Efficient Coalgebraic Partition Refinement

Stefan Milius

Joint work with:

Ulrich Dorsch, Lutz Schröder, Thorsten Wißmann

Friedrich-Alexander-Universität Erlangen-Nürnberg

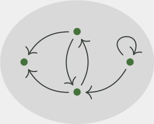
IFIP WG 1.3 Meeting
July 5, 2018

Efficient Coalgebraic Partition Refinement

Efficient Coalgebraic Partition Refinement

1. Coalgebras:

State based
systems

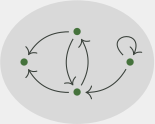


Labels, Non-Determinism,
Probabilities, Automata,
... and their combinations!

Efficient Coalgebraic Partition Refinement

1. Coalgebras:

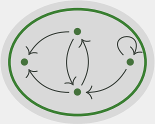
State based systems



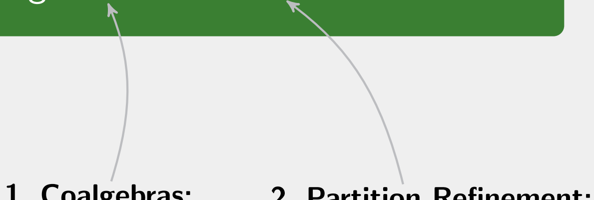
Labels, Non-Determinism,
Probabilities, Automata,
... and their combinations!

2. Partition Refinement:

Successively distinguish different behaviour

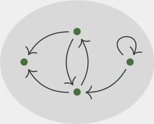


Efficient Coalgebraic Partition Refinement



1. Coalgebras:

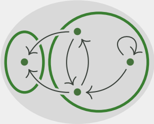
State based systems



Labels, Non-Determinism,
Probabilities, Automata,
… and their combinations!

2. Partition Refinement:

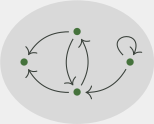
Successively distinguish different behaviour



Efficient Coalgebraic Partition Refinement

1. Coalgebras:

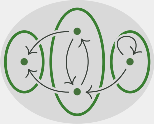
State based systems



Labels, Non-Determinism,
Probabilities, Automata,
... and their combinations!

2. Partition Refinement:

Successively distinguish different behaviour



Efficient Coalgebraic Partition Refinement

3. Efficiency:

(a) Incrementally compute partitions

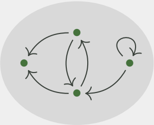
(b) Complexity Analysis

$$\mathcal{O}(m \cdot \log n)$$

Edges States

1. Coalgebras:

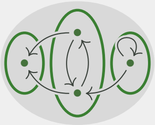
State based systems



Labels, Non-Determinism,
Probabilities, Automata,
... and their combinations!

2. Partition Refinement:

Successively distinguish different behaviour



Share Common Structure & Ideas

Similar Run-Time

Variations in Details

Share Common Structure & Ideas

Similar Run-Time

Variations in Details

Deterministic Finite Automata

$n \cdot \log n$	$\lvert A \rvert \cdot n \cdot \log n$
Hopcroft '71	Gries '73 Knuutila '01

Share Common Structure & Ideas

Deterministic Finite Automata

$n \cdot \log n$ Hopcroft '71

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

Transition Systems

$m \cdot \log n$ Paige, Tarjan '87

Similar Run-Time

Variations in Details

Share Common Structure & Ideas

Similar Run-Time

Variations in Details

Deterministic Finite Automata

$n \cdot \log n$ Hopcroft '71

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

Transition Systems

$m \cdot \log n$

Paige, Tarjan '87

Segala Systems

$m \cdot n \cdot (\log m + \log n)$

Baier, Engelen, Majster-Cederbaum '00

Share Common Structure & Ideas

Similar Run-Time

Variations in Details

Deterministic Finite Automata

$n \cdot \log n$ Hopcroft '71

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

Transition Systems

$m \cdot \log n$ Paige, Tarjan '87

Segala Systems

$m \cdot n \cdot (\log m + \log n)$ Baier, Engelen, Majster-Cederbaum '00

Labelled Transition Systems

$m \cdot \log n$ Valmari '09

Share Common Structure & Ideas

Similar Run-Time

Variations in Details

Deterministic Finite Automata

$n \cdot \log n$ Hopcroft '71

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

Transition Systems

$m \cdot \log n$

Paige, Tarjan '87

Segala Systems

$m \cdot n \cdot (\log m + \log n)$

Baier, Engelen, Majster-Cederbaum '00

Weighted Systems
"Markov Chain Lumping"

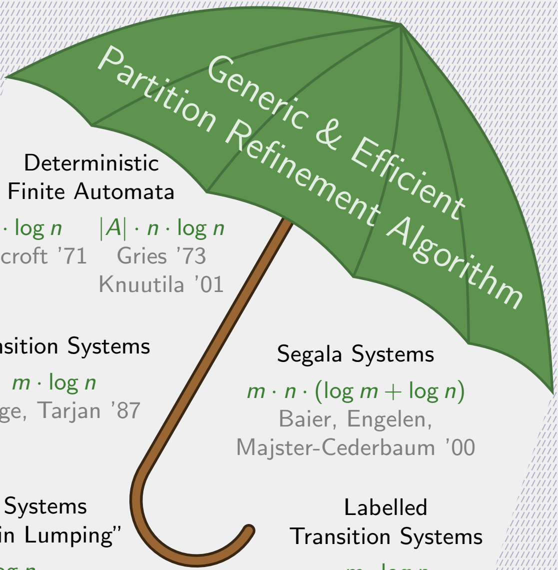
$m \cdot \log n$

Valmari, Franceschinis '10

Labelled Transition Systems

$m \cdot \log n$

Valmari '09



Generic & Efficient Partition Refinement Algorithm

Deterministic Finite Automata

$n \cdot \log n$ Hopcroft '71

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

Transition Systems

$m \cdot \log n$

Paige, Tarjan '87

Segala Systems

$m \cdot n \cdot (\log m + \log n)$

Baier, Engelen, Majster-Cederbaum '00

Weighted Systems "Markov Chain Lumping"

$m \cdot \log n$

Valmari, Franceschinis '10

Labelled Transition Systems

$m \cdot \log n$

Valmari '09

Ingredient 1: Factorizations

Equivalence Relations $\cong$ Quotients $\cong$ Partitions

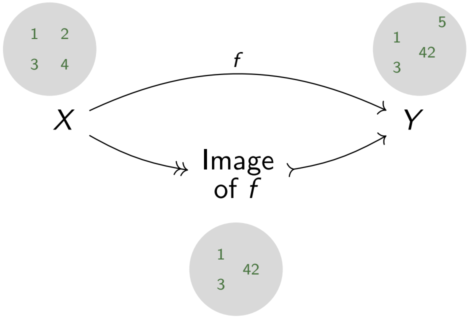
Kernels $\cong$ Regular Epimorphisms

Ingredient 1: Factorizations

Equivalence Relations ≅ Quotients ≅ Partitions

Kernels ≅ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations



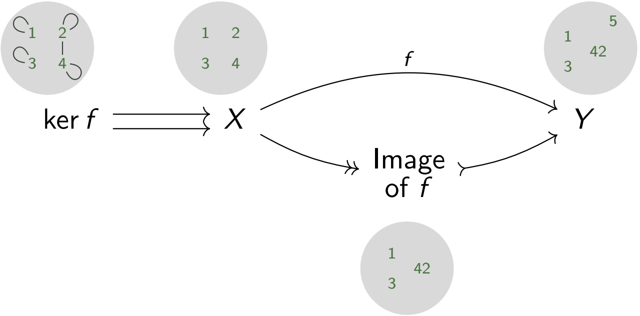
$$\begin{tikzcd}
X \arrow[rr, bend left, "f"] \arrow[dr, two heads] & & Y \\
& \text{Image of } f \arrow[ur, tail] &
\end{tikzcd}$$

Ingredient 1: Factorizations

Equivalence Relations $\cong$ Quotients $\cong$ Partitions

Kernels $\cong$ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations



$$\begin{tikzcd}
\ker f \arrow[r, shift left] \arrow[r, shift right] & X \arrow[rr, bend left, "f"] \arrow[r, two heads] & \text{Image of } f \arrow[r, tail] & Y
\end{tikzcd}$$

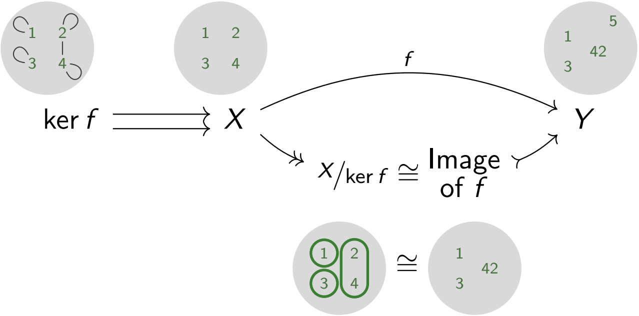
$$\ker f = \{(x_1, x_2) \in X^2 \mid f(x_1) = f(x_2)\}$$

