An Introduction To Coalgebra

Thorsten Wißmann
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
`thorsten-wissmann.de`

Logic Mentoring Workshop @CSL23
Februrary 17, 2023, Warsaw, Poland

Motivation

Goals

- Convey the basic ideas of coalgebra
- Motivation for looking for abstract patterns
- Avoid redundancy in mathematical results

Instances of
F-Coalgebras

Transition Systems

Deterministic Automata

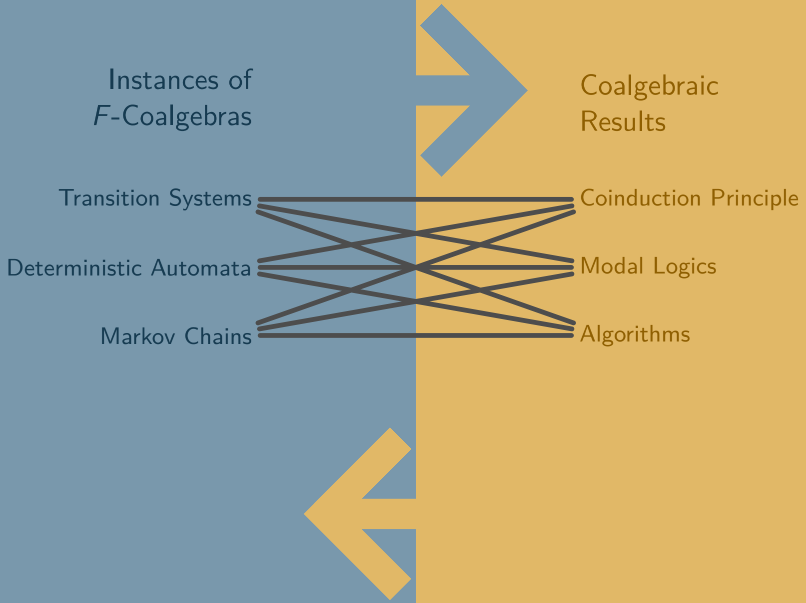
Markov Chains

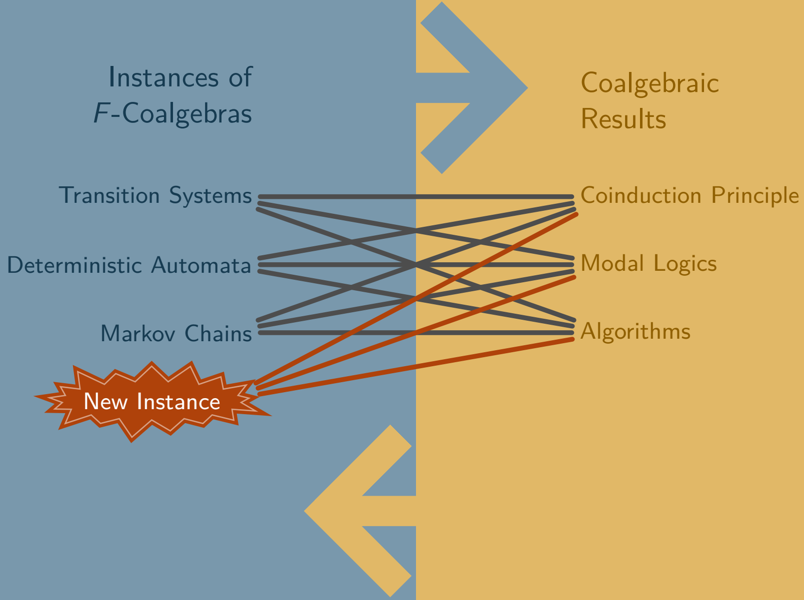
Coalgebraic
Results

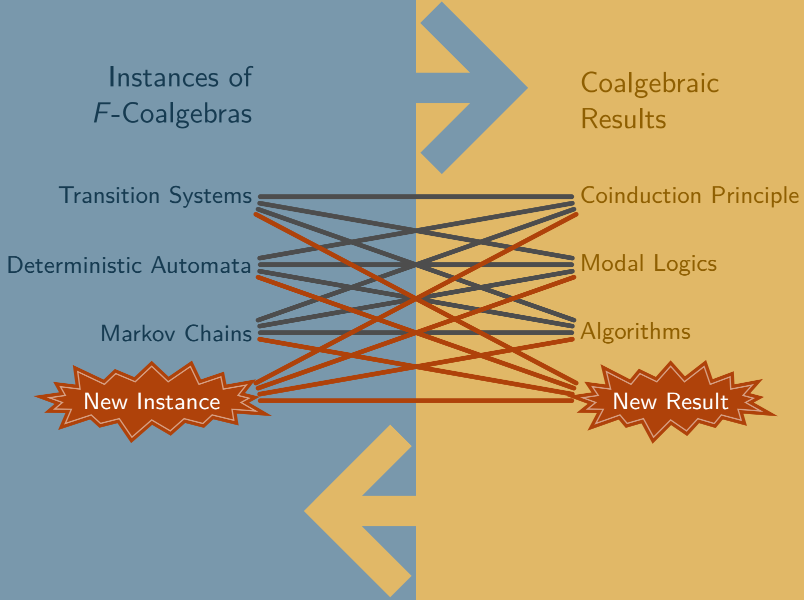
Coinduction Principle

Modal Logics

Algorithms


Instances of
F-Coalgebras
Coalgebraic
Results
Transition Systems
Deterministic Automata
Markov Chains
Coinduction Principle
Modal Logics
Algorithms


Instances of
F-Coalgebras
Coalgebraic
Results
Transition Systems
Deterministic Automata
Markov Chains
New Instance
Coinduction Principle
Modal Logics
Algorithms


Instances of
F-Coalgebras
Coalgebraic
Results
Transition Systems
Deterministic Automata
Markov Chains
New Instance
Coinduction Principle
Modal Logics
Algorithms
New Result

Definition: Functor $F\colon \mathsf{Set} \to \mathsf{Set}$

- Each set X is sent to a set FX
- Each map $f\colon X \to Y$ is sent to a map $Ff\colon FX \to FY$
  - Preserves identities: $F\mathrm{id}_X = \mathrm{id}_{FX}\colon FX \to FX$
  - Preserves composition: $F(g \circ f) = Fg \circ Ff$

    $F(Z \xleftarrow{g \circ f} X) = F(Z \xleftarrow{g} Y) \circ F(Y \xleftarrow{f} X)$

