State-based Simulation of Linear Course-of-value Iteration

Baltasar Trancón y Widemann

University of Bayreuth, Germany

11th CMCS, Tallinn, Estonia
2012-03-31 / -04-01

1. **Introduction**
   - General Theory
   - Special Case: Linear History
2. **Simulation**
   - Main Definitions
   - Limit Cases
   - Average Cases
3. **Conclusion**

Preliminaries

Notational Conventions

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

Initial F-algebra $(\mu F, \mathrm{in}_F : F\mu F \to \mu F)$

Final F-coalgebra $(\nu F, \mathrm{out}_F : \nu F \to F\nu F)$

Lambek's $\mathrm{in}_F, \mathrm{out}_F$ are isomorphisms

Ordinary (Co)Iteration, Categorically

Catamorphism

Every F-algebra $(A, \varphi : FA \to A)$ induces a unique
homomorphism $(\!|\varphi|\!)_A : \mu F \to A$
satisfying $(\!|\varphi|\!)_A \circ \mathsf{in}_F = \varphi \circ F(\!|\varphi|\!)_A$

Anamorphism

Every F-coalgebra $(B, \varphi : B \to FB)$ induces a unique
homomorphism $[\!(\varphi)\!]_B : B \to \nu F$
satisfying $\mathsf{out}_F \circ [\!(\varphi)\!]_B = F[\!(\varphi)\!]_B \circ \varphi$

Ordinary (Co)Iteration, Categorically

Catamorphism

Every F-algebra $(A, \varphi : FA \to A)$ induces a unique

homomorphism $(\!|\varphi|\!)_A : \mu F \to A$

satisfying $(\!|\varphi|\!)_A \circ \mathsf{in}_F = \varphi \circ F(\!|\varphi|\!)_A$

Anamorphism

Every F-coalgebra $(B, \varphi : B \to FB)$ induces a unique

homomorphism $[\!(\varphi)\!]_B : B \to \nu F$

satisfying $\mathsf{out}_F \circ [\!(\varphi)\!]_B = F[\!(\varphi)\!]_B \circ \varphi$

Course-of-value Iteration, Categorically

Some More Functors

- Add a *colouring* C to F

$$F_C^\times = C \times F(-)$$

- Strange auxiliary definitions (see below for intuition)

$$F^! C = \nu F_C^\times \qquad F^!(h : C \to D) = [\![(h \times \mathrm{id}_{F^? C}) \circ \mathrm{out}_{F_C^\times}]\!]_{F_D^\times}$$

$$F^? = F F^!$$

Histomorphism (Uustalu and Vene 1999)

- Every $F^?$-algebra $(C, \varphi : F^? C \to C)$ induces a unique

homomorphism $\{\!|\varphi|\!\}_F : \mu F \to C$

satisfying $\{\!|\varphi|\!\}_F \circ \mathrm{in}_F = \varphi \circ F[\![\langle \{\!|\varphi|\!\}_F, \mathrm{in}_F^{-1} \rangle]\!]_{F_C^\times}$

namely $\{\!|\varphi|\!\}_F = \pi_1 \circ \mathrm{out}_{F_C^\times} \circ (\!| \mathrm{out}_{F_C^\times}^{-1} \circ \langle \varphi, \mathrm{id}_{F^? C} \rangle |\!)_F$

Course-of-value Iteration, Categorically

Some More Functors

- Add a *colouring* C to F

$$F_C^\times = C \times F(-)$$

- Strange auxiliary definitions (see below for intuition)

$$F^! C = \nu F_C^\times \qquad F^!(h : C \to D) = [\![(h \times \mathrm{id}_{F^? C}) \circ \mathrm{out}_{F_C^\times}]\!]_{F_D^\times}$$

$$F^? = F F^!$$

Histomorphism (Uustalu and Vene 1999)

