Efficient and Modular Coalgebraic Partition Refinement

Thorsten Wißmann

University Erlangen-Nürnberg

Joint work with:

Hans-Peter Deifel, Ulrich Dorsch, Stefan Milius, Lutz Schröder

- Published in Concur 2017
- Submitted To LMCS
- Ongoing Implementation

Highlights of Logic, Games and Automata

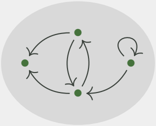
September 19, 2018

Efficient and Modular Coalgebraic Partition Refinement

Efficient and Modular Coalgebraic Partition Refinement

Coalgebras:

State based systems



Labels, Non-Determinism, Probabilities, Automata, ...

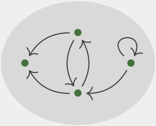
Efficient and Modular Coalgebraic Partition Refinement

Modularity:

Combine system types by $\circ$, $\times$, $+$

Coalgebras:

State based systems



Labels, Non-Determinism, Probabilities, Automata, ...

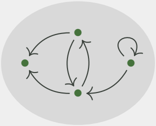
Efficient and Modular Coalgebraic Partition Refinement

Modularity:

Combine system types by ∘, ×, +

Coalgebras:

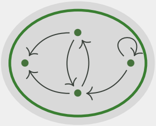
State based systems



Labels, Non-Determinism, Probabilities, Automata, …

Partition Refinement:

Successively distinguish different behaviour



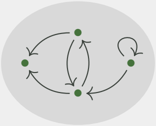
Efficient and Modular Coalgebraic Partition Refinement

Modularity:

Combine system types by $\circ$, $\times$, $+$

Coalgebras:

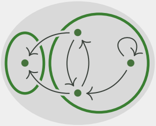
State based systems



Labels, Non-Determinism, Probabilities, Automata, ...

Partition Refinement:

Successively distinguish different behaviour



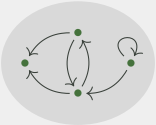
Efficient and Modular Coalgebraic Partition Refinement

Modularity:

Combine system types by $\circ$, $\times$, $+$

Coalgebras:

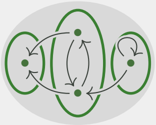
State based systems



Labels, Non-Determinism, Probabilities, Automata, ...

Partition Refinement:

Successively distinguish different behaviour



Efficient and Modular Coalgebraic Partition Refinement

Efficiency:

(a) Incrementally compute partitions

(b) Complexity Analysis

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

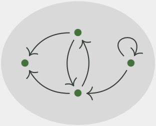
Edges States

Modularity:

Combine system types by $\circ$, $\times$, $+$

Coalgebras:

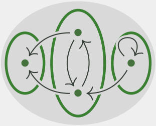
State based systems



Labels, Non-Determinism, Probabilities, Automata, ...

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

$n \cdot \log n$ Hopcroft '71

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

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

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

Efficient & Modular 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

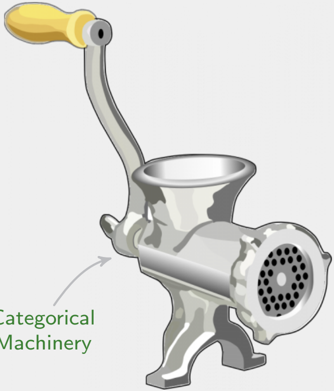
Weighted Systems "Markov Chain Lumping"