Examples

Definition: Functor $F\colon \mathsf{Set} \to \mathsf{Set}$

- Each set X is sent to a set FX
- Each map $f\colon X \to Y$ is sent to a map $Ff\colon FX \to FY$
  - Preserves identities: $F\mathrm{id}_X = \mathrm{id}_{FX}\colon FX \to FX$
  - Preserves composition: $F(g \circ f) = Fg \circ Ff$

    $F(Z \xleftarrow{g \circ f} X) = F(Z \xleftarrow{g} Y) \circ F(Y \xleftarrow{f} X)$

Examples

- $FX = \mathcal{P}X$, $\mathcal{P}f(S) := \{f(x) \mid x \in S\}$

Definition: Functor $F\colon \mathsf{Set} \to \mathsf{Set}$

- Each set X is sent to a set FX
- Each map $f\colon X \to Y$ is sent to a map $Ff\colon FX \to FY$
  - Preserves identities: $F\mathrm{id}_X = \mathrm{id}_{FX}\colon FX \to FX$
  - Preserves composition: $F(g \circ f) = Fg \circ Ff$

    $F(Z \xleftarrow{g\circ f} X) = F(Z \xleftarrow{g} Y) \circ F(Y \xleftarrow{f} X)$

Examples

- $FX = \mathcal{P}X$, $\mathcal{P}f(S) := \{f(x) \mid x \in S\}$
- $FX = A \times X$, $Ff(a, x) = (a, f(x))$

Definition: Functor $F\colon \mathsf{Set} \to \mathsf{Set}$

- Each set X is sent to a set FX
- Each map $f\colon X \to Y$ is sent to a map $Ff\colon FX \to FY$
  - Preserves identities: $F\mathrm{id}_X = \mathrm{id}_{FX}\colon FX \to FX$
  - Preserves composition: $F(g \circ f) = Fg \circ Ff$

    $F(Z \xleftarrow{g\circ f} X) = F(Z \xleftarrow{g} Y) \circ F(Y \xleftarrow{f} X)$

Examples

- $FX = \mathcal{P}X$, $\mathcal{P}f(S) := \{f(x) \mid x \in S\}$
- $FX = A \times X$, $Ff(a, x) = (a, f(x))$
- $FX = X^A$, $Ff(t) = f \circ t$, using $t\colon A \to X$

Definition: Functor $F\colon \mathsf{Set} \to \mathsf{Set}$

- Each set X is sent to a set FX
- Each map $f\colon X \to Y$ is sent to a map $Ff\colon FX \to FY$
  - Preserves identities: $F\mathrm{id}_X = \mathrm{id}_{FX}\colon FX \to FX$
  - Preserves composition: $F(g \circ f) = Fg \circ Ff$

    $F(Z \xleftarrow{g\circ f} X) = F(Z \xleftarrow{g} Y) \circ F(Y \xleftarrow{f} X)$

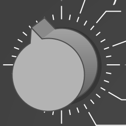
Examples

- $FX = \mathcal{P}X$, $\mathcal{P}f(S) := \{f(x) \mid x \in S\}$
- $FX = A \times X$, $Ff(a, x) = (a, f(x))$
- $FX = X^A$, $Ff(t) = f \circ t$, using $t\colon A \to X$
- $FX = \mathbb{R}^{(X)} := \{\mu\colon X \to \mathbb{R} \mid \mu(x) \neq 0 \text{ for finitely many } x \in X\}$
  $\mathbb{R}^{(f)}(\mu) = (y \mapsto \sum_{x\in X, f(x)=y} \mu(x)) \quad \in \mathbb{R}^{(Y)}, \quad \mu \in \mathbb{R}^{(X)}$

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$

$F\colon \mathsf{Set} \to \mathsf{Set}$



- $\mathcal{P}$
- $2 \times (-)^A$
- $\mathbb{R}^{(-)}$
- $F\colon \mathsf{Pos}$ ↺
- $F\colon \mathsf{Nom}$ ↺

