We can define the coalgebra number for the pebbling comonad exactly as done before for the Ehrenfeucht-Fraïssé comonad.
A slightly more subtle argument is needed to show:

Theorem

For the pebbling comonad $\mathbb{P}_k$, the coalgebra number of $\mathcal{A}$ corresponds precisely to the tree-width of $\mathcal{A}$.

Where we are

We now have a considerable number of examples of game comonads corresponding to various notions of model comparison game:

- pebbling comonad
- EF comonad
- modal comonad
- comonads for hybrid logic and other extensions of basic modal logic
- guarded quantifier comonads (atom, loose and clique guards)
- generalized quantifier comonads
- pathwidth comonad
- hypertree comonad

Where we are

We now have a considerable number of examples of game comonads corresponding to various notions of model comparison game:

- pebbling comonad
- EF comonad
- modal comonad
- comonads for hybrid logic and other extensions of basic modal logic
- guarded quantifier comonads (atom, loose and clique guards)
- generalized quantifier comonads
- pathwidth comonad
- hypertree comonad

In each case, we have tight connections with logical fragments, and with combinatorial invariants.

Where we are

We now have a considerable number of examples of game comonads corresponding to various notions of model comparison game:

- pebbling comonad
- EF comonad
- modal comonad
- comonads for hybrid logic and other extensions of basic modal logic
- guarded quantifier comonads (atom, loose and clique guards)
- generalized quantifier comonads
- pathwidth comonad
- hypertree comonad

In each case, we have tight connections with logical fragments, and with combinatorial invariants.

The connections involve both the coKleisli categories and the Eilenberg-Moore categories of the comonads.

Where we are

We now have a considerable number of examples of game comonads corresponding to various notions of model comparison game:

- pebbling comonad
- EF comonad
- modal comonad
- comonads for hybrid logic and other extensions of basic modal logic
- guarded quantifier comonads (atom, loose and clique guards)
- generalized quantifier comonads
- pathwidth comonad
- hypertree comonad

In each case, we have tight connections with logical fragments, and with combinatorial invariants.

The connections involve both the coKleisli categories and the Eilenberg-Moore categories of the comonads.

We get direct descriptions of the coalgebras in terms of *comonadic forgetful functors*. These are important both for formulating bisimulation, and for the connection with combinatorial invariants.

Summary table

$\mathbb{C}_k$	**Logic**	$\kappa^{\mathbb{C}}$	$\rightarrow_k^{\mathbb{C}}$	$\leftrightarrow_k^{\mathbb{C}}$	$\cong_k^{\mathbb{C}}$
$\mathbb{E}_k$ [AS21]	**FOL** w/ qr $\leq k$	tree-depth	✓	✓	✓
$\mathbb{P}_k$ [ADW17]	k-variable logic	treewidth $+1$	✓	✓	✓
$\mathbb{M}_k$ [AS21]	**ML** w/ md $\leq k$	sync. tree-depth	✓	✓	✓
$\mathbb{G}_k^{\mathfrak{g}}$ [AM20]	$\mathfrak{g}$-guarded logic w/ width $\leq k$	guarded treewidth	✓	✓	?
$\mathbb{H}_{n,k}$ [CD20]	k-variable logic w/ $\mathbf{Q}_n$-quantifiers	n-ary general treewidth	✓	✓	✓
$\mathbb{PR}_k$	k-variable logic restricted-$\wedge$	pathwidth $+1$	✓	?	?
$\mathbb{LG}_k$	k-conjunct guarded logic	hypertree-width	✓	?	?

Current developments

- First wave: establishing the paradigm, finding many examples.
- Culmination in an axiomatic framework of *arboreal categories* and *arboreal covers*.
- Second wave: an emerging landscape, "dividing lines" beginning to appear, structural features.
  - General versions of model-theoretic results such as preservation theorems: Rossman's homomorphism preservation theorems, van Benthem-Rosen, etc.
  - Uniform proofs of preservation theorems in the finite and infinite cases: "model theory without compactness".
  - Structural features of comonads (idempotence, bisimilar companions property), and their significance for computational tractability.
  - Lovasz-type theorems on counting homomorphisms.
  - Combinatorial parameters: concrete cases, axiomatic approach via density comonads.
  - Classification of constraint satisfaction problems: use of sheaf-theoretic and cohomological methods originally developed for quantum contextuality.

