An Invitation to Game Comonads, day 3: Coalgebras and Combinatorial Parameters [a]

Tomáš Jakl

6 August 2025

ESSLLI 2025, Bochum

[a] This project has received funding from the EU's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101111373.

Funded by the European Union

Summary of Day 2

Was discussed:

- Model comparison games capture relationships in logic.
- Forth-only versions of some games modelled semantically as

$$G(A) \to B$$

- These constructions satisfies axioms of a comonad $(G, \varepsilon, (\cdot)^*)$:

$$\varepsilon_A^* = \mathrm{id}_{G(A)} \qquad \varepsilon_B \circ f^* = f \qquad (g \circ f^*)^* = g^* \circ f^*$$

Obvious questions:

- What can we use from the theory of (co)monads?
- Generic proofs by employing categorical tools?

Functors

(from day 1)

Recall, functors are "homomorphisms of categories".

A functor $F\colon \mathscr{C} \to \mathscr{D}$ is given by

- a mapping on objects $F\colon \mathrm{Ob}(\mathscr{C}) \to \mathrm{Ob}(\mathscr{D})$
- a mapping on morphisms, for every $A, B \in \mathscr{C}$,

$$F\colon \mathscr{C}(A, B) \to \mathscr{D}(F(A), F(B))$$

which preserves identities and compositions:

$$F(\mathrm{id}_A) = \mathrm{id}_{F(A)}$$
$$F(f \circ g) = F(f) \circ F(g)$$

Example: comonads extend to functors!

Given a comonad $(G, \varepsilon, (\cdot)^*)$ on $\mathscr{C}$, define

$$f \colon A \to B \quad \longmapsto \quad G(f) \colon G(A) \to G(B)$$

Example: comonads extend to functors!

Given a comonad $(G, \varepsilon, (\cdot)^*)$ on $\mathscr{C}$, define

$$f\colon A \to B \quad \longmapsto \quad \begin{aligned} &G(f)\colon G(A) \to G(B) \\ &G(f) = (f \circ \varepsilon_A)^* \end{aligned}$$

G is a functor $\mathscr{C} \to \mathscr{C}$ as

$$G(\mathrm{id}_A) = (\mathrm{id}_A \circ \varepsilon_A)^* = (\varepsilon_A)^* = \mathrm{id}_{G(A)}$$

$$\begin{aligned} G(f) \circ G(g) &= (f \circ \varepsilon)^* \circ (g \circ \varepsilon)^* = (f \circ \varepsilon \circ (g \circ \varepsilon)^*)^* \\ &= (f \circ g \circ \varepsilon)^* = G(f \circ g) \end{aligned}$$

Example: comonads extend to functors!

Given a comonad $(G, \varepsilon, (\cdot)^*)$ on $\mathscr{C}$, define

$$f\colon A \to B \quad \longmapsto \quad \begin{array}{l} G(f)\colon G(A) \to G(B) \\ G(f) = (f \circ \varepsilon_A)^* \end{array}$$

G is a functor $\mathscr{C} \to \mathscr{C}$ as

$$G(\mathrm{id}_A) = (\mathrm{id}_A \circ \varepsilon_A)^* = (\varepsilon_A)^* = \mathrm{id}_{G(A)}$$

$$\begin{aligned} G(f) \circ G(g) &= (f \circ \varepsilon)^* \circ (g \circ \varepsilon)^* = (f \circ \varepsilon \circ (g \circ \varepsilon)^*)^* \\ &= (f \circ g \circ \varepsilon)^* = G(f \circ g) \end{aligned}$$

Example

For $h\colon A \to B$ in $\mathbf{Str}(\sigma)$, the functor $\mathbb{E}_k(h)\colon \mathbb{E}_k(A) \to \mathbb{E}_k(B)$ maps $[a_1, \ldots, a_n]$ to $[h(a_1), \ldots, h(a_n)]$.

Eilenberg–Moore coalgebras

Given a comonad $(G, \varepsilon, (\cdot)^*)$ on $\mathscr{C}$, for every $A \in \mathrm{Ob}(G)$, define the **comultiplication**

$$\delta_A \colon G(A) \to G(G(A))$$

as the morphism $(\mathrm{id}_{G(A)})^*$.

