Algebraic–Coalgebraic Recursion Theory of History-Dependent Dynamical System Models

Baltasar Trancón y Widemann, Michael Hauhs

Algebraic–Coalgebraic Recursion Theory of History-Dependent Dynamical System Models

Baltasar Trancón y Widemann[1][2] and Michael Hauhs[2]

[1] Programming Languages and Compilers
Ilmenau University of Technology, Germany
baltasar.trancon@tu-ilmenau.de

[2] Ecological Modelling
University of Bayreuth, Germany
michael.hauhs@uni-bayreuth.de

Abstract. We investigate the common recursive structure of history-dependent dynamic models in science and engineering. We give formal semantics in terms of a hybrid algebraic–coalgebraic scheme, namely course-of-value iteration. This theoretical approach yields categories of observationally equivalent model representations with precise semantic relationships. Along the initial–final axis of these categories, history dependence can appear both literally and transformed into instantaneous state. The framework can be connected to philosophical and epistemological discourse on one side, and to algorithmic considerations for computational modeling on the other.

1 Introduction

Models of system dynamics are a cornerstone of science and engineering. They relate the future of a system to its present and/or past. An obvious qualitative distinction is whether observation of the present (*state*) alone suffices to predict or modify the future, or whether information about the past (*history*) is necessary. This question can be discussed on the philosophical level, or on the mathematical level, potentially leading to theoretical frameworks and tools for the working scientist and engineer.

In this paper, we explore the mathematical option, and present a formalization that puts the two model classes on equal footing. Specifically, they shall be demonstrated to form not a dichotomy, but a continuum along the initial–final axis of suitable categories of models, constructed from first principles of algebraic–coalgebraic recursion theory.

In philosophical terms, this framework gives precise semantic relationships between more and less history-dependent models, and thus renders the two most common objections against history-dependent modeling obsolete: arguments from naïve reductionism (invoking Laplace's Daemon) and arguments from parsimony (invoking Occam's Razor).

We begin with motivating examples that demonstrate the pervasive occurrence, and the vastly different relative reputability, of the two modeling approaches across various disciplines. The purpose of this digression is to demonstrate the broad applicability of our proposed framework, which is hard to see directly from the fairly modest formal results. Readers more narrowly interested in theoretical matters are encouraged to skip ahead to section 2.

1.1 Basic Scientific Example: Simple Harmonic Oscillator

A simple harmonic oscillator is an ideal point mass m moving frictionlessly along a line and acted on by a restoring force proportional, with positive coefficient k, to its displacement x. An empirical study of (real approximations of) many such systems might reveal that, with good accuracy, *three* displacements observed at snapshots spaced equally in time, with a small delay δ, are related according to the following model formula:

$$x_{t+\delta} = \left(2 - \tfrac{k}{m}\delta^2\right)x_t - x_{t-\delta} \tag{1a}$$

On the other hand, with slightly less accuracy, *two* observations at snapshots with a small delay δ are related according to

$$\begin{pmatrix} x_{t+\delta} \\ v_{t+\delta} \end{pmatrix} = \begin{pmatrix} 1 & \delta \\ -\frac{k}{m}\delta & 1 \end{pmatrix} \begin{pmatrix} x_t \\ v_t \end{pmatrix} \tag{1b}$$

provided that the "virtual" observable $v = \mathrm{d}x/\mathrm{d}t$ is added to the data set.

From the fact that apparently each system has a period of $T = 2\pi/\omega$ where $\omega = \sqrt{k/m}$, one might get the intuition that the matrix entries in (1b) are actually linear approximations of trigonometric functions for $\delta \to 0$. Indeed, whereas the preceding models are only accurate for $\delta \ll T$, the following model gives the exact dynamics of the system:

$$\begin{pmatrix} x_{t+\delta} \\ v_{t+\delta} \end{pmatrix} = \begin{pmatrix} \cos(\omega\delta) & \omega^{-1}\sin(\omega\delta) \\ -\omega\sin(\omega\delta) & \cos(\omega\delta) \end{pmatrix} \begin{pmatrix} x_t \\ v_t \end{pmatrix} \tag{1c}$$

Elementary theoretical physics tells us that all three models given above contain a grain of the same truth, namely that they can be derived from the characteristic linear differential equation of the system:

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \omega^2 x = 0 \tag{2}$$

This differential equation has a family of solutions of the form $x_t = A\sin(\omega t + \varphi)$ for arbitrary A and φ. The models are then obtained as follows:

- Model (1c) by substitution of $t+\delta$ for t and various trigonometric identities.
- Model (1b) by linear approximation:

$$x_{t+\delta} \approx x_t + \delta v_t \qquad\qquad v_{t+\delta} \approx v_t + \delta a_t \tag{3a}$$

  where $a = \mathrm{d}v/\mathrm{d}t = \mathrm{d}^2x/\mathrm{d}t^2 = -\omega^2 x$ according to (2).
- Model (1a) by slightly different linear approximation eliminating v:

$$x_{t+\delta} \approx x_t + \delta v_{t+\delta} \qquad v_{t+\delta} \approx v_t + \delta a_t \qquad v_t \approx \frac{x_t - x_{t-\delta}}{\delta} \tag{3b}$$

1.2 Complex Scientific Example: ARMA

History dependence in scientific modeling can also take more assertive forms, where explicit dependence on past values is not only taken at face value, but featured as the principal methodological design concept.

The Box–Jenkins approach [1] focuses on the *auto-regressive moving average* (ARMA) class of stochastic models. These can be thought of as filters that add statistical autocorrelation to a discrete input signal in a controlled way, by linear dependence on past output (auto-regressive) and input (moving average) values.

$$y_t = \underbrace{\phi_1 y_{t-1} + \cdots + \phi_p y_{t-p}}_{\text{AR}} + x_t + \underbrace{\theta_1 x_{t-1} + \cdots + \theta_q x_{t-q}}_{\text{MA}} \tag{4a}$$

The theory is phrased as transformation of bilaterally infinite stochastic processes, although deterministic variants, with at best pseudorandom input, are used for actual simulation. In practice, the models are used both *directly*, to simulate data with a prescribed autocorrelation structure, and *inversely*, to estimate the model coefficients that describe observed output data optimally, by minimizing the variance of the implied, unobserved random input.

Linear combinations of past values are formulated neatly in an operator calculus on sequences, namely as formal power series of the *backshift* operator B, defined as $(Bx)_t = x_{t-1}$, which gives the following compact formula:

$$\underbrace{(1 - \Phi)}_{\text{AR}}\, y = \underbrace{(1 + \Theta)}_{\text{MA}}\, x \quad \text{where} \quad \Phi = \sum_{k=1} \phi_k B^k \quad \text{and} \quad \Theta = \sum_{k=1} \theta_k B^k \tag{4b}$$

In the basic form, both power series are finite, and the summations have upper bounds. But on one hand, there are transformations of finitary pure AR into infinitary pure MA models, and of finitary pure MA into infinitary pure AR models. And on the other hand, many kinds of real-world modeling problems call for extensions, in particular the ARIMA class, where the ouput of the ARMA filter is subsequently *integrated*, either a whole number of times (ARIMA proper) or *fractionally* (FARIMA) [2]. Since integration is linear and invertible, the resulting model can be viewed not only as a composition of two sequence transforms, but also as an ARMA model where a corresponding *differencing* step is composed with the auto-regressive part, in terms of a formal backward differencing operator:

$$\underbrace{(1 - \Phi)}_{\text{AR}} \underbrace{(1 - B)^d}_{\text{I}}\, y = \underbrace{(1 + \Theta)}_{\text{MA}}\, x \tag{4c}$$

Differencing can be raised to arbitrary real powers d by Newton's generalized binomial theorem:

$$(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k \qquad \text{where} \qquad \binom{d}{k} = \prod_{i=0}^{k-1} \frac{d - i}{1 + i}$$

This power series is easily seen to vanish only for positive integer d; hence the flagship models of FARIMA class, particularly effective for the structure of realistic time series data [5], depend on *actually infinite* history. Of course, simulations approximate by padding with zeroes, and luckily the consequent behavioral error tends to vanish in few steps.

1.3 Complex Engineering Example: TFM

The trace function method (TFM) [8] is a mathematically fundamental approach to behavioral description of software components. Components are taken to interact at their interface in discrete events. Data flows through the values of input and output variables, controlled by the environment and the component, respectively, at the time of an event.

