Efficient Coalgebraic Partition Refinement

Thorsten Wißmann

Joint work with:
Ulrich Dorsch, Stefan Milius, Lutz Schröder

Friedrich-Alexander-Universität Erlangen-Nürnberg

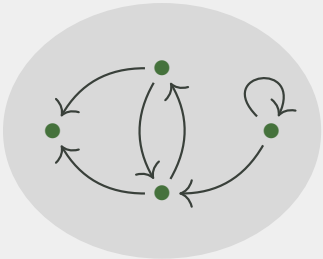
CONCUR 2017
September 7, 2017

Efficient Coalgebraic Partition Refinement

Efficient Coalgebraic Partition Refinement

1. Coalgebras:

State based systems

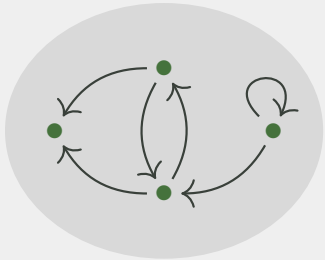


Labels, Non-Determinism,
Probabilities, Automata,
... and their combinations!

Efficient Coalgebraic Partition Refinement

1. Coalgebras:

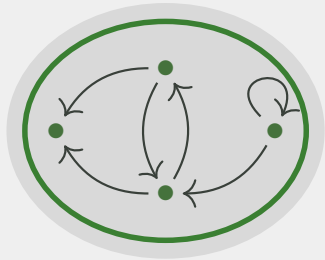
State based systems



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

2. Partition Refinement:

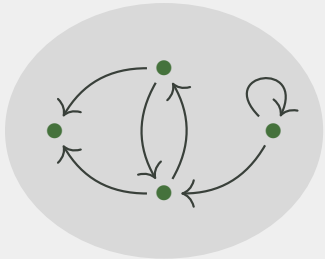
Successively distinguish different behaviour



Efficient Coalgebraic Partition Refinement

1. Coalgebras:

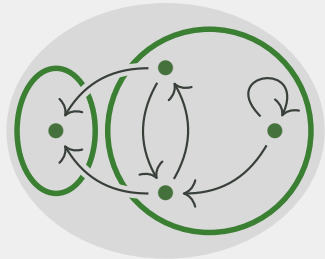
State based systems



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

2. Partition Refinement:

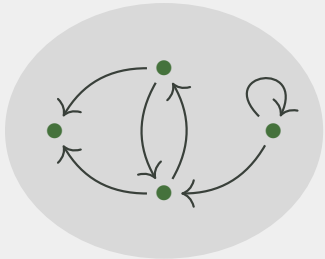
Successively distinguish different behaviour



Efficient Coalgebraic Partition Refinement

1. Coalgebras:

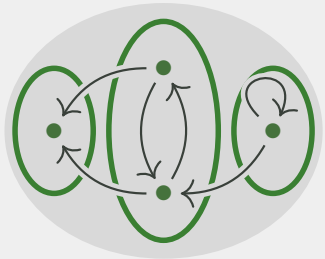
State based systems



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

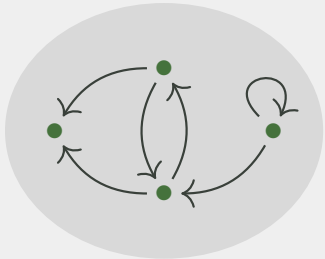
(b) Complexity Analysis

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

Edges States

1. Coalgebras:

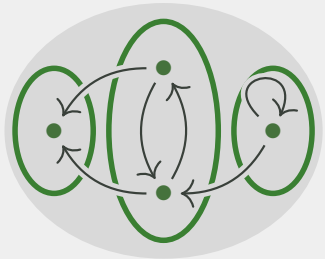
State based systems



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

$n \cdot \log n$ 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"

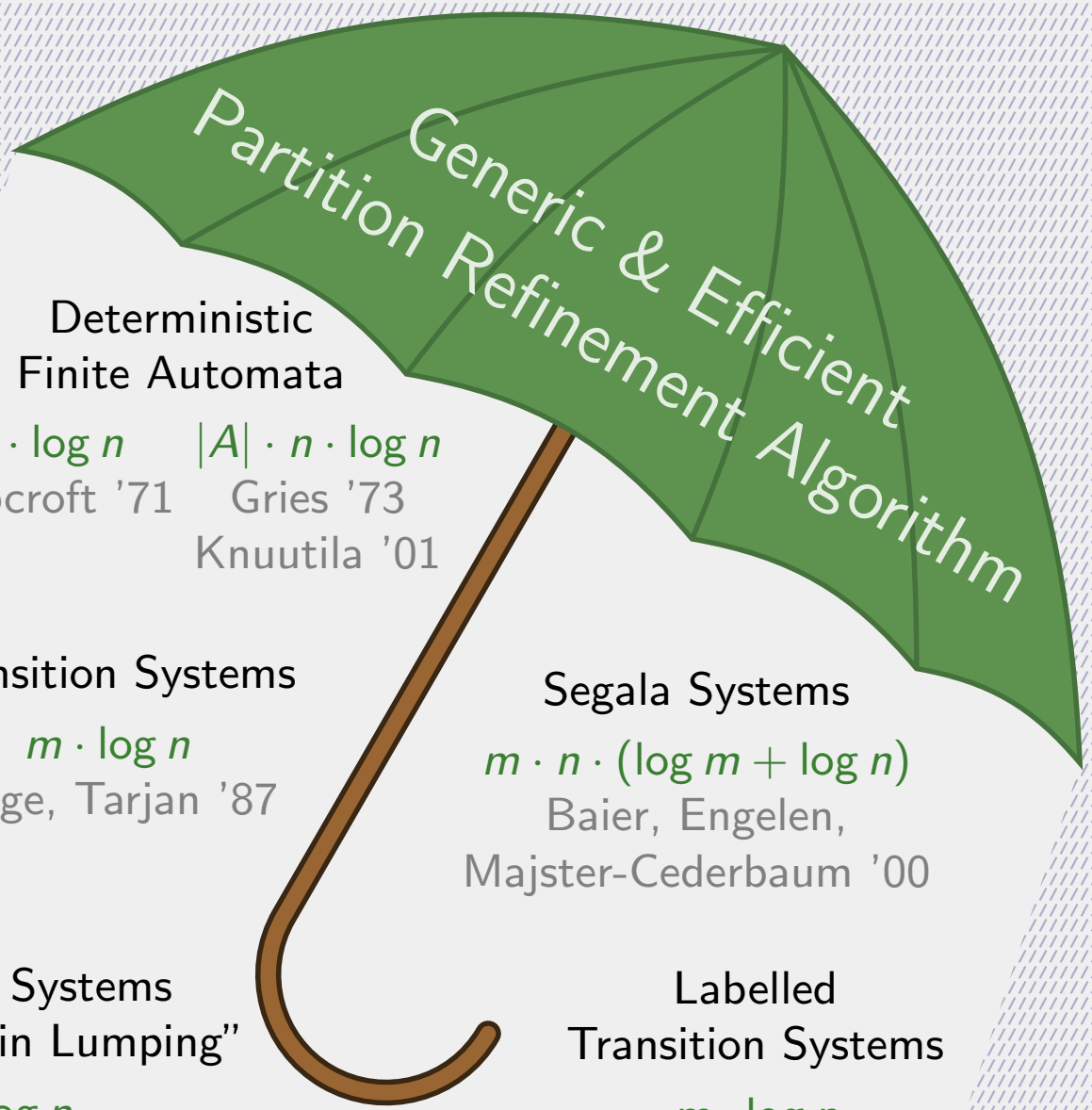
$m \cdot \log n$

Valmari, Franceschinis '10

Labelled Transition Systems

$m \cdot \log n$

Valmari '09



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

$$\text{Equivalence Relations} \cong \text{Quotients} \cong \text{Partitions}$$

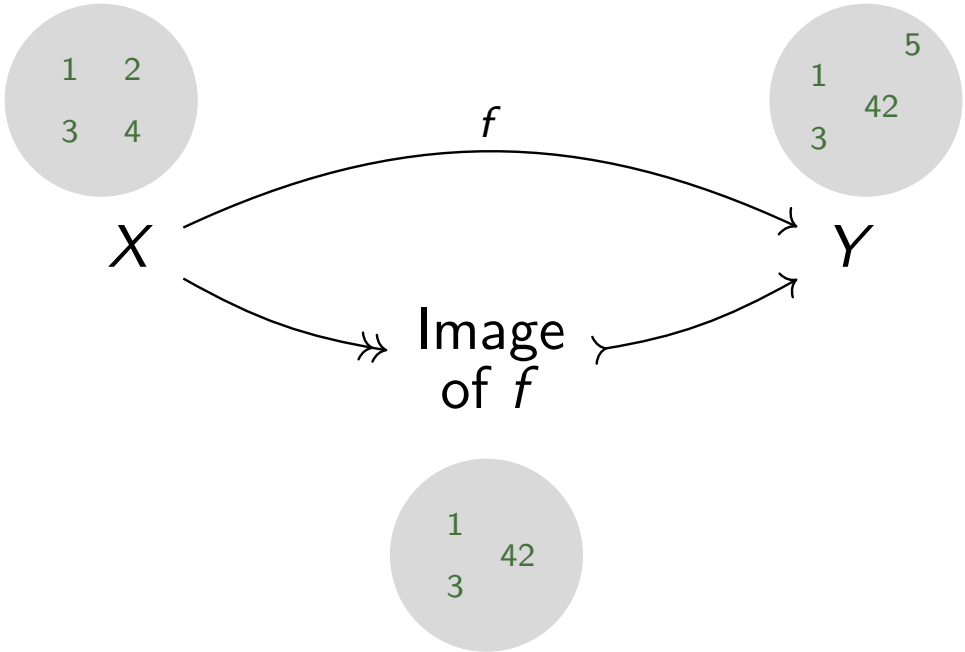
$$\text{Kernels} \cong \text{Regular Epimorphisms}$$

Ingredient 1: Factorizations

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

Kernels $\cong$ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations

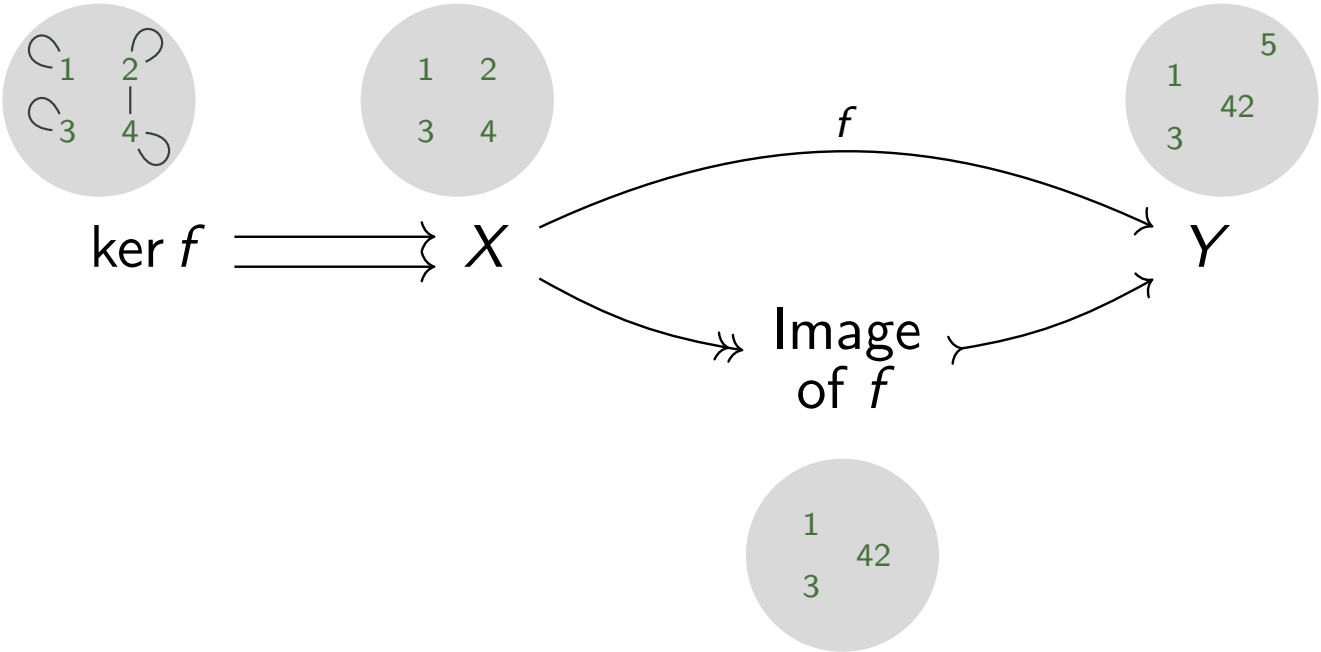


Ingredient 1: Factorizations

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

Kernels $\cong$ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations



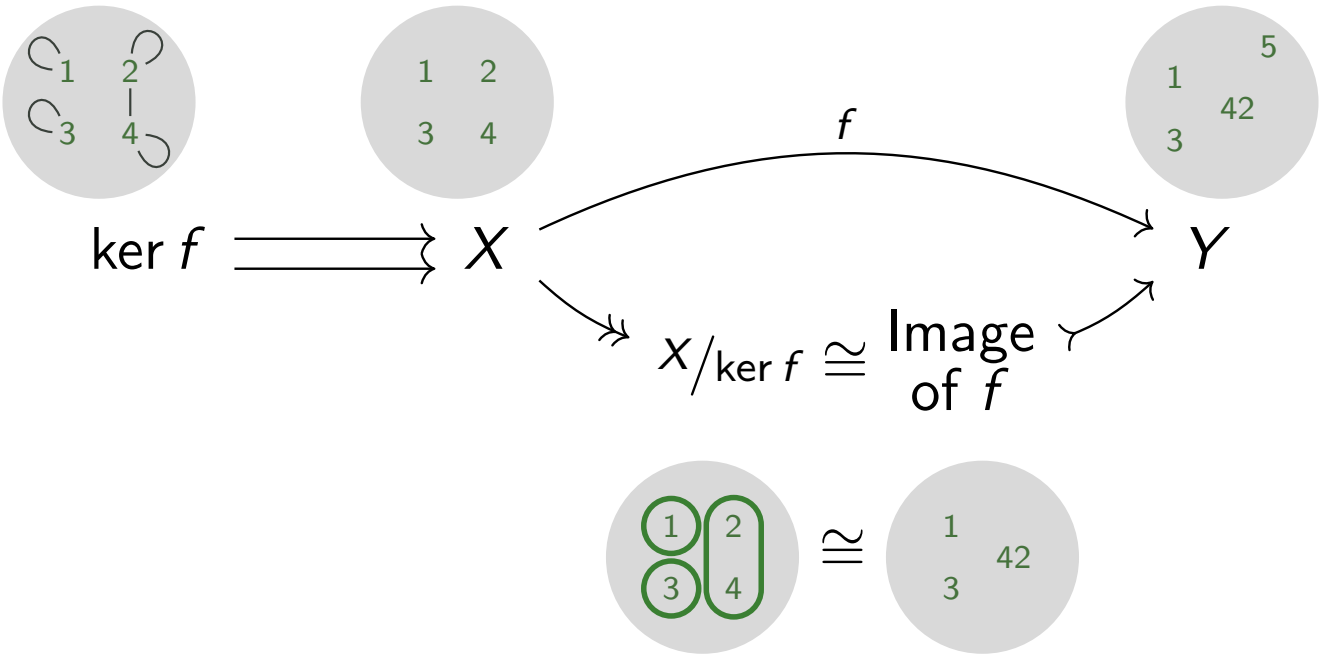
$$\ker f = \{(x_1, x_2) \in X^2 \mid f(x_1) = f(x_2)\}$$