Given a comonad $(G, \varepsilon, (\cdot)^*)$ on $\mathscr{C}$, for every $A \in \mathrm{Ob}(G)$, define the **comultiplication**

$$\delta_A \colon G(A) \to G(G(A))$$

as the morphism $(\mathrm{id}_{G(A)})^*$.

Then, a morphism $\alpha \colon A \to G(A)$ is a G-**coalgebra on** A if

$$\begin{array}{ccc} A & & \\ {\scriptstyle\alpha}\downarrow & \searrow^{\mathrm{id}} & \\ G(A) & \xrightarrow{\varepsilon_A} & A \end{array} \qquad\qquad \begin{array}{ccc} A & \xrightarrow{\alpha} & G(A) \\ {\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\delta_A} \\ G(A) & \xrightarrow{G(\alpha)} & G(G(A)) \end{array}$$

(i.e. $\varepsilon_A \circ \alpha = \mathrm{id}$ and $\delta_A \circ \alpha = G(\alpha) \circ \alpha$)

Origins – Dual notions

Algebras as functions in **Set**

$$F(A) \to A$$

E.g. for signature $\Sigma = \{\vee, \neg\}$ and $F(A) = (A \times A) \uplus A$

$$\text{functions } (A \times A) \uplus A \to A \quad \approx \quad \Sigma\text{-algebras } (A, \vee, \neg)$$

Origins – Dual notions

Algebras as functions in **Set**

$$F(A) \to A$$

E.g. for signature $\Sigma = \{\vee, \neg\}$ and $F(A) = (A \times A) \uplus A$

$$\text{functions } (A \times A) \uplus A \to A \quad \approx \quad \Sigma\text{-algebras } (A, \vee, \neg)$$

Correspondence (Σ signature, $\mathcal{E}$ equations) $\longleftrightarrow$ monads T

$$\mathbf{Alg}(\Sigma, \mathcal{E}) \cong \left\{ T(A) \xrightarrow{\alpha} A \;\middle|\; \begin{array}{ccc} A & \xrightarrow{\eta_A} & T(A) \\ & \overset{\text{id}}{\searrow} & \downarrow \alpha \\ & & A \end{array} \quad \text{and} \quad \begin{array}{ccc} T^2(A) & \xrightarrow{T(\alpha)} & T(A) \\ \mu \downarrow & & \downarrow \alpha \\ T(A) & \xrightarrow{\alpha} & A \end{array} \right\}$$

Warming up with the List comonad

Define a comonad on **Set**

$$\texttt{List}\colon \mathrm{Ob}(\textbf{Set}) \to \mathrm{Ob}(\textbf{Set}), \quad A \mapsto \{\,[a_1, \ldots, a_n] \mid a_i \in A\}$$

The counit is

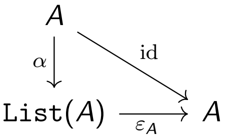
$$\varepsilon_A\colon \texttt{List}(A) \to A, \qquad [a_1, \ldots, a_n] \mapsto a_n$$

and, for a function $f\colon \texttt{List}(A) \to B$, define

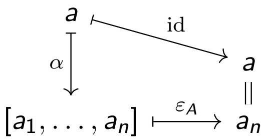
$$f^*\colon \texttt{List}(A) \to \texttt{List}(B)$$

by $[a_1, \ldots, a_n] \mapsto [b_1, \ldots, b_n]$ where $b_i = f([a_1, \ldots, a_i])$

List-**coalgebras, the first axiom**



imposes



List-coalgebras, the second axiom

$$\begin{array}{ccc}
A & \xrightarrow{\quad\alpha\quad} & \texttt{List}(A) \\
{\scriptstyle\alpha}\downarrow & & \downarrow{\scriptstyle\delta_A} \\
\texttt{List}(A) & \xrightarrow{\texttt{List}(\alpha)} & \texttt{List}(\texttt{List}(A))
\end{array}$$