- Every $F^?$-algebra $(C, \varphi : F^? C \to C)$ induces a unique

homomorphism $\{\!|\varphi|\!\}_F : \mu F \to C$

satisfying $\{\!|\varphi|\!\}_F \circ \mathrm{in}_F = \varphi \circ F[\![\langle \{\!|\varphi|\!\}_F, \mathrm{in}_F^{-1} \rangle]\!]_{F_C^\times}$

namely $\{\!|\varphi|\!\}_F = \pi_1 \circ \mathrm{out}_{F_C^\times} \circ (\!| \mathrm{out}_{F_C^\times}^{-1} \circ \langle \varphi, \mathrm{id}_{F^? C} \rangle |\!)_F$

Course-of-value Iteration, Categorically

Some More Functors

- Add a *colouring* C to F

$$F_C^\times = C \times F(-)$$

- Strange auxiliary definitions (see below for intuition)

$$F^! C = \nu F_C^\times \qquad F^!(h : C \to D) = [\![(h \times \mathrm{id}_{F^? C}) \circ \mathrm{out}_{F_C^\times}]\!]_{F_D^\times}$$

$$F^? = F F^!$$

Histomorphism (Uustalu and Vene 1999)

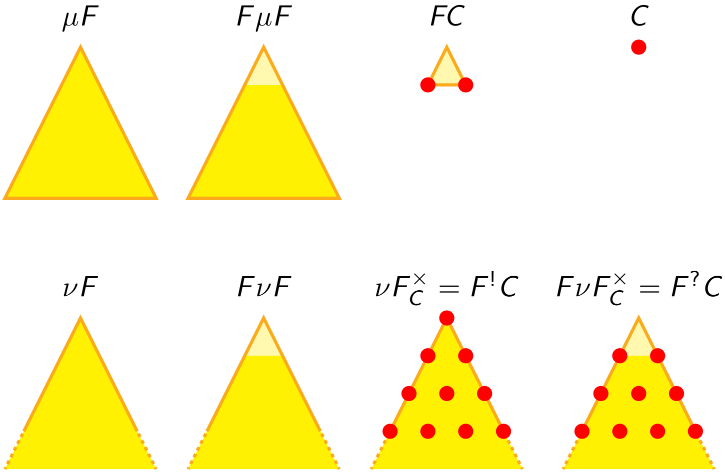
- Every $F^?$-algebra $(C, \varphi : F^? C \to C)$ induces a unique

homomorphism $\{\!|\varphi|\!\}_F : \mu F \to C$

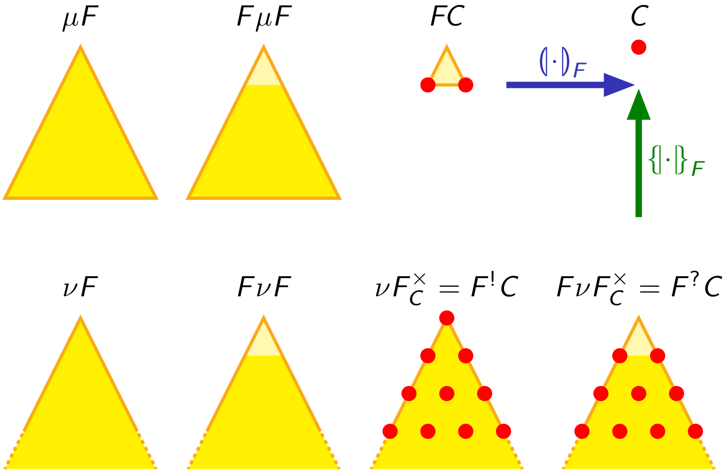
satisfying $\{\!|\varphi|\!\}_F \circ \mathrm{in}_F = \varphi \circ F[\!(\langle \{\!|\varphi|\!\}_F, \mathrm{in}_F^{-1} \rangle)\!]_{F_C^\times}$

namely $\{\!|\varphi|\!\}_F = \pi_1 \circ \mathrm{out}_{F_C^\times} \circ (\!|\mathrm{out}_{F_C^\times}^{-1} \circ \langle \varphi, \mathrm{id}_{F^? C} \rangle|\!)_F$