F-Coalgebra

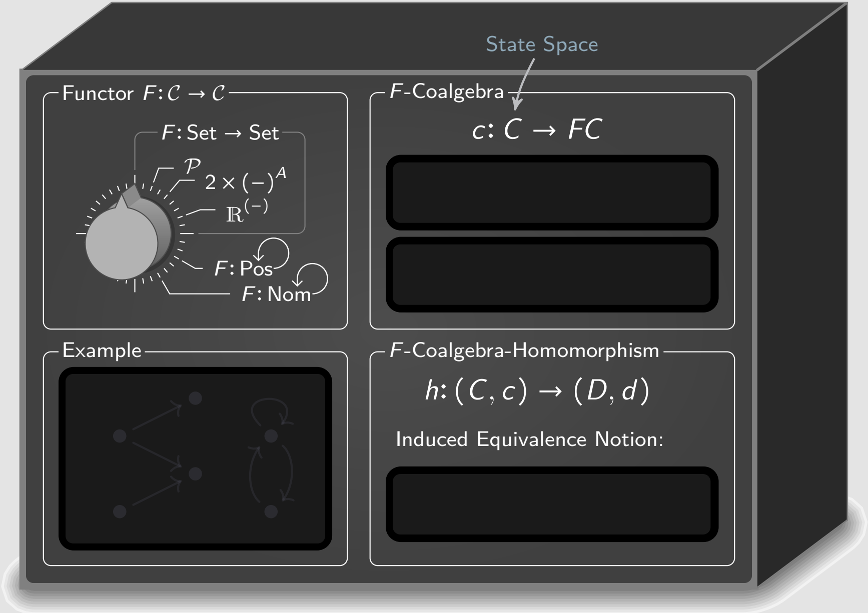
$$c\colon C \to FC$$

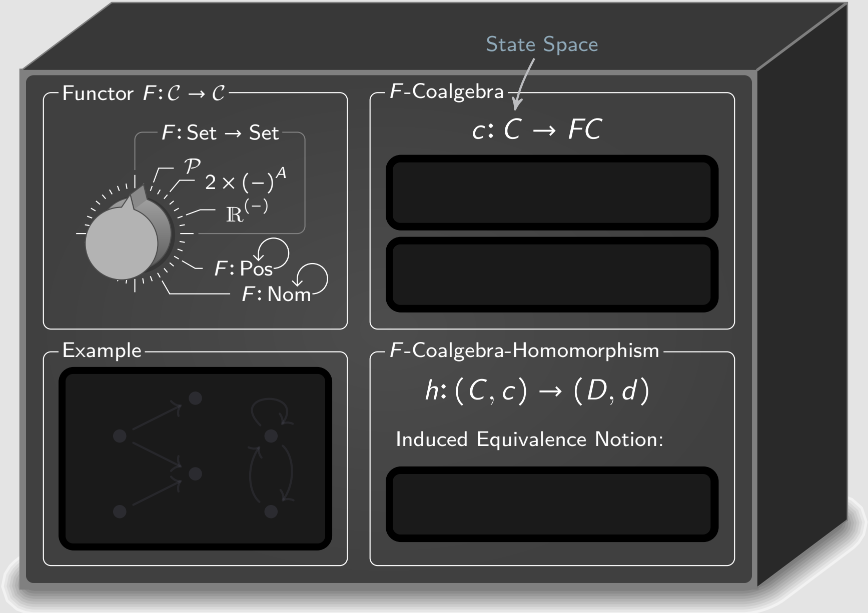
Example

F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

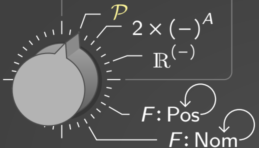

State Space
Functor $F\colon \mathcal{C} \to \mathcal{C}$
$F\colon \mathsf{Set} \to \mathsf{Set}$
$\mathcal{P}$
$2 \times (-)^A$
$\mathbb{R}^{(-)}$
$F\colon \mathsf{Pos}$
$F\colon \mathsf{Nom}$
F-Coalgebra
$c\colon C \to FC$
Example
F-Coalgebra-Homomorphism
$h\colon (C, c) \to (D, d)$
Induced Equivalence Notion:


State Space
Functor $F\colon \mathcal{C} \to \mathcal{C}$
$F\colon \mathrm{Set} \to \mathrm{Set}$
$\mathcal{P}$
$2 \times (-)^A$
$\mathbb{R}^{(-)}$
$F\colon \mathrm{Pos}$
$F\colon \mathrm{Nom}$
F-Coalgebra
$c\colon C \to FC$
Example
F-Coalgebra-Homomorphism
$h\colon (C, c) \to (D, d)$
Induced Equivalence Notion:

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$

$F\colon \mathsf{Set} \to \mathsf{Set}$



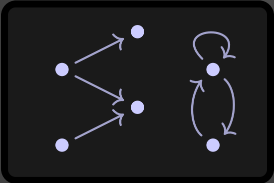
F-Coalgebra

$$c\colon C \to FC$$

$$c\colon C \to \mathcal{P}C$$

“Transition System”

Example



F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

“Bisimilarity”

Coalgebra Homomorphism $h\colon (C, c) \to (D, d)$

$$\begin{array}{ccc} C & \xrightarrow{\;c\;} & FC \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle Fh} \\ D & \xrightarrow{\;d\;} & FD \end{array}$$

Coalgebra Homomorphism $h\colon (C, c) \to (D, d)$

$$\begin{array}{ccc} C & \xrightarrow{\;c\;} & FC \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle Fh} \\ D & \xrightarrow{\;d\;} & FD \end{array}$$

$x, y \in C$ behaviourally equivalent

$\Longleftrightarrow$ exists $h\colon (C, c) \to (D, d)$ with $h(x) = h(y)$

Coalgebra Homomorphism
$h\colon (C, c) \to (D, d)$

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle Fh} \\ D & \xrightarrow{d} & FD \end{array}$$

$x, y \in C$ behaviourally equivalent

$\iff$ exists $h\colon (C, c) \to (D, d)$ with $h(x) = h(y)$

Example: $x, y \in C \xrightarrow{c} \mathcal{P}C$ behaviourally equiv. $\iff$ x,y bisimilar

Coalgebra Homomorphism $h\colon (C, c) \to (D, d)$

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle Fh} \\ D & \xrightarrow{d} & FD \end{array}$$

$x, y \in C$ behaviourally equivalent

$\iff$ exists $h\colon (C, c) \to (D, d)$ with $h(x) = h(y)$

Example: $x, y \in C \xrightarrow{c} \mathcal{P}C$ behaviourally equiv. $\iff$ x,y bisimilar

$\Rightarrow$ If $h(x) = h(y)$, then $R = \{(x', y') \in C \times C \mid h(x') = h(y')\}$ is a bisimulation.

Coalgebra Homomorphism $h\colon (C, c) \to (D, d)$

$$\begin{array}{ccc} C & \xrightarrow{\;c\;} & FC \\ {\scriptstyle h}\big\downarrow & & \big\downarrow{\scriptstyle Fh} \\ D & \xrightarrow{\;d\;} & FD \end{array}$$

$x, y \in C$ behaviourally equivalent

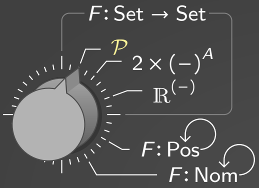
$\iff$ exists $h\colon (C, c) \to (D, d)$ with $h(x) = h(y)$

Example: $x, y \in C \xrightarrow{c} \mathcal{P}C$ behaviourally equiv. $\iff$ x,y bisimilar

- $\Rightarrow$ If $h(x) = h(y)$, then $R = \{(x', y') \in C \times C \mid h(x') = h(y')\}$ is a bisimulation.
- $\Leftarrow$ Define $D = C$ modulo bisimilarity and $h\colon C \to D$ sends each $x \in C$ to its equivalence class. Then, h is a $\mathcal{P}$-coalgebra homomorphism.

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$



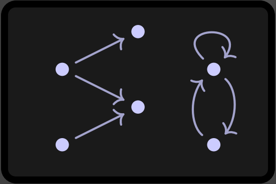
F-Coalgebra

$$c\colon C \to FC$$

$$c\colon C \to \mathcal{P}C$$

"Transition System"

Example

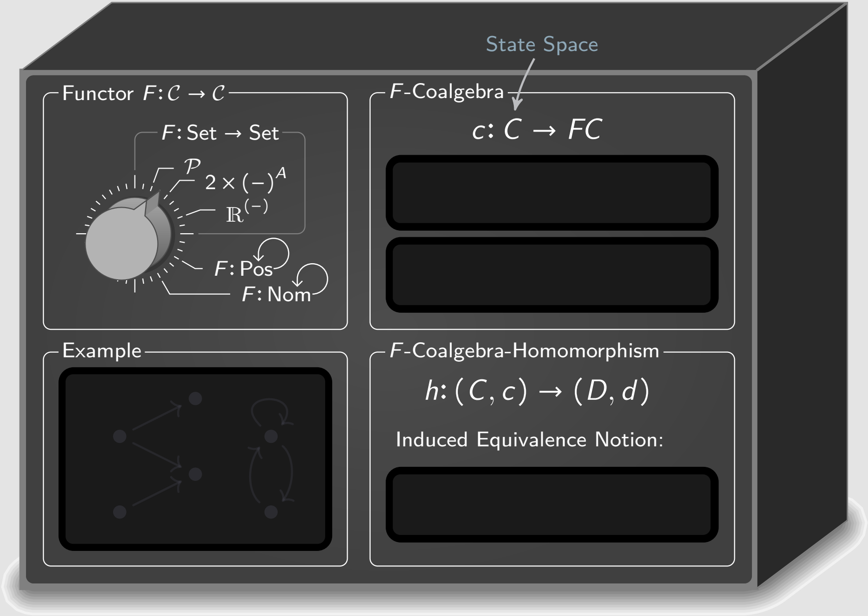


F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

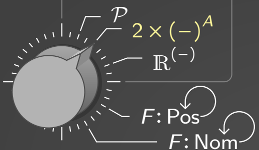
"Bisimilarity"