`List`-coalgebras, the second axiom

$$
\begin{array}{ccc}
A & \xrightarrow{\quad\alpha\quad} & \mathtt{List}(A) \\
{\scriptstyle\alpha}\big\downarrow & & \big\downarrow{\scriptstyle\delta_A} \\
\mathtt{List}(A) & \xrightarrow{\mathtt{List}(\alpha)} & \mathtt{List}(\mathtt{List}(A))
\end{array}
$$

imposes

$$
\begin{array}{ccc}
a & \xmapsto{\quad\alpha\quad} & [a_1, \ldots, a_n] \\
 & & \big\downarrow{\scriptstyle\delta_A} \\
{\scriptstyle\alpha}\big\downarrow & & [[a_1], [a_1, a_2], \ldots, [a_1, \ldots, a_n]] \\
 & & \shortparallel \\
[a_1, \ldots, a_n] & \xmapsto{\mathtt{List}(\alpha)} & [\alpha(a_1), \ldots, \alpha(a_n)]
\end{array}
$$

Therefore

$$\alpha(a_i) = [a_1, \ldots, a_i]$$

$\texttt{List}$-coalgebras, the second axiom: forest order

For, $w, w' \in \texttt{List}(A)$, write

$$w \sqsubseteq w' \quad \text{for} \quad w \text{ is a prefix of } w'$$

For, $w, w' \in \texttt{List}(A)$, write

$$w \sqsubseteq w' \quad \text{for} \quad w \text{ is a prefix of } w'$$

Consequently,

- If $\alpha(a) = [a_1, \ldots, a_n]$ then $\alpha(a_i) \sqsubseteq \alpha(a_j)$ iff $i \leq j$.

List-coalgebras, the second axiom: forest order

For, $w, w' \in \texttt{List}(A)$, write

$$w \sqsubseteq w' \quad \text{for} \quad w \text{ is a prefix of } w'$$

Consequently,

- If $\alpha(a) = [a_1, \ldots, a_n]$ then $\alpha(a_i) \sqsubseteq \alpha(a_j)$ iff $i \leq j$.
- The set $\{a_1, \ldots, a_n\}$ is a <u>chain</u> in the $\leq_\alpha$-order where

$$a \leq_\alpha a' \iff \alpha(a) \sqsubseteq \alpha(a')$$

$\texttt{List}$-coalgebras, the second axiom: forest order

For, $w, w' \in \texttt{List}(A)$, write

$$w \sqsubseteq w' \quad \text{for} \quad w \text{ is a prefix of } w'$$

Consequently,

- If $\alpha(a) = [a_1, \ldots, a_n]$ then $\alpha(a_i) \sqsubseteq \alpha(a_j)$ iff $i \leq j$.
- The set $\{a_1, \ldots, a_n\}$ is a <u>chain</u> in the $\leq_\alpha$-order where

$$a \leq_\alpha a' \iff \alpha(a) \sqsubseteq \alpha(a')$$

- $\leq_\alpha$ defines a **forest order**:
  - $(A, \leq_\alpha)$ is a poset
  - $\forall a \in A \quad {\downarrow}a = \{x \in A \mid x \leq_\alpha a\}$ is a finite chain.

$\texttt{List}$-coalgebras, recovering from forest orders

For a poset $(A, \leq)$ where $\leq$ is a forest order, define

$$\alpha_{\leq} \colon A \to \texttt{List}(A)$$

by setting

$$\alpha_{\leq}(a) = [a_1, \ldots, a_n]$$

where

$$\downarrow a = \{a_1, \ldots, a_n\} \quad \text{is the chain} \quad a_1 < \cdots < a_n = a$$

Exercise

The mapping $\alpha_{\leq}$ is a $\texttt{List}$-coalgebra.

Proposition

For any set $A \in \mathbf{Set}$, there is a bijective correspondence between

- *coalgebras $A \to \mathtt{List}(A)$*
- *forest orders $\leq$ on A*

Proof.

It is enough to observe that $\alpha = \alpha_{\leq_\alpha}$ and $\leq = \leq_{\alpha_\leq}$. □

Exercise

Draw the tree for the `List`-coalgebra $(\{0, 1, 2, 3\}, \alpha)$ where

$$\alpha(0) = [0] \qquad \alpha(2) = [0, 1, 2]$$
$$\alpha(1) = [0, 1] \qquad \alpha(3) = [0, 3]$$