Ingredient 1: Factorizations

Equivalence Relations ≅ Quotients ≅ Partitions

Kernels ≅ Regular Epimorphisms

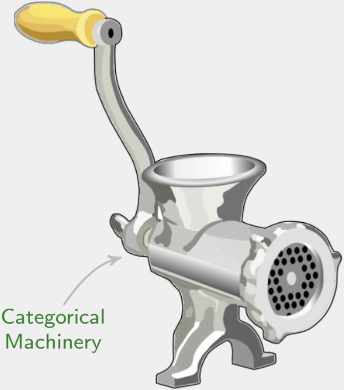
Category with (Regular Epi,Mono)-Factorizations



$$\begin{tikzcd}
\ker f \arrow[r, shift left] \arrow[r, shift right] & X \arrow[rr, bend left, "f"] \arrow[dr, two heads] & & Y \\
& & X/_{\ker f} \cong \text{Image of } f \arrow[ur, tail] &
\end{tikzcd}$$

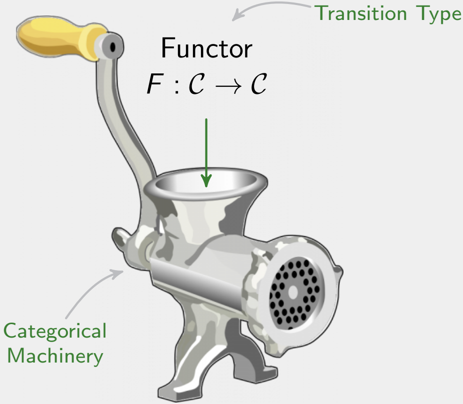
$$\ker f = \{(x_1, x_2) \in X^2 \mid f(x_1) = f(x_2)\}$$

Ingredient 2: Coalgebra – Generic state based systems



Categorical Machinery

Ingredient 2: Coalgebra – Generic state based systems



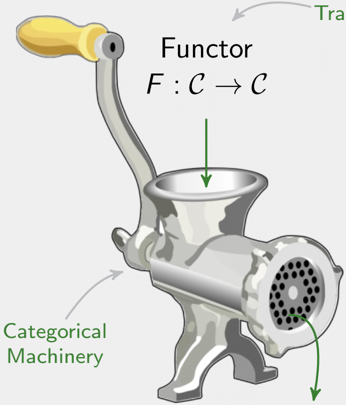
Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Categorical Machinery

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

Functor

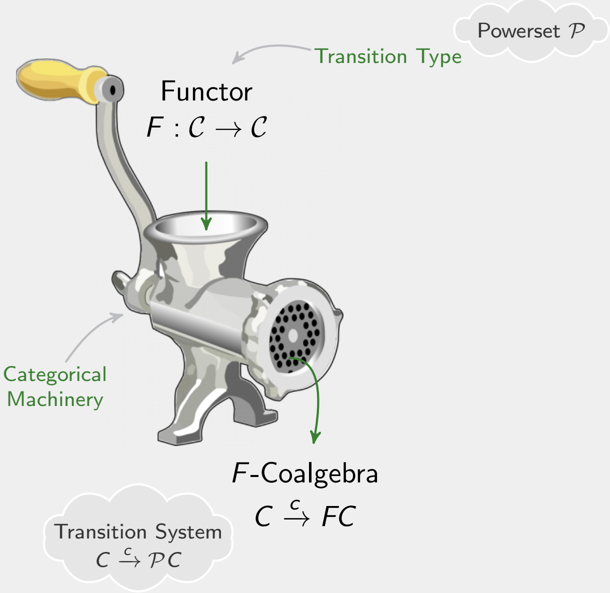
$F : \mathcal{C} \to \mathcal{C}$

Categorical Machinery

F-Coalgebra

$C \xrightarrow{c} FC$

Ingredient 2: Coalgebra – Generic state based systems



Powerset $\mathcal{P}$

Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Categorical Machinery

F-Coalgebra

$C \xrightarrow{c} FC$

Transition System

$C \xrightarrow{c} \mathcal{P}C$

Ingredient 2: Coalgebra – Generic state based systems

Powerset $\mathcal{P}$

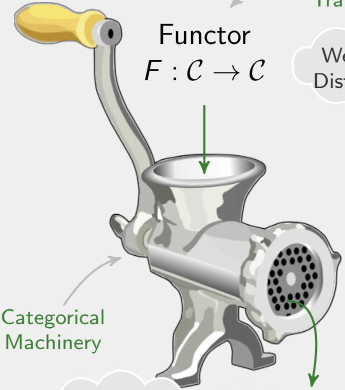
Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Weights $\mathbb{R}^{(-)}$