State Space
Functor $F\colon \mathcal{C} \to \mathcal{C}$
$F\colon \mathsf{Set} \to \mathsf{Set}$
$\mathcal{P}$
$2 \times (-)^A$
$\mathbb{R}^{(-)}$
$F\colon \mathsf{Pos}$
$F\colon \mathsf{Nom}$
F-Coalgebra
$c\colon C \to FC$
Example
F-Coalgebra-Homomorphism
$h\colon (C, c) \to (D, d)$
Induced Equivalence Notion:

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$

$F\colon \mathsf{Set} \to \mathsf{Set}$



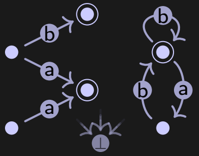
F-Coalgebra

$$c\colon C \to FC$$

$$c\colon C \to 2 \times C^A$$

"DFA"

Example



F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

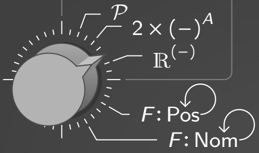
Induced Equivalence Notion:

"Same Language"

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$

$F\colon \mathsf{Set} \to \mathsf{Set}$



F-Coalgebra

$$c\colon C \to FC$$

Example

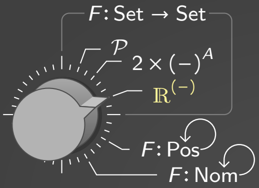
F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$



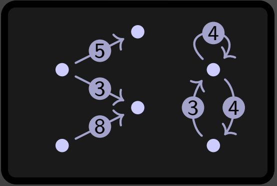
F-Coalgebra

$$c\colon C \to FC$$

$$c\colon C \to \mathbb{R}^{(C)}$$

"Markov Chain"

Example



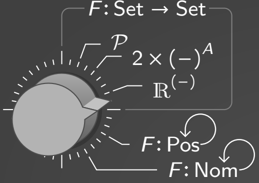
F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

"Weighted Bisimil."

Functor $F\colon \mathcal{C} \to \mathcal{C}$



F-Coalgebra

State Space

$$c\colon C \to FC$$

[redacted]

Example

[redacted]

F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

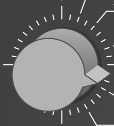
Induced Equivalence Notion:

[redacted]

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$

$F\colon \mathsf{Set} \to \mathsf{Set}$



- $\mathcal{P}$
- $2 \times (-)^A$
- $\mathbb{R}^{(-)}$
- $F\colon \mathsf{Pos}$
- $F\colon \mathsf{Nom}$

F-Coalgebra

$$c\colon C \to FC$$

Example

F-Coalgebra-Homomorphism

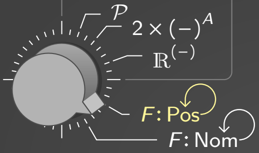
$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$

$F\colon \mathsf{Set} \to \mathsf{Set}$



F-Coalgebra

$c\colon C \to FC$

$c\colon C \to FX$

"monotone"

Example

F-Coalgebra-Homomorphism

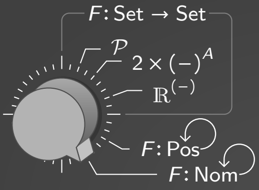
$h\colon (C, c) \to (D, d)$

Induced Equivalence Notion:

"Equiv. + Monotone"

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$



F-Coalgebra

$$c\colon C \to FC$$

Example

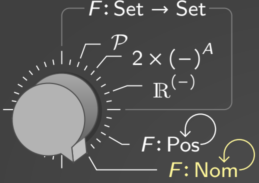
F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

State Space

Functor $F\colon \mathcal{C} \to \mathcal{C}$



F-Coalgebra

$$c\colon C \to FC$$

$$c\colon C \to FX$$

"Equivariant"

Example

F-Coalgebra-Homomorphism

$$h\colon (C, c) \to (D, d)$$

Induced Equivalence Notion:

"Equiv. modulo $\equiv_\alpha$"

Coinduction

Ideas

- Canonical domain of semantics of F-coalgebras
- Use coalgebras to define (possibly infinite) behaviour

The final F-coalgebra (T, t) is an F-coalgebra such that...

$$\begin{array}{rcccl} \text{for all coalgebras} & C & \xrightarrow{c} & FC & \\ \text{there exists a unique } h & \downarrow & & \downarrow & Fh \\ & T & \xrightarrow{t} & FT & \end{array}$$

The final F-coalgebra (T, t) is an F-coalgebra such that...

$$\begin{array}{rccc} \text{for all coalgebras} & C & \xrightarrow{\;c\;} & FC \\ \text{there exists a unique } h & \big\downarrow & & \big\downarrow Fh \\ & T & \xrightarrow{\;t\;} & FT \end{array}$$

Final F-coalgebra $\Longleftrightarrow$ greatest fixpoint νF

$x, y \in C$ behaviourally equivalent $\Longleftrightarrow$ x, y identified in νF

The final F-coalgebra (T, t) is an F-coalgebra such that...

$$\begin{array}{ccc} \text{for all coalgebras} \quad C & \xrightarrow{\;c\;} & FC \\ \text{there exists a unique } h \downarrow & & \downarrow Fh \\ T & \xrightarrow{\;t\;} & FT \end{array}$$

Final F-coalgebra $\Leftrightarrow$ greatest fixpoint νF

$x, y \in C$ behaviourally equivalent $\Leftrightarrow$ x, y identified in νF

Examples

- $FX = A \times X \Rightarrow \nu F = A^{\mathbb{N}}$

The final F-coalgebra (T, t) is an F-coalgebra such that...

for all coalgebras $C \xrightarrow{c} FC$

there exists a unique h: $h: C \to T$, $Fh: FC \to FT$, $T \xrightarrow{t} FT$

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ \downarrow h & & \downarrow Fh \\ T & \xrightarrow{t} & FT \end{array}$$