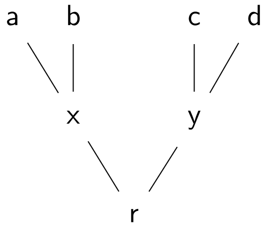
Exercise

Draw the tree for the `List`-coalgebra $(\{0, 1, 2, 3\}, \alpha)$ where

$$\alpha(0) = [0] \qquad \alpha(2) = [0, 1, 2]$$
$$\alpha(1) = [0, 1] \qquad \alpha(3) = [0, 3]$$

Exercise

Define the `List` coalgebra corresponding to the following tree



Morphisms of G-coalgebras

G-coalgebras form the **Eilenberg–Moore category**

$$\mathbf{EM}(G)$$

with

- Objects: (A, α) where $\alpha \colon A \to G(A)$ is a G-coalgebra
- Morphisms: $(A, \alpha) \to (B, \beta)$ are morphisms $f \colon A \to B$ in $\mathscr{C}$ such that

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

Exercise: Check that $\mathbf{EM}(G)$ is a category.

Example: morphisms of $\texttt{List}$-coalgebras

Proposition

Given $\texttt{List}$-coalgebras (A, α), (B, β) and a function $f \colon A \to B$, the following are equivalent:

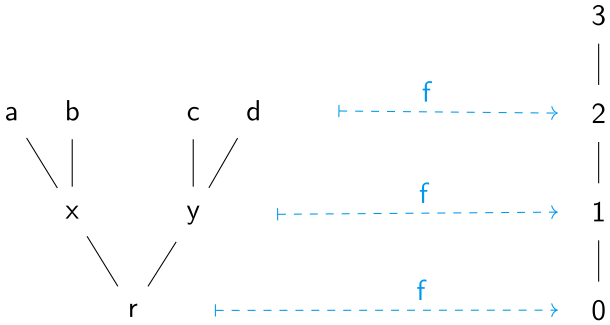
- *f is a coalgebra morphism* $(A, \alpha) \to (B, \beta)$, *i.e.*

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

- *f is a* **forest morphism** $(A, \leq_\alpha) \to (B, \leq_\beta)$ *i.e.*
  - *f preserves roots (i.e. minimal elements)*
  - $a \prec a' \implies f(a) \prec f(a')$

  where $a \prec a'$ iff $a < a'$ and $a \leq z \leq a'$ implies $a = z$ or $a' = z$.

Example



Theorem

The category $\mathbf{EM}(\mathtt{List})$ *is isomorphic to the category of forest orders and forest morphism.*

Coalgebras of $\mathbb{E}_k, \mathbb{P}_k, \mathbb{M}_k$

$\mathbb{E}_k$-coalgebras

Proposition

There is a bijection between coalgebras $\alpha\colon A \to \mathbb{E}_k(A)$ *and*

compatible forest orders $\leq$ on A of depth at most k

that is, relations $\leq$ on A such that

(T1) $\leq$ *is a forest order*

(T2) $\downarrow a$ *has at most* $\leq k$ *elements, for every* $a \in A$

(T3) $(a_1, \ldots, a_n) \in R^A$ *implies* $a_i \leq a_j$ *or* $a_j \leq a_i$ $\quad (\forall i, j)$

Proposition

There is a bijection between coalgebras $\alpha\colon A \to \mathbb{E}_k(A)$ and

compatible forest orders $\leq$ on A of depth at most k

that is, relations $\leq$ on A such that

(T1) $\leq$ *is a forest order*

(T2) $\downarrow a$ *has at most $\leq k$ elements, for every $a \in A$*

(T3) $(a_1, \ldots, a_n) \in R^A$ *implies* $a_i \leq a_j$ *or* $a_j \leq a_i$ $(\forall i, j)$

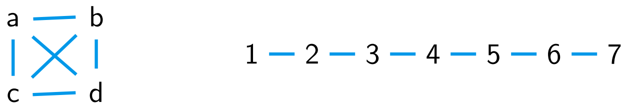
Proof.

$(a_1, \ldots, a_n) \in R^A$ implies $(\alpha(a_1), \ldots, \alpha(a_n)) \in R^{\mathbb{E}_k(A)}$ i.e.