All Those Functorial Data Structures Pictured



All Those Functorial Data Structures Pictured



Special Case

Restriction to Simplest of Functors

- Peano functor

$$F = N = 1 + (-) : \mathbf{Set} \to \mathbf{Set}$$

- Data structures simplify

$$\mu N = \mathbb{N} \qquad\qquad NC = 1 + C$$

$$\nu N = \mathbb{N}_\infty \qquad N^{!}C = C^{+\infty} \qquad N^{?}C \cong C^{\infty}$$

- Ordinary iteration simplifies

$$\varphi = [z, s] \implies (\!|\varphi|\!)_N(n) = s^n(z)$$

- Course-of-value iteration simplifies

$$\{\!|\varphi|\!\}_N(n) \cong \varphi\Big(\{\!|\varphi|\!\}_N(n-1), \ldots, \{\!|\varphi|\!\}_N(0)\Big)$$

- Applications beyond Fibonacci & friends
  - Black-box components in software engineering
  - Empirics/statistics of dynamical systems

Special Case

Restriction to Simplest of Functors

- Peano functor

$$F = N = 1 + (-) : \mathbf{Set} \to \mathbf{Set}$$

- Data structures simplify

$$\mu N = \mathbb{N} \qquad\qquad NC = 1 + C$$

$$\nu N = \mathbb{N}_\infty \qquad N^! C = C^{+\infty} \qquad N^? C \cong C^\infty$$

- Ordinary iteration simplifies

$$\varphi = [z, s] \implies (\!|\varphi|\!)_N(n) = s^n(z)$$

- Course-of-value iteration simplifies

$$\{\!|\varphi|\!\}_N(n) \cong \varphi\Big(\{\!|\varphi|\!\}_N(n-1), \dots, \{\!|\varphi|\!\}_N(0)\Big)$$

- Applications beyond Fibonacci & friends
  - Black-box components in software engineering
  - Empirics/statistics of dynamical systems

Special Case

Restriction to Simplest of Functors

- Peano functor

$$F = N = 1 + (-) : \mathbf{Set} \to \mathbf{Set}$$

- Data structures simplify

$$\mu N = \mathbb{N} \qquad NC = 1 + C$$

$$\nu N = \mathbb{N}_\infty \qquad N^! C = C^{+\infty} \qquad N^? C \cong C^\infty$$

- Ordinary iteration simplifies

$$\varphi = [z, s] \implies (\!|\varphi|\!)_N(n) = s^n(z)$$

- Course-of-value iteration simplifies

$$\{\!|\varphi|\!\}_N(n) \cong \varphi\Big(\{\!|\varphi|\!\}_N(n-1), \dots, \{\!|\varphi|\!\}_N(0)\Big)$$

- Applications beyond Fibonacci & friends
  - Black-box components in software engineering
  - Empirics/statistics of dynamical systems

Special Case

Restriction to Simplest of Functors

- Peano functor

$$F = N = 1 + (-) : \mathbf{Set} \to \mathbf{Set}$$

- Data structures simplify

$$\mu N = \mathbb{N} \qquad\qquad NC = 1 + C$$

$$\nu N = \mathbb{N}_\infty \qquad N^! C = C^{+\infty} \qquad N^? C \cong C^\infty$$

- Ordinary iteration simplifies

$$\varphi = [z, s] \implies (\!|\varphi|\!)_N(n) = s^n(z)$$

- Course-of-value iteration simplifies

$$\{\!|\varphi|\!\}_N(n) \cong \varphi\Big(\{\!|\varphi|\!\}_N(n-1), \dots, \{\!|\varphi|\!\}_N(0)\Big)$$

- Applications beyond Fibonacci & friends
  - Black-box components in software engineering
  - Empirics/statistics of dynamical systems