Distributions $\mathcal{D}$



Categorical Machinery

Markov Chain

$C \xrightarrow{c} \mathbb{R}^{C}$

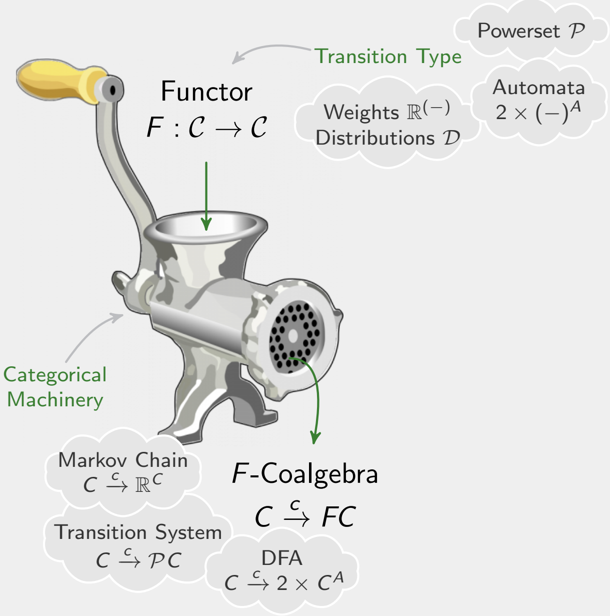
Transition System

$C \xrightarrow{c} \mathcal{P}C$

F-Coalgebra

$C \xrightarrow{c} FC$

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Powerset $\mathcal{P}$

Weights $\mathbb{R}^{(-)}$

Distributions $\mathcal{D}$

Automata

$2 \times (-)^A$

Categorical Machinery

F-Coalgebra

$C \xrightarrow{c} FC$

Markov Chain

$C \xrightarrow{c} \mathbb{R}^C$

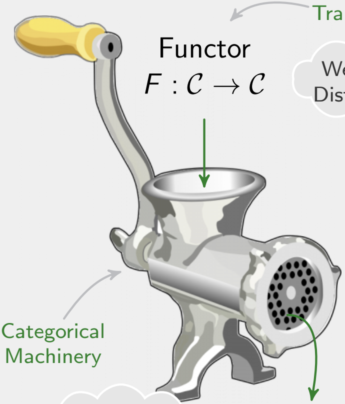
Transition System

$C \xrightarrow{c} \mathcal{P}C$

DFA

$C \xrightarrow{c} 2 \times C^A$

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Powerset $\mathcal{P}$

Simple Segala
$\mathcal{P}(A \times -) \cdot \mathcal{D}$

Weights $\mathbb{R}^{(-)}$
Distributions $\mathcal{D}$

Automata
$2 \times (-)^A$

Categorical Machinery

F-Coalgebra

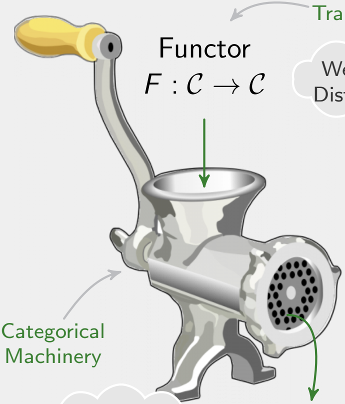
$C \xrightarrow{c} FC$

Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

Transition System
$C \xrightarrow{c} \mathcal{P}C$

DFA
$C \xrightarrow{c} 2 \times C^A$

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Powerset $\mathcal{P}$

Weights $\mathbb{R}^{(-)}$
Distributions $\mathcal{D}$

Automata
$2 \times (-)^A$

Simple Segala
$\mathcal{P}(A \times -) \cdot \mathcal{D}$

General Segala
$\mathcal{P} \cdot \mathcal{D}(A \times -)$

Categorical Machinery

F-Coalgebra

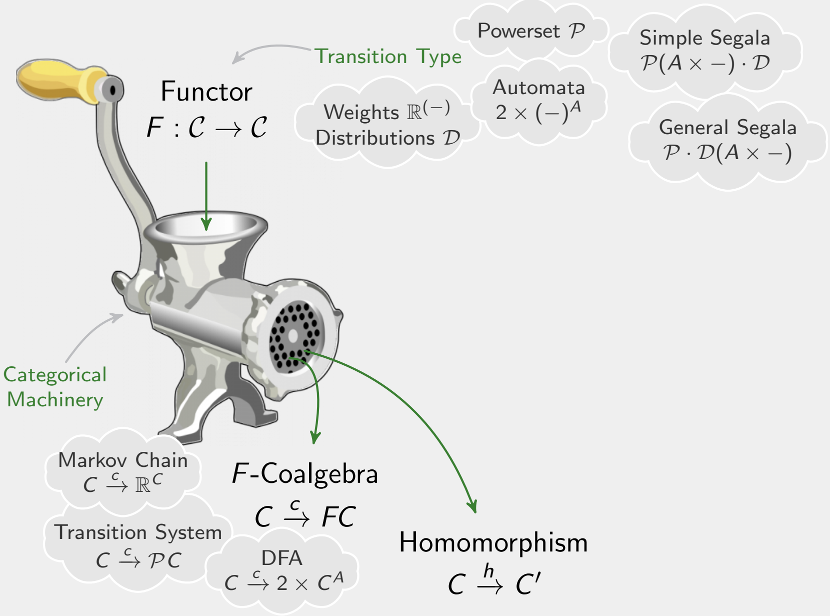
$C \xrightarrow{c} FC$

Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

Transition System
$C \xrightarrow{c} \mathcal{P}C$

DFA
$C \xrightarrow{c} 2 \times C^A$

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Powerset $\mathcal{P}$

Weights $\mathbb{R}^{(-)}$
Distributions $\mathcal{D}$

Automata
$2 \times (-)^A$

Simple Segala
$\mathcal{P}(A \times -) \cdot \mathcal{D}$

General Segala
$\mathcal{P} \cdot \mathcal{D}(A \times -)$

Categorical Machinery

F-Coalgebra

$C \xrightarrow{c} FC$

Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

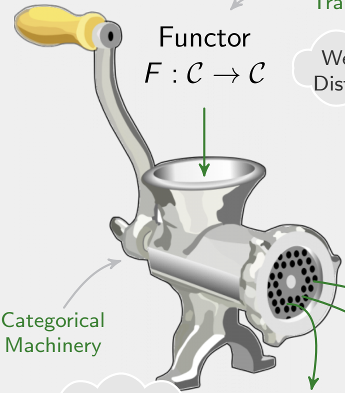
Transition System
$C \xrightarrow{c} \mathcal{P}C$

DFA
$C \xrightarrow{c} 2 \times C^A$

Homomorphism

$C \xrightarrow{h} C'$

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

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

Powerset $\mathcal{P}$

Weights $\mathbb{R}^{(-)}$
Distributions $\mathcal{D}$

Automata
$2 \times (-)^A$

Simple Segala
$\mathcal{P}(A \times -) \cdot \mathcal{D}$

General Segala
$\mathcal{P} \cdot \mathcal{D}(A \times -)$

Categorical Machinery

F-Coalgebra
$C \xrightarrow{c} FC$

Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

Transition System
$C \xrightarrow{c} \mathcal{P}C$

DFA
$C \xrightarrow{c} 2 \times C^A$

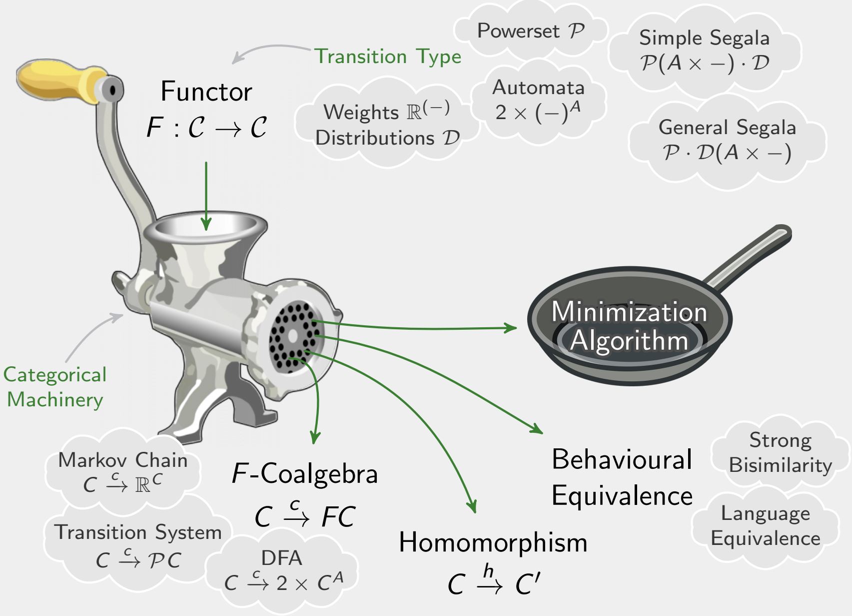
Homomorphism
$C \xrightarrow{h} C'$

Behavioural Equivalence

Strong Bisimilarity

Language Equivalence

Ingredient 2: Coalgebra – Generic state based systems



Transition Type

Functor

$F : \mathcal{C} \to \mathcal{C}$

Powerset $\mathcal{P}$

Weights $\mathbb{R}^{(-)}$
Distributions $\mathcal{D}$

Automata
$2 \times (-)^A$

Simple Segala
$\mathcal{P}(A \times -) \cdot \mathcal{D}$

General Segala
$\mathcal{P} \cdot \mathcal{D}(A \times -)$

Categorical Machinery

Minimization Algorithm

F-Coalgebra

$C \xrightarrow{c} FC$

Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

Transition System
$C \xrightarrow{c} \mathcal{P}C$

DFA
$C \xrightarrow{c} 2 \times C^A$

Homomorphism

$C \xrightarrow{h} C'$

Behavioural Equivalence

Strong Bisimilarity

Language Equivalence

The Coalgebraic Task

For a functor $F : \mathcal{C} \to \mathcal{C}$

$$\begin{tikzcd}
\text{Given a coalgebra} & C \arrow[r, "c"] \arrow[d, two heads, "h"'] & FC \arrow[d, two heads, "Fh"] \\
\text{find the } \textbf{simple} \text{ quotient} & C' \arrow[r, "c'"] & FC'
\end{tikzcd}$$

no proper quotient

all equivalent behaviours identified

The Coalgebraic Task

For a functor $F : \mathcal{C} \to \mathcal{C}$

$$\begin{tikzcd}
\text{Given a coalgebra} & C \arrow[r, "c"] \arrow[d, two heads, "h"'] & FC \arrow[d, two heads, "Fh"] \\
\text{find the \textbf{simple} quotient} & C' \arrow[r, "c'"] & FC'
\end{tikzcd}$$

no proper quotient

all equivalent behaviours identified

Instance

For $2 \times (-)^A$: $\begin{tikzcd} \mathsf{Set} \arrow[loop right] \end{tikzcd}$

Automata minimization

The Coalgebraic Task

For a functor $F : \mathcal{C} \to \mathcal{C}$

Given a coalgebra

find the **simple** quotient

$$\begin{tikzcd} C \arrow[r, "c"] \arrow[d, two heads, "h"'] & FC \arrow[d, two heads, "Fh"] \\ C' \arrow[r, "c'"] & FC' \end{tikzcd}$$

no proper quotient

all equivalent behaviours identified

Instance

Instance

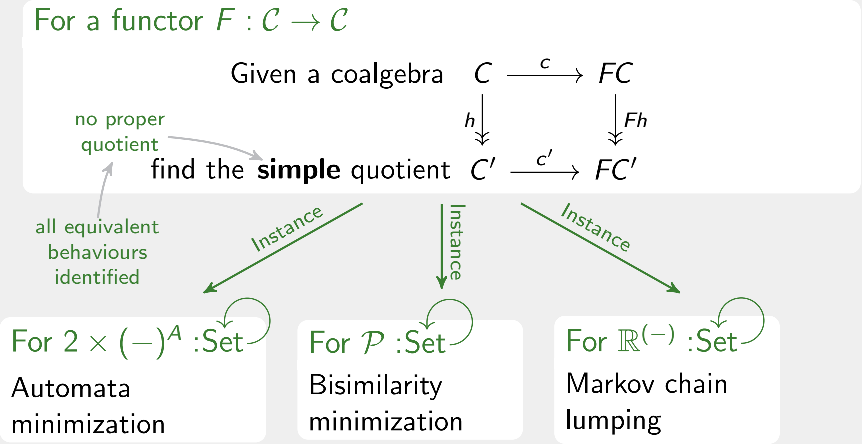
For $2 \times (-)^A$:Set

Automata minimization

For $\mathcal{P}$:Set

Bisimilarity minimization

The Coalgebraic Task



For a functor $F : \mathcal{C} \to \mathcal{C}$

Given a coalgebra / find the **simple** quotient

$$\begin{tikzcd} C \arrow[r, "c"] \arrow[d, two heads, "h"'] & FC \arrow[d, two heads, "Fh"] \\ C' \arrow[r, "c'"] & FC' \end{tikzcd}$$