- $\alpha(a_i) \sqsubseteq \alpha(a_j)$ or $\alpha(a_j) \sqsubseteq \alpha(a_i)$ $(\forall i, j)$
- $(\varepsilon(\alpha(a_1)), \ldots, \varepsilon(\alpha(a_n))) = (a_1, \ldots, a_n) \in R^A$ ✓ (automatic) □

Exercise

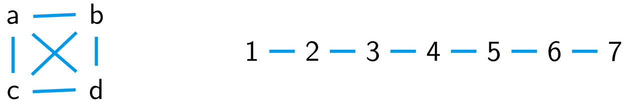
Given graphs



what are the minimal k such that they admit an $\mathbb{E}_k$-coalgebra?

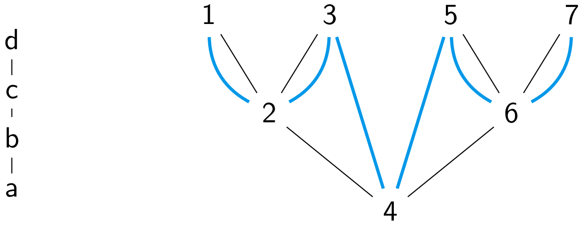
Exercise

Given graphs



what are the minimal k such that they admit an $\mathbb{E}_k$-coalgebra?

Answer



Proposition

There is a bijection between coalgebras $\alpha\colon A \to \mathbb{P}_k(A)$ *and*

compatible k**-pebble forest orders** $\leq, p$ **on** A

that is, relations $\leq$ *and pebbling functions* $p\colon A \to \{1, \ldots, k\}$ *satisfying*

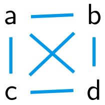
(T1) $\leq$ *is a forest order*

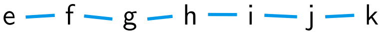
(T3') $(a_1, \ldots, a_n) \in R^A$ *implies*

- $a_i \leq a_j$ *or* $a_j \leq a_i$ $(\forall i, j)$.
- $\forall z \quad a_i < z \leq a_j \implies p(a_i) \neq p(z)$

Exercise

Given graphs

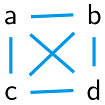


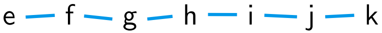


what are the minimal k such that they admit an $\mathbb{P}_k$-coalgebra?

Exercise

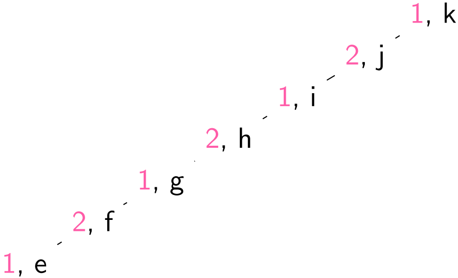
Given graphs





what are the minimal k such that they admit an $\mathbb{P}_k$-coalgebra?

Answer



Proposition

There is coalgebra $\alpha\colon (A, a) \to \mathbb{M}_k(A, a)$ *iff*

(A, a) **is a synchronization tree of depth at most** k

i.e., for every $x \in A$*, there is a unique path of length* $\leq k$

$$a \xrightarrow{R_1} a_1 \xrightarrow{R_2} \dots \xrightarrow{R_n} x$$

Proposition

There is coalgebra $\alpha\colon (A, a) \to \mathbb{M}_k(A, a)$ *iff*

(A, a) **is a synchronization tree of depth at most** k

i.e., for every $x \in A$*, there is a unique path of length* $\leq k$

$$a \xrightarrow{R_1} a_1 \xrightarrow{R_2} \ldots \xrightarrow{R_n} x$$

In fact, synchronization trees are automatically forest ordered:

$$x \prec y \iff (x, y) \in R^A \quad \text{for a (unique) binary } R \in \sigma$$

Theorem (Abramsky–Shah, 2021)

$\mathbf{EM}(\mathbb{E}_k)$ *is isomorphic to the category with*

- *objects:* σ*-structures with a compatible forest order of depth at most* k
- *morphisms:* *homomorphisms of* σ*-structures that are also forest morphisms.*

Theorem (Abramsky–Shah, 2021)

$\mathbf{EM}(\mathbb{P}_k)$ *is isomorphic to the category with*

- *objects:* σ*-structures with a compatible* k*-pebble forest order*
- *morphisms:* *homomorphisms of* σ*-structures that are forest morphisms and preserve the pebbling function.*

Theorem (Abramsky–Shah, 2021)