Special Case

Restriction to Simplest of Functors

- Peano functor

$$F = N = 1 + (-) : \mathbf{Set} \to \mathbf{Set}$$

- Data structures simplify

$$\mu N = \mathbb{N} \qquad\qquad NC = 1 + C$$

$$\nu N = \mathbb{N}_\infty \qquad N^! C = C^{+\infty} \qquad N^? C \cong C^\infty$$

- Ordinary iteration simplifies

$$\varphi = [z, s] \implies (\!|\varphi|\!)_N(n) = s^n(z)$$

- Course-of-value iteration simplifies

$$\{\!|\varphi|\!\}_N(n) \cong \varphi\Big(\{\!|\varphi|\!\}_N(n-1), \dots, \{\!|\varphi|\!\}_N(0)\Big)$$

- Applications beyond Fibonacci & friends
  - Black-box components in software engineering
  - Empirics/statistics of dynamical systems

Ordinary N-iteration as Loop Program

```
procedure iter(z, s, n):
 1 var state := z;
 2 for i := 1 to n do
 3   state := s(state)
 4 end;
 5 return state.
```

Evaluation

- Line 3 takes advantage of results needed only once
  - valid for ordinary iteration
  - invalid for course-of-value iteration
- COV iteration must remember more of input/output
  - finitely much? how much? how organized?
- General theory desirable

Ordinary N-iteration as Loop Program

```
procedure iter(z, s, n):
 1 var state := z;
 2 for i := 1 to n do
 3   state := s(state)
 4 end;
 5 return state.
```

Evaluation

- Line 3 takes advantage of results needed only once
  - valid for ordinary iteration
  - invalid for course-of-value iteration
- COV iteration must remember more of input/output
  - finitely much? how much? how organized?
- General theory desirable

Simulation Defined

Definition (State System, Factoring, Epi)

- A C-**state system** is a triple (S, σ, τ) with

  state space S
  abstraction $\sigma : N^{?}C \to S$
  transition $\tau : C \times S \to S$

- State system (S, σ, τ) **factors** $\varphi : N^{?}C \to C$ iff there is $\widetilde{\varphi} : S \to C$ such that

$$\begin{tikzcd}
 & C^{\infty} \arrow[dl, "\sigma"'] \arrow[dr, "\varphi"] & \\
S \arrow[rr, "\widetilde{\varphi}"'] & & C
\end{tikzcd}
\qquad
\begin{tikzcd}
 & S \arrow[r, "\langle\widetilde{\varphi},\mathrm{id}\rangle"] & C \times S \arrow[dr, "\tau"] & \\
C^{\infty} \arrow[ur, "\sigma"] \arrow[dr, "\langle\varphi,\mathrm{id}\rangle"'] \arrow[rrr, dotted, "\delta"] & & & S \\
 & C \times C^{\infty} \arrow[r, "\mathrm{cons}"'] & C^{\infty} \arrow[ur, "\sigma"'] &
\end{tikzcd}$$

- (S, σ, τ) is called **epi-state system** iff σ is epi
  - making $\widetilde{\varphi}$ unique

Simulation Defined

Definition (State System, Factoring, Epi)

- A C-**state system** is a triple (S, σ, τ) with

  state space S

  abstraction $\sigma : N^{?}C \to S$

  transition $\tau : C \times S \to S$

- State system (S, σ, τ) **factors** $\varphi : N^{?}C \to C$ iff there is $\widetilde{\varphi} : S \to C$ such that

$$
\begin{tikzcd}
& C^{\infty} \arrow[dl, "\sigma"'] \arrow[dr, "\varphi"] & \\
S \arrow[rr, "\widetilde{\varphi}"'] & & C
\end{tikzcd}
\qquad
\begin{tikzcd}
& S \arrow[r, "\langle\widetilde{\varphi},\mathrm{id}\rangle"] & C \times S \arrow[dr, "\tau"] & \\
C^{\infty} \arrow[ur, "\sigma"] \arrow[rrr, dotted, "\delta"] \arrow[dr, "\langle\varphi,\mathrm{id}\rangle"'] & & & S \\
& C \times C^{\infty} \arrow[r, "\mathrm{cons}"'] & C^{\infty} \arrow[ur, "\sigma"'] &
\end{tikzcd}
$$