Final F-coalgebra $\iff$ greatest fixpoint νF

$x, y \in C$ behaviourally equivalent $\iff$ x, y identified in νF

Examples

- $FX = A \times X \Rightarrow \nu F = A^{\mathbb{N}}$

For $FX = \mathbb{N} \times X$

$C := \mathbb{N} \times \mathbb{N}$

$c\colon \mathbb{N} \times \mathbb{N} \to F(\mathbb{N} \times \mathbb{N})$
$c(k, n) = (k, (n, k + n))$

The final F-coalgebra (T, t) is an F-coalgebra such that...

for all coalgebras $C \xrightarrow{c} FC$

there exists a unique h

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle Fh} \\ T & \xrightarrow{t} & FT \end{array}$$

Final F-coalgebra $\iff$ greatest fixpoint νF

$x, y \in C$ behaviourally equivalent $\iff$ x, y identified in νF

Examples

- $FX = A \times X \Rightarrow \nu F = A^{\mathbb{N}}$

For $FX = \mathbb{N} \times X$

$C := \mathbb{N} \times \mathbb{N}$

$c \colon \mathbb{N} \times \mathbb{N} \to F(\mathbb{N} \times \mathbb{N})$

$c(k, n) = (k, (n, k + n))$

What is $h(1, 1) \in \nu F = \mathbb{N}^{\mathbb{N}}$*?*

The final F-coalgebra (T, t) is an F-coalgebra such that...

for all coalgebras

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ {\scriptstyle\text{there exists a unique } h}\downarrow & & \downarrow{\scriptstyle Fh} \\ T & \xrightarrow{t} & FT \end{array}$$

Final F-coalgebra $\iff$ greatest fixpoint νF

$x, y \in C$ behaviourally equivalent $\iff$ x, y identified in νF

Examples

- $FX = A \times X \Rightarrow \nu F = A^{\mathbb{N}}$
- $FX = X^A \Rightarrow \nu F = 1$

For $FX = \mathbb{N} \times X$

$C := \mathbb{N} \times \mathbb{N}$

$c \colon \mathbb{N} \times \mathbb{N} \to F(\mathbb{N} \times \mathbb{N})$
$c(k, n) = (k, (n, k + n))$

What is $h(1, 1) \in \nu F = \mathbb{N}^{\mathbb{N}}$*?*

The final F-coalgebra (T, t) is an F-coalgebra such that...

$$\begin{array}{rccc} \text{for all coalgebras} & C & \xrightarrow{c} & FC \\ \text{there exists a unique } h & \downarrow & & \downarrow Fh \\ & T & \xrightarrow{t} & FT \end{array}$$

Final F-coalgebra $\Longleftrightarrow$ greatest fixpoint νF

$x, y \in C$ behaviourally equivalent $\Longleftrightarrow$ x, y identified in νF

Examples

- $FX = A \times X \Rightarrow \nu F = A^{\mathbb{N}}$
- $FX = X^A \Rightarrow \nu F = 1$
- $FX = 2 \times X^A \Rightarrow \nu F = 2^{A^*}$

For $FX = \mathbb{N} \times X$

$C := \mathbb{N} \times \mathbb{N}$

$c \colon \mathbb{N} \times \mathbb{N} \to F(\mathbb{N} \times \mathbb{N})$
$c(k, n) = (k, (n, k + n))$

What is $h(1, 1) \in \nu F = \mathbb{N}^{\mathbb{N}}$*?*

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof

$$\begin{array}{ccc} T & \xrightarrow{t} & FT \\ {\scriptstyle t}\downarrow & & \downarrow{\scriptstyle Ft} \\ FT & \xrightarrow{Ft} & FFT \end{array}$$

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

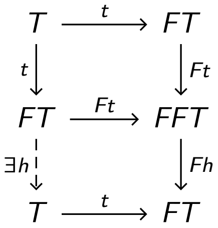
Proof

$$
\begin{array}{ccc}
T & \xrightarrow{\quad t \quad} & FT \\
{\scriptstyle t}\downarrow & & \downarrow{\scriptstyle Ft} \\
FT & \xrightarrow{\quad Ft \quad} & FFT \\
{\scriptstyle \exists h}\downarrow & & \downarrow{\scriptstyle Fh} \\
T & \xrightarrow{\quad t \quad} & FT
\end{array}
$$

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof

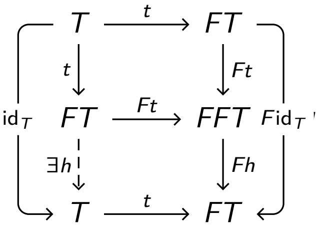


- $h \circ t \colon (T, t) \to (T, t)$ is an F-coalgebra homomorphism

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof

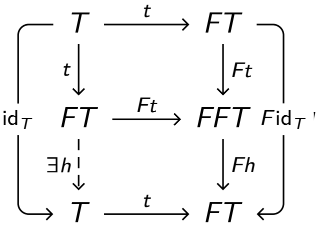


- $h \circ t \colon (T, t) \to (T, t)$ is an F-coalgebra homomorphism
- $\mathrm{id}_T \colon (T, t) \to (T, t)$, too

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof



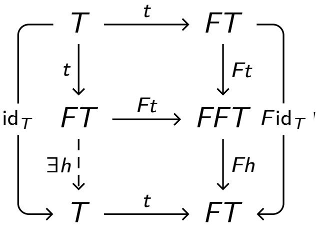
- $h \circ t \colon (T, t) \to (T, t)$ is an F-coalgebra homomorphism
- $\mathrm{id}_T \colon (T, t) \to (T, t)$, too

$\Longrightarrow$ $h \circ t = \mathrm{id}_T$ by uniqueness

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof



- $h \circ t\colon (T, t) \to (T, t)$ is an F-coalgebra homomorphism
- $\mathrm{id}_T\colon (T, t) \to (T, t)$, too

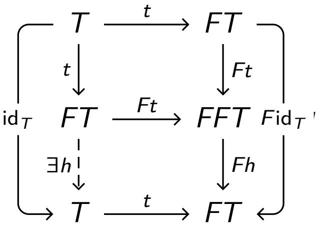
$\Longrightarrow$ $h \circ t = \mathrm{id}_T$ by uniqueness

- $t \circ h = Fh \circ Ft = F(h \circ t)$
  $= F\mathrm{id}_T = \mathrm{id}_{FT}$ □

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof



- $h \circ t \colon (T, t) \to (T, t)$ is an F-coalgebra homomorphism
- $\mathrm{id}_T \colon (T, t) \to (T, t)$, too

$\Longrightarrow$ $h \circ t = \mathrm{id}_T$ by uniqueness

- $t \circ h = Fh \circ Ft = F(h \circ t)$
  $= F\mathrm{id}_T = \mathrm{id}_{FT}$ □

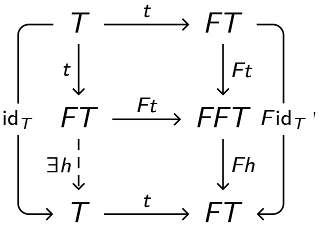
Cantor's Theorem

Any map $C \to \mathcal{P}C$ is not surjective

Lambek's Lemma

If the coalgebra (T, t) is final, then t is an isomorphism.