Current developments

- First wave: establishing the paradigm, finding many examples.
- Culmination in an axiomatic framework of *arboreal categories* and *arboreal covers*.
- Second wave: an emerging landscape, "dividing lines" beginning to appear, structural features.
  - General versions of model-theoretic results such as preservation theorems: Rossman's homomorphism preservation theorems, van Benthem-Rosen, etc.
  - Uniform proofs of preservation theorems in the finite and infinite cases: "model theory without compactness".
  - Structural features of comonads (idempotence, bisimilar companions property), and their significance for computational tractability.
  - Lovasz-type theorems on counting homomorphisms.
  - Combinatorial parameters: concrete cases, axiomatic approach via density comonads.
  - Classification of constraint satisfaction problems: use of sheaf-theoretic and cohomological methods originally developed for quantum contextuality.

Survey paper: Structure and Power: an emerging synthesis, SA, in *Fundamenta Informaticae* 2022, also `arXiv:2206.07393`

Arboreal Categories and HPT

- We axiomatize the notion of a category with intrinsic tree structure in a very general setting, assuming only a factorization system and some colimits.
- The whole pattern of results described in our examples can be carried out at this abstract level.
- For example, we can define back-and-forth games, and prove their equivalence to bisimulations, at the abstract level.
- This framework has been used to give a proof of a general form of Rossman's Equirank Homomorphism Preservation Theorem, which is a *tour de force* of (finite) model theory.
- This leads to an Equivariable HPT.

CSP and the Feder-Vardi Conjecture

Given a finite relational structure B over a finite relational vocabulary σ, the *constraint satisfaction problem* $\mathrm{CSP}(B)$ is to decide, for an *instance* given by a finite σ-structure A, whether there is a homomorphism $A \to B$.

CSP and the Feder-Vardi Conjecture

Given a finite relational structure B over a finite relational vocabulary σ, the *constraint satisfaction problem* $\mathrm{CSP}(B)$ is to decide, for an *instance* given by a finite σ-structure A, whether there is a homomorphism $A \to B$.

The Feder-Vardi Conjecture (1993):

For every B, $\mathrm{CSP}(B)$ is either polynomial-time solvable, or NP-complete.

CSP and the Feder-Vardi Conjecture

Given a finite relational structure B over a finite relational vocabulary σ, the *constraint satisfaction problem* $\mathrm{CSP}(B)$ is to decide, for an *instance* given by a finite σ-structure A, whether there is a homomorphism $A \to B$.

The Feder-Vardi Conjecture (1993):

For every B, $\mathrm{CSP}(B)$ is either polynomial-time solvable, or NP-complete.

Evidence: known problems at the time were either NP-complete (e.g. 3-SAT), or fell into two "islands of tractability":

- bounded width – exactly solvable by local consistency, or
- "subgroup problems"

CSP and the Feder-Vardi Conjecture

Given a finite relational structure B over a finite relational vocabulary σ, the *constraint satisfaction problem* $\mathrm{CSP}(B)$ is to decide, for an *instance* given by a finite σ-structure A, whether there is a homomorphism $A \to B$.

The Feder-Vardi Conjecture (1993):

> For every B, $\mathrm{CSP}(B)$ is either polynomial-time solvable, or NP-complete.

Evidence: known problems at the time were either NP-complete (e.g. 3-SAT), or fell into two "islands of tractability":

- bounded width – exactly solvable by local consistency, or
- "subgroup problems"

The two tractable classes identified by Feder and Vardi appeared to be quite different in character.

CSP and the Feder-Vardi Conjecture

Given a finite relational structure B over a finite relational vocabulary σ, the *constraint satisfaction problem* $\mathrm{CSP}(B)$ is to decide, for an *instance* given by a finite σ-structure A, whether there is a homomorphism $A \to B$.