Ingredient 1: Factorizations

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

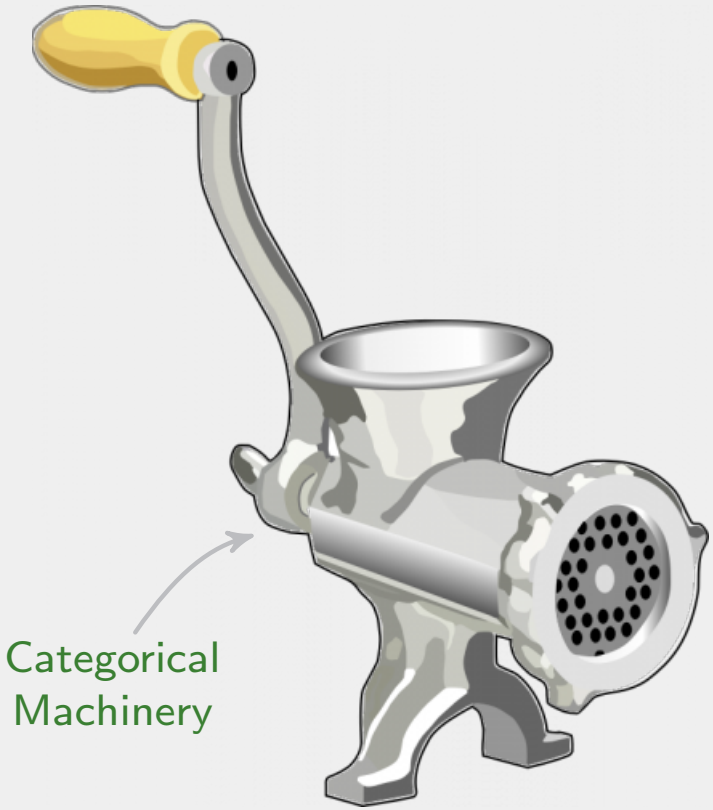
Kernels $\cong$ Regular Epimorphisms

Category with (Regular Epi,Mono)-Factorizations

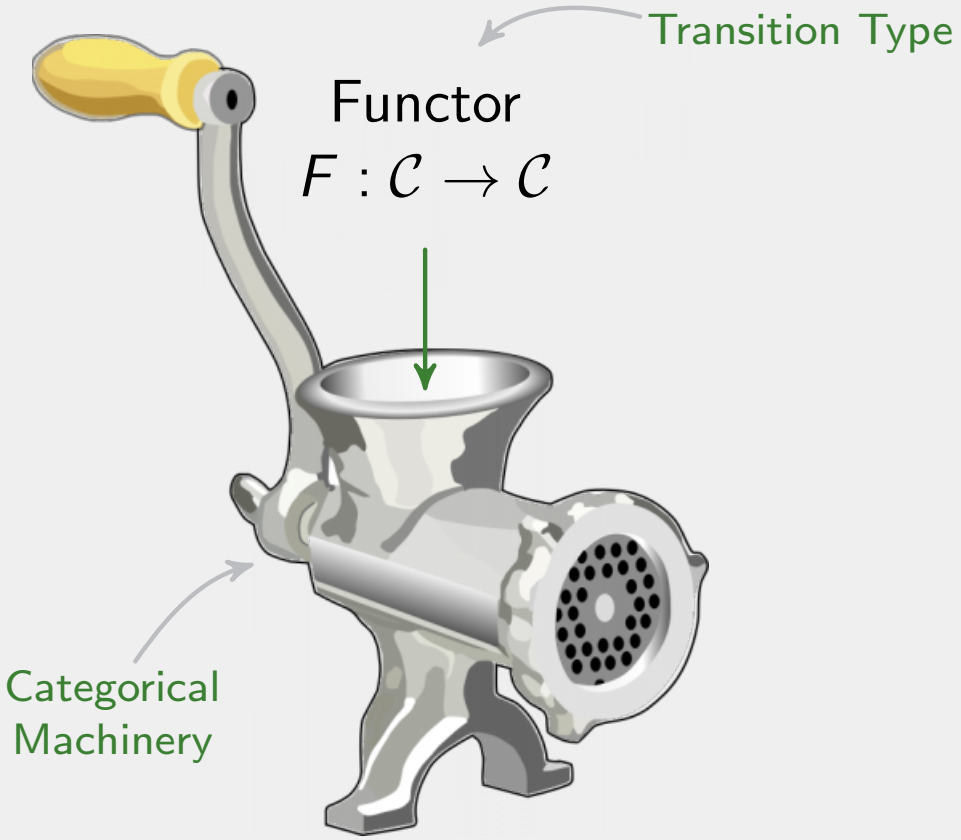


$$\ker f = \{(x_1, x_2) \in X^2 \mid f(x_1) = f(x_2)\}$$

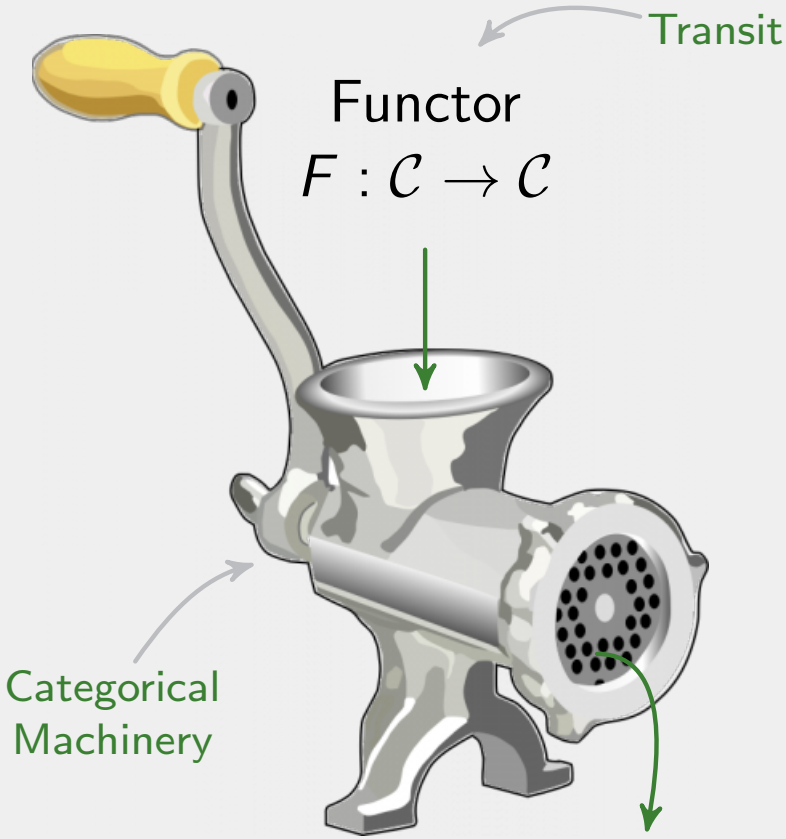
Ingredient 2: Coalgebra – Generic state based systems



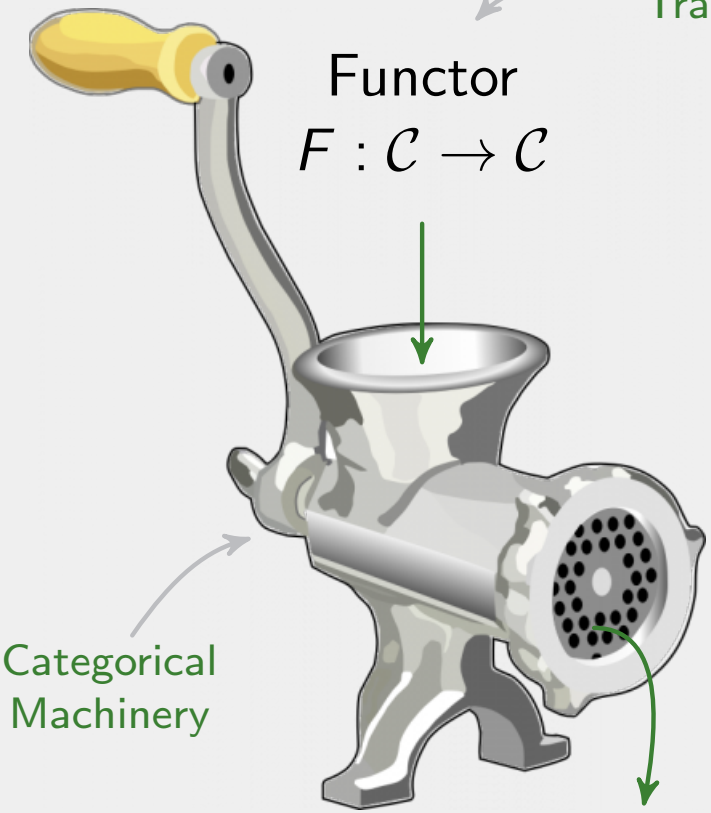
Ingredient 2: Coalgebra – Generic state based systems



Ingredient 2: Coalgebra – Generic state based systems



Ingredient 2: Coalgebra – Generic state based systems



Powerset $\mathcal{P}$

Transition Type

Functor

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

Categorical Machinery

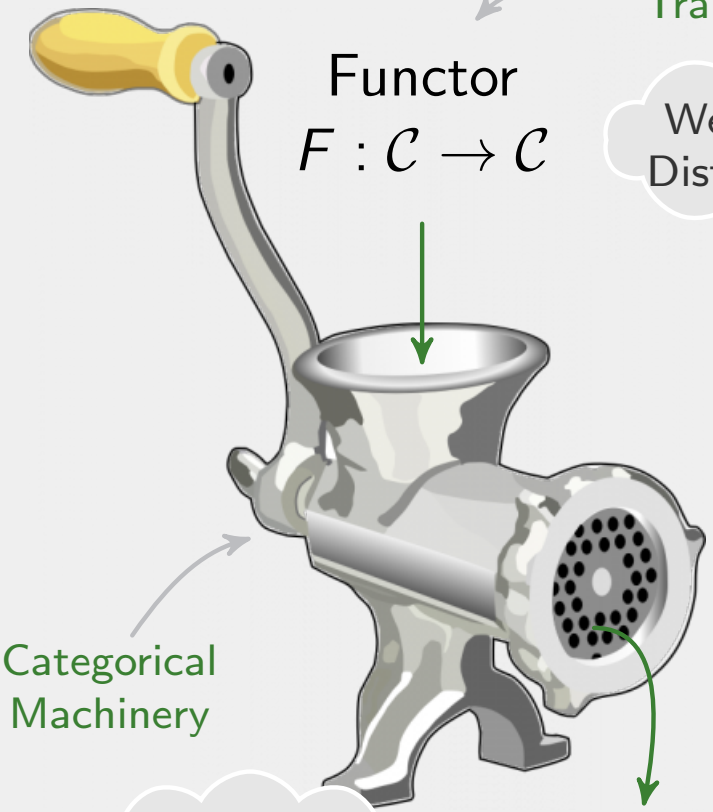
F-Coalgebra

$C \xrightarrow{c} FC$

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 $\mathcal{D}$

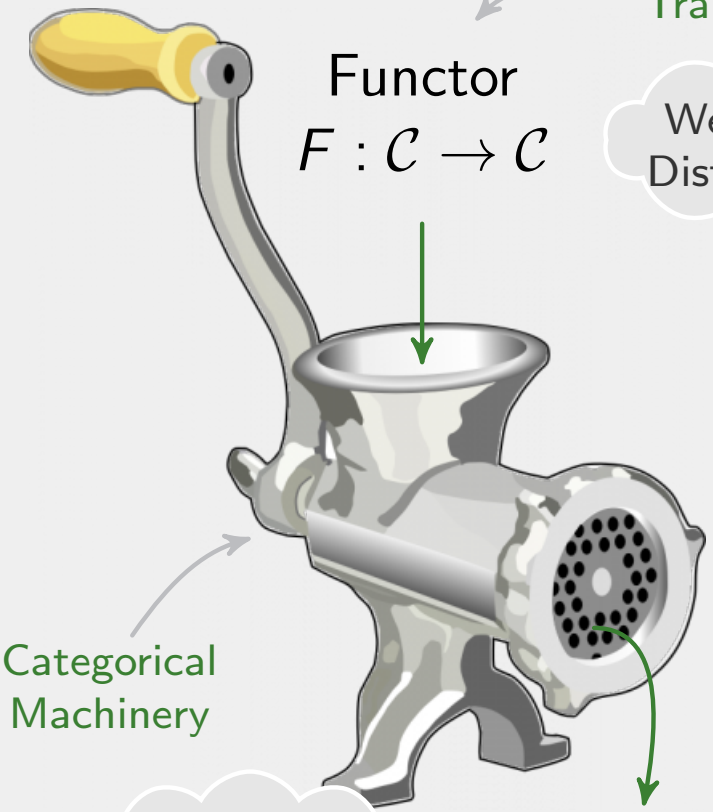
Categorical Machinery

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

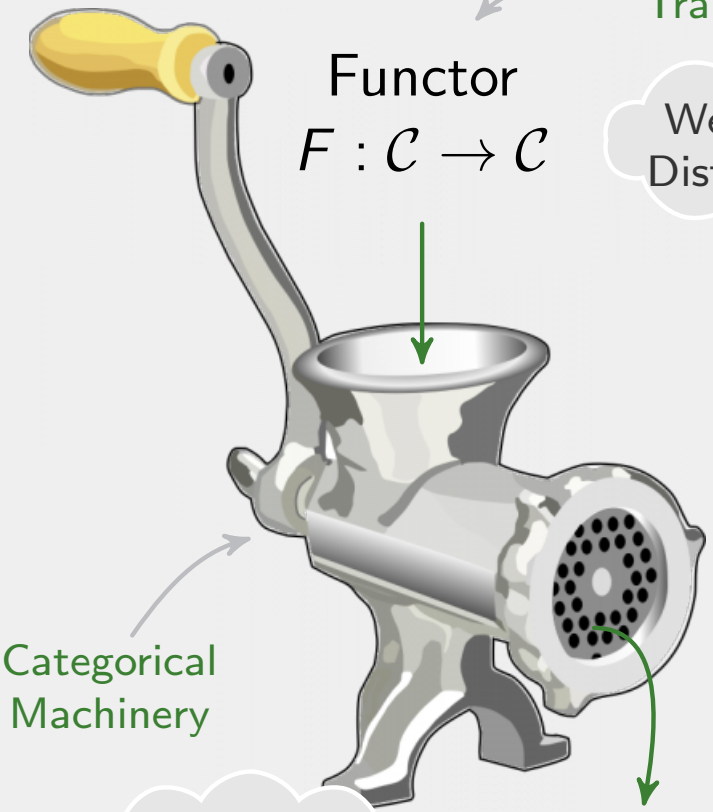
Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