Proof



- $h \circ t \colon (T, t) \to (T, t)$ is an F-coalgebra homomorphism
- $\mathrm{id}_T \colon (T, t) \to (T, t)$, too

$\Longrightarrow$ $h \circ t = \mathrm{id}_T$ by uniqueness

- $t \circ h = Fh \circ Ft = F(h \circ t)$
  $= F\mathrm{id}_T = \mathrm{id}_{FT}$ □

Cantor's Theorem

Any map $C \to \mathcal{P}C$ is not surjective

$\Longrightarrow$

There is no final $\mathcal{P}$-coalgebra

Coalgebraic Modal Logic

Ideas

- Define a set of modalities $\heartsuit_1, \heartsuit_2, \ldots$ "suitable" for F
- We build formulas using these modalities...
- and F-coalgebras are the models
- Formulas ϕ hold at a subset of the states $[\![\phi]\!] \subseteq C$ of a coalgebra $c\colon C \to FC$

Syntax

$\phi, \psi ::= \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \heartsuit\phi$

Syntax

$\phi, \psi ::= \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \heartsuit\phi$

Semantics $[\![\phi]\!] \subseteq C$ in a coalgebra $c\colon C \to FC$

$[\![\top]\!] = C,\qquad [\![\bot]\!] = \emptyset,\qquad [\![\neg\phi]\!] = C \setminus [\![\phi]\!],$

$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!],\qquad [\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!],$

Syntax

$\phi, \psi ::= \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \heartsuit\phi$

Semantics $[\![\phi]\!] \subseteq C$ in a coalgebra $c \colon C \to FC$

$[\![\top]\!] = C,$ $\quad [\![\bot]\!] = \emptyset,$ $\quad [\![\neg\phi]\!] = C \setminus [\![\phi]\!],$

$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!],$ $\quad [\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!],$

(Unary) Predicate Lifting $[\![\heartsuit]\!]$

$$[\![\heartsuit]\!] \subseteq F2 \iff 2^C \to 2^{FC} \text{ natural in } C$$

Syntax

$\phi, \psi ::= \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \heartsuit\phi$

Semantics $[\![\phi]\!] \subseteq C$ in a coalgebra $c\colon C \to FC$

$[\![\top]\!] = C, \qquad [\![\bot]\!] = \emptyset, \qquad [\![\neg\phi]\!] = C \setminus [\![\phi]\!],$

$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!], \qquad [\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!],$

$[\![\heartsuit\phi]\!] = \{x \in C \mid F\chi_{[\![\phi]\!]}(c(x)) \in [\![\heartsuit]\!]\} \qquad \chi_{[\![\phi]\!]}\colon C \to 2$

(Unary) Predicate Lifting $[\![\heartsuit]\!]$

$$[\![\heartsuit]\!] \subseteq F2 \quad \Longleftrightarrow \quad 2^C \to 2^{FC} \text{ natural in } C$$

Syntax

$\phi, \psi ::= \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \heartsuit\phi$

Semantics $[\![\phi]\!] \subseteq C$ in a coalgebra $c\colon C \to FC$

$[\![\top]\!] = C, \qquad [\![\bot]\!] = \emptyset, \qquad [\![\neg\phi]\!] = C \setminus [\![\phi]\!],$

$[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!], \qquad [\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!],$

$[\![\heartsuit\phi]\!] = \{x \in C \mid F\chi_{[\![\phi]\!]}(c(x)) \in [\![\heartsuit]\!]\} \qquad \chi_{[\![\phi]\!]}\colon C \to 2$

(Unary) Predicate Lifting $[\![\heartsuit]\!]$

$$[\![\heartsuit]\!] \subseteq F2 \iff 2^C \to 2^{FC} \text{ natural in } C$$

Examples

- $FX = \mathcal{P}X$: $[\![\Diamond]\!] = \{s \in \mathcal{P}2 \mid 1 \in s\}$ $[\![\Box]\!] = \{s \in \mathcal{P}2 \mid 0 \notin s\}$
- $FX = \mathbb{R}^{(X)}$: $[\![\langle \geq r \rangle]\!] = \{\mu \in \mathbb{R}^{(2)} \mid \mu(1) \geq r\}$

Theorem: Adequacy

If x, y are behaviourally equivalent,
then $x \in [\![\phi]\!] \leftrightarrow y \in [\![\phi]\!]$ for all formulas ϕ

Theorem: Adequacy

If x, y are behaviourally equivalent,
then $x \in [\![\phi]\!] \leftrightarrow y \in [\![\phi]\!]$ for all formulas ϕ

For converse direction: separating set of predicate liftings, e.g. $\Box$

Theorem: Adequacy

If x, y are behaviourally equivalent,
then $x \in [\![\phi]\!] \leftrightarrow y \in [\![\phi]\!]$ for all formulas ϕ

For converse direction: separating set of predicate liftings, e.g. $\Box$

Implementations & Extensions

- Is a given formula ϕ satisfiable? (or provable from)
- Fixpoint Logics (mu-calculus, including CTL)
- Semantics for infinite games (Büchi-Automata)
- Combine functors to form new logics (e.g. combining standard $\Box/\Diamond$ with probabilities)

Software COOL: `https://git8.cs.fau.de/software/cool`

Minimization Algorithms

Task

- Given a finite $c\colon C \to FC$ ...
- ... which states have equivalent behaviour?