$\mathbf{EM}(\mathbb{M}_k)$ *is isomorphic to the category with*

- *objects:* *synchronization trees of depth at most k*
- *morphisms:* *homomorphisms of* σ*-structures that are also forest morphisms.*

Combinatorial parameters

We have seen that, for a structure A,

$$A \text{ admits a } G\text{-coalgebra} \quad (\text{of the form } A \to G(A))$$

corresponds to the fact that

$$A \text{ admits a 'nice' decomposition.}$$

Our examples:

Comonad	The corresponding decomposition
$\mathbb{E}_k$	compatible forest order of depth $\leq k$
$\mathbb{P}_k$	compatible k-pebble forest order
$\mathbb{M}_k$	synch. tree of depth $\leq k$

In fact, these notions are well-known in combinatorics.

Tree-depth [Nesetril–Ossona de Mendez, 2006]

A **forest cover** of a graph A is a forest $(T, \leq)$ and an injective function $f\colon A \to T$ such that

$$\text{if } (v, w) \in E^A, \text{ then either } f(v) \leq f(w) \text{ or } f(w) \leq f(v).$$

Write

$$\mathrm{td}(A) \leq k$$

if there exists a forest cover $(T, \leq)$ of A such that the size of $\downarrow x$ is at most k, for any $x \in T$.

Tree-depth [Nesetril–Ossona de Mendez, 2006]

A **forest cover** of a graph A is a forest $(T, \leq)$ and an injective function $f\colon A \to T$ such that

$$\text{if } (v, w) \in E^A, \text{ then either } f(v) \leq f(w) \text{ or } f(w) \leq f(v).$$

Write

$$\mathrm{td}(A) \leq k$$

if there exists a forest cover $(T, \leq)$ of A such that the size of $\downarrow x$ is at most k, for any $x \in T$.

Theorem (Abramsky–Shah, 2018 & 2021)

$\mathrm{td}(A) \leq k$ *iff* A *admits a* $\mathbb{E}_k$*-coalgebra*

Tree-width [Robertson–Seymour, 1986]

A **tree decomposition** of a graph A is a function $f\colon T \to \mathcal{P}(A)$, from a <u>tree</u> $(T, \leq)$ to subsets of A such that

- $\forall v \in A \quad \exists x \in T$ such that $v \in f(x)$,
- $\forall (u, v) \in E^A \quad \exists x \in T$ such that $\{u, v\} \subseteq f(x)$, and
- if $v \in f(x) \cap f(y)$, then $v \in f(z)$ for all z on the unique path between x and y in T.

Write

$$\mathrm{tw}(A) < k,$$

if there exists a tree decomposition $f\colon T \to \mathcal{P}(A)$ of such that $|f(x)| \leq k$ for every $x \in T$.

Tree-width [Robertson–Seymour, 1986]

A **tree decomposition** of a graph A is a function $f\colon T \to \mathcal{P}(A)$, from a <u>tree</u> $(T, \leq)$ to subsets of A such that

- $\forall v \in A \quad \exists x \in T$ such that $v \in f(x)$,
- $\forall (u, v) \in E^A \quad \exists x \in T$ such that $\{u, v\} \subseteq f(x)$, and
- if $v \in f(x) \cap f(y)$, then $v \in f(z)$ for all z on the unique path between x and y in T.

Write

$$\mathrm{tw}(A) < k,$$

if there exists a tree decomposition $f\colon T \to \mathcal{P}(A)$ of such that $|f(x)| \leq k$ for every $x \in T$.

Theorem (Abramsky–Dawar–Wang, 2017)

$\mathrm{tw}(A) < k$ *iff* A *admits a* $\mathbb{P}_k$*-coalgebra*

Applications

Applications in CSP and logic

Lemma

If $\mathrm{tw}(A) < k$ *and Duplicator has a winning strategy in the* k*-pebble forth-only game from* A *to* B *then there exists a homomorphism* $A \to B$.

Applications in CSP and logic

Lemma

If $\mathrm{tw}(A) < k$ *and Duplicator has a winning strategy in the* k*-pebble forth-only game from* A *to* B *then there exists a homomorphism* $A \to B$.

Proof.