$m \cdot \log n$ — Valmari, Franceschinis '10

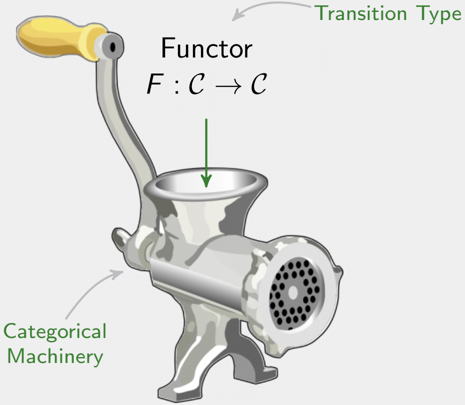
Labelled Transition Systems

$m \cdot \log n$ — Valmari '09

Coalgebra – Generic state based systems



Coalgebra – Generic state based systems

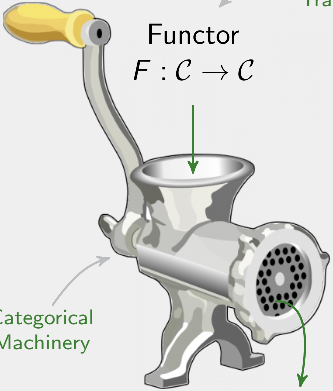


Coalgebra – Generic state based systems

Transition Type

Functor

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



Categorical Machinery

F-Coalgebra

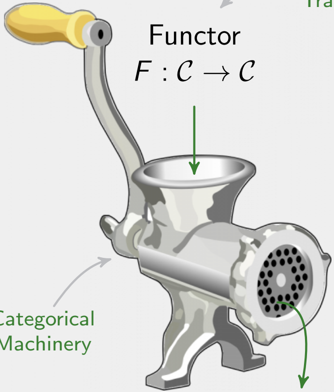
$C \xrightarrow{c} FC$

Coalgebra – Generic state based systems

Powerset $\mathcal{P}$

Transition Type

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

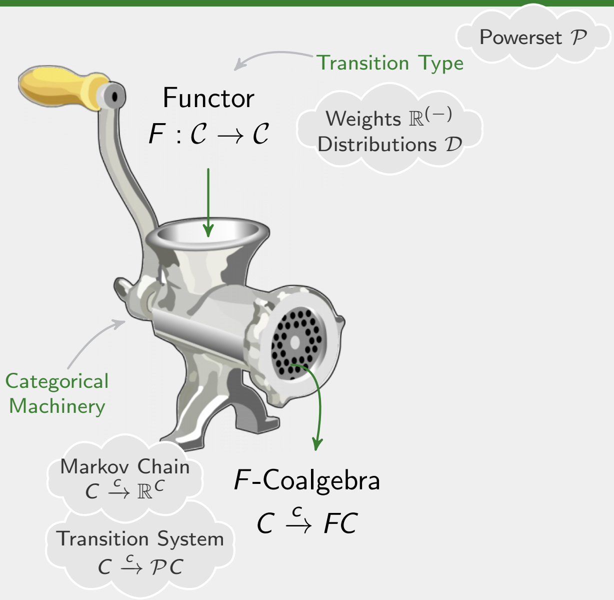


Categorical Machinery

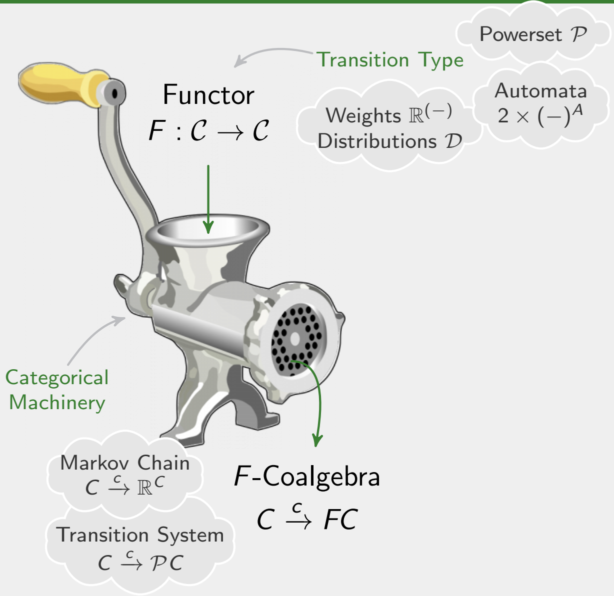
F-Coalgebra
$C \xrightarrow{c} FC$

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

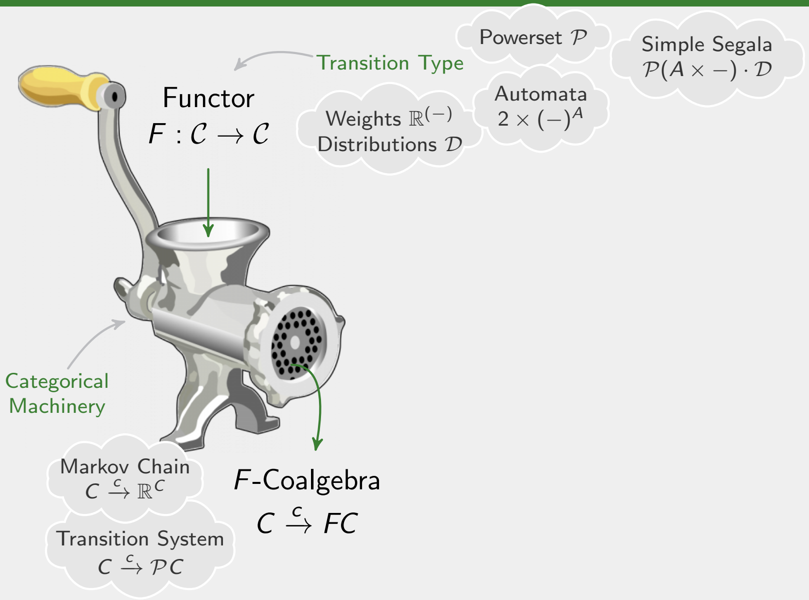
Coalgebra – Generic state based systems



Coalgebra – Generic state based systems



Coalgebra – Generic state based systems



Coalgebra – Generic state based systems

Transition Type

Functor

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

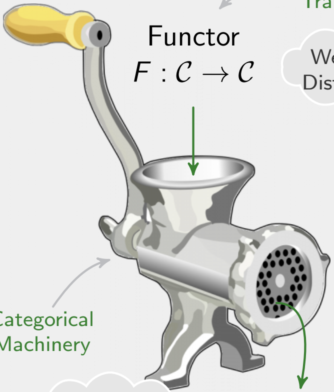
Powerset $\mathcal{P}$

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

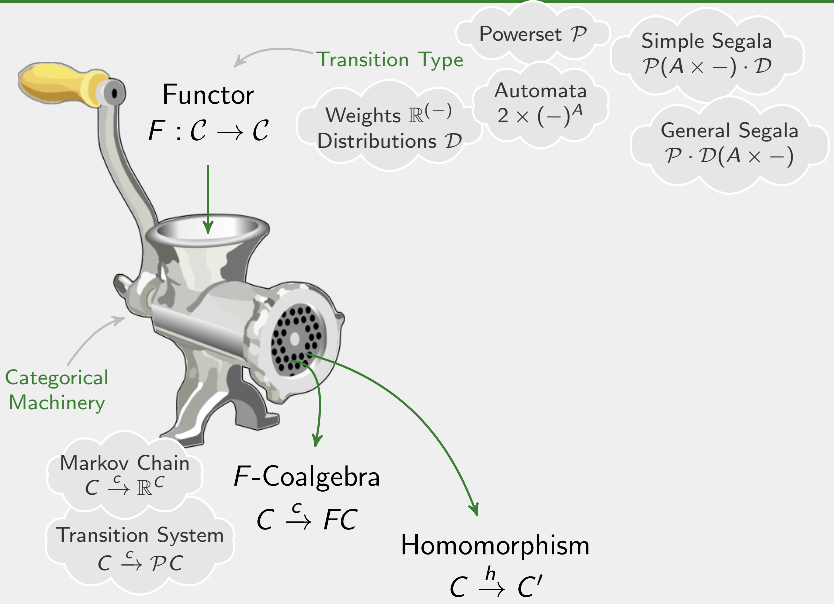
Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