no proper quotient

all equivalent behaviours identified

For $2 \times (-)^A$: Set

Automata minimization

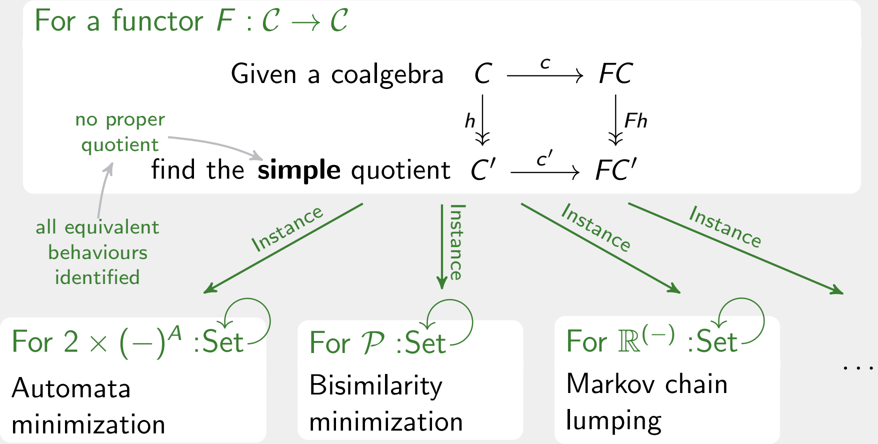
For $\mathcal{P}$: Set

Bisimilarity minimization

For $\mathbb{R}^{(-)}$: Set

Markov chain lumping

The Coalgebraic Task



For a functor $F : \mathcal{C} \to \mathcal{C}$

$$\begin{tikzcd}
\text{Given a coalgebra} & C \arrow[r, "c"] \arrow[d, two heads, "h"'] & FC \arrow[d, two heads, "Fh"] \\
\text{find the } \textbf{simple} \text{ quotient} & C' \arrow[r, "c'"] & FC'
\end{tikzcd}$$

no proper quotient

all equivalent behaviours identified

For $2 \times (-)^A$:Set

Automata minimization

For $\mathcal{P}$:Set

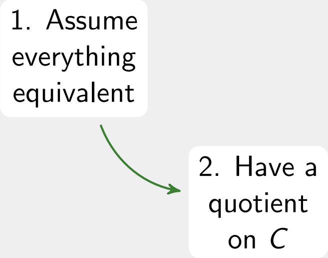
Bisimilarity minimization

For $\mathbb{R}^{(-)}$:Set

Markov chain lumping

…

1. Assume everything equivalent



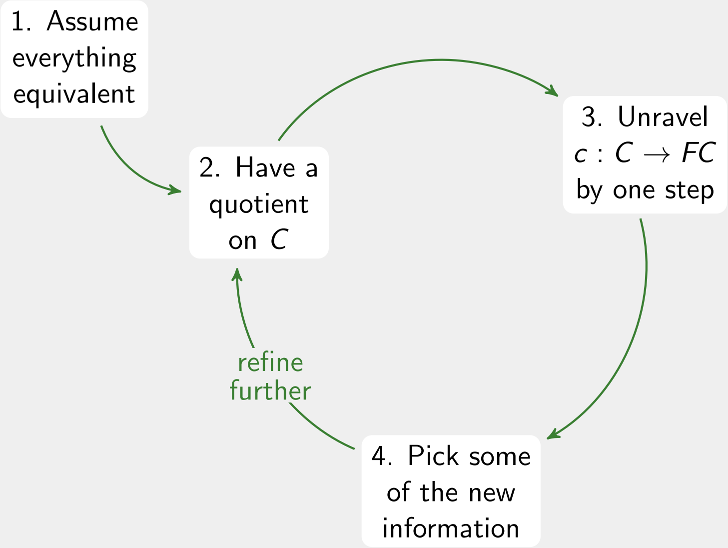
1. Assume everything equivalent

2. Have a quotient on C

1. Assume everything equivalent

2. Have a quotient on C

3. Unravel $c : C \to FC$ by one step



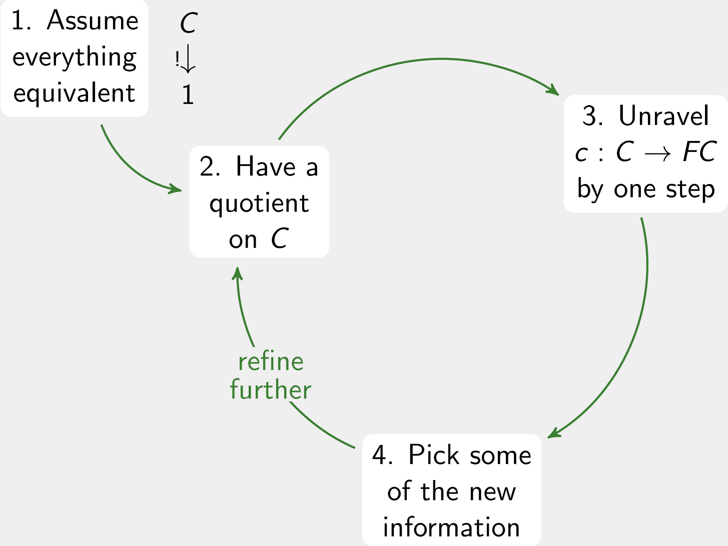
1. Assume everything equivalent

2. Have a quotient on C

3. Unravel $c : C \to FC$ by one step

4. Pick some of the new information

refine further



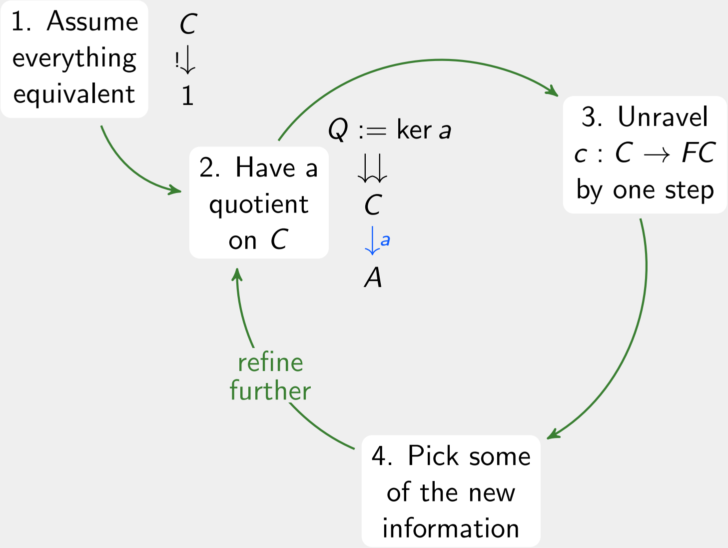
1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