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

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

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

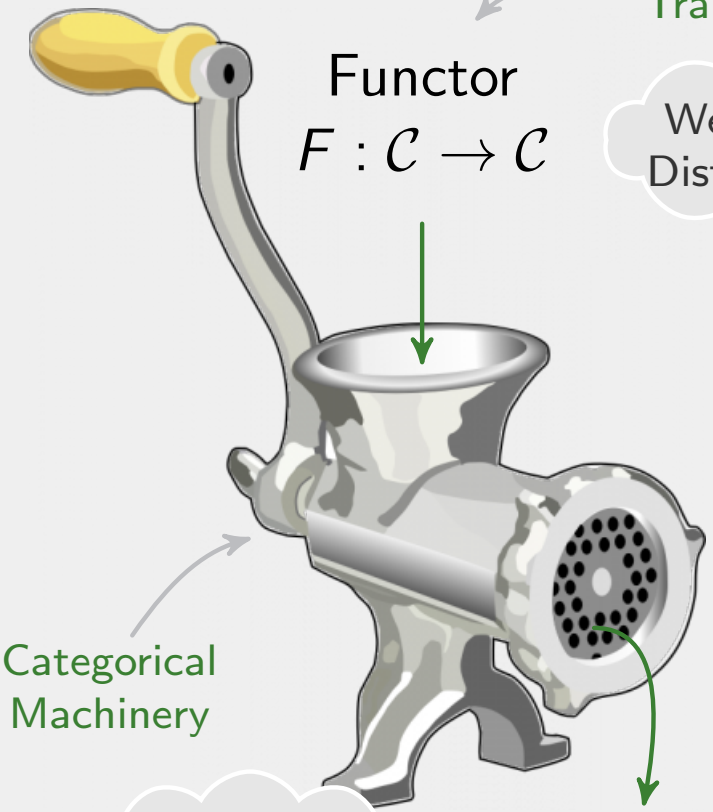
Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

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

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

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
$2 \times (-)^A$

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

Categorical Machinery

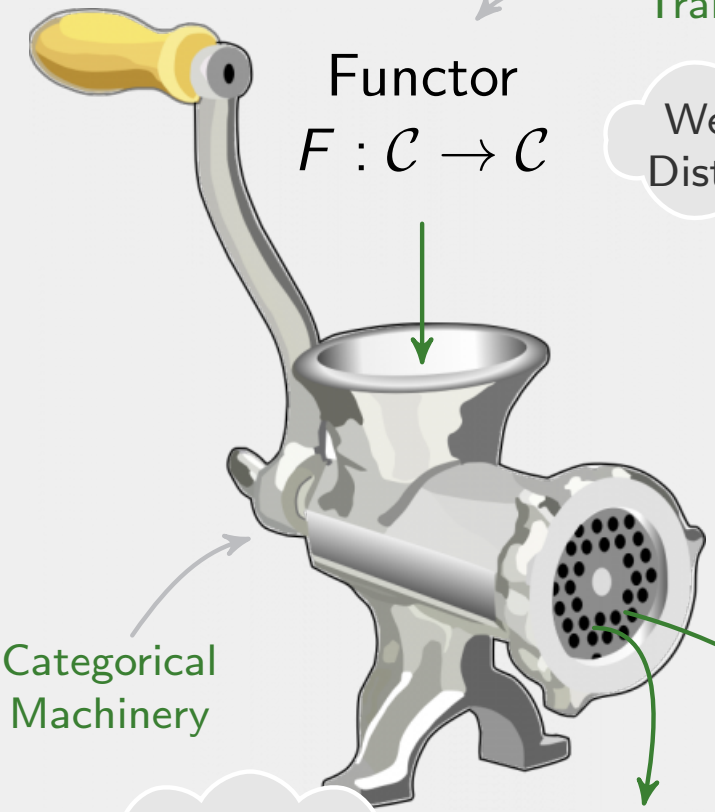
Markov Chain
$C \xrightarrow{c} \mathbb{R}^C$

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

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

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
$2 \times (-)^A$

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

Categorical Machinery

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

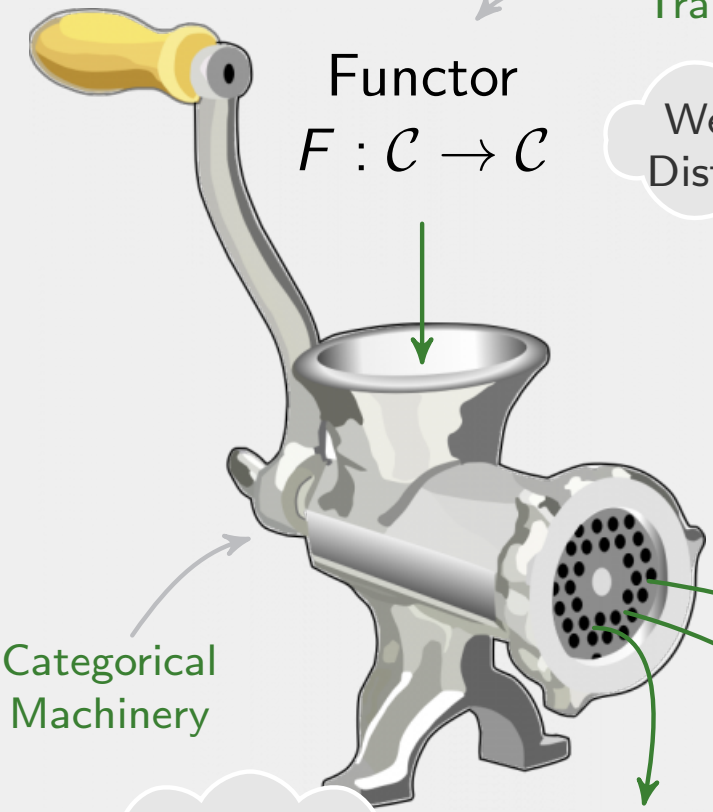
Transition System
$C \xrightarrow{c} \mathcal{P}C$