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

F-Coalgebra

$C \xrightarrow{c} FC$

Coalgebra – Generic state based systems



Coalgebra – Generic state based systems

Powerset $\mathcal{P}$

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

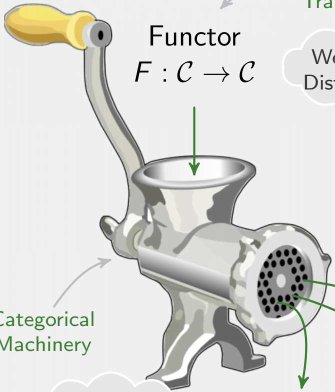
Transition Type

Automata $2 \times (-)^A$

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

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

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



Categorical Machinery

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

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

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

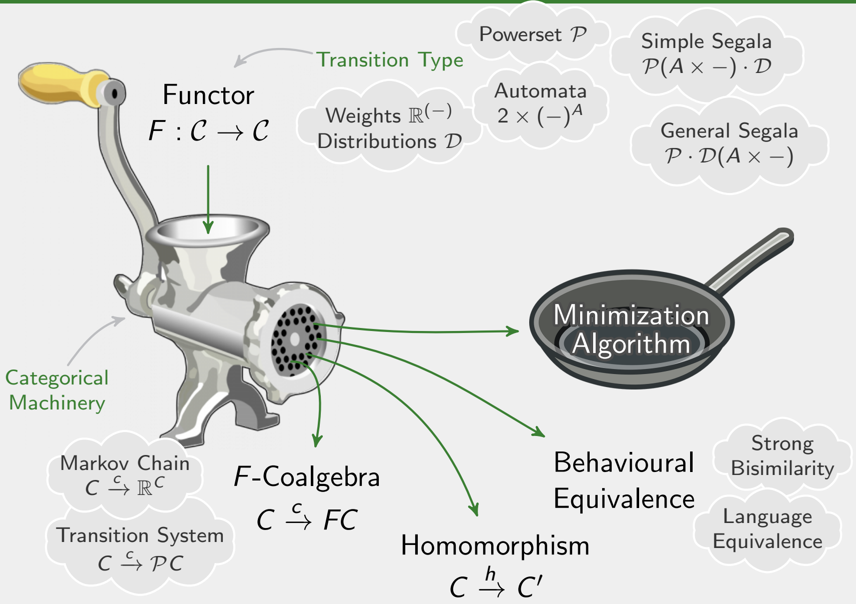
Homomorphism $C \xrightarrow{h} C'$

Behavioural Equivalence

Strong Bisimilarity

Language Equivalence

Coalgebra – Generic state based systems



Construction on a category $\mathcal{C}$

Construction on a category $\mathcal{C}$

Efficiency

$F\colon \mathsf{Set} \to \mathsf{Set}$ is zippable
&
F has a refinement interface

$\Longrightarrow$ (Pseudo-Code)

Minimization runs in $\mathcal{O}((m+n)\cdot \log n)$

(m: Edges, n: States)

Construction on a category $\mathcal{C}$

Functor F

F-coalgebra

→ Coalgebraic Partition Refinement → Minimized Coalgebra

Efficiency

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

&

F has a refinement interface

$\Longrightarrow$ (Pseudo-Code)

Minimization runs in $\mathcal{O}((m+n)\cdot \log n)$

(m: Edges, n: States)

Modularity

$$F, H ::= F \cdot H \mid F \times H \mid F + H \mid C \mid \mathcal{P} \mid \mathcal{B}_{\mathrm{f}} \mid G^{(-)}$$

(C: Constant Set, G: Group)

Algorithm on Sets

Functor Term

F

F-coalgebra

Coalgebraic Partition Refinement

Haskell Implementation

Minimized Coalgebra

Efficiency

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

&

F has a refinement interface

$\Longrightarrow$ (Pseudo-Code)

Minimization runs in $\mathcal{O}((m+n)\cdot \log n)$

Edges (m), States (n)

Modularity

$$F, H ::= F\cdot H \mid F\times H \mid F+H \mid C \mid \mathcal{P} \mid \mathcal{B}_{\mathrm{f}} \mid G^{(-)}$$

Constant Set (C), Group (G)

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	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m_{\mathcal{P}}\cdot n\cdot\log(m_{\mathcal{P}}\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$\geq$	$(m_{\mathcal{P}}+m_{\mathcal{D}}+n)$ $\cdot\ \log(m_{\mathcal{P}}+n)$

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	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m_{\mathcal{P}}\cdot n\cdot\log(m_{\mathcal{P}}\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$\geq$	$(m_{\mathcal{P}}+m_{\mathcal{D}}+n)$ $\cdot\ \log(m_{\mathcal{P}}+n)$

Generic & Efficient

System	Functor	[illegible]…hm		Our instantiation
Transition Systems	$\mathcal{P}$	[illegible]	$=$	$(m+n)\cdot\log n$
LTS	$\mathcal{P}(A\times -)$	[illegible] $n)$ Dovi[illegible]azza, Po[illegible]criti '04 $(m+$[illegible]$\log n$ [illegible]lmari '0[illegible]	$=$	$(m$[illegible]$n)\cdot\log(m+n)$
Markov Chains	$\mathbb{R}^{(-)}$	$(m+$[illegible]$\log n$ Valm[illegible]i, France[illegible]hinis '10	$=$	$(m+n)\cdot\log n$
DFA	$2\times(-$[illegible]	$n\cdot\log$[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	$\approx$	$\lvert A\rvert\cdot n\cdot\log n$ $+\lvert A\rvert\cdot n\cdot\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m_{\mathcal{P}}\cdot n\cdot\log(m_{\mathcal{P}}\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$\geq$	$(m_{\mathcal{P}}+m_{\mathcal{D}}+n)$ $\cdot\log(m_{\mathcal{P}}+n)$