2. Have a quotient on C

3. Unravel $c : C \to FC$ by one step

4. Pick some of the new information



1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

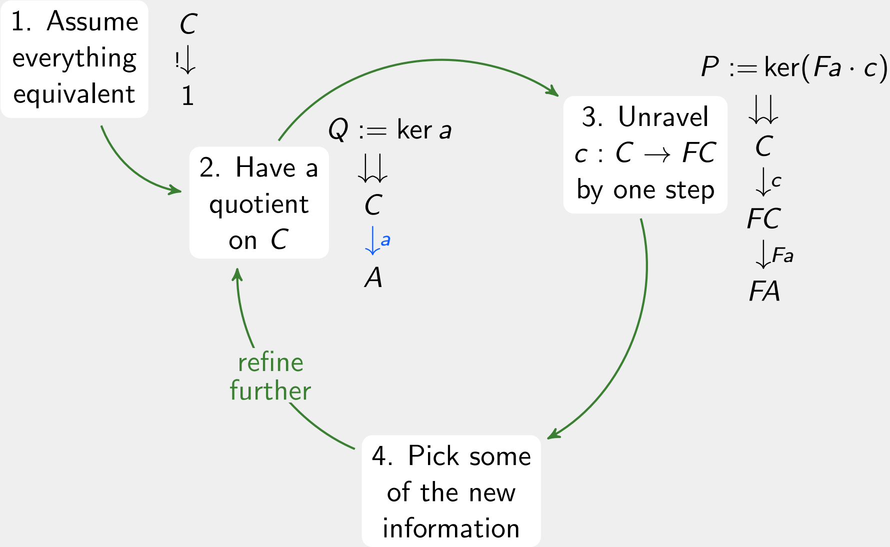
2. Have a quotient on C

$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \\ A \end{tikzcd}$$

3. Unravel $c : C \to FC$ by one step

4. Pick some of the new information

refine further



1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

2. Have a quotient on C

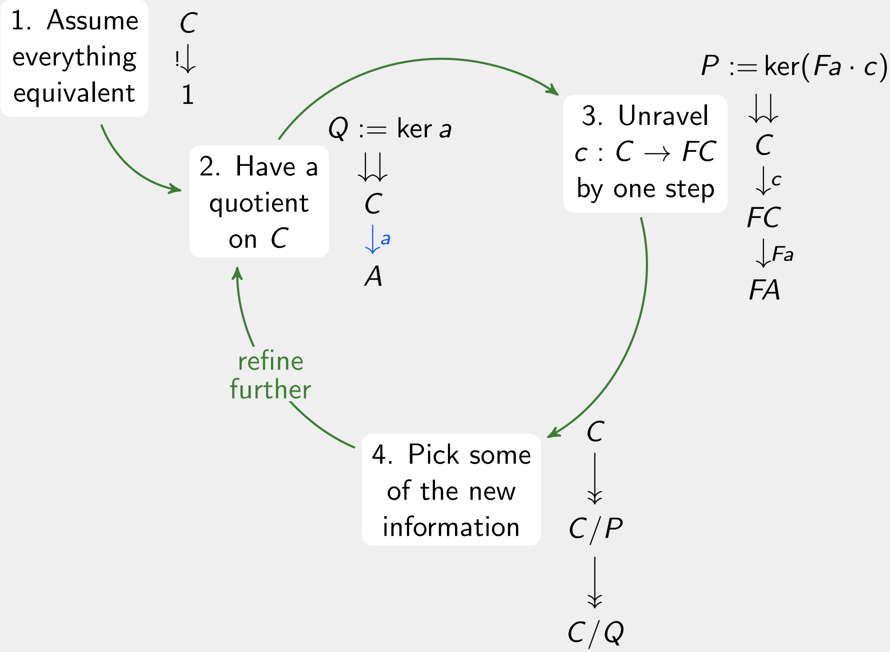
$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \\ A \end{tikzcd}$$

3. Unravel $c : C \to FC$ by one step

$$\begin{tikzcd} P := \ker(Fa \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "c"] \\ FC \arrow[d, "Fa"] \\ FA \end{tikzcd}$$

4. Pick some of the new information

refine further



1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

2. Have a quotient on C

$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \\ A \end{tikzcd}$$

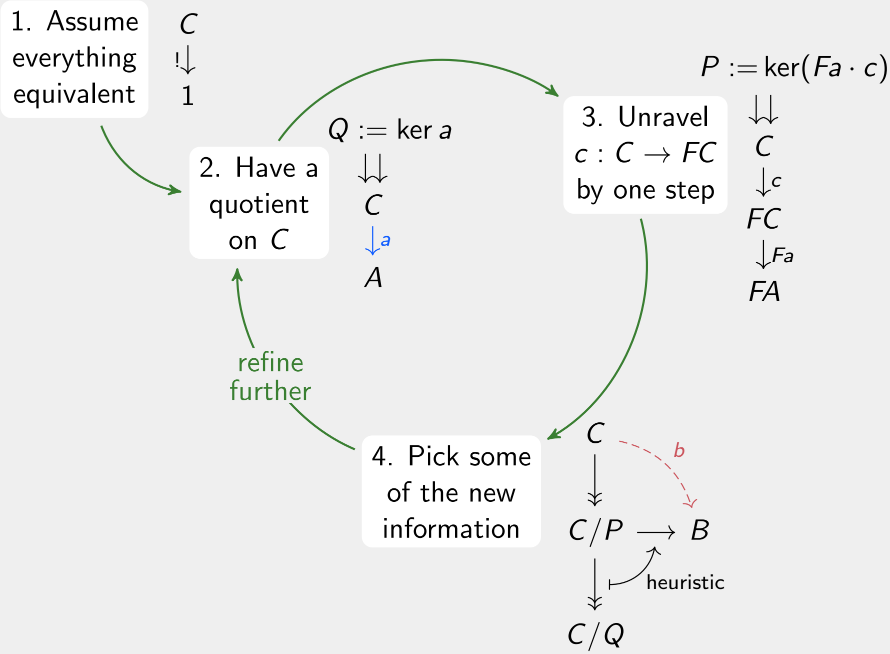
3. Unravel $c : C \to FC$ by one step

$$\begin{tikzcd} P := \ker(Fa \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "c"] \\ FC \arrow[d, "Fa"] \\ FA \end{tikzcd}$$

4. Pick some of the new information

$$\begin{tikzcd} C \arrow[d, two heads] \\ C/P \arrow[d, two heads] \\ C/Q \end{tikzcd}$$

refine further



1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

2. Have a quotient on C

$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \\ A \end{tikzcd}$$

3. Unravel $c : C \to FC$ by one step

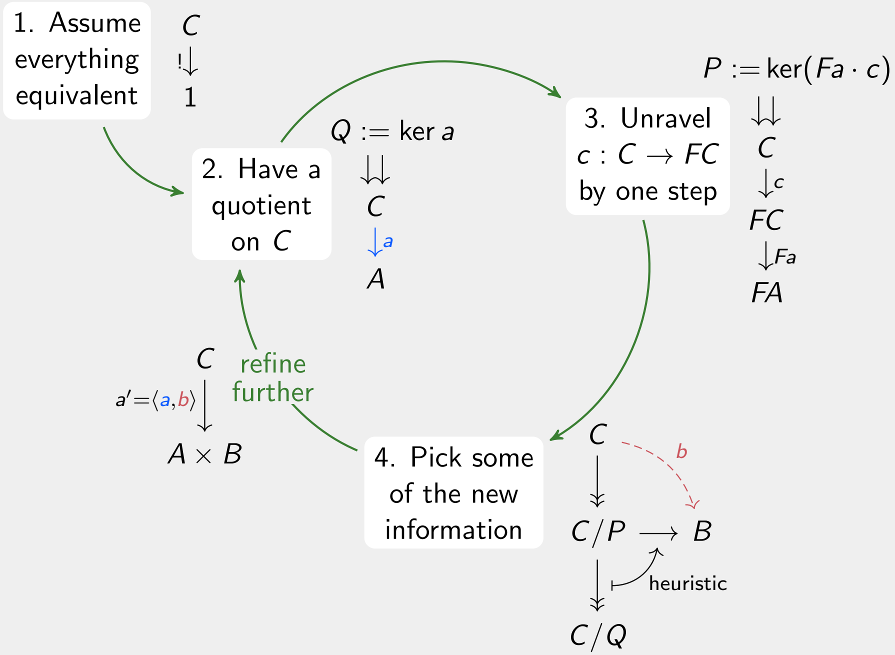
$$\begin{tikzcd} P := \ker(Fa \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "c"] \\ FC \arrow[d, "Fa"] \\ FA \end{tikzcd}$$

4. Pick some of the new information

$$\begin{tikzcd} C \arrow[d, two heads] \arrow[rd, dashed, bend left, "b"] & \\ C/P \arrow[r] \arrow[d, two heads] & B \\ C/Q & \end{tikzcd}$$

heuristic

refine further



1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

2. Have a quotient on C

$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \\ A \end{tikzcd}$$

3. Unravel $c : C \to FC$ by one step

$$\begin{tikzcd} P := \ker(Fa \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "c"] \\ FC \arrow[d, "Fa"] \\ FA \end{tikzcd}$$