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

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 $\mathcal{P}$

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

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

Automata
$2 \times (-)^A$

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
$C \xrightarrow{c} FC$

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

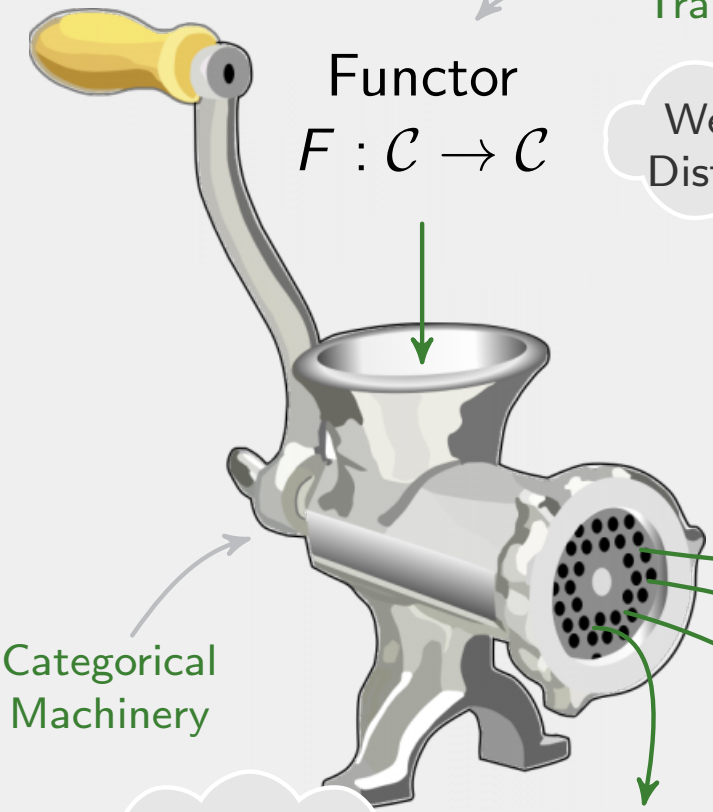
Behavioural Equivalence

Strong Bisimilarity

Language Equivalence

Homomorphism
$C \xrightarrow{h} C'$

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
$2 \times (-)^A$

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

Categorical Machinery

Minimization Algorithm

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

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

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

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

Behavioural Equivalence

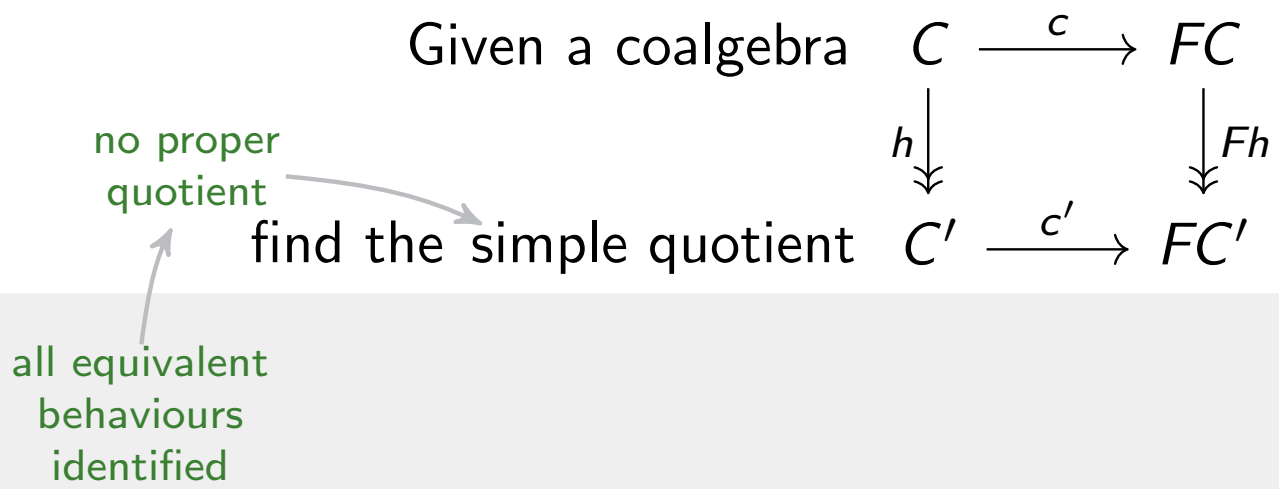
Strong Bisimilarity

Language Equivalence

Homomorphism
$C \xrightarrow{h} C'$

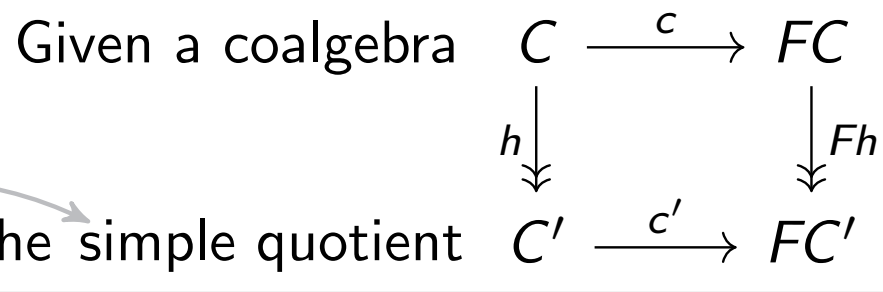
The Coalgebraic Task

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



The Coalgebraic Task

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