The Feder-Vardi Conjecture (1993):

For every B, $\mathrm{CSP}(B)$ is either polynomial-time solvable, or NP-complete.

Evidence: known problems at the time were either NP-complete (e.g. 3-SAT), or fell into two "islands of tractability":

- bounded width – exactly solvable by local consistency, or
- "subgroup problems"

The two tractable classes identified by Feder and Vardi appeared to be quite different in character.

This conjecture was recently proved (independently) by Bulatov and Zhuk (*c.* 2017).

Contextuality

In a nutshell: contextually arises where we have a family of overlapping pieces of data which is *locally consistent, but globally inconsistent*.

Contextuality

In a nutshell: contextually arises where we have a family of overlapping pieces of data which is *locally consistent, but globally inconsistent*.

Illustration: local consistency

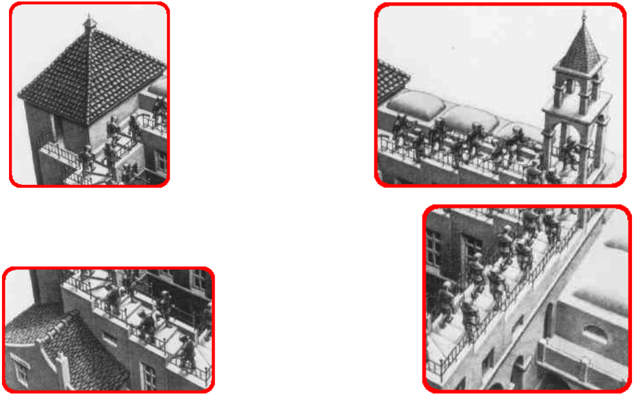
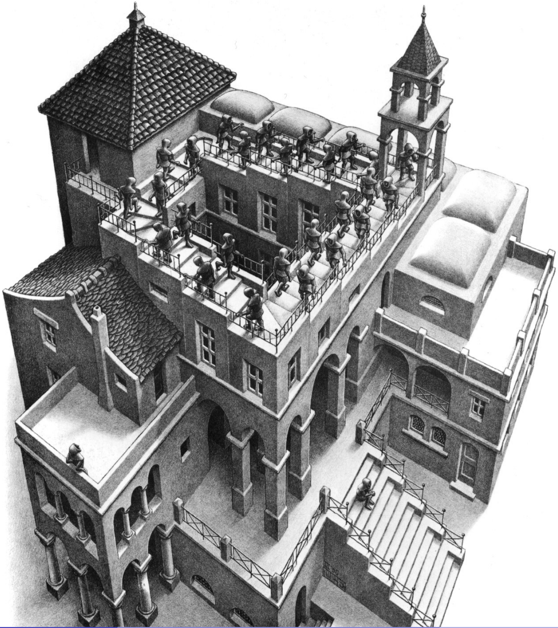


Illustration: global inconsistency



Topology of Paradox

- Clearly, the staircase *as a whole* cannot exist in the real world. Nonetheless, the constituent parts make sense *locally*.
- Quantum contextuality shows that the logical structure of quantum mechanics exhibits exactly these features of *local consistency*, but *global inconsistency*.
- This can happen because *not all variables can be measured at the same time* (non-commuting observables).
- We note that Escher's work was inspired by the *Penrose stairs*.
- Indeed, these figures provide more than a mere analogy. Penrose has studied the topological "twisting" in these figures using cohomology. This is quite analogous to our use of sheaf cohomology to capture the logical twisting in contextuality.
- Recent cross-over of these ideas into Constraint Satisfaction and structure isomorphism (refinements of Weisfeiler-Leman).

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Given an approximation level k, we test if there are solutions for all $\leq k$-element subsets of the instance.

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Given an approximation level k, we test if there are solutions for all $\leq k$-element subsets of the instance.

More precisely, this amounts to the existence of a non-empty family S of homomorphisms $f : C \to B$, where C is an induced substructure of A with $|C| \leq k$.

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Given an approximation level k, we test if there are solutions for all $\leq k$-element subsets of the instance.