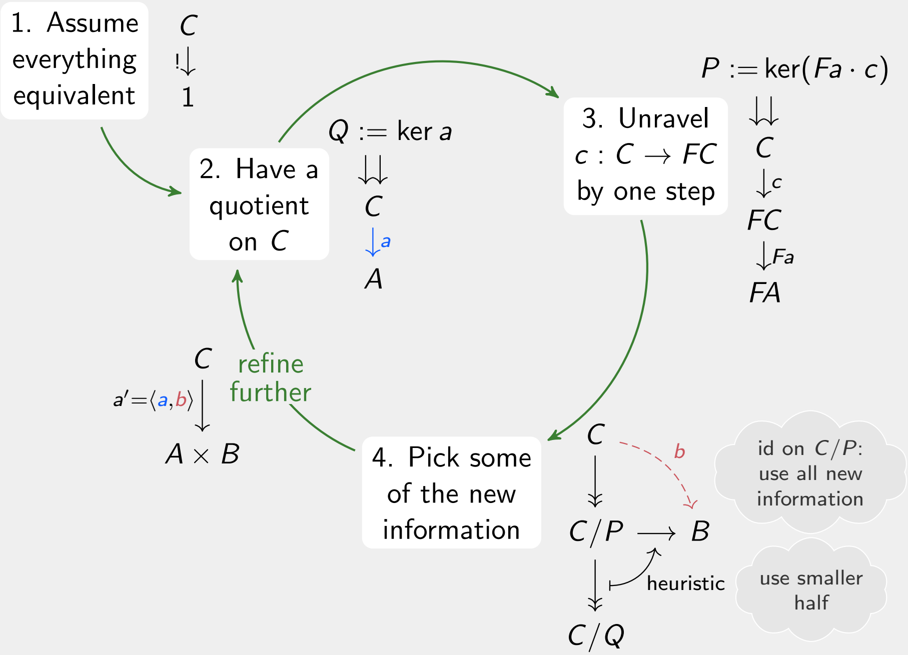
4. Pick some of the new information

$$\begin{tikzcd} C \arrow[d, two heads] \arrow[dr, dashed, bend left, "b"] & \\ C/P \arrow[r] \arrow[d, two heads] & B \\ C/Q \end{tikzcd}$$

heuristic

refine further

$$\begin{tikzcd} C \arrow[d, "a' = \langle a, b \rangle"'] \\ A \times B \end{tikzcd}$$



1. Assume everything equivalent

$$\begin{tikzcd} C \arrow[d, "!"'] \\ 1 \end{tikzcd}$$

2. Have a quotient on C

$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \\ A \end{tikzcd}$$

3. Unravel $c : C \to FC$ by one step

$$\begin{tikzcd} P := \ker(Fa \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "c"] \\ FC \arrow[d, "Fa"] \\ FA \end{tikzcd}$$

4. Pick some of the new information

$$\begin{tikzcd} C \arrow[d, two heads] \arrow[dr, dashed, bend left, "b"] & \\ C/P \arrow[r] \arrow[d, two heads] & B \\ C/Q & \end{tikzcd}$$

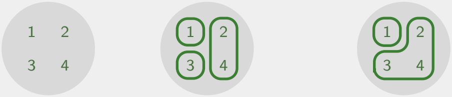
id on C/P: use all new information

use smaller half

refine further

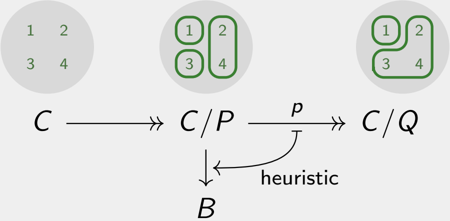
$$\begin{tikzcd} C \arrow[d, "a' = \langle a, b \rangle"'] \\ A \times B \end{tikzcd}$$

Heuristic



$$\begin{tikzcd}
C \arrow[r, two heads] & C/P \arrow[r, two heads, "p", ""{name=P, below}] \arrow[d, ""{name=D}] & C/Q \\
& B &
\arrow[from=P, to=D, bend left, "\text{heuristic}"]
\end{tikzcd}$$

Heuristic

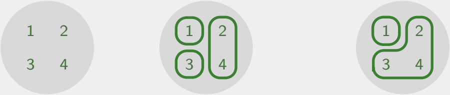


$$\begin{tikzcd} C \arrow[r, two heads] & C/P \arrow[r, two heads, "p"{name=P}] \arrow[d, ""{name=D}] & C/Q \\ & B & \arrow[from=P, to=D, bend left, "\text{heuristic}"] \end{tikzcd}$$

Use all new information

$B = C/P \rightsquigarrow$ Final Chain algorithm König, Küpper '14

Heuristic



$$\begin{tikzcd}
C \arrow[r, two heads] & C/P \arrow[r, two heads, "p", ""{name=P, below}] \arrow[d, ""{name=D}] & C/Q \\
& B &
\arrow[from=P, to=D, bend left, "\text{heuristic}"]
\end{tikzcd}$$

Use all new information

$B = C/P \rightsquigarrow$ Final Chain algorithm König, Küpper '14

Process the smaller half

Surrounding block in C/Q

Let $S \in C/P$, such that $2 \cdot |S| \leq |p(S)|$

$$B = \{\overset{\{3\}}{\text{ChosenBlock}}, \overset{\{2, 4\}}{\text{SameSurroundingBlock}}, \overset{\{1\}}{\text{RemainingBlocks}}\}$$

Assume

- Finitely complete category $\mathcal{C}$
- (RegularEpi,Mono)-factorisations
- F mono-preserving

Theorem (Correctness)

$$\begin{tikzcd}
C \arrow[r, "c"] \arrow[d, two heads] & FC \arrow[d] \\
C/P_i \arrow[r] & F(C/Q_i)
\end{tikzcd}$$

Assume

- Finitely complete category $\mathcal{C}$
- (RegularEpi,Mono)-factorisations
- F mono-preserving

Theorem (Correctness)

$$\begin{tikzcd}
C \arrow[r, "c"] \arrow[d, two heads] & FC \arrow[d] \\
C/P_i \arrow[r] & F(C/Q_i)
\end{tikzcd}$$

If $P_i \cong Q_i$, then this

1. is a coalgebra
2. has no proper quotient

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \\
A
\end{tikzcd}$$

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[rr] & & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & & A \times B
\end{tikzcd}$$

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[rr] & & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] \\
FA
\end{tikzcd}$$

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"'] \arrow[rr] & & C \arrow[d, "a'=\langle a,b\rangle"'] \\
A & & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & & ?? \cap ?? \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"'] & & C \arrow[d, "c"'] \\
FC \arrow[d, "Fa"'] \arrow[rr] & & FC \arrow[d, "F\langle a,b\rangle"'] \\
FA & & F(A \times B)
\end{tikzcd}$$

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[r] & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & ?? \cap ?? \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] \arrow[r] & C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] & FC \arrow[d, "F\langle a,b\rangle"] \\
FA & F(A \times B)
\end{tikzcd}$$

Question: When is $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$?

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[rr] & & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & & ?? \cap ?? \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] \arrow[rr] & & C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] & & FC \arrow[d, "F\langle a,b\rangle"] \\
FA & & F(A \times B)
\end{tikzcd}$$

Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[r] & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & ?? \cap ?? \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] \arrow[r] & C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] & FC \arrow[d, "F\langle a,b\rangle"] \\
FA & F(A \times B)
\end{tikzcd}$$

Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

$$\begin{tikzcd}
F(L+R) \arrow[d] \\
F(L+1) \times F(1+R)
\end{tikzcd}$$

injective and

"zippable"

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[rr] & & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & & ?? \cap ?? \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] \arrow[rr] & & C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] & & FC \arrow[d, "F\langle a,b\rangle"] \\
FA & & F(A \times B)
\end{tikzcd}$$

Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

$$\begin{tikzcd}
F(L+R) \arrow[d] \\
F(L+1) \times F(1+R)
\end{tikzcd}$$

injective and $\ker a \cup \ker b$ an equivalence

“zippable”

Functors F zippable, if

$$F(L+R) \xrightarrow{\text{unzip}} F(L+1) \times F(1+R) \text{ is monic.}$$

E.g. Id, Constants, $\times$, $+$, $\hookrightarrow$, $M^{(-)}$, part. additive

Examples for sets $L = \{a_1, a_2, a_3\}$, $R = \{b_1, b_2\}$, $1 = \{\_\}$

$$a_1\ a_2\ b_1\ a_3\ b_2 \xmapsto{\text{unzip}} \begin{array}{l}(\,a_1\ \ a_2\ \ \_\ \ a_3\ \ \_\,,\\ \ \ \_\ \ \ \_\ \ b_1\ \ \_\ \ b_2)\end{array}$$

$(-)^*$ is zippable

$$\{a_1, a_2, b_1\} \xmapsto{\text{unzip}} \begin{array}{l}(\{a_1, a_2, \_\},\\ \quad \{\_, b_1\})\end{array}$$

$\mathcal{P}$ is zippable

Functors F zippable, if

$$F(L+R) \xrightarrow{\mathrm{unzip}} F(L+1) \times F(1+R) \text{ is monic.}$$

E.g. Id, Constants, $\times$, $+$, $\hookrightarrow$, $M^{(-)}$, part. additive

Examples for sets $L = \{a_1, a_2, a_3\}$, $R = \{b_1, b_2\}$, $1 = \{\_\}$

$$\begin{tikzcd} a_1\ a_2\ b_1\ a_3\ b_2 \arrow[d, mapsto, "\mathrm{unzip}"] \\ (\,a_1\ \ a_2\ \ \_\ \ a_3\ \ \_\,,\ \ \_\ \ \_\ \ b_1\ \ \_\ \ b_2) \end{tikzcd}$$