no proper quotient

all equivalent behaviours identified

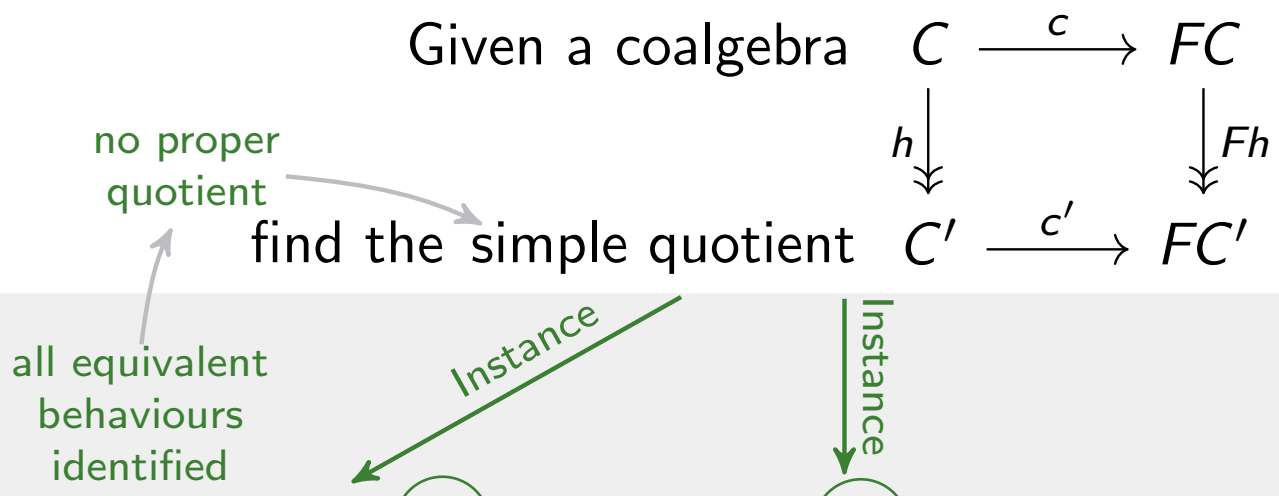
Instance

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

Automata minimization

The Coalgebraic Task

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



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

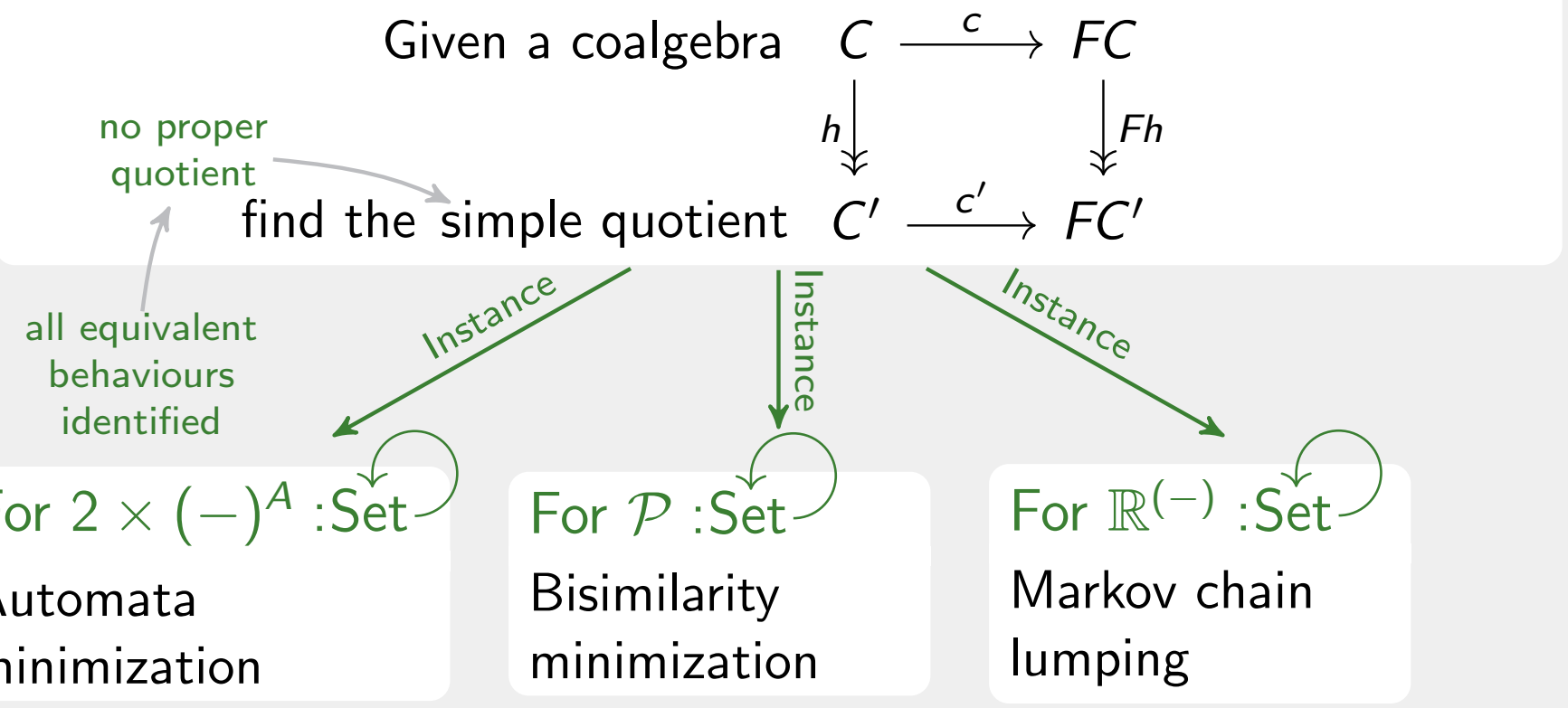
Automata minimization

For $\mathcal{P}$:Set

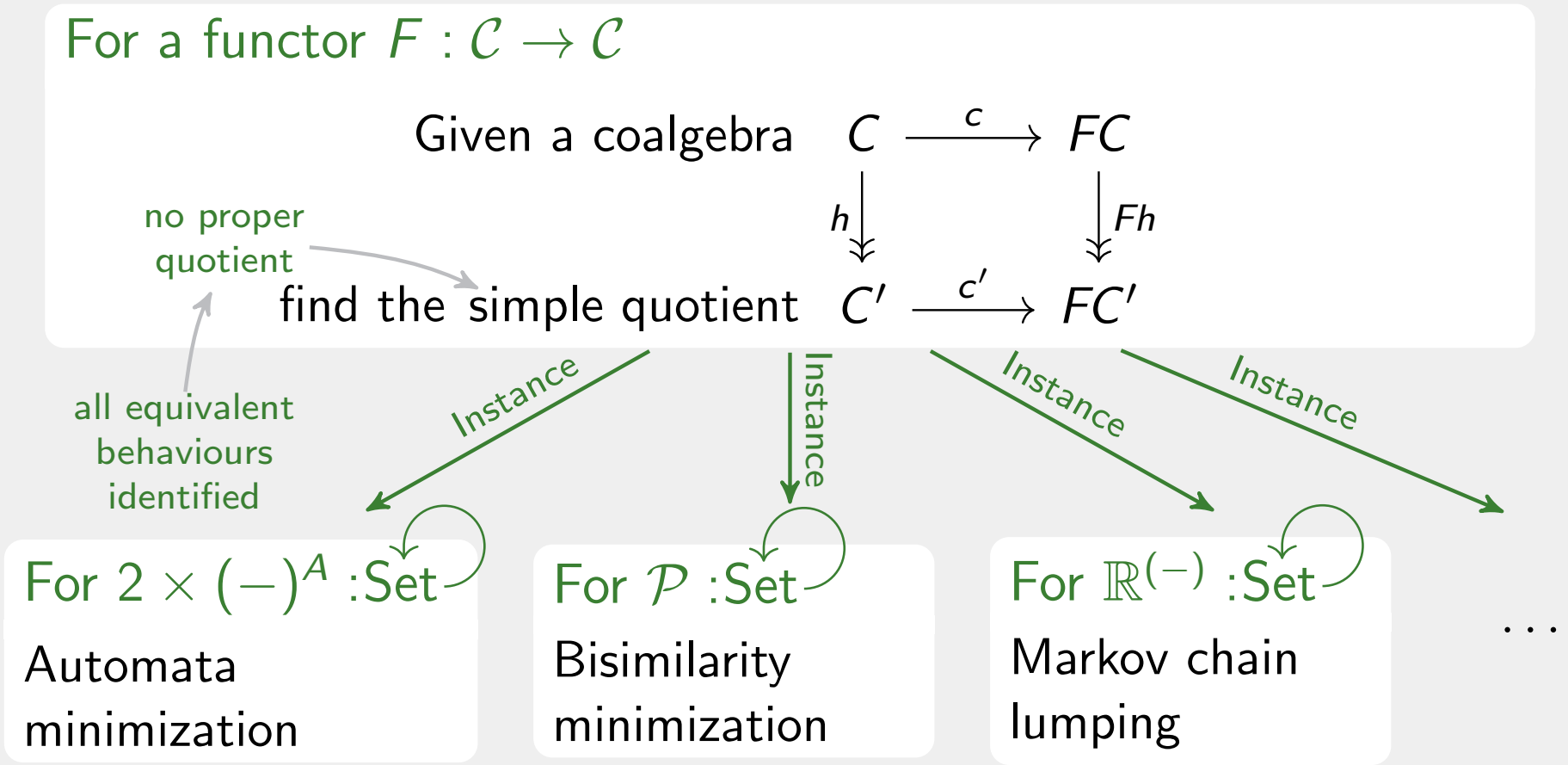
Bisimilarity minimization

The Coalgebraic Task

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

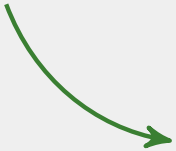


The Coalgebraic Task

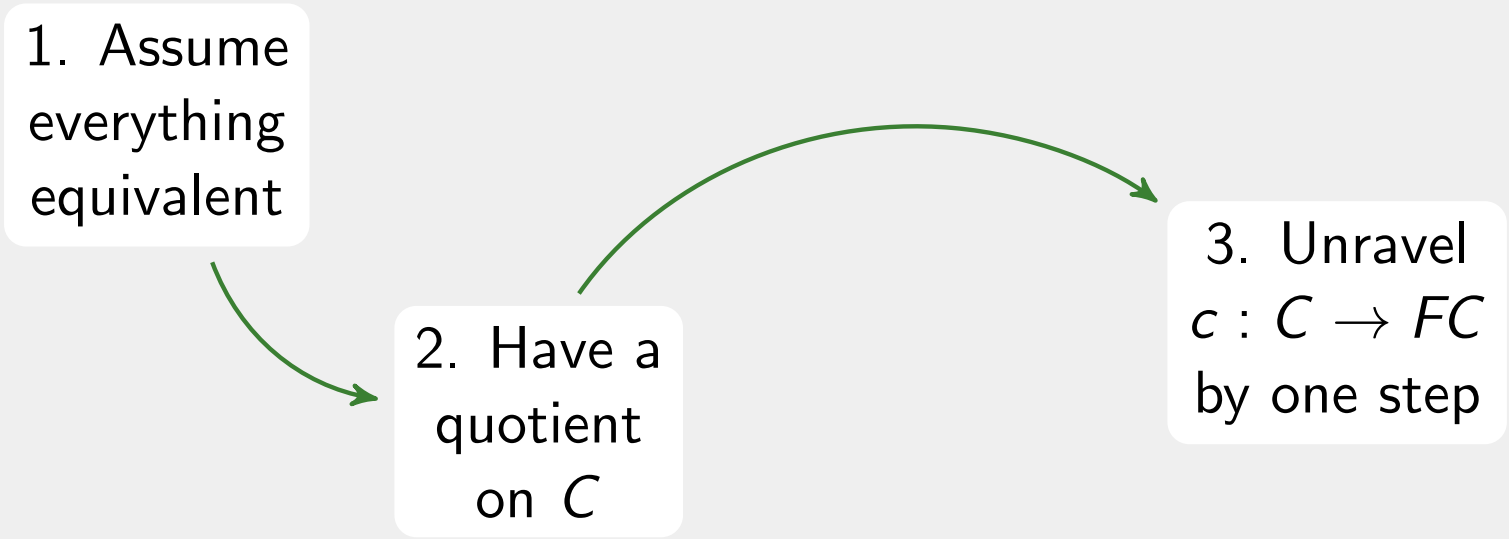


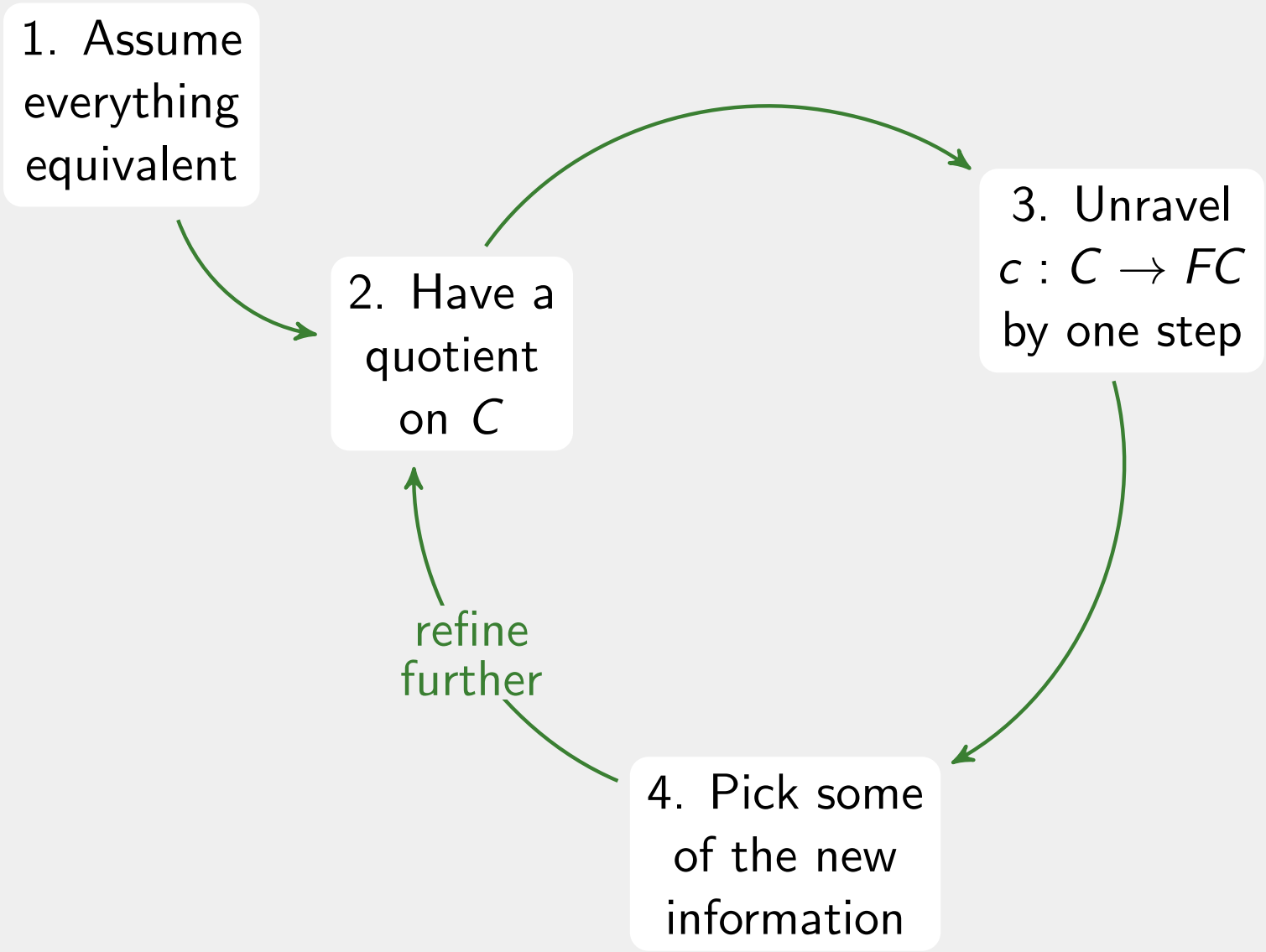
1. Assume everything equivalent

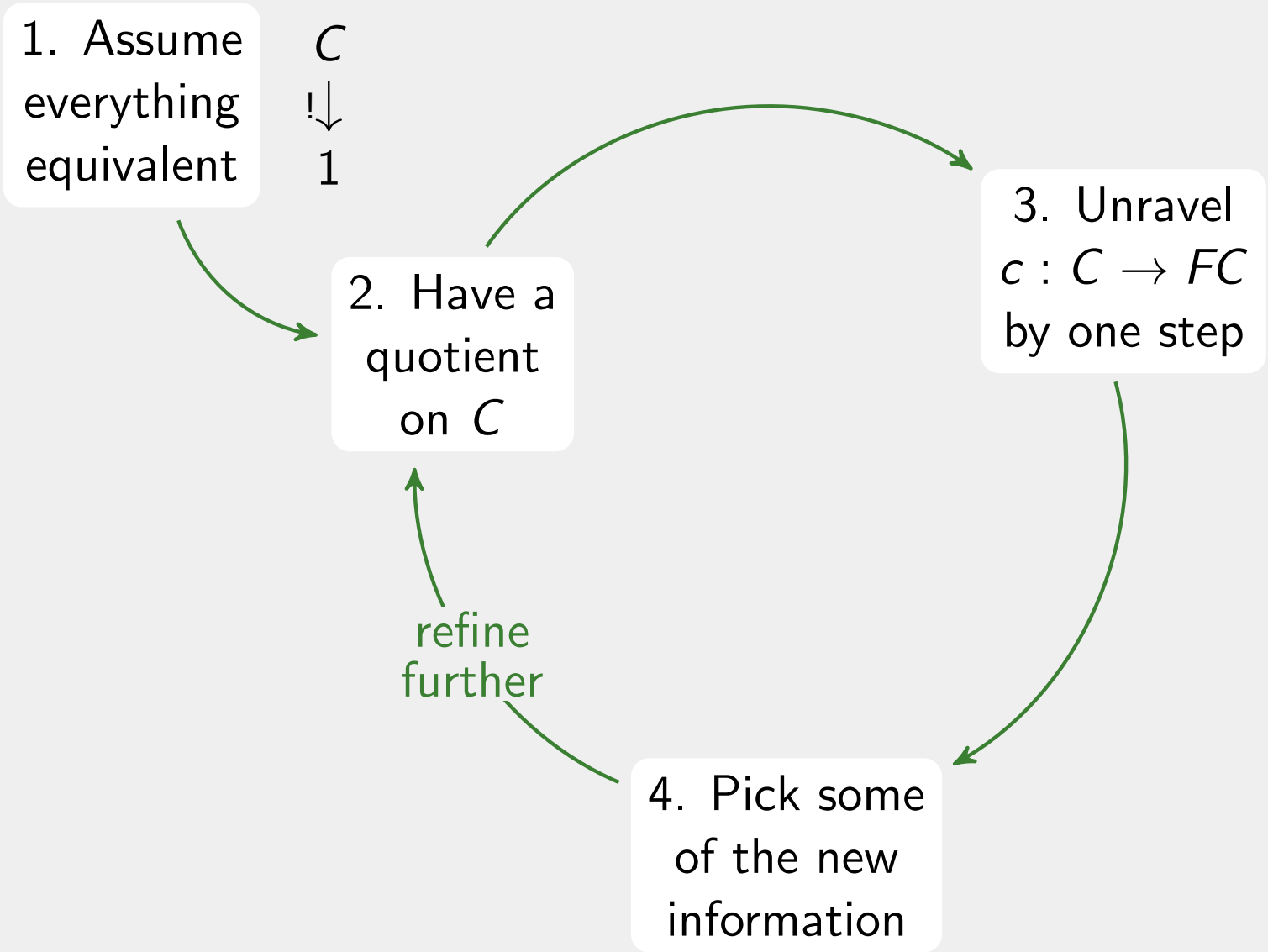
1. Assume everything equivalent

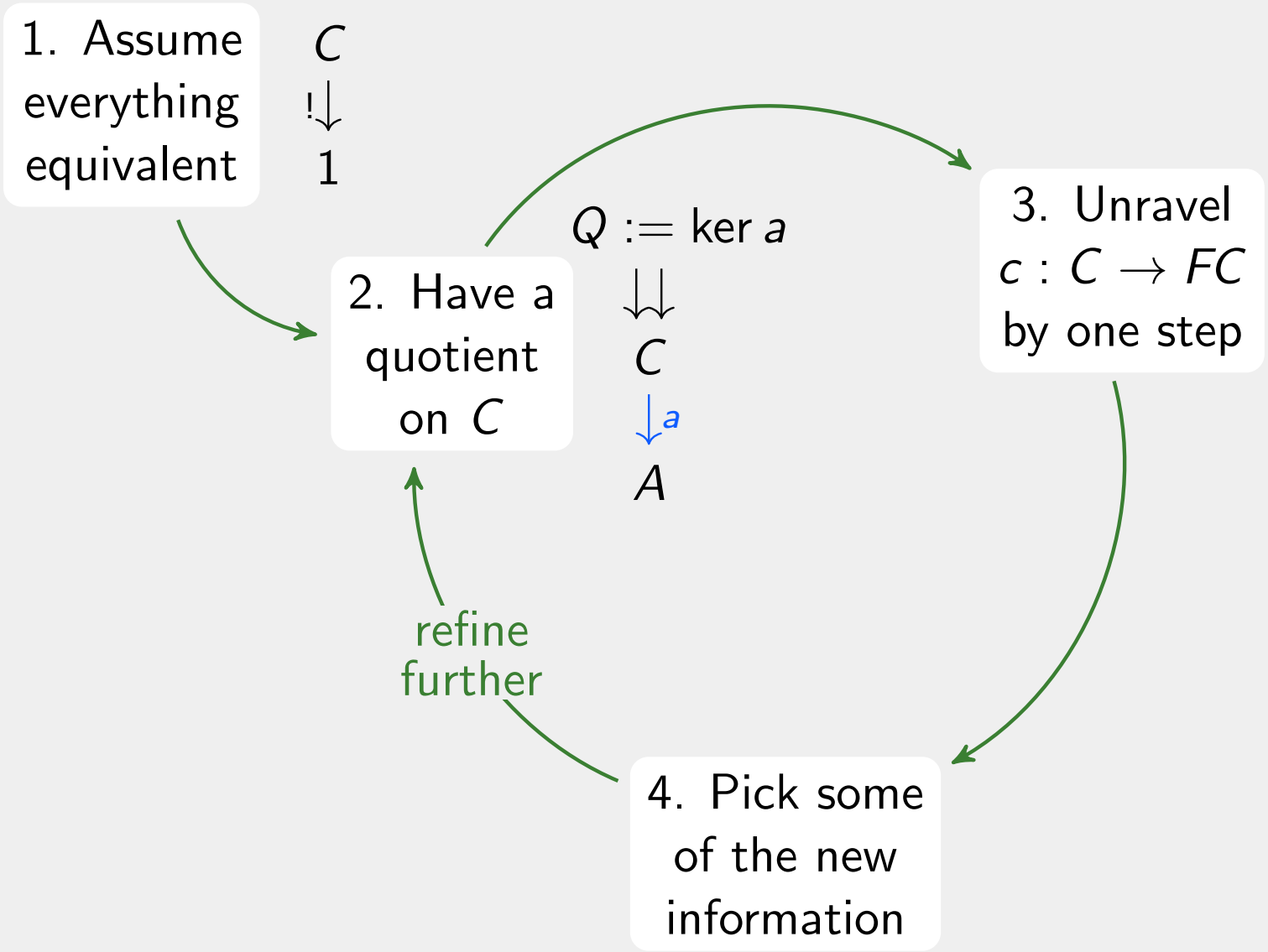


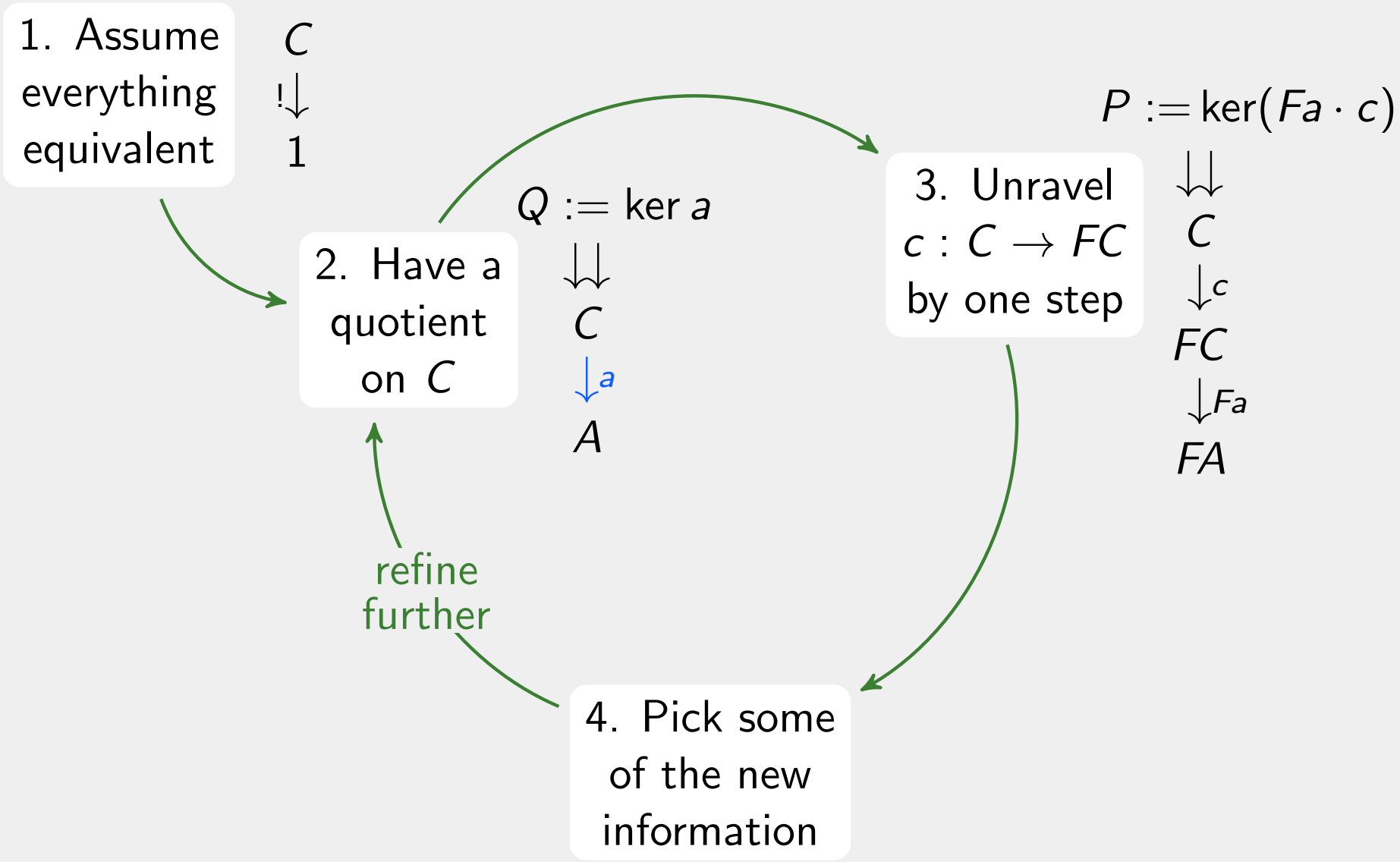
2. Have a quotient on C

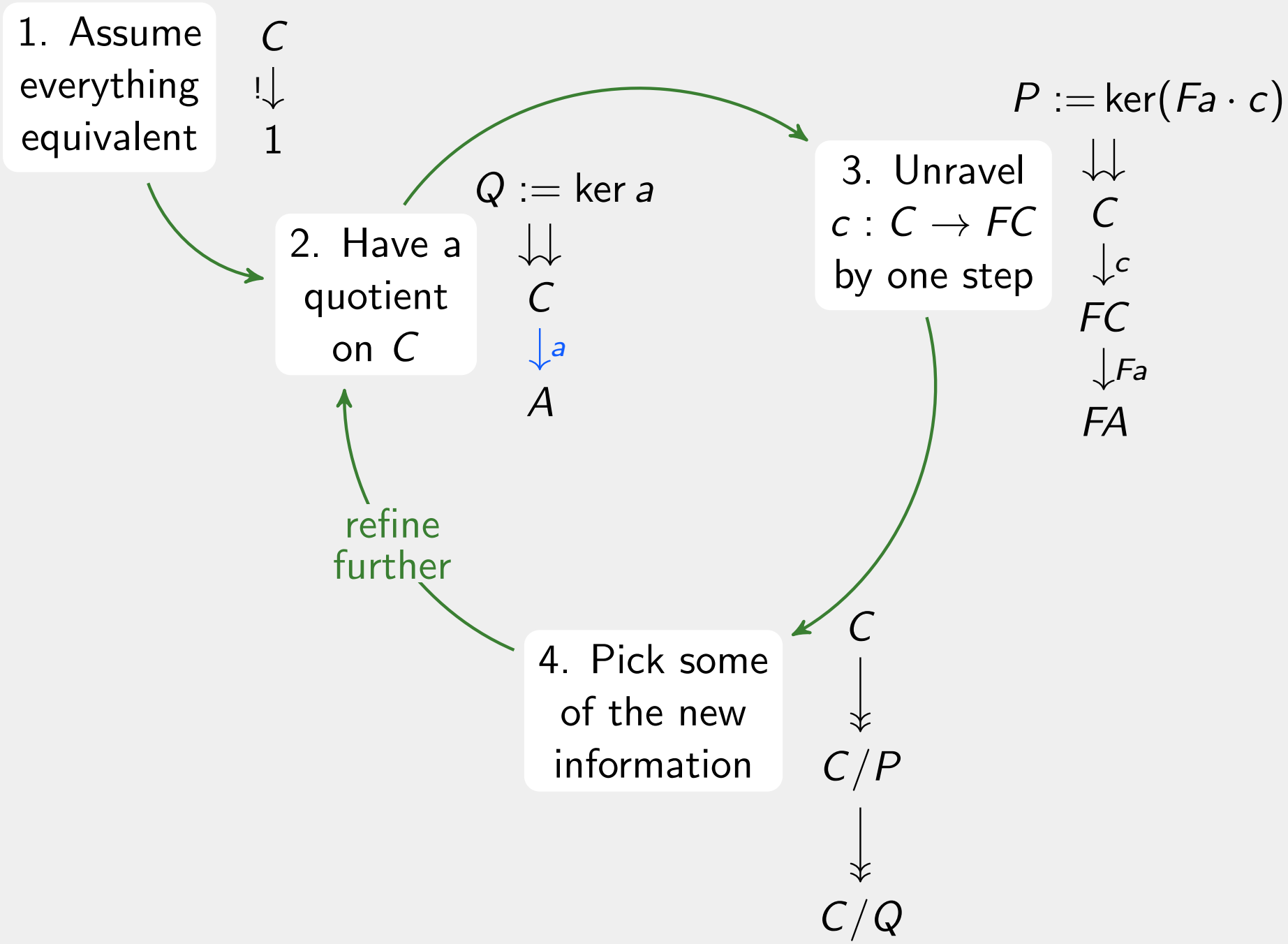

1. Assume everything equivalent
2. Have a quotient on C
3. Unravel $c : C \to FC$ by one step

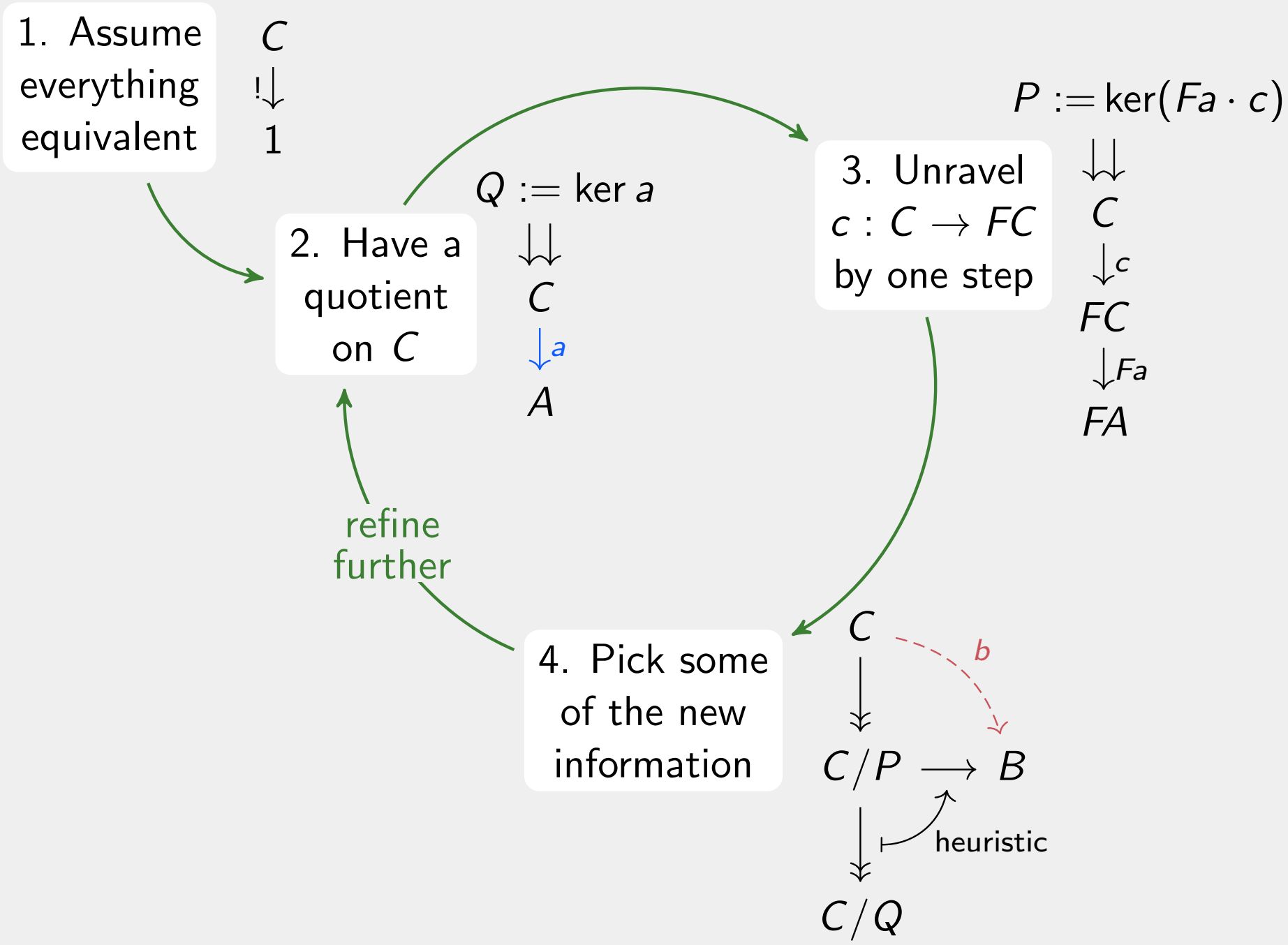

1. Assume everything equivalent
2. Have a quotient on C
3. Unravel $c : C \to FC$ by one step
4. Pick some of the new information
refine further