- (S, σ, τ) is called **epi-state system** iff σ is epi
  - making $\widetilde{\varphi}$ unique

Simulation Defined

Definition (State System, Factoring, Epi)

- A C-**state system** is a triple (S, σ, τ) with

  state space S
  abstraction $\sigma : N^? C \to S$
  transition $\tau : C \times S \to S$

- State system (S, σ, τ) **factors** $\varphi : N^? C \to C$ iff there is $\widetilde{\varphi} : S \to C$ such that

$$
\begin{tikzcd}
 & C^\infty \arrow[dl, "\sigma"'] \arrow[dr, "\varphi"] & \\
S \arrow[rr, "\widetilde{\varphi}"'] & & C
\end{tikzcd}
\qquad
\begin{tikzcd}
 & S \arrow[r, "\langle\widetilde{\varphi}{,}\mathrm{id}\rangle"] & C \times S \arrow[dr, "\tau"] & \\
C^\infty \arrow[ur, "\sigma"] \arrow[rrr, dotted, "\delta"] \arrow[dr, "\langle\varphi{,}\mathrm{id}\rangle"'] & & & S \\
 & C \times C^\infty \arrow[r, "\mathrm{cons}"'] & C^\infty \arrow[ur, "\sigma"'] &
\end{tikzcd}
$$

- (S, σ, τ) is called **epi-state system** iff σ is epi
  - making $\widetilde{\varphi}$ unique

Simulation Proven

Theorem (State-Based Simulation)

A state system (S, σ, τ) that factors $\varphi : N^{?}C \to C$ can simulate it.

$$\{\!|\varphi|\!\}_N = \pi_1 \circ (\!|\underbrace{\langle \pi_1, \tau \rangle \circ \langle \widetilde{\varphi}, \mathrm{id}_S \rangle \circ [\sigma \circ \iota_1, \pi_2]}_{\rho}|\!)_N$$

Proof Idea.

Substitution into characteristic universal property. □

1. Introduction
   - General Theory
   - Special Case: Linear History
2. Simulation
   - Main Definitions
   - Limit Cases
   - Average Cases
3. Conclusion

A Whole Category of State Systems

Category of Factoring (Epi-)State Systems

Objects (epi-)state systems factoring φ

Morphisms $h : S_1 \to S_2$ such that

$$\begin{tikzcd} & C^\infty \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\ S_1 \arrow[rr, "h"'] & & S_2 \\ & C^\infty \arrow[ul, "\delta_1"] \arrow[ur, "\delta_2"'] & \end{tikzcd}$$

- Simultaneous coslices σ and δ under C^∞
- Initial object $(C^\infty, \mathrm{id}, \mathrm{cons})$ – maximal, **syntactic** system
  - history at face value, no abstraction
  - state space too large in practice
- Final object $(\mathrm{Coimg}(\varphi), \dots)$ – minimal, **semantic** system
  - epi only! (general case open)
  - answers question what must be remembered in theory
  - quotient structure too hard in practice

A Whole Category of State Systems

Category of Factoring (Epi-)State Systems

Objects (epi-)state systems factoring φ

Morphisms $h : S_1 \to S_2$ such that

$$\begin{tikzcd}
& C^\infty \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
S_1 \arrow[rr, "h"'] & & S_2 \\
& C^\infty \arrow[ul, "\delta_1"] \arrow[ur, "\delta_2"'] &
\end{tikzcd}$$

- Simultaneous coslices σ and δ under C^∞
- Initial object $(C^\infty, \mathrm{id}, \mathrm{cons})$ – maximal, **syntactic** system
  - history at face value, no abstraction
  - state space too large in practice
- Final object $(\mathrm{Coimg}(\varphi), \dots)$ – minimal, **semantic** system
  - epi only! (general case open)
  - answers question what must be remembered in theory
  - quotient structure too hard in practice