More instances:
further system types
& categories

Generic & Efficient

System	Functor	[illegible]		Our instantiation
Transition Systems	$\mathcal{P}$	[illegible]	$=$	$(m+n)\cdot\log n$
LTS	[illegible]	[illegible], Dovier, Piazza, Policriti '04; $(m+n)\cdot\log n$, Valmari '0[illegible]	$=$	$(m+n)\cdot\log(m+n)$
Markov Chains	[illegible]	$(m+n)\cdot\log n$, 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	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m_{\mathcal{P}}\cdot n\cdot\log(m_{\mathcal{P}}\cdot n)$, Baier, Engelen, Majster-Cederbaum'00	$\geq$	$(m_{\mathcal{P}}+m_{\mathcal{D}}+n)\cdot\log(m_{\mathcal{P}}+n)$

More instances: further system types & categories

Compare to existing concrete implementations

Generic & Efficient

System	Functor	…hm		Our instantiation
Transition Systems	$\mathcal{P}$		=	$(m+n)\cdot\log n$
LTS		Dovi…, …iazza, Po…criti '04 $(m+n)\cdot\log n$ Valmari '0…	=	$(m+n)\cdot\log(m+n)$
Mar… Chains		$(m+n)\cdot\log n$ Valm…, France…hinis '10	=	$(m+n)\cdot\log n$
DFA	$2\times(-)$…	$n\cdot\log$… for fix… Hopcrof…		…$\log n$
	$2\times\mathcal{P}(A\times$…	$\lvert A\rvert\cdot n$… Gries '73, K…		…$\log n$ …$\log\lvert A\rvert$
Segala Systems	$\mathcal{P}(A\times-)\cdot\mathcal{D}$	$m_{\mathcal{P}}\cdot n\cdot\log(m$… Baier, Engelen, Majster-Cederbaum'00		…$+m_{\mathcal{D}}+n)$ $\cdot\log(m_{\mathcal{P}}+n)$

More instances: further system types & categories

Compare to existing concrete implementations

Generic & Efficient

See poster and tinyurl.com/coalgebra

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	$\mathcal{P}(A\times -)\cdot\mathcal{D}$	$m_{\mathcal{P}}\cdot n\cdot\log(m_{\mathcal{P}}\cdot n)$ Baier, Engelen, Majster-Cederbaum'00	$\geq$	$(m_{\mathcal{P}}+m_{\mathcal{D}}+n)$ $\cdot\ \log(m_{\mathcal{P}}+n)$

Appendix ...

The Coalgebraic Task

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

Given a coalgebra $C \xrightarrow{c} FC$

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

find the simple quotient $C' \xrightarrow{c'} FC'$

no proper quotient

all equivalent behaviours identified

The Coalgebraic Task

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

Given a coalgebra $C \xrightarrow{c} FC$

find the simple quotient $C' \xrightarrow{c'} FC'$

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

no proper quotient

all equivalent behaviours identified

Instance

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

Automata minimization

The Coalgebraic Task

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

Given a coalgebra $C \xrightarrow{c} FC$

find the simple quotient $C' \xrightarrow{c'} FC'$

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

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 $C \xrightarrow{c} FC$

find the simple quotient $C' \xrightarrow{c'} FC'$

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

no proper quotient

all equivalent behaviours identified

Instance

Instance

Instance

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

Given a coalgebra $C \xrightarrow{c} FC$

find the simple quotient $C' \xrightarrow{c'} FC'$

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

simple: no proper quotient

quotient: all equivalent behaviours identified

Instance →

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

Automata minimization

For $\mathcal{P}$:Set

Bisimilarity minimization

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

Markov chain lumping

. . .

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

$$a_1\, a_2\, b_1\, a_3\, b_2 \overset{\text{unzip}}{\longmapsto} (a_1 a_2 \,\_\, a_3 \,\_, \;\; \_\,\_\, b_1 \,\_\, b_2)$$

$(-)^*$ is zippable

$$\{a_1, a_2, b_1\} \overset{\text{unzip}}{\longmapsto} (\{a_1, a_2, \_\}, \;\; \{\_, b_1\})$$

$\mathcal{P}$ is 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 $L = \{a_1, a_2, a_3\}$, $R = \{b_1, b_2\}$, $1 = \{\_\}$

$$a_1\, a_2\, b_1\, a_3\, b_2 \overset{\text{unzip}}{\longmapsto} (a_1 a_2\, \_\, a_3\, \_,\ \_\,\_\, b_1\, \_\, b_2)$$

$(-)^*$ is zippable

$$\{a_1, a_2, b_1\} \overset{\text{unzip}}{\longmapsto} (\{a_1, a_2, \_\},\ \{\_, b_1\})$$

$\mathcal{P}$ is zippable

$$\{\{a_1, b_1\}, \{a_2, b_2\}\} \overset{\text{unzip}}{\longmapsto} (\{\{a_1, \_\}, \{a_2, \_\}\},\ \{\{\_, b_1\}, \{\_, b_2\}\}) \overset{\text{unzip}}{\longmapsfrom} \{\{a_1, b_2\}, \{a_2, b_1\}\}$$