Similar Ideas

Similar Run-Time

System type hard-wired

Similar Ideas

Similar Run-Time

System type hard-wired

Deterministic finite Automata

$n \cdot \log n$ — Hopcroft '71

$|A| \cdot n \cdot \log n$ — Gries '73, Knuutila '01

Similar Ideas

Similar Run-Time

System type hard-wired

Deterministic finite Automata

$n \cdot \log n$ — Hopcroft '71

$|A| \cdot n \cdot \log n$ — Gries '73, Knuutila '01

(Labelled) Transition Systems

$m \cdot \log n$ — Paige, Tarjan '87, Valmari '09

Similar Ideas

Similar Run-Time

System type hard-wired

Deterministic finite Automata

$n \cdot \log n$ Hopcroft '71

$|A| \cdot n \cdot \log n$ Gries '73, Knuutila '01

(Labelled) Transition Systems

$m \cdot \log n$

Paige, Tarjan '87

Valmari '09

Weighted Systems "Markov Chain Lumping"

$m \cdot \log n$

Valmari, Franceschinis '10

Similar Ideas

Similar Run-Time

System type hard-wired

Deterministic finite Automata

$n \cdot \log n$ — Hopcroft '71

$|A| \cdot n \cdot \log n$ — Gries '73, Knuutila '01

(Labelled) Transition Systems

$m \cdot \log n$

Paige, Tarjan '87

Valmari '09

Segala Systems

$m_{\text{dist}} \cdot \log m_{\text{acts}}$

Groote, Verduzco, de Vink '18

Weighted Systems "Markov Chain Lumping"

$m \cdot \log n$

Valmari, Franceschinis '10

Similar Ideas

Similar Run-Time

System type hard-wired

Deterministic finite Automata

$n \cdot \log n$ — Hopcroft '71

$|A| \cdot n \cdot \log n$ — Gries '73, Knuutila '01

(Labelled) Transition Systems

$m \cdot \log n$

Paige, Tarjan '87

Valmari '09

Segala Systems

$m_{\text{dist}} \cdot \log m_{\text{acts}}$

Groote, Verduzco, de Vink '18

Weighted Systems
"Markov Chain Lumping"

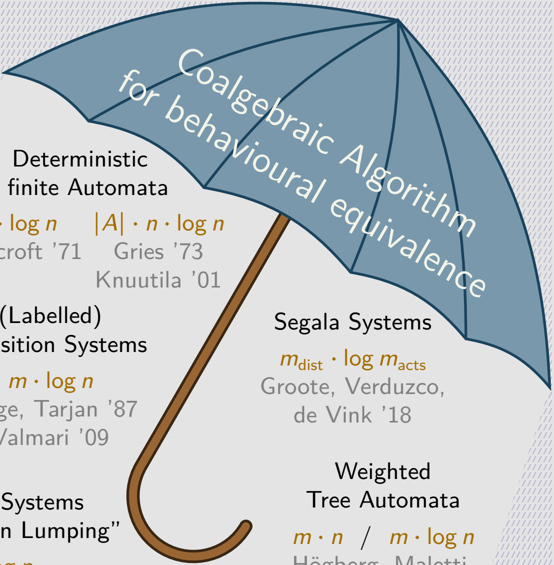
$m \cdot \log n$

Valmari, Franceschinis '10

Weighted Tree Automata

$m \cdot n$ / $m \cdot \log n$

Högberg, Maletti, May '07



Deterministic finite Automata

$n \cdot \log n$ — Hopcroft '71

$|A| \cdot n \cdot \log n$ — Gries '73, Knuutila '01

(Labelled) Transition Systems

$m \cdot \log n$

Paige, Tarjan '87
Valmari '09

Weighted Systems "Markov Chain Lumping"

$m \cdot \log n$

Valmari, Franceschinis '10

Segala Systems

$m_{\text{dist}} \cdot \log m_{\text{acts}}$

Groote, Verduzco, de Vink '18

Weighted Tree Automata

$m \cdot n$ / $m \cdot \log n$

Högberg, Maletti, May '07

Final Chain: sequence of partitions / equivalence relations

x, y behaviourally equivalent $\iff \forall k \in \mathbb{N}\colon p_k(x) = p_k(y)$

- $p_0\colon C \to 1$ (all states equivalent)

Final Chain: sequence of partitions / equivalence relations

x, y behaviourally equivalent $\iff \forall\, k \in \mathbb{N}\colon p_k(x) = p_k(y)$

- $p_0\colon C \to 1$ (all states equivalent)
- $p_1\colon C \xrightarrow{c} FC \xrightarrow{Fp_0} F1$ (final vs. non-final)

Final Chain: sequence of partitions / equivalence relations

x, y behaviourally equivalent $\iff \forall k \in \mathbb{N}\colon p_k(x) = p_k(y)$

- $p_0\colon C \to 1$ (all states equivalent)
- $p_1\colon C \xrightarrow{c} FC \xrightarrow{Fp_0} F1$ (final vs. non-final)
- $p_2\colon C \xrightarrow{c} FC \xrightarrow{Fp_1} FF1$

Definition: kernel pair of $f\colon X \to Y$

$\ker(f) := \{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$

Final Chain: sequence of partitions / equivalence relations

x, y behaviourally equivalent $\iff \forall k \in \mathbb{N}\colon p_k(x) = p_k(y)$

- $p_0\colon C \to 1$ (all states equivalent)
- $p_1\colon C \xrightarrow{c} FC \xrightarrow{Fp_0} F1$ (final vs. non-final)
- $p_2\colon C \xrightarrow{c} FC \xrightarrow{Fp_1} FF1$

$\vdots$

Definition: kernel pair of $f\colon X \to Y$

$\ker(f) := \{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$

Final Chain: sequence of partitions / equivalence relations

x, y behaviourally equivalent $\iff \forall k \in \mathbb{N}\colon p_k(x) = p_k(y)$

- $p_0\colon C \to 1$ (all states equivalent)
- $p_1\colon C \xrightarrow{c} FC \xrightarrow{Fp_0} F1$ (final vs. non-final)
- $p_2\colon C \xrightarrow{c} FC \xrightarrow{Fp_1} FF1$

$\vdots$

- $p_{k+1}\colon C \xrightarrow{c} FC \xrightarrow{p_k} FF^k 1 = F^{k+1} 1$ for $p_k\colon C \to F^k 1$

Definition: kernel pair of $f\colon X \to Y$

$\ker(f) := \{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$

Final Chain: sequence of partitions / equivalence relations

x, y behaviourally equivalent $\iff \forall k \in \mathbb{N}\colon p_k(x) = p_k(y)$

- $p_0\colon C \to 1$ (all states equivalent)
- $p_1\colon C \xrightarrow{c} FC \xrightarrow{Fp_0} F1$ (final vs. non-final)
- $p_2\colon C \xrightarrow{c} FC \xrightarrow{Fp_1} FF1$

$\vdots$

- $p_{k+1}\colon C \xrightarrow{c} FC \xrightarrow{p_k} FF^k 1 = F^{k+1} 1$ for $p_k\colon C \to F^k 1$

Definition: kernel pair of $f\colon X \to Y$

$\ker(f) := \{(x_1, x_2) \in X \times X \mid f(x_1) = f(x_2)\}$

Theorem

As soon as p_k and p_{k+1} describe the same relation ($\ker p_k = \ker p_{k+1}$), then p_k describes behavioural equivalence.