More precisely, this amounts to the existence of a non-empty family S of homomorphisms $f : C \to B$, where C is an induced substructure of A with $|C| \leq k$.

This is subject to the following conditions:

- **down-closure**: If $f : C \to B \in S$ and $C' \subseteq C$, then $f|_{C'} : C' \to B \in S$.
- **forth condition**: If $f : C \to B \in S$, $|C| < k$, and $a \in A$, then for some $f' : C \cup \{a\} \to B \in S$, $f'|_C = f$.

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Given an approximation level k, we test if there are solutions for all $\leq k$-element subsets of the instance.

More precisely, this amounts to the existence of a non-empty family S of homomorphisms $f : C \to B$, where C is an induced substructure of A with $|C| \leq k$.

This is subject to the following conditions:

- **down-closure**: If $f : C \to B \in S$ and $C' \subseteq C$, then $f|_{C'} : C' \to B \in S$.
- **forth condition**: If $f : C \to B \in S$, $|C| < k$, and $a \in A$, then for some $f' : C \cup \{a\} \to B \in S$, $f'|_C = f$.

This is equivalent to the existence of a winning strategy for Duplicator in the existential k-pebble game from A to B.

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Given an approximation level k, we test if there are solutions for all $\leq k$-element subsets of the instance.

More precisely, this amounts to the existence of a non-empty family S of homomorphisms $f : C \to B$, where C is an induced substructure of A with $|C| \leq k$.

This is subject to the following conditions:

- **down-closure**: If $f : C \to B \in S$ and $C' \subseteq C$, then $f|_{C'} : C' \to B \in S$.
- **forth condition**: If $f : C \to B \in S$, $|C| < k$, and $a \in A$, then for some $f' : C \cup \{a\} \to B \in S$, $f'|_C = f$.

This is equivalent to the existence of a winning strategy for Duplicator in the existential k-pebble game from A to B.

Notation: $A \to_k B$.

Local consistency as strong k-consistency: the Kolaitis-Vardi representation

Given an approximation level k, we test if there are solutions for all $\leq k$-element subsets of the instance.

More precisely, this amounts to the existence of a non-empty family S of homomorphisms $f : C \to B$, where C is an induced substructure of A with $|C| \leq k$.

This is subject to the following conditions:

- **down-closure**: If $f : C \to B \in S$ and $C' \subseteq C$, then $f|_{C'} : C' \to B \in S$.
- **forth condition**: If $f : C \to B \in S$, $|C| < k$, and $a \in A$, then for some $f' : C \cup \{a\} \to B \in S$, $f'|_C = f$.

This is equivalent to the existence of a winning strategy for Duplicator in the existential k-pebble game from A to B.

Notation: $A \to_k B$.

This fits perfectly into the sheaf-theoretic language used to capture contextuality by Abramsky-Brandenburger et al!

Global sections and cohomology

A *global section* is a family of partial homomorphisms $\{s_C : C \to B\}_{C \subseteq A, |C| \leq k}$ which agrees on overlaps:

$$\forall C, C' : \;\; s_C|_{C \cap C'} = s_{C'}|_{C \cap C'}$$

Global sections and cohomology

A *global section* is a family of partial homomorphisms $\{s_C : C \to B\}_{C \subseteq A, |C| \leq k}$ which agrees on overlaps:

$$\forall C, C' : \;\; s_C|_{C \cap C'} = s_{C'}|_{C \cap C'}$$

Global sections are in $1-1$ correspondence with homomorphisms $A \to B$.

Global sections and cohomology

A *global section* is a family of partial homomorphisms $\{s_C : C \to B\}_{C \subseteq A, |C| \leq k}$ which agrees on overlaps:

$$\forall C, C' : \;\; s_C|_{C \cap C'} = s_{C'}|_{C \cap C'}$$

Global sections are in $1-1$ correspondence with homomorphisms $A \to B$.

In previous work with Barbosa, Mansfield, Kishida and Lal (ABKLM) I used *cohomology* to characterise contextuality, *i.e.* the *non-existence* of such a homomorphism.