1. tree-width $< k$ gives a coalgebra $A \to \mathbb{P}_k(A)$
2. a winning strategy gives $\mathbb{P}_k(A) \to B$
3. we compose $A \to \mathbb{P}_k(A) \to B$ □

Applications in CSP and logic

Lemma

If $\mathrm{tw}(A) < k$ *and Duplicator has a winning strategy in the* k*-pebble forth-only game from* A *to* B *then there exists a homomorphism* $A \to B$*.*

Proof.

1. tree-width $< k$ gives a coalgebra $A \to \mathbb{P}_k(A)$
2. a winning strategy gives $\mathbb{P}_k(A) \to B$
3. we compose $A \to \mathbb{P}_k(A) \to B$ □

Observation: Works for arbitrary comonads!

Applications in CSP and logic

Lemma

If $\mathrm{td}(A) \leq k$ and Duplicator has a winning strategy in the k-round Ehrenfeucht–Fraïssé forth-only game from A to B then there exists a homomorphism $A \to B$.

Lemma

For a synchronisation tree (A, a) of depth $\leq k$, if Duplicator has a winning strategy in the k-round simulation game from (A, a) to (B, b) then there exists a homomorphism $(A, a) \to (B, b)$.

Although, these are not so difficult to prove directly from the definitions.

Applications in combinatorics

There is a "comonad morphism" $\mathbb{E}_k \Rightarrow \mathbb{P}_k$, given by

$$\mathbb{E}_k(A) \xrightarrow{\lambda_A} \mathbb{P}_k(A)$$

$$[a_1, \ldots, a_n] \longmapsto [(1, a_1), (2, a_2), \ldots, (n, a_n)]$$

Lemma

For every σ-structure A, $\mathrm{tw}(A) + 1 \leq \mathrm{td}(A)$.

Applications in combinatorics

There is a "comonad morphism" $\mathbb{E}_k \Rightarrow \mathbb{P}_k$, given by

$$\mathbb{E}_k(A) \xrightarrow{\lambda_A} \mathbb{P}_k(A)$$

$$[a_1, \ldots, a_n] \longmapsto [(1, a_1), (2, a_2), \ldots, (n, a_n)]$$

Lemma

For every σ-structure A, $\mathrm{tw}(A) + 1 \leq \mathrm{td}(A)$.

Proof sketch.

Assume there is a coalgebra $A \xrightarrow{\alpha} \mathbb{E}_k(A)$.

Then, the composition

$$A \xrightarrow{\alpha} \mathbb{E}_k(A) \xrightarrow{\lambda_A} \mathbb{P}_k(A)$$

is a coalgebra too, by the axioms of comonad morphisms. □

Bonus slides

Revisiting the Chandra–Merlin correspondence

Recall the construction

$$\mathbf{M}\colon \mathrm{PP} \to \mathbf{Str}_{fin}(\sigma)$$

transforming φ in steps

1. variable renaming $\Rightarrow$ unique variable usage
2. prenex normal form $\Rightarrow$ $\exists x_1, \ldots, x_n \, (A_1 \wedge \cdots \wedge A_m)$
3. $\mathbf{M}(\varphi)$ on set $\{x_1, \ldots, x_n\}$ with relations as in $A_1, \ldots, A_m$

Revisiting the Chandra–Merlin correspondence

Recall the construction

$$\mathbf{M}\colon \mathrm{PP} \to \mathbf{Str}_{\mathit{fin}}(\sigma)$$

transforming φ in steps

1. variable renaming $\Rightarrow$ unique variable usage
2. prenex normal form $\Rightarrow$ $\exists x_1, \ldots, x_n \, (A_1 \wedge \cdots \wedge A_m)$
3. $\mathbf{M}(\varphi)$ on set $\{x_1, \ldots, x_n\}$ with relations as in $A_1, \ldots, A_m$

Theorem

- $\exists\ A \to \mathbb{E}_k(A) \quad \Longleftrightarrow \quad A \cong \mathbf{M}(\varphi)$ *for some* $\varphi \in \mathrm{PP}_k$
- $\exists\ A \to \mathbb{P}_k(A) \quad \Longleftrightarrow \quad A \cong \mathbf{M}(\varphi)$ *for some* $\varphi \in \mathrm{PP}^k$

Proof idea.

Quantifier nesting $\leftrightarrow$ tree order

Variable usage $\leftrightarrow$ pebbling function $\square$