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

~~Composition~~

~~Quotients~~

Functor encoding

- internal weights W, $w : FX \to \mathcal{P}X \to W$
- edge labels L
- $\flat : FX \to \mathcal{B}_\mathrm{f}(L \times X)$
- update : $\mathcal{B}_\mathrm{f}(L) \times W \longrightarrow W \times F(2 \times 2) \times W$

edges into S — weight of C — weight of S — $F\langle \chi_S, \chi_C \rangle$ — 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:	$G^{(2)}$	$\mathcal{B}_\mathrm{f}2$	$\mathcal{D}2$	$\mathbb{N}$	$F_\Sigma 2$
$w(C)$, $C \subseteq Y$:	$G\chi_C$	$\mathcal{B}_\mathrm{f}\chi_C$	$\mathcal{D}\chi_C$	$\lvert C \cap (-) \rvert$	$F_\Sigma \chi_C$

1. Assume everything equivalent

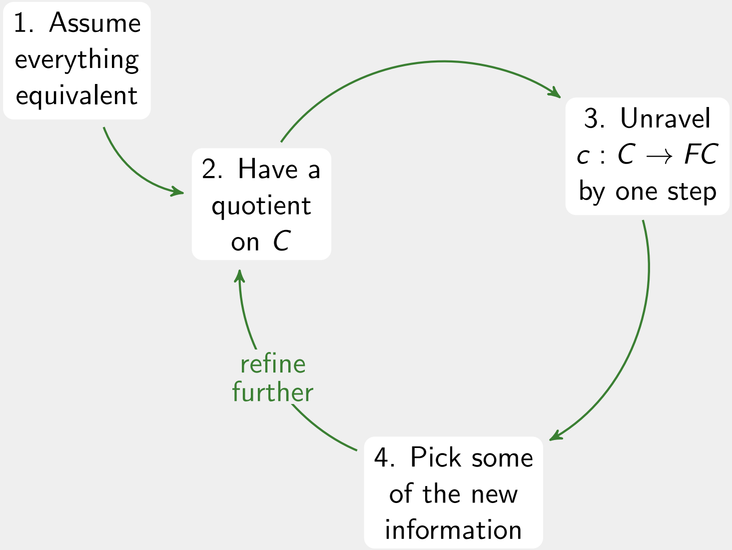
1. Assume everything equivalent

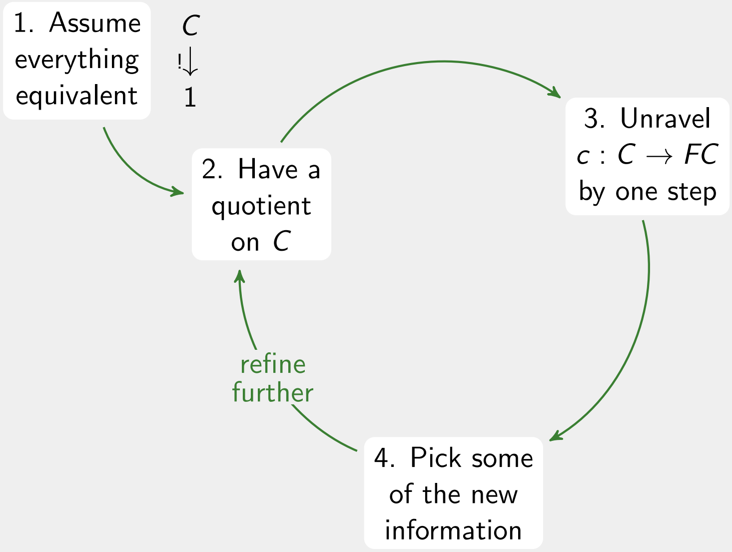
2. Have a quotient on C

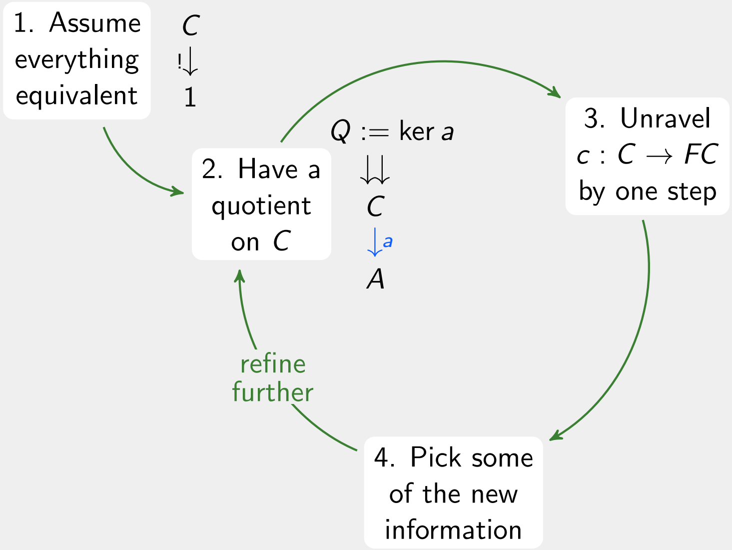
1. Assume everything equivalent

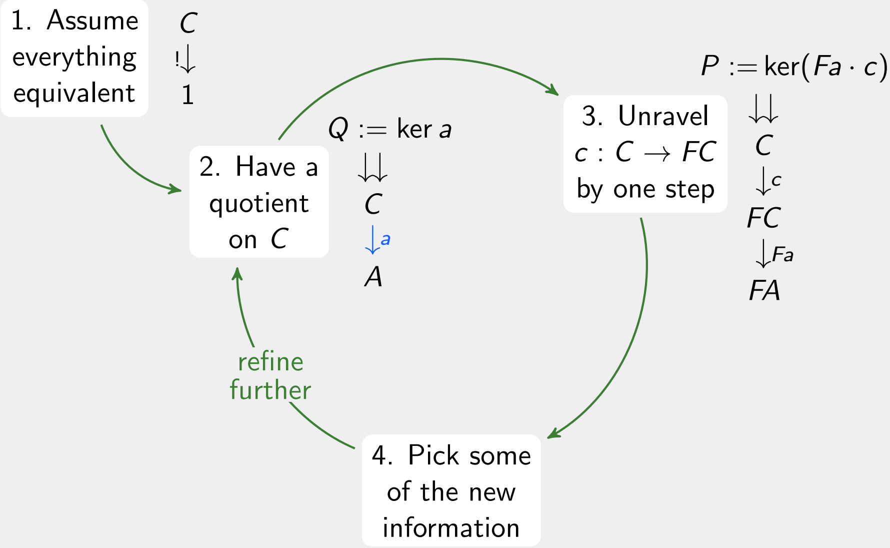
2. Have a quotient on C

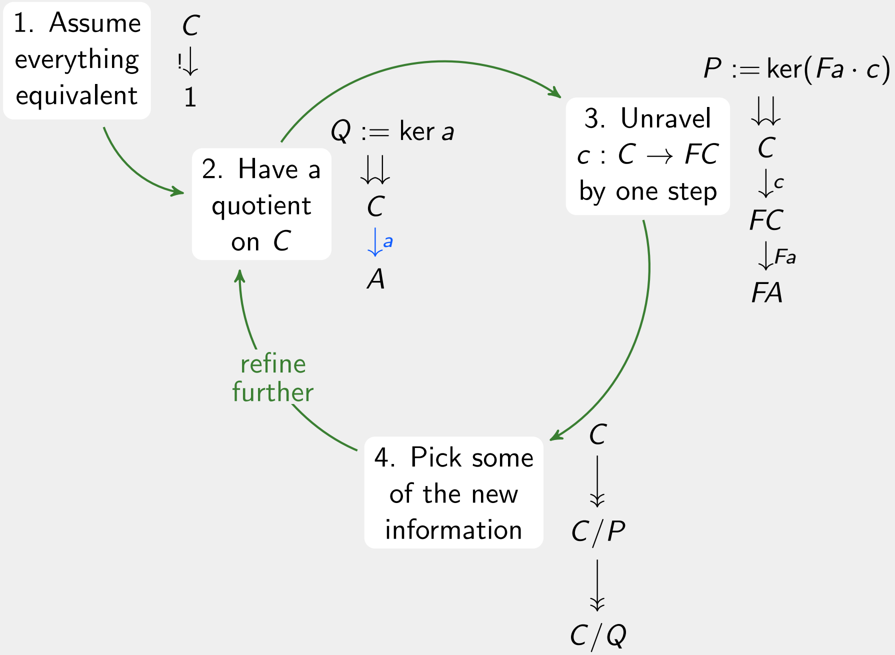
3. Unravel $c : C \to FC$ by one step

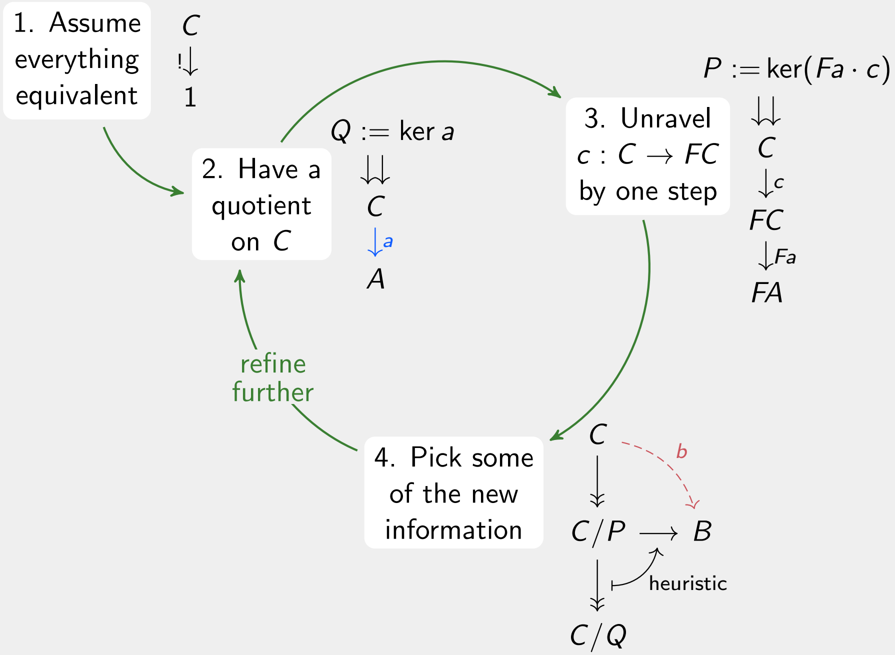

1. Assume everything equivalent
2. Have a quotient on C
3. Unravel $c : C \to FC$ by one step
4. Pick some of the new information
refine further