Global sections and cohomology

A *global section* is a family of partial homomorphisms $\{s_C : C \to B\}_{C \subseteq A, |C| \leq k}$ which agrees on overlaps:

$$\forall C, C' : \;\; s_C|_{C \cap C'} = s_{C'}|_{C \cap C'}$$

Global sections are in $1-1$ correspondence with homomorphisms $A \to B$.

In previous work with Barbosa, Mansfield, Kishida and Lal (ABKLM) I used *cohomology* to characterise contextuality, *i.e.* the *non-existence* of such a homomorphism.

This gives linear-algebraic approximations to global sections: a family $\{r_C\}_C$ where r_C is a $\mathbb{Z}$*-linear combination* of local sections, satisfying the compatibility (*i.e.* agree-on-overlap) conditions, suitably extended to linear combinations.

Global sections and cohomology

A *global section* is a family of partial homomorphisms $\{s_C : C \to B\}_{C \subseteq A, |C| \leq k}$ which agrees on overlaps:

$$\forall C, C' : \;\; s_C|_{C \cap C'} = s_{C'}|_{C \cap C'}$$

Global sections are in $1-1$ correspondence with homomorphisms $A \to B$.

In previous work with Barbosa, Mansfield, Kishida and Lal (ABKLM) I used *cohomology* to characterise contextuality, *i.e.* the *non-existence* of such a homomorphism.

This gives linear-algebraic approximations to global sections: a family $\{r_C\}_C$ where r_C is a $\mathbb{Z}$*-linear combination* of local sections, satisfying the compatibility (*i.e.* agree-on-overlap) conditions, suitably extended to linear combinations.

Given $s : C_0 \to B$, we can ask if it has an extension to such a $\mathbb{Z}$-linear family $\{r_C\}$, with $r_{C_0} = 1 \cdot s$.

Global sections and cohomology

A *global section* is a family of partial homomorphisms $\{s_C : C \to B\}_{C \subseteq A, |C| \leq k}$ which agrees on overlaps:

$$\forall C, C' : \;\; s_C|_{C \cap C'} = s_{C'}|_{C \cap C'}$$

Global sections are in $1 - 1$ correspondence with homomorphisms $A \to B$.

In previous work with Barbosa, Mansfield, Kishida and Lal (ABKLM) I used *cohomology* to characterise contextuality, *i.e.* the *non-existence* of such a homomorphism.

This gives linear-algebraic approximations to global sections: a family $\{r_C\}_C$ where r_C is a $\mathbb{Z}$-*linear combination* of local sections, satisfying the compatibility (*i.e.* agree-on-overlap) conditions, suitably extended to linear combinations.

Given $s : C_0 \to B$, we can ask if it has an extension to such a $\mathbb{Z}$-linear family $\{r_C\}$, with $r_{C_0} = 1 \cdot s$.

We can use this test to filter out those local sections from the k-consistency approximation which *do not have* such extensions, getting a sharper approximation.

Cohomological k-consistency

Key insight by Adam O' Conghaile: this cohomological refinement of k-consistency is *efficiently computable*!

(Since the predicate "s has a $\mathbb{Z}$-linear extension" translates into solvability of a polynomial size system of $\mathbb{Z}$-linear equations).

Cohomological k-consistency

Key insight by Adam O' Conghaile: this cohomological refinement of k-consistency is *efficiently computable*!
(Since the predicate "s has a $\mathbb{Z}$-linear extension" translates into solvability of a polynomial size system of $\mathbb{Z}$-linear equations).

So we get a new refined approximation algorithm for $\mathrm{CSP}(B)$.

Cohomological k-consistency

Key insight by Adam O' Conghaile: this cohomological refinement of k-consistency is *efficiently computable*!
(Since the predicate "s has a $\mathbb{Z}$-linear extension" translates into solvability of a polynomial size system of $\mathbb{Z}$-linear equations).

So we get a new refined approximation algorithm for $\mathrm{CSP}(B)$.

A key property which follows from the ABKLM results on contextuality:

Proposition

Cohomological k-consistency is exact for affine templates B (i.e. solving linear equations over finite rings).