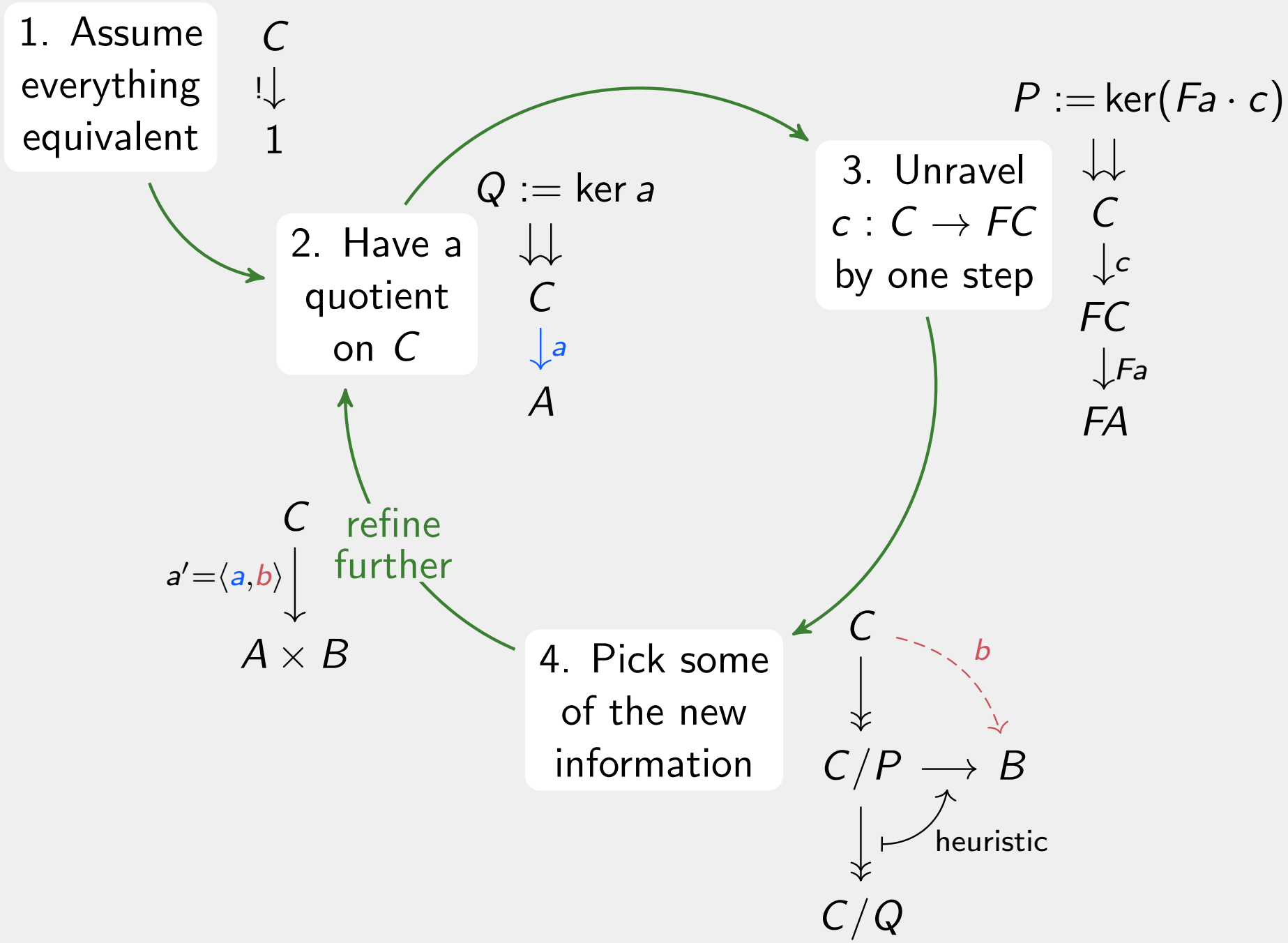

1. Assume everything equivalent
$C \xrightarrow{!} 1$
2. Have a quotient on C
3. Unravel $c : C \to FC$ by one step
4. Pick some of the new information
refine further

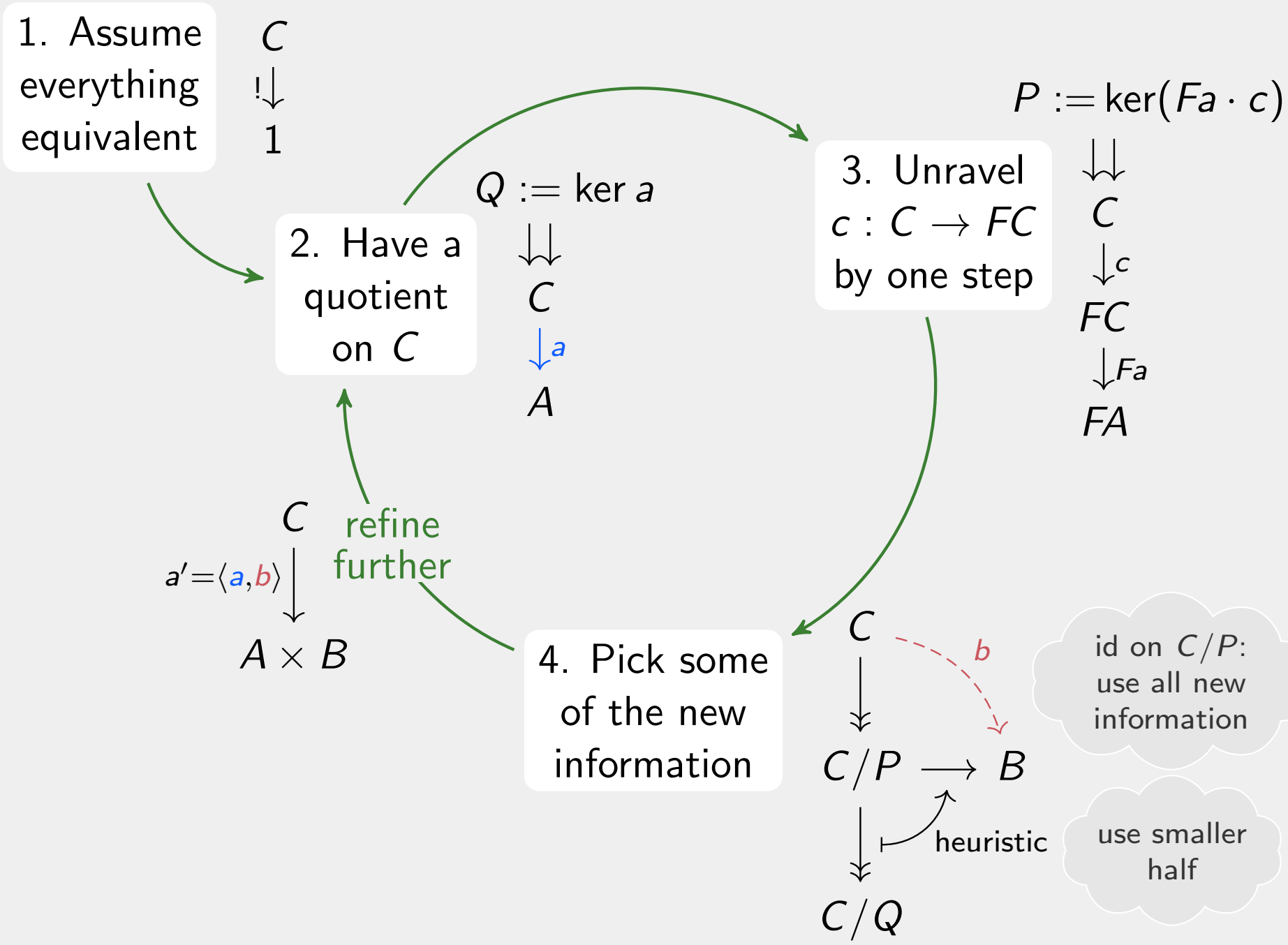

1. Assume everything equivalent
C
!
1
2. Have a quotient on C
Q := ker a
C
a
A
3. Unravel c : C → FC by one step
4. Pick some of the new information
refine further


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

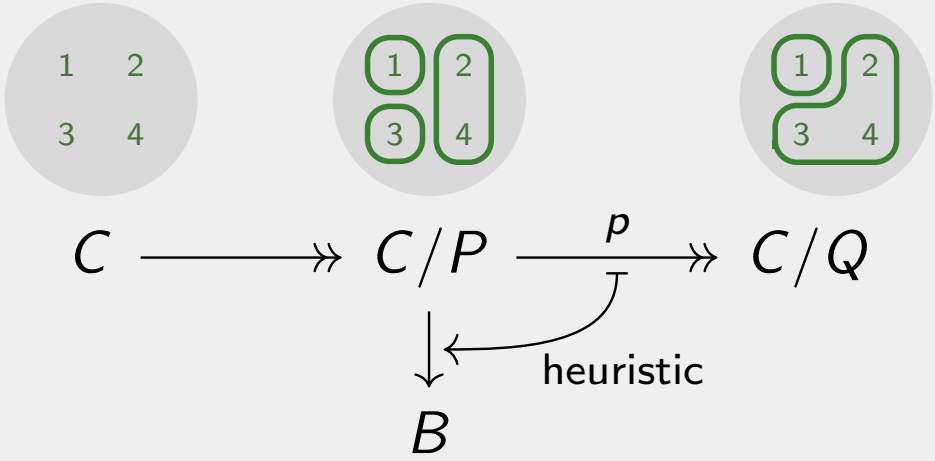

1. Assume everything equivalent
C
!
1
2. Have a quotient on C
Q := ker a
C
a
A
3. Unravel c : C → FC by one step
P := ker(Fa · c)
C
c
FC
Fa
FA
4. Pick some of the new information
refine further
C
C/P
C/Q


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
refine further
C
b
C/P
B
heuristic
C/Q

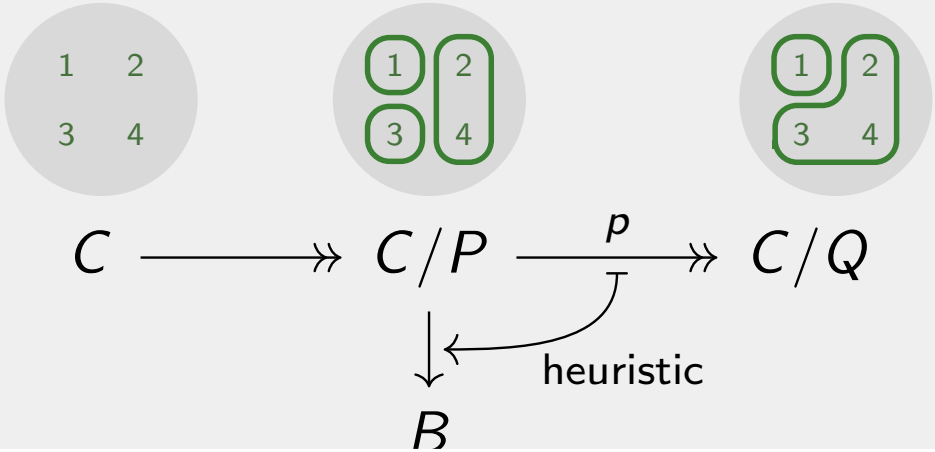

1. Assume everything equivalent
C
!
1
2. Have a quotient on C
Q := ker a
C
a
A
3. Unravel c : C → FC by one step
P := ker(Fa · c)
C
c
FC
Fa
FA
4. Pick some of the new information
C
b
C/P
B
heuristic
C/Q
refine further
C
a′=⟨a,b⟩
A × B


1. Assume everything equivalent
C
!
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
refine further
C
a′=⟨a,b⟩
A × B
C
b
C/P
B
heuristic
C/Q
id on C/P: use all new information
use smaller half

Heuristic



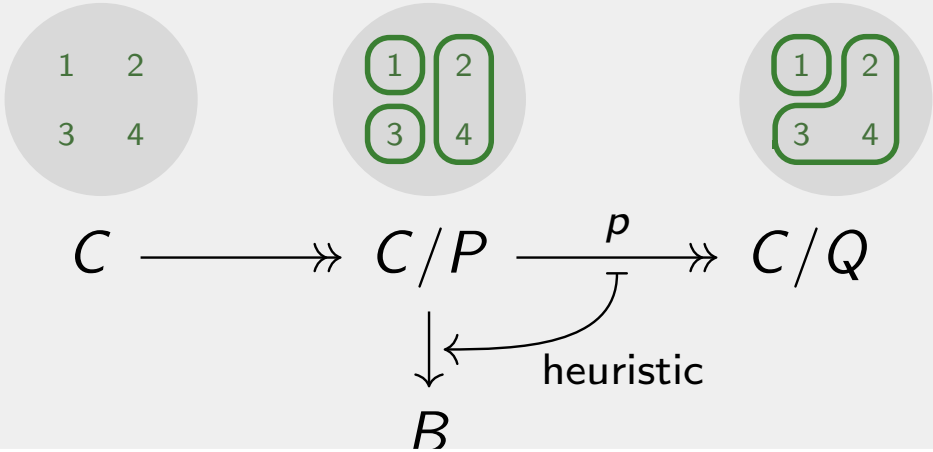
Heuristic



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

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

Assume

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

Theorem (Correctness)

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ \downarrow & & \downarrow \\ C/P_i & \longrightarrow & F(C/Q_i) \end{array}$$