A Whole Category of State Systems

Category of Factoring (Epi-)State Systems

Objects (epi-)state systems factoring φ

Morphisms $h : S_1 \to S_2$ such that

$$\begin{tikzcd}
& C^\infty \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
S_1 \arrow[rr, "h"'] & & S_2 \\
& C^\infty \arrow[ul, "\delta_1"] \arrow[ur, "\delta_2"'] &
\end{tikzcd}$$

- Simultaneous coslices σ and δ under C^∞
- Initial object $(C^\infty, \mathrm{id}, \mathrm{cons})$ – maximal, **syntactic** system
  - history at face value, no abstraction
  - state space too large in practice
- Final object $(\mathrm{Coimg}(\varphi), \dots)$ – minimal, **semantic** system
  - epi only! (general case open)
  - answers question what must be remembered in theory
  - quotient structure too hard in practice

A Whole Category of State Systems

Category of Factoring (Epi-)State Systems

Objects (epi-)state systems factoring φ

Morphisms $h : S_1 \to S_2$ such that

$$\begin{tikzcd}
& C^\infty \arrow[dl, "\sigma_1"'] \arrow[dr, "\sigma_2"] & \\
S_1 \arrow[rr, "h"'] & & S_2 \\
& C^\infty \arrow[ul, "\delta_1"] \arrow[ur, "\delta_2"'] &
\end{tikzcd}$$

- Simultaneous coslices σ and δ under C^∞
- Initial object $(C^\infty, \mathrm{id}, \mathrm{cons})$ – maximal, **syntactic** system
  - history at face value, no abstraction
  - state space too large in practice
- Final object $(\mathrm{Coimg}(\varphi), \dots)$ – minimal, **semantic** system
  - epi only! (general case open)
  - answers question what must be remembered in theory
  - quotient structure too hard in practice

Universal Implementation for Bounded Memory

Definition (Regular Course-of-value Iteration)

Operation $\varphi : N^? C \to C$ is k-**regular** iff there is $\widehat{\varphi} : C^k \to C$ and $h \in C^k$ such that

$$\varphi = \widehat{\varphi} \circ \mathrm{take}(k) \circ \mathrm{append}(h)$$

Theorem

A FIFO buffer of size k gives rise to a state system factoring any k-regular operation.

$$S = C^k \qquad \sigma = \mathrm{take}(k) \circ \mathrm{append}(h)$$

$$\widetilde{\varphi} = \widehat{\varphi} \qquad \tau(c_0, (c_1, \dots, c_k)) = (c_0, \dots, c_{k-1})$$

Example (Fibonacci)

$$C = \mathbb{N} \qquad k = 2 \qquad h = (1, -1) \qquad \widehat{\varphi}(a, b) = a + b$$

Simulation of $\{\!|\varphi|\!\}_N = \mathrm{fib}$ specifies standard linear algorithm!

Universal Implementation for Bounded Memory

Definition (Regular Course-of-value Iteration)

Operation $\varphi : N^? C \to C$ is k-**regular** iff there is $\widehat{\varphi} : C^k \to C$ and $h \in C^k$ such that

$$\varphi = \widehat{\varphi} \circ \mathrm{take}(k) \circ \mathrm{append}(h)$$

Theorem

A FIFO buffer of size k gives rise to a state system factoring any k-regular operation.

$$S = C^k \qquad \sigma = \mathrm{take}(k) \circ \mathrm{append}(h)$$

$$\widetilde{\varphi} = \widehat{\varphi} \qquad \tau(c_0, (c_1, \dots, c_k)) = (c_0, \dots, c_{k-1})$$

Example (Fibonacci)

$$C = \mathbb{N} \qquad k = 2 \qquad h = (1, -1) \qquad \widehat{\varphi}(a, b) = a + b$$

Simulation of $\{\!|\varphi|\!\}_N = \mathrm{fib}$ specifies standard linear algorithm!