With stronger assumptions:

Efficient Coalgebraic Partition Refinement

- Assume that $F\colon \mathsf{Set} \to \mathsf{Set}$ is "zippable"
- Assume that F has a "refinement interface"

$\implies$ An F-coalgebra with n states and m transitions is minimized in $\mathcal{O}((m+n)\cdot\log n)$

Software COPAR: `https://git8.cs.fau.de/software/copar`

System Type	Functor FX	Run time $(m \geq n)$		Dedicated Algorithm	
Transition Systems	$\mathcal{P}_f X$	$m \cdot \log n$	=	$m \cdot \log n$	Paige, Tarjan 1987
Markov Chains	$\mathbb{R}^{(X)}$	$m \cdot \log n$	=	$m \cdot \log n$	Valmari, Franceschinis 2010
Determi-nistic Automata	$2 \times X^A$ $(A$ fixed$)$	$n \cdot \log n$	=	$n \cdot \log n$	Hopcroft 1971
Colour Refinement	$\mathcal{B}X = \mathbb{N}^{(X)}$	$m \cdot \log n$	=	$m \cdot \log n$	Berkholz, Bonsma, Grohe 2017

System Type	Functor FX	Run time $(m \geq n)$		Dedicated Algorithm	
Transition Systems	$\mathcal{P}_f X$	$m \cdot \log n$	$=$	$m \cdot \log n$	Paige, Tarjan 1987
LTS	$\mathbb{P}_f(\mathbb{N} \times X)$	$m \cdot \log m$	$=$	$m \cdot \log m$	Dovier, Piazza, Policriti 2004
			$>$	$m \cdot \log n$	Valmari 2009
Markov Chains	$\mathbb{R}^{(X)}$	$m \cdot \log n$	$=$	$m \cdot \log n$	Valmari, Franceschinis 2010
Deterministic Automata	$2 \times X^A$ (A fixed)	$n \cdot \log n$	$=$	$n \cdot \log n$	Hopcroft 1971
	$2 \times \mathcal{P}_f(A \times X)$	$\lvert A \rvert \cdot n \cdot \log n$	$=$	$\lvert A \rvert \cdot n \cdot \log n$	Gries 1973/Knuutila 2001
Segala Systems	$\mathcal{P}_f(A \times \mathcal{D}X)$	$m_{\mathcal{D}} \cdot \log m_{\mathcal{P}_f}$	$<$	$m \cdot \log n$	Baier, Engelen, Majster-Cederbaum 2000
			$=$	$m_{\mathcal{D}} \cdot \log m_{\mathcal{P}_f}$	Groote, Verduzco, de Vink 2018
Colour Refinement	$\mathcal{B}X = \mathbb{N}^{(X)}$ New	$m \cdot \log n$	$=$	$m \cdot \log n$	Berkholz, Bonsma, Grohe 2017
Weighted Tree-Automata (Backwards Bisimulation)	$M^{(\Sigma X)}$ M uncancellable	$m \cdot \log^2 m$	$<$	$m \cdot n$	Högberg, Maletti, May 2007
	$M^{(\Sigma X)}$ M cancellable	$m \cdot \log m$	$=$ Σ fest	$m \cdot \log n$	Högberg, Maletti, May 2007

Conclusions

- Coalgebra: reasoning about coinductive data
- Generic methods allow to distinguish *general pattern* and *type-specifics*
- Genericity and speed don't exclude each other!

Thank You!

[BBG17] Christoph Berkholz, Paul S. Bonsma, Martin Grohe. "Tight Lower and Upper Bounds for the Complexity of Canonical Colour Refinement". In: *Theory Comput. Syst.* 60.4 (2017), pp. 581–614. DOI: 10.1007/s00224-016-9686-0. URL: https://doi.org/10.1007/s00224-016-9686-0.

[BEM00] Christel Baier, Bettina Engelen, Mila Majster-Cederbaum. "Deciding Bisimilarity and Similarity for Probabilistic Processes". In: *J. Comput. Syst. Sci.* 60 (2000), pp. 187–231.

[DPP04] Agostino Dovier, Carla Piazza, Alberto Policriti. "An efficient algorithm for computing bisimulation equivalence". In: *Theor. Comput. Sci.* 311.1-3 (2004), pp. 221–256.

[Gri73] David Gries. "Describing an algorithm by Hopcroft". In: *Acta Informatica* 2 (1973), pp. 97–109. ISSN: 1432-0525.

[GVdV18] Jan Friso Groote, Jao Rivera Verduzco, Erik P. de Vink. "An Efficient Algorithm to Determine Probabilistic Bisimulation". In: *Algorithms* 11.9 (2018), p. 131. DOI: 10.3390/a11090131. URL: https://doi.org/10.3390/a11090131.

[HMM07] Johanna Högberg, Andreas Maletti, Jonathan May. "Bisimulation Minimisation for Weighted Tree Automata". In: *Developments in Language Theory, DLT 2007*. Vol. 4588. LNCS. Springer, 2007, pp. 229–241. ISBN: 978-3-540-73207-5. DOI: 10.1007/978-3-540-73208-2. URL: https://doi.org/10.1007/978-3-540-73208-2.

[Hop71] John Hopcroft. "An $n \log n$ algorithm for minimizing states in a finite automaton". In: *Theory of Machines and Computations*. Academic Press, 1971, pp. 189–196.

[Knu01] Timo Knuutila. "Re-describing an algorithm by Hopcroft". In: *Theor. Comput. Sci.* 250 (2001), pp. 333–363. ISSN: 0304-3975.

[PT87] Robert Paige, Robert E. Tarjan. "Three partition refinement algorithms". In: *SIAM J. Comput.* 16.6 (1987), pp. 973–989.

[Val09] Antti Valmari. "Bisimilarity Minimization in $\mathcal{O}(m \log n)$ Time". In: *Applications and Theory of Petri Nets, PETRI NETS 2009*. Vol. 5606. LNCS. Springer, 2009, pp. 123–142. ISBN: 978-3-642-02423-8.

[VF10] Antti Valmari, Giuliana Franceschinis. "Simple $\mathcal{O}(m \log n)$ Time Markov Chain Lumping". In: *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2010*. Vol. 6015. LNCS. Springer, 2010, pp. 38–52.