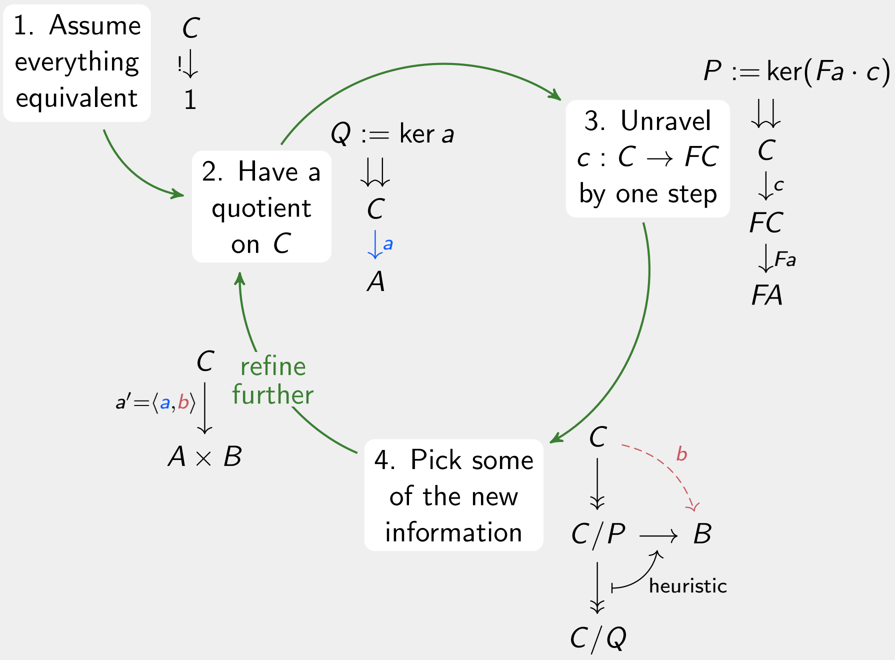

1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
3. Unravel $c : C \to FC$ by one step
4. Pick some of the new information
refine further

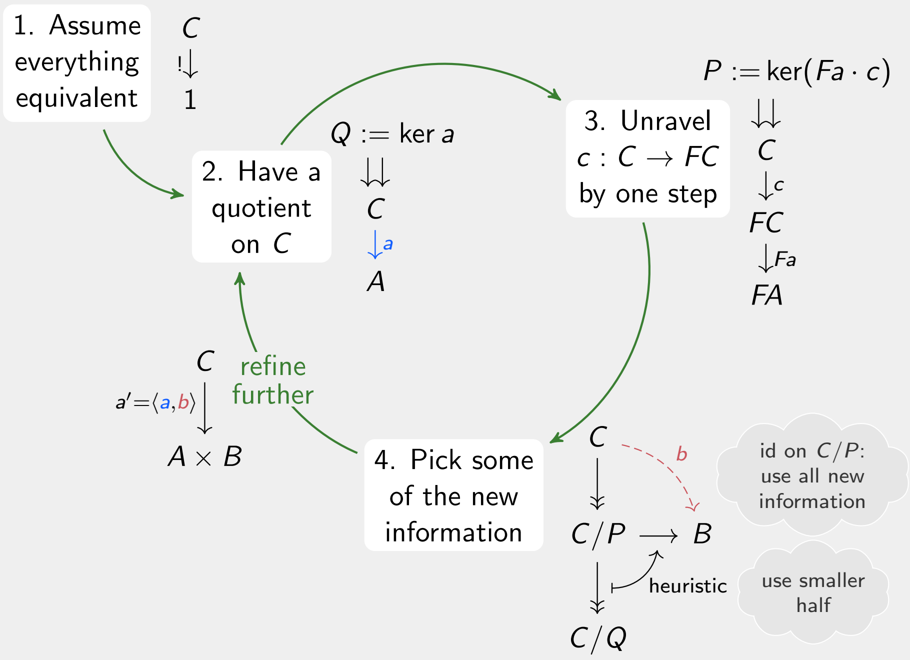

1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
$Q := \ker a \rightrightarrows C \xrightarrow{a} A$
3. Unravel $c : C \to FC$ by one step
4. Pick some of the new information
refine further


1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
$Q := \ker a \rightrightarrows C \xrightarrow{a} A$
3. Unravel $c : C \to FC$ by one step
$P := \ker(Fa \cdot c) \rightrightarrows C \xrightarrow{c} FC \xrightarrow{Fa} FA$
4. Pick some of the new information
refine further


1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
$Q := \ker a$
$C \xrightarrow{a} A$
3. Unravel $c : C \to FC$ by one step
$P := \ker(Fa \cdot c)$
$C \xrightarrow{c} FC \xrightarrow{Fa} FA$
4. Pick some of the new information
$C \twoheadrightarrow C/P \twoheadrightarrow C/Q$
refine further


1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
$Q := \ker a$
$C \xrightarrow{a} A$
3. Unravel $c : C \to FC$ by one step
$P := \ker(Fa \cdot c)$
$C \xrightarrow{c} FC \xrightarrow{Fa} FA$
4. Pick some of the new information
C
$C/P \longrightarrow B$
b
heuristic
C/Q
refine further


1. Assume everything equivalent
C
!
1
2. Have a quotient on C
Q := ker a
C
a
A
3. Unravel c : C → FC by one step
P := ker(Fa · c)
C
c
FC
Fa
FA
4. Pick some of the new information
C
b
C/P
B
heuristic
C/Q
refine further
C
a′=⟨a,b⟩
A × B


1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
$Q := \ker a$
$C \xrightarrow{a} A$
3. Unravel $c : C \to FC$ by one step
$P := \ker(Fa \cdot c)$
$C \xrightarrow{c} FC \xrightarrow{Fa} FA$
4. Pick some of the new information
$C \to C/P \to C/Q$
$C/P \to B$
b
heuristic
id on C/P: use all new information
use smaller half
refine further
$a' = \langle a, b \rangle$
$C \to A \times B$

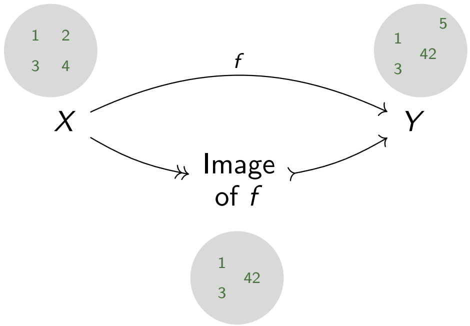
Factorizations

Equivalence Relations	$\cong$	Quotients $\cong$ Partitions
Kernels	$\cong$	Regular Epimorphisms