$(-)^*$ is zippable

$$\begin{tikzcd} \{a_1, a_2, b_1\} \arrow[d, mapsto, "\mathrm{unzip}"] \\ (\{a_1, a_2, \_\}, \{\_, b_1\}) \end{tikzcd}$$

$\mathcal{P}$ is zippable

$$\begin{tikzcd} \{\{a_1, b_1\}, \{a_2, b_2\}\} \arrow[dr, mapsto, "\mathrm{unzip}"'] & & \{\{a_1, b_2\}, \{a_2, b_1\}\} \arrow[dl, mapsto, "\mathrm{unzip}"] \\ & (\{\{a_1, \_\}, \{a_2, \_\}\}, \{\{\_, b_1\}, \{\_, b_2\}\}) & \end{tikzcd}$$

$\mathcal{P}\mathcal{P}$ is not zippable

~~Composition~~

~~Quotients~~

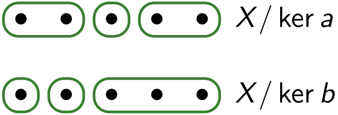
$$A \xleftarrow{a} X \xrightarrow{b} B$$

$\ker a \cup \ker b$ is an equivalence in Set

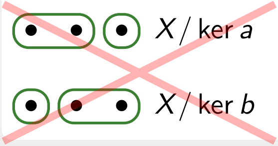
$\Leftrightarrow$ $\ker a \cup \ker b$ transitive

$\Leftrightarrow$ $\forall x \in X : [x]_a \subseteq [x]_b$ or $[x]_a \supseteq [x]_b$

Example



Non-Example



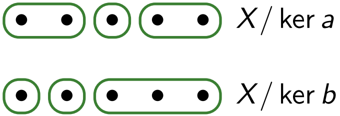
$$A \xleftarrow{a} X \xrightarrow{b} B$$

$\ker a \cup \ker b$ is an equivalence in Set

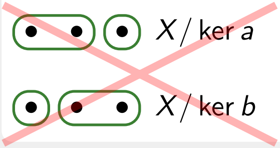
$\Leftrightarrow$ $\ker a \cup \ker b$ transitive

$\Leftrightarrow$ $\forall x \in X : [x]_a \subseteq [x]_b$ or $[x]_a \supseteq [x]_b$

Example



Non-Example



Heuristics **respecting compound blocks**:

$$\begin{tikzcd} C \arrow[r, two heads] & C/P \arrow[r, two heads, "p"{name=P}] \arrow[d, "k"'{name=K}] & C/Q \\ & B & \end{tikzcd}$$

heuristic

s.th. $\ker k \cup \ker p$ equivalence

E.g. **use all new information** and **process the smaller half**

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[rr] & & C \arrow[d, "a'=\langle a,b\rangle"] \\
A & & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & & {\color{red} ?? \cap ??} \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] & & C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] \arrow[rr] & & FC \arrow[d, "F\langle a,b\rangle"] \\
FA & & F(A \times B)
\end{tikzcd}$$

Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

$$\begin{tikzcd}
F(L+R) \arrow[d] \\
F(L+1) \times F(1+R)
\end{tikzcd}$$

injective and $\ker a \cup \ker b$ an equivalence

“zippable”

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{tikzcd} Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "a"] \arrow[r] & C \arrow[d, "a'=\langle a,b\rangle"] \\ A & A \times B \end{tikzcd}$$

$$\begin{tikzcd} P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & P \cap \ker(Fb \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\ C \arrow[d, "c"] \arrow[r] & C \arrow[d, "c"] \\ FC \arrow[d, "Fa"] & FC \arrow[d, "F\langle a,b\rangle"] \\ FA & F(A \times B) \end{tikzcd}$$

Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

$$\begin{tikzcd} F(L+R) \arrow[d] \\ F(L+1) \times F(1+R) \end{tikzcd}$$

injective and $\ker a \cup \ker b$ an equivalence

“zippable”

Efficiency: Incremental Partitions

Concrete Algorithm?

Incremental partitions

$$\begin{tikzcd}
Q := \ker a \arrow[d, shift left] \arrow[d, shift right] & Q \cap \ker b \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "a"] \arrow[r] & C \arrow[d, "a' = \langle a, b\rangle"] \\
A & A \times B
\end{tikzcd}$$

$$\begin{tikzcd}
P := \ker(c \cdot Fa) \arrow[d, shift left] \arrow[d, shift right] & P \cap \ker(Fb \cdot c) \arrow[d, shift left] \arrow[d, shift right] \\
C \arrow[d, "c"] & C \arrow[d, "c"] \\
FC \arrow[d, "Fa"] \arrow[r] & FC \arrow[d, "F\langle a, b\rangle"] \\
FA & F(A \times B)
\end{tikzcd}$$

Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

$$\begin{tikzcd}
F(L+R) \arrow[d] \\
F(L+1) \times F(1+R)
\end{tikzcd}$$

injective and $\ker a \cup \ker b$ an equivalence

"zippable"

Setting for complexity analysis

Category:	Heuristic:	Functor:
Set	smaller half	zippable & **encoding**

Setting for complexity analysis

Category:	Heuristic:	Functor:
Set	smaller half	zippable & **encoding**

Assumption: Functor encoding

- Refinement interface { `type W`, `type L`, `init()`, `w()`, `update()` }
- coalgebra structure as edges with labels

$$C \xrightarrow{c} FC \xrightarrow{\flat} \mathcal{P}(L \times C)$$

⇒ compute “smaller half” intersections in linear time

Functor encoding

- type of weights W, type of edge labels L
- weight function $w : \mathcal{P}X \times FX \to W$
- $\flat : FX \to \mathcal{B}_\mathrm{f}(L \times X)$
- $\mathrm{init} : F1 \times \mathcal{B}_\mathrm{f}(L) \to W$
- $\mathrm{update} : \underbrace{\mathcal{B}_\mathrm{f}(L)}_{\text{edges into } S} \times \underbrace{W}_{\text{weight of } C} \longrightarrow \underbrace{W}_{\text{weight of } S} \times \underbrace{F(2 \times 2)}_{F\langle \chi_S, \chi_C \rangle} \times \underbrace{W}_{\text{weight of } C \setminus S}$

Functor:	$G^{(-)}$	$\mathcal{B}_\mathrm{f}$	$\mathcal{D}$	$\mathcal{P}$	F_Σ
Labels L:	G	$\mathbb{N}$	$[0,1]$	1	$\mathbb{N}$
Weights W:	$G^{(2)}$	$\mathcal{B}_\mathrm{f}2$	$\mathcal{D}2$	$\mathbb{N}$	$F_\Sigma 2$
$w(C)$, $C \subseteq X$:	$G\chi_C$	$\mathcal{B}_\mathrm{f}\chi_C$	$\mathcal{D}\chi_C$	$\lvert C \cap (-) \rvert$	$F_\Sigma \chi_C$

Setting for complexity analysis

Category:	Heuristic:	Functor:
Set	smaller half	zippable & **encoding**

Assumption: Functor encoding

- Refinement interface { `type W`, `type L`, `init()`, `w()`, `update()` }
- coalgebra structure as edges with labels

$$C \xrightarrow{c} FC \xrightarrow{\flat} \mathcal{P}(L \times C)$$

$\Rightarrow$ compute "smaller half" intersections in linear time

Setting for complexity analysis

Category:	Heuristic:	Functor:
Set	smaller half	zippable & **encoding**

Assumption: Functor encoding

- Refinement interface { `type W`, `type L`, `init()`, `w()`, `update()` }
- coalgebra structure as edges with labels

$$C \xrightarrow{c} FC \xrightarrow{\flat} \mathcal{P}(L \times C)$$

$\Rightarrow$ compute "smaller half" intersections in linear time

Theorem

Overall complexity: $\mathcal{O}((m+n)\cdot \log n)$ for $n = |C|$, $m = \sum_{x\in C} |\flat c(x)|$

System	Functor	Concrete algorithm		Our instantiation
Transition Systems	$\mathcal{P}$	$(m+n)\cdot\log n$ Paige, Tarjan '87	$=$	$(m+n)\cdot\log n$
LTS	$\mathcal{P}(A\times -)$	$(m+n)\cdot\log(m+n)$ Dovier, Piazza, Policriti '04	$=$	$(m+n)\cdot\log(m+n)$
		$(m+n)\cdot\log m$ Valmari '09	$<$	
Markov Chains	$\mathbb{R}^{(-)}$	$(m+n)\cdot\log n$ Valmari, Franceschinis '10	$=$	$(m+n)\cdot\log n$
DFA	$2\times(-)^A$	$n\cdot\log n$ for fixed A, Hopcroft '71	$=$	$n\cdot\log n$
	$2\times\mathcal{P}(A\times -)$	$\lvert A\rvert\cdot n\cdot\log n$, Gries '73, Knuutila '01	$\approx$	$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m\cdot n\cdot\log(m\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$>$	$(m+n)\cdot\log(m+n)$