Cohomological k-consistency

Key insight by Adam O' Conghaile: this cohomological refinement of k-consistency is *efficiently computable*!
(Since the predicate "s has a $\mathbb{Z}$-linear extension" translates into solvability of a polynomial size system of $\mathbb{Z}$-linear equations).

So we get a new refined approximation algorithm for $\mathrm{CSP}(B)$.

A key property which follows from the ABKLM results on contextuality:

Proposition

Cohomological k-consistency is exact for affine templates B (i.e. solving linear equations over finite rings).

So cohomological k-consistency captures the two main tractable classes identified by Feder-Vardi in a unified fashion.

Cohomological k-consistency

Key insight by Adam O' Conghaile: this cohomological refinement of k-consistency is *efficiently computable*!
(Since the predicate "s has a $\mathbb{Z}$-linear extension" translates into solvability of a polynomial size system of $\mathbb{Z}$-linear equations).

So we get a new refined approximation algorithm for $\mathrm{CSP}(B)$.

A key property which follows from the ABKLM results on contextuality:

Proposition

Cohomological k-consistency is exact for affine templates B (i.e. solving linear equations over finite rings).

So cohomological k-consistency captures the two main tractable classes identified by Feder-Vardi in a unified fashion.

With Adam, Rui and Anuj, we are currently working on determining the exact power of cohomological k-consistency:

Question

Is cohomological k-consistency exact for all tractable cases?

Cohomological k-equivalence

The whole story adapts quite straightforwardly to yield cohomological refinements of the Weisfeiler-Leman approximations to graph and structure isomorphism.

Cohomological k-equivalence

The whole story adapts quite straightforwardly to yield cohomological refinements of the Weisfeiler-Leman approximations to graph and structure isomorphism.

Recall that these correspond to equivalence in k-variable logic with counting $\mathbb{C}_k$.

Cohomological k-equivalence

The whole story adapts quite straightforwardly to yield cohomological refinements of the Weisfeiler-Leman approximations to graph and structure isomorphism.

Recall that these correspond to equivalence in k-variable logic with counting $\mathbb{C}_k$.

Moreover, the result on completeness of cohomological k-consistency for affine templates is leveraged to show that $\equiv_k^{\mathbb{Z}}$ is discriminating enough to defeat two important families of counter-examples:

- the CFI (Cai-Furer-Immerman) construction used to show that $\mathbb{C}_k$ is not strong enough to characterise polynomial time, and
- the constructions due to Lichter and Dawar et al. which are used to show similar results for linear algebraic extensions of $\mathbb{C}_k$.

Cohomological k-equivalence

The whole story adapts quite straightforwardly to yield cohomological refinements of the Weisfeiler-Leman approximations to graph and structure isomorphism.

Recall that these correspond to equivalence in k-variable logic with counting $\mathbb{C}_k$.

Moreover, the result on completeness of cohomological k-consistency for affine templates is leveraged to show that $\equiv_k^{\mathbb{Z}}$ is discriminating enough to defeat two important families of counter-examples:

- the CFI (Cai-Furer-Immerman) construction used to show that $\mathbb{C}_k$ is not strong enough to characterise polynomial time, and
- the constructions due to Lichter and Dawar et al. which are used to show similar results for linear algebraic extensions of $\mathbb{C}_k$.

References:

- `https://arxiv.org/abs/2206.15253` (AOC paper to appear in MFCS)
- `https://arxiv.org/abs/2206.12156` (SA notes)

Samson Abramsky, Anuj Dawar, and Pengming Wang.
The pebbling comonad in finite model theory.
In *Logic in Computer Science (LICS), 2017 32nd Annual ACM/IEEE Symposium on*, pages 1–12. IEEE, 2017.

Samson Abramsky and Dan Marsden.
Comonadic semantics for guarded fragments, 2020.
In LiCS 2021.

Samson Abramsky and Nihil Shah.
Relating structure and power: Comonadic semantics for computational resources, 2021.
In *Journal of Logic and Computation.*

Adam Ó Conghaile and Anuj Dawar.
Game comonads & generalised quantifiers, 2020.