Factorizations

Equivalence Relations	$\cong$	Quotients $\cong$ Partitions
Kernels	$\cong$	Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations

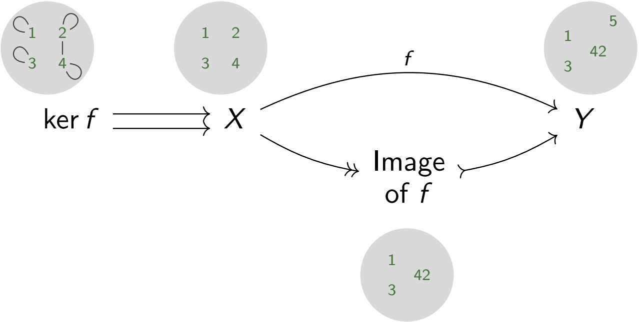


Factorizations

Equivalence Relations ≅ Quotients ≅ Partitions

Kernels ≅ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations



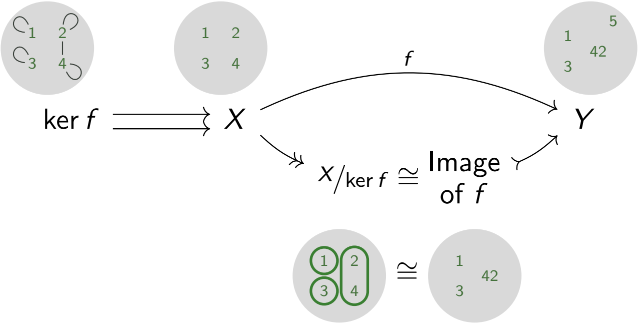
$\ker f = \{(x_1, x_2) \in X^2 \mid f(x_1) = f(x_2)\}$

Factorizations

Equivalence Relations ≅ Quotients ≅ Partitions

Kernels ≅ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations



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

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,

$$C \xrightarrow{c} FG\,C, \quad C \overset{c'}{\dashrightarrow} FD, \quad FD \xrightarrow{Fd} FG\,C \qquad \rightsquigarrow \qquad D \overset{d}{\hookrightarrow} GC$$

Genericity: Composition

If F finitary,

$$\begin{array}{ccc} C & \xrightarrow{\;c\;} & FG\,C \\ & \overset{c'}{\searrow} & \uparrow Fd \\ & & FD \end{array} \qquad \rightsquigarrow \qquad D \overset{d}{\hookrightarrow} GC$$

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,

$$\begin{array}{ccc} C & \xrightarrow{c} & FG\,C \\ & \overset{c'}{\searrow} & \uparrow Fd \\ & & FD \end{array} \qquad \rightsquigarrow \qquad D \overset{d}{\hookrightarrow} GC$$

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

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

Examples

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

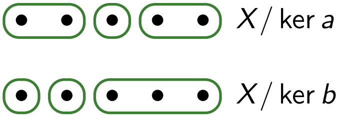
$$A \xleftarrow{a} X \xrightarrow{b} B$$

$\ker a \cup \ker b$ a kernel in Set

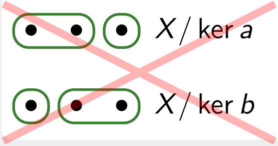
$\Leftrightarrow$ $\ker a \cup \ker b$ transitive

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

Example



Non-Example



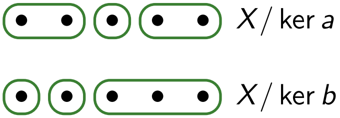
$$A \xleftarrow{a} X \xrightarrow{b} B$$

$\ker a \cup \ker b$ a kernel in Set

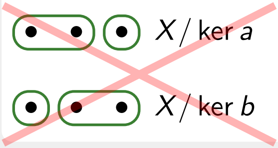
$\Leftrightarrow$ $\ker a \cup \ker b$ transitive

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

Example



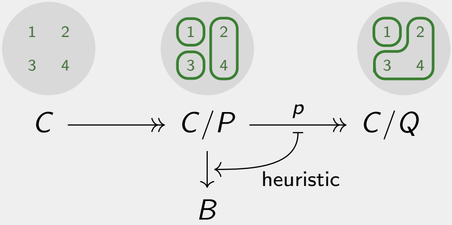
Non-Example



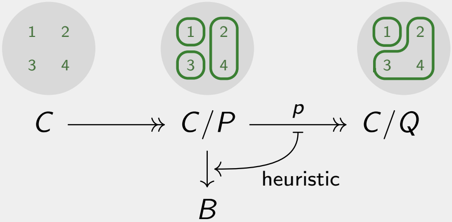
Process smaller half for $X \overset{f}{\twoheadrightarrow} F \overset{g}{\twoheadrightarrow} G$

Find $x \in X$, with $S := [x]_f$, $C := [x]_{gf}$, such that $2 \cdot |S| \leq |C|$.
Return $\langle \chi_S, \chi_C \rangle : X \to 2 \times 2$

Heuristic



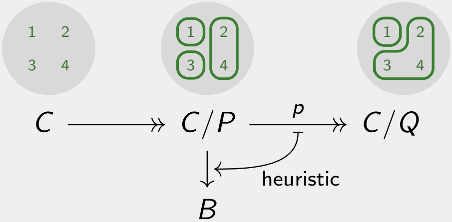
Heuristic



Use all new information

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

Heuristic



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 $2 \cdot |S| \leq |p(S)|$

$$B = \{\underbrace{\text{ChosenBlock}}^{\{3\}}, \overbrace{\text{SameSurroundingBlock}}^{\{2,4\}}, \overbrace{\text{RemainingBlocks}}^{\{1\}}\}$$

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.