System	Functor	Concrete algorithm		Our instantiation
Transition Systems	$\mathcal{P}$	$(m+n)\cdot\log n$ Paige, Tarjan '87	$=$	$(m+n)\cdot\log n$
LTS	$\mathcal{P}(A\times -)$	$(m+n)\cdot\log(m+n)$ Dovier, Piazza, Policriti '04 $(m+n)\cdot\log n$ Valmari '09	$=$	$(m+n)\cdot\log(m+n)$
Markov Chains	$\mathbb{R}^{(-)}$	$(m+n)\cdot\log n$ Valmari, Franceschinis '10	$=$	$(m+n)\cdot\log n$
DFA	$2\times(-)^A$	$n\cdot\log n$ for fixed A, Hopcroft '71	$=$	$n\cdot\log n$
	$2\times\mathcal{P}(A\times -)$	$\lvert A\rvert\cdot n\cdot\log n$, Gries '73, Knuutila '01	$\approx$	$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m\cdot n\cdot\log(m\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$>$	$(m+n)\cdot\log(m+n)$

Generic & Efficient

System	Functor	[illegible]		Our instantiation
Transition Systems	$\mathcal{P}$	[illegible]	$=$	$(m+n)\cdot\log n$
LTS	$\mathcal{P}(A\times -)$	[illegible] Dovier, Piazza, Policriti '04 $(m+n)\cdot\log n$ Valmari '09	$=$	$(m+n)\cdot\log(m+n)$
Markov Chains	$\mathbb{R}^{(-)}$	$(m+n)\cdot\log n$ Valmari, Franceschinis '10	$=$	$(m+n)\cdot\log n$
DFA	$2\times(-)^A$	$n\cdot\log n$ for fixed A, Hopcroft '71	$=$	$n\cdot\log n$
	$2\times\mathcal{P}(A\times -)$	$\lvert A\rvert\cdot n\cdot\log n$, Gries '73, Knuutila '01	$\approx$	$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m\cdot n\cdot\log(m\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$>$	$(m+n)\cdot\log(m+n)$

More instances:
further system types
& categories

Generic & Efficient

System	Functor	[illegible]hm		Our instantiation
Transition Systems	$\mathcal{P}$	[illegible]	=	$(m+n)\cdot\log n$
LTS	[illegible]	[illegible] Dovi[illegible]azza, P[illegible]riti '04 $(m+$[illegible] log n [illegible]lmari '0[illegible]	=	$(m+n)\cdot\log(m+n)$
Mark[illegible] Chains	[illegible]	$(m+$[illegible] log n Valm[illegible] France[illegible]hinis '10	=	$(m+n)\cdot\log n$
DFA	$2\times(-$[illegible]	$n\cdot$ lo[illegible] for fixed A, Hopcroft '71	=	$n\cdot\log n$
	$2\times\mathcal{P}(A\times$[illegible]	$\lvert A\rvert\cdot n\cdot\log n$, Gries '73, Knuutila '01	≈	$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m\cdot n\cdot\log(m\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	>	$(m+n)\cdot\log(m+n)$

More instances:
further system types
& categories

Implementation:
functor & system
as the input

Generic & Efficient

System	Functor	hm		Our instantiation
Transition Systems	$\mathcal{P}$		=	$(m+n)\cdot\log n$
LTS		Dovie iazza, Po criti ’04 $(m+$ log n imari ’0	=	m $n)\cdot\log(m+n)$
Mark Chains		$(m+$ log n Valm , rance inis ’10	=	$(m+n)\cdot\log n$
DFA	$2\times(-$	$n\cdot$ lo f Ho		$n\cdot\log n$
	$2\times\mathcal{P}(A\times$	$\lvert A$ Gries ’		$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m\cdot n\cdot$ lo Baier, Engelen, Majster-Cederbaum’00	>	$(m+n)\cdot\log(m+n)$

More instances:
further system types
& categories

Implementation:
functor & system
as the input

Generic & Efficient

Compare to
existing concrete
implementations

System	Functor	Concrete algorithm		Our instantiation
Transition Systems	$\mathcal{P}$	$(m+n)\cdot\log n$ Paige, Tarjan '87	$=$	$(m+n)\cdot\log n$
LTS	$\mathcal{P}(A\times -)$	$(m+n)\cdot\log(m+n)$ Dovier, Piazza, Policriti '04	$=$	$(m+n)\cdot\log(m+n)$
		$(m+n)\cdot\log m$ Valmari '09	$<$	
Markov Chains	$\mathbb{R}^{(-)}$	$(m+n)\cdot\log n$ Valmari, Franceschinis '10	$=$	$(m+n)\cdot\log n$
DFA	$2\times(-)^A$	$n\cdot\log n$ for fixed A, Hopcroft '71	$=$	$n\cdot\log n$
	$2\times\mathcal{P}(A\times -)$	$\lvert A\rvert\cdot n\cdot\log n$, Gries '73, Knuutila '01	$\approx$	$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m\cdot n\cdot\log(m\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$>$	$(m+n)\cdot\log(m+n)$

Appendix …

Genericity: Initial partiton

Given

$$C \xrightarrow{c} FC$$

Usual partition refinement algorithms

Return coarsest partition compatible with c, refining $C \overset{\kappa}{\twoheadrightarrow} \mathcal{I}$

Genericity: Initial partiton

Given

$$C \xrightarrow{c} FC$$

Usual partition refinement algorithms

Return coarsest partition compatible with c, refining $C \overset{\kappa}{\twoheadrightarrow} \mathcal{I}$

$$\Updownarrow$$

Coalgebraic partition refinement for $\mathcal{I} \times F$

For the coalgebra $C \xrightarrow{\langle \kappa, c \rangle} \mathcal{I} \times FC$

Genericity: Composition

If F finitary,

$$C \xrightarrow{c} FG\,C$$

Genericity: Composition

If F finitary,

$$C \xrightarrow{c} FG\,C \qquad \rightsquigarrow \qquad D \xhookrightarrow{d} GC$$

Genericity: Composition

If F finitary,

$$\begin{tikzcd}
C \arrow[r, "c"] \arrow[dr, dashed, "c'"'] & FG\,C & & D \arrow[r, hook, "d"] & GC \\
& FD \arrow[u, "Fd"'] & & &
\end{tikzcd}
\qquad \rightsquigarrow$$

Genericity: Composition

If F finitary,

$$\begin{tikzcd} C \arrow[r, "c"] \arrow[dr, dashed, "c'"'] & FG\,C & \rightsquigarrow & D \arrow[r, hook, "d"] & GC \\ & FD \arrow[u, "Fd"'] & & & \end{tikzcd}$$

A coalgebra on Set^2 for the functor $(X, Y) \mapsto (FY, GX)$:

$$(C, D) \xrightarrow{(c',d)} (FD, GC)$$

Genericity: Composition

If F finitary,

$$\begin{tikzcd}
C \arrow[r, "c"] \arrow[dr, dashed, "c'"'] & FG\,C & \rightsquigarrow & D \arrow[r, hook, "d"] & GC \\
& FD \arrow[u, "Fd"'] & & &
\end{tikzcd}$$

A coalgebra on Set^2 for the functor $(X, Y) \mapsto (FY, GX)$:

$$(C, D) \xrightarrow{(c', d)} (FD, GC)$$

Examples

$$\mathcal{P} \cdot (A \times (-)) \qquad (2 \times \mathcal{P}) \cdot (A \times (-))$$
$$\mathcal{P} \cdot (A \times (-)) \cdot \mathcal{D} \qquad \mathcal{P} \cdot \mathcal{D} \cdot (A \times (-)) \qquad \ldots$$

References

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

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

David Gries. “Describing an algorithm by Hopcroft”. In: **Acta Inf.** 2 (1973), pp. 97–109. ISSN: 1432-0525.

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.

Barbara König, Sebastian Küpper. “Generic Partition Refinement Algorithms for Coalgebras and an Instantiation to Weighted Automata”. In: **Theoretical Computer Science, IFIP TCS 2014**. Vol. 8705. LNCS. Springer, 2014, pp. 311–325. ISBN: 978-3-662-44601-0.

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

Robert Paige, Pobert Tarjan. “Three partition refinement algorithms”. In: **SIAM J. Comput.** 16.6 (1987), pp. 973–989.

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.

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.