Universal Implementation for Bounded Memory

Definition (Regular Course-of-value Iteration)

Operation $\varphi : N^? C \to C$ is k-**regular** iff there is $\widehat{\varphi} : C^k \to C$ and $h \in C^k$ such that

$$\varphi = \widehat{\varphi} \circ \mathsf{take}(k) \circ \mathsf{append}(h)$$

Theorem

A FIFO buffer of size k gives rise to a state system factoring any k-regular operation.

$$S = C^k \qquad \sigma = \mathsf{take}(k) \circ \mathsf{append}(h)$$

$$\widetilde{\varphi} = \widehat{\varphi} \qquad \tau(c_0, (c_1, \dots, c_k)) = (c_0, \dots, c_{k-1})$$

Example (Fibonacci)

$$C = \mathbb{N} \qquad k = 2 \qquad h = (1, -1) \qquad \widehat{\varphi}(a, b) = a + b$$

Simulation of $\{\!|\varphi|\!\}_N = \mathsf{fib}$ specifies standard linear algorithm!

Summary

- Course-of-value (cov) iteration is a convenient, mostly conservative extension of ordinary iteration
  - linear case: discrete system dynamics with path dependence
- Cov iteration (histomorphisms) remembers subarguments *and* the corresponding results
  - conceptually infinitely much
  - linear case: all past I/O
  - generally difficult to compute in a loop
- State systems are, in a sense, homomorphic models of cov iteration
  - reduce to ordinary iteration
  - epi $\Longrightarrow$ unique model operation
  - category with axis initial$\leftrightarrow$final
  - retrieve standard algorithms for average (regular) case

Bibliography

Uustalu, Tarmo and Varmo Vene (1999). "Primitive (co)recursion and course-of-value (co)iteration, categorically". In: *Informatica* 10.1, pp. 5–26.

No (Obvious) Final State System Without Epi

$$
\begin{tikzcd}
K \arrow[r, "\kappa_2"] \arrow[d, "\kappa_1"'] & C^\infty \arrow[d, "\varphi"] \\
C^\infty \arrow[r, "\varphi"'] & C
\end{tikzcd}
\qquad
\begin{tikzcd}
K \arrow[r, "\kappa_1", "\kappa_2"'] & C^\infty \arrow[r, "\sigma"] \arrow[dr, "\varphi"'] & S \arrow[d, "\exists! \widetilde{\varphi}"] \\
 & & C
\end{tikzcd}
$$

No (Obvious) Final State System Without Epi

$$
\begin{tikzcd}
K \arrow[r, "\kappa_2"] \arrow[d, "\kappa_1"'] & C^\infty \arrow[d, "\varphi"] \\
C^\infty \arrow[r, "\varphi"'] & C
\end{tikzcd}
\qquad
\begin{tikzcd}
K \arrow[r, "\kappa_1", "\kappa_2"'] & C^\infty \arrow[r, "\sigma"] \arrow[d, "\sigma'"'] \arrow[dr, "\varphi"'] & S \arrow[d, "\exists!\widetilde{\varphi}"] \\
 & S' \arrow[r, "\widetilde{\varphi}'"'] & C
\end{tikzcd}
$$

No (Obvious) Final State System Without Epi

$$
\begin{tikzcd}
K \arrow[r, "\kappa_2"] \arrow[d, "\kappa_1"'] & C^\infty \arrow[d, "\varphi"] \\
C^\infty \arrow[r, "\varphi"'] & C
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
K \arrow[r, "\kappa_1", "\kappa_2"'] & C^\infty \arrow[r, "\sigma"] \arrow[d, "\sigma'"'] & S \arrow[d, "\exists!\widetilde{\varphi}"] \\
 & S' \arrow[ur, "\exists!h"] \arrow[r, "\widetilde{\varphi}'"'] & C
\end{tikzcd}
$$