Assume

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

Theorem (Correctness)

$$\begin{array}{ccc} C & \xrightarrow{c} & FC \\ \downarrow & & \downarrow \\ C/P_i & \longrightarrow & F(C/Q_i) \end{array}$$

If $P_i \cong Q_i$, then this

1. is a coalgebra
2. has no proper quotient

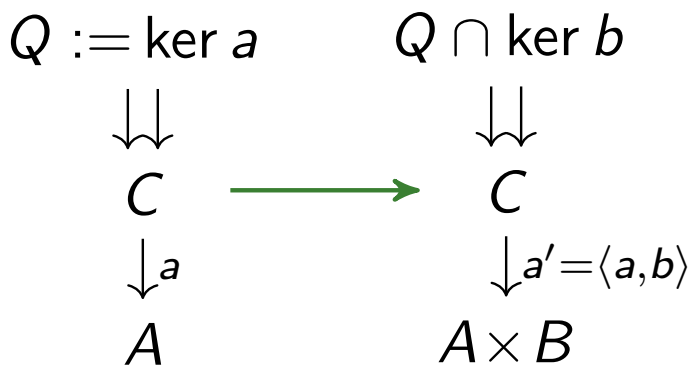
Efficiency: Incremental Partitions

Incremental partitions

$$
\begin{array}{l}
Q := \ker a \\
\quad \twoheaddownarrow \\
\quad C \\
\quad \downarrow a \\
\quad A
\end{array}
$$

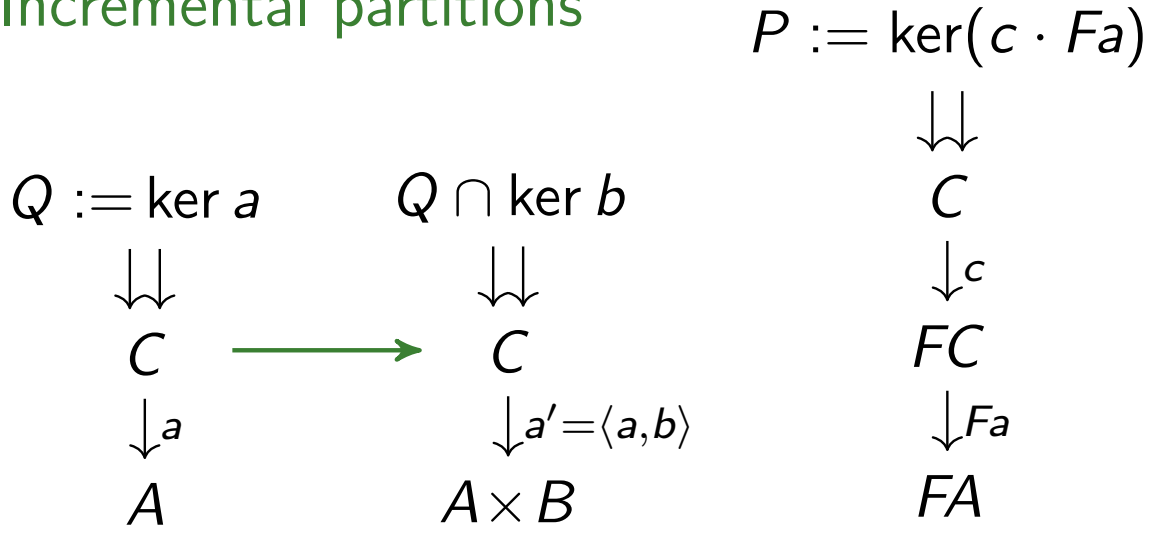
Efficiency: Incremental Partitions

Incremental partitions



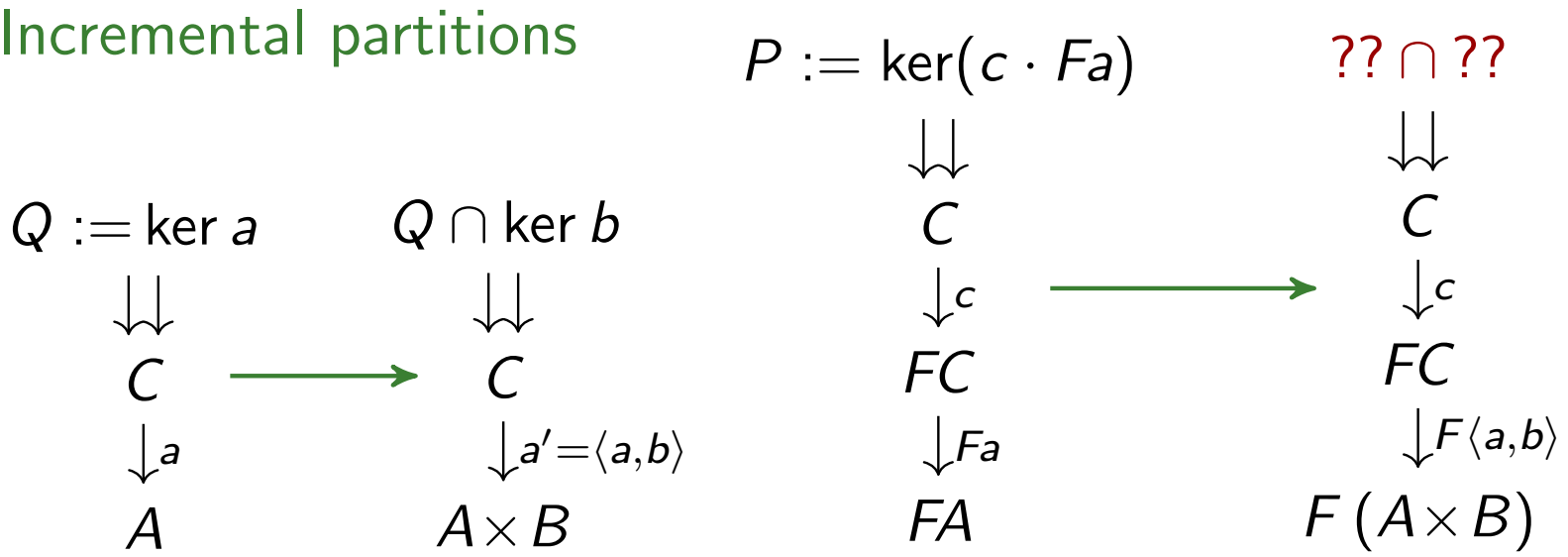
Efficiency: Incremental Partitions

Incremental partitions



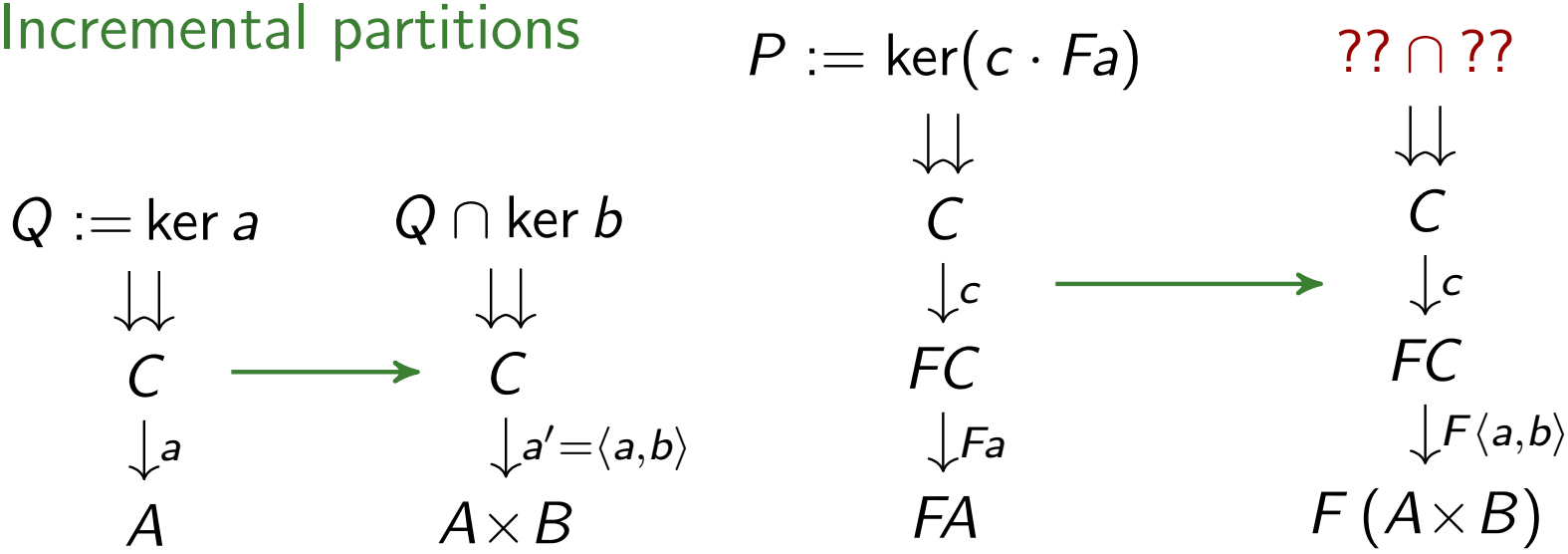
Efficiency: Incremental Partitions

Incremental partitions



Efficiency: Incremental Partitions

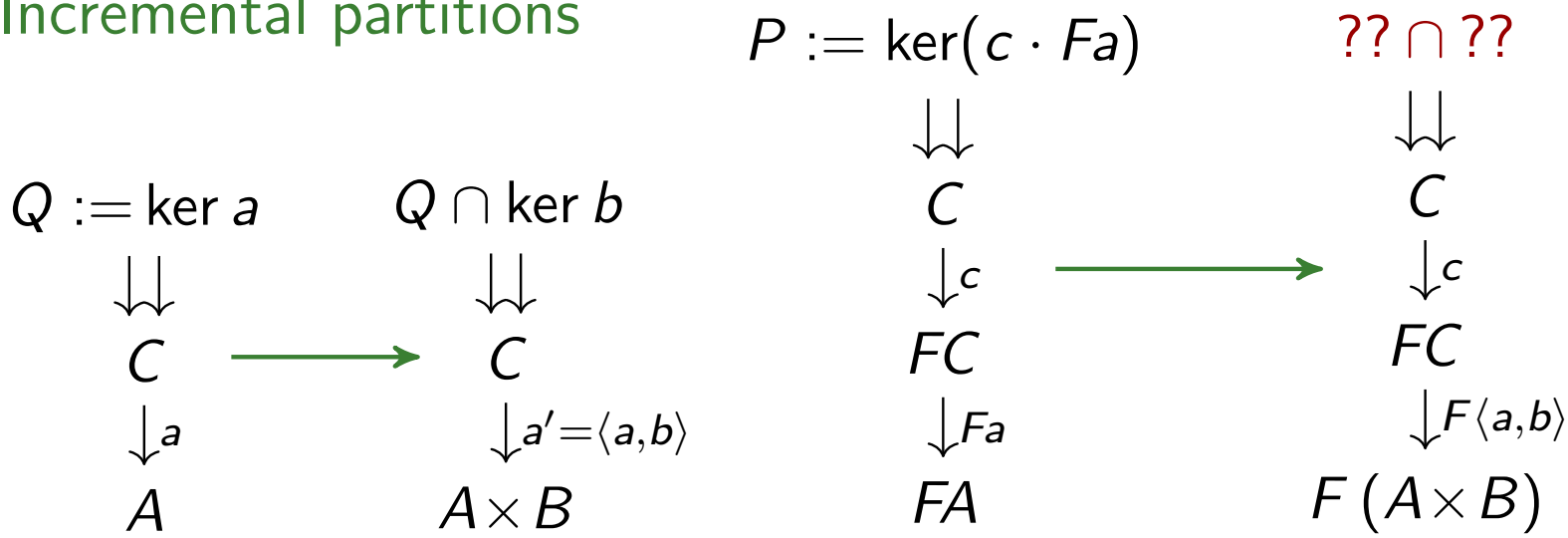
Incremental partitions



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

Efficiency: Incremental Partitions

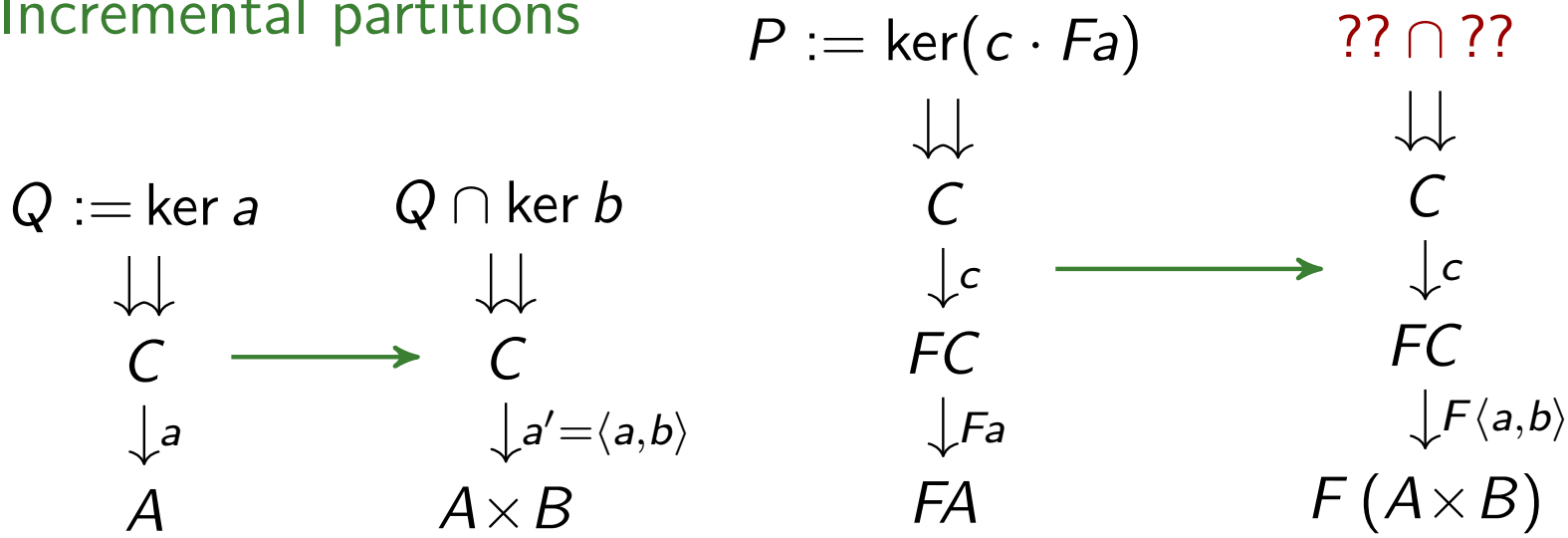
Incremental partitions



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

Efficiency: Incremental Partitions

Incremental partitions



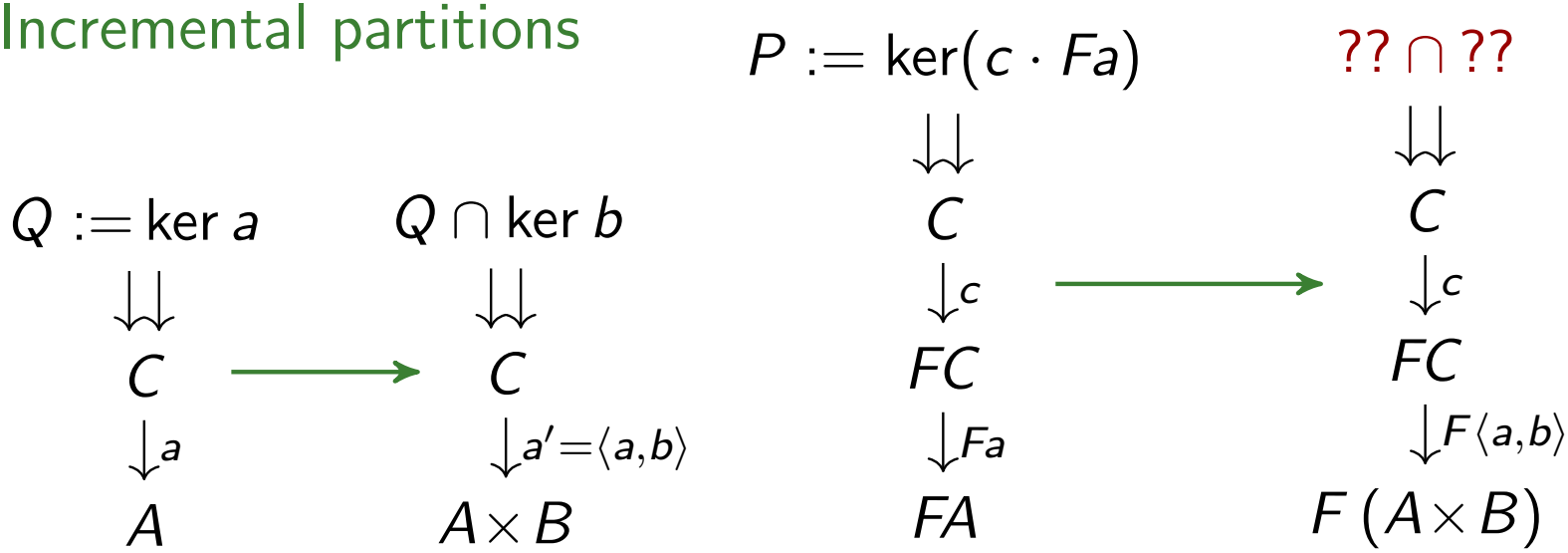
Theorem: In Set, $\ker F\langle a, b\rangle = \ker\langle Fa, Fb\rangle$ if

$$\begin{array}{c} F(L+R) \\ \downarrow \quad \text{injective} \\ F(L+1) \times F(1+R) \end{array} \quad \text{and}$$

"zippable"

Efficiency: Incremental Partitions

Incremental partitions



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

$$\begin{array}{c} F(L+R) \\ \downarrow \\ F(L+1) \times F(1+R) \end{array}$$

injective and

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

"zippable"

Efficiency: Incremental Partitions

Incremental partitions

$$\begin{array}{ccc} Q := \ker a & & Q \cap \ker b \\ \downdownarrows & & \downdownarrows \\ C & \longrightarrow & C \\ \downarrow a & & \downarrow a' = \langle a, b \rangle \\ A & & A \times B \end{array}$$

$$\begin{array}{ccc} P := \ker(c \cdot Fa) & & P \cap \ker(Fb \cdot c) \\ \downdownarrows & & \downdownarrows \\ C & & C \\ \downarrow c & \longrightarrow & \downarrow c \\ FC & & FC \\ \downarrow Fa & & \downarrow F\langle a,b \rangle \\ FA & & F(A \times B) \end{array}$$

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

$$\begin{array}{c} F(L+R) \\ \downarrow \quad \text{injective} \\ F(L+1) \times F(1+R) \end{array} \quad \text{and} \quad \begin{array}{c} \ker a \cup \ker b \\ \text{an equivalence} \end{array}$$

"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

- coalgebra structure as edges with labels

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

- Refinement interface $\{$ `update()`, `init()`, `weight()` $\}$

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

Setting for complexity analysis

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

Assumption: Functor encoding

- coalgebra structure as edges with labels

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

- Refinement interface $\{$ `update()`, `init()`, `weight()` $\}$

$\Rightarrow$ 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	hm		Our instantiation
Transition Systems	$\mathcal{P}$		$=$	$(m+n)\cdot\log n$
LTS	$\mathcal{P}(A\times -)$	$(m+n)\cdot\log n$ 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	$>$	$(m+n)\cdot\log(m+n)$

More instances: further system types & categories

Generic & Efficient

System	Functor	hm		Our instantiation
Transition Systems	$\mathcal{P}$		$=$	$(m+n)\cdot\log n$
LTS		Dovier, Piazza, Policriti '04	$=$	$(m+n)\cdot\log(m+n)$
		$(m+n)\cdot\log n$ Valmari '09		
Mark Chains		$(m+n)\cdot\log n$ Valmari, France, Tinnis '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

Implementation: functor & system as the input

Generic & Efficient

System	Functor	hm		Our instantiation
Transition Systems	$\mathcal{P}$		$=$	$(m+n)\cdot\log n$
LTS		Dovier, Piazza, Policriti '04	$=$	$(m+n)\cdot\log(m+n)$
		$(m+n)\cdot\log n$ Valmari '09		
Mark Chains		$(m+n)\cdot\log n$ Valmari, France, Tinnis '10	$=$	$(m+n)\cdot\log n$
DFA	$2\times(-)^A$	$n\cdot\log$ 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	$\mathcal{P}(A\times-)\cdot\mathcal{D}$	$m\cdot n\cdot$ Baier, Engelen, Majster-Cederbaum'00	$>$	$(m+n)\cdot\log(m+n)$

More instances: further system types & categories

Implementation: functor & system as the input

Compare to existing concrete implementations

Generic & Efficient

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 $C \overset{\kappa}{\twoheadrightarrow} \mathcal{I}$

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,

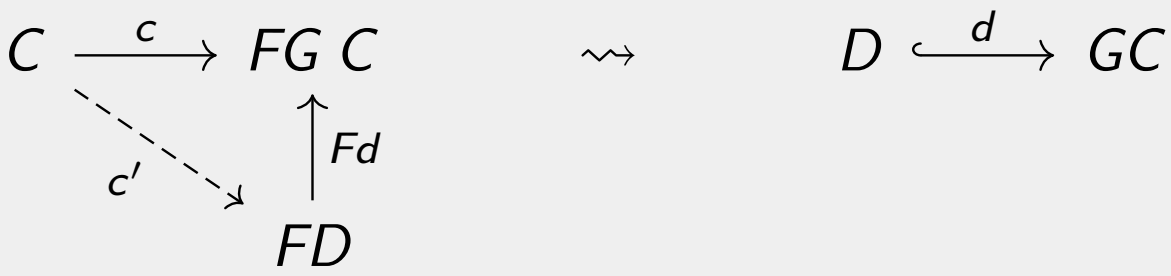
$$C \xrightarrow{c} FG\,C \qquad \rightsquigarrow \qquad D \xhookrightarrow{d} GC$$

Genericity: Composition

If F finitary,

Genericity: Composition

If F finitary,

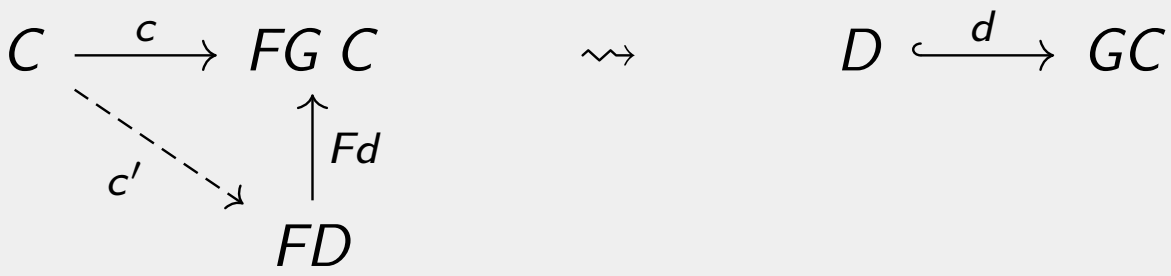


A coalgebra on Set^2 for the functor $(X, Y) \mapsto (FY, GX)$:

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

Genericity: Composition

If F finitary,



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

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

Examples

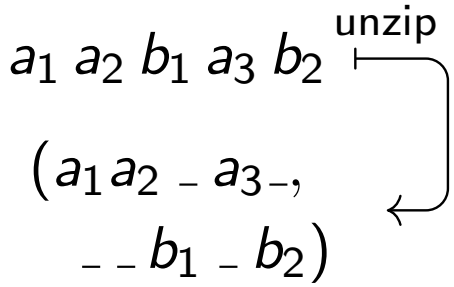
$$\mathcal{P} \cdot (A \times (-)) \qquad (2 \times \mathcal{P}) \cdot (A \times (-))$$
$$\mathcal{P} \cdot (A \times (-)) \cdot \mathcal{D} \qquad \mathcal{P} \cdot \mathcal{D} \cdot (A \times (-)) \qquad \dots$$

Functors F zippable, if

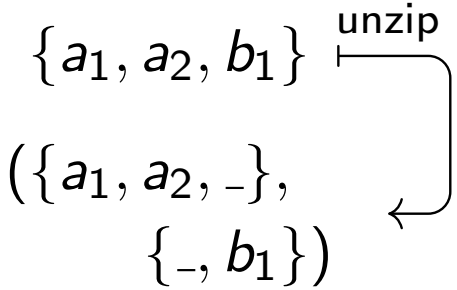
$$F(L+R) \xrightarrow{\text{unzip}} F(L+1) \times F(1+R) \text{ is monic.}$$

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

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



$(-)^*$ is zippable



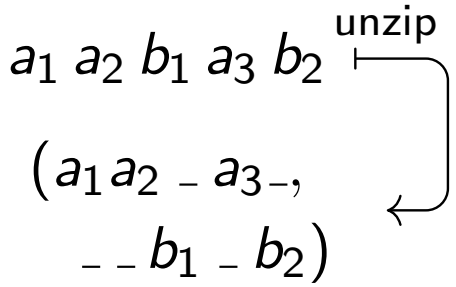
$\mathcal{P}$ is zippable

Functors F zippable, if

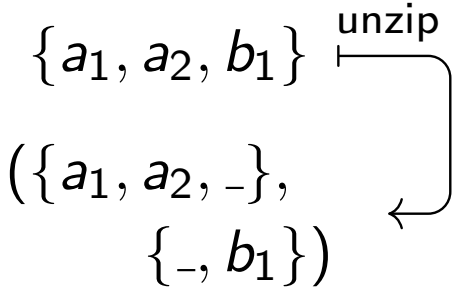
$$F(L+R) \xrightarrow{\text{unzip}} F(L+1) \times F(1+R) \text{ is monic.}$$

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

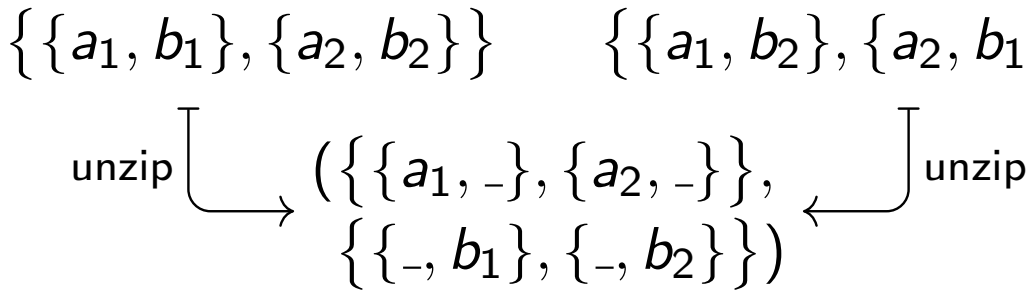
Examples for sets $L = \{a_1, a_2, a_3\}$, $R = \{b_1, b_2\}$, $1 = \{\_\}$



$(-)^*$ is zippable



$\mathcal{P}$ is zippable



$\mathcal{PP}$ is not zippable

~~Composition~~

~~Quotients~~

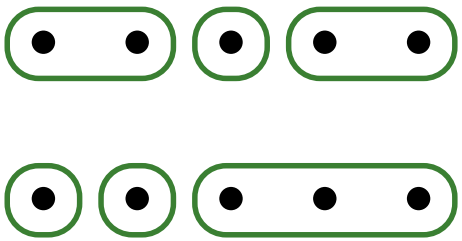
$$A \xleftarrow{a} X \xrightarrow{b} B$$

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

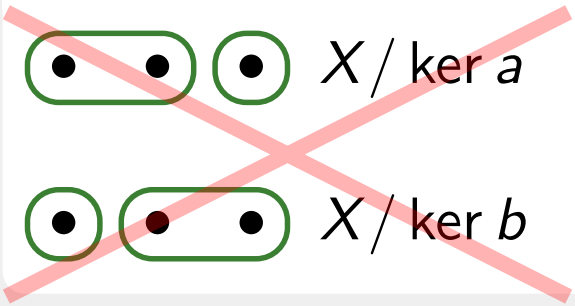
$\Leftrightarrow$ $\ker a \cup \ker b$ transitive

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

Example



Non-Example



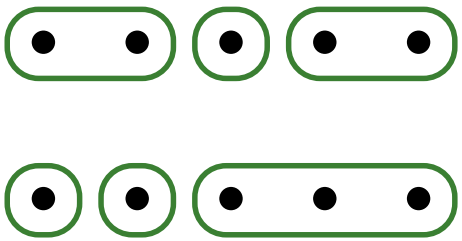
$$A \xleftarrow{a} X \xrightarrow{b} B$$

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

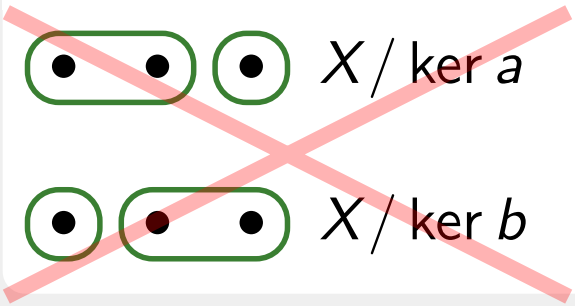
$\Leftrightarrow$ $\ker a \cup \ker b$ transitive

$\Leftrightarrow$ $\forall x \in X : [x]_a \subseteq [x]_b$ 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 $\langle \chi_S, \chi_C \rangle : X \to 2 \times 2$

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:	$G^{(2)}$	$\mathcal{B}_{\mathrm{f}}2$	$\mathcal{D}2$	$\mathbb{N}$	$F_\Sigma 2$
$w(C)$, $C \subseteq Y$:	$G\chi_C$	$\mathcal{B}_{\mathrm{f}}\chi_C$	$\mathcal{D}\chi_C$	$\lvert C \cap (-)\rvert$	$F_\Sigma \chi_C$

References

[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 Inf.** 2 (1973), pp. 97–109. ISSN: 1432-0525.

[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.

[KK14] 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.

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

[PT87] Robert Paige, Pobert 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.