The behavior of the component is represented in terms of sequences of events (*traces*). Traces come in two flavours: a *complete* trace has input and output values for all events; an *incomplete* trace omits output values for the latest event. Valid responses of the component are given as a collection of maps (optionally set-valued for nondeterminism) from incomplete traces to output values, one per output variable. The set of valid complete traces, the semantic object of behavior, is defined recursively: The empty trace is valid; a nonempty trace is valid if and only if its latest outputs can be reconstructed from its incomplete form, and the rest of the trace is valid.

As a toy example, a "stubborn" vending machine that offers a choice of either coffee (c) or tea (t) but, when asked for coffee the third time in a row, produces tea instead, can be described as a component with one input and one output variable, each of type $\{c, t\}$, and the output function

$$f(c,(c,c),(c,c),\dots) = t \qquad \text{otherwise} \qquad f(x_n,(y_{n-1},x_{n-1}),\dots) = x_n \tag{5}$$

where output precedes input, and the latest, incomplete event is leftmost.

TFM shines particularly for behavior that is far too complex and/or irregular to be discussed here in passing. See [8, 13] for worked-out examples.

1.4 Discussion

The succession of three oscillator models, as given in section 1.1, creates the impression of the growth of scientific knowledge along the following lines:

1. Empirical models establish candidates for causal relationships between past and future values of directly observable variables. Such models are necessarily approximations.
2. In theoretically informed models, history of directly observable variables is "explained away" by reference to auxiliary, indirectly observable variables. The rate of change of some variable often turns out to be a good candidate for an auxiliary variable; hence the usefulness of differential equations.
3. If done in the right way, approximative models can eventually be replaced by exact, albeit idealized, history-independent models: *dynamical systems.*

Classical physics has had its great successes in progressing from stage 1 (typically via stage 2) to stage 3 for many systems. On the other hand, most practically relevant models in self-styled "complex system" sciences seem to be stuck in stage 1. For instance, the Box–Jenkins framework, as given in section 1.2, is not only resilient to replacement by more state-oriented models, but actually spreading virulently across disciplines: Though originally developed for economics, it is considered state of the art in environmental sciences as well [4, 6]. The success of these models, their empirical and heuristical nature notwithstanding, is due to their pragmatic relevance as powerful forecasting tools [1].

In engineering applications, TFM even demonstrates a reversal of reputability of modeling with and without history dependence: A description in terms of traces of history is more abstract, and *precisely hence* more durable, reliable and valuable information than any more concrete, state-based "implementation". This view conforms to the software engineering doctrine that the "what" should be stated, not the "how"; here applied to the question of *memory*.

With the benefit of theoretical hindsight, the approximations (3a) and (3b) that justify the empirical relations (1a) and (1b), respectively, may seem strangely ad-hoc. But the situation in complex system sciences is often such that modeling and simulation precedes theory, or even may take its place indefinitely [7]. Hence, in absence of theoretical justification, we need a neutral approach to judge the epistemological quality of models. Since, as Rosen [9] observed, scientific modeling is essentially about the discovery of *recursive* functional relationships in data, thus elucidating how the future can be understood as entailed by the past, it seems only reasonable to turn to (categorial) recursion theory for answers.

2 Algebraic–Coalgebraic Recursion Theory

We recall some basic notions of categorial recursion theory. Since the key construct for the present work is from [15], we adopt that paper's notation with a few exceptions; in particular we write the more conventional $\iota_1, \ldots, \iota_n$ and $\pi_1, \ldots, \pi_n$ for n-ary coproduct injections and product projections, respectively. We also write $C^{a..b}$ for the set of streams (sequences) of Cs of lengths a to b, inclusively, where $a, b \in \mathbb{N} \cup \{\omega\}$. Note that the symbol ω henceforth refers to the limit ordinal, not the angular velocity as in section 1.1.

We consider algebras and coalgebras for certain simple endofunctors on **Set**. Unless explicitly restricted to **Set**, results generalize implcitly to other distributive categories, and endofunctors that satisfy the small print. We take the liberty to call any morphisms in this category "maps". We assume that, for a given functor F of interest, an initial F-algebra $(\mu F, \mathrm{in}_F : F(\mu F) \to \mu F)$ and a final F-coalgebra $(\nu F, \mathrm{out}_F : \nu F \to F(\nu F))$ exist. By Lambek's Lemma, in_F and its dual out_F are isomorphisms.

The following three definitions and their properties are a minimal excerpt of [15] for our present needs. Note that we have added references to applications in scientific modeling from our own work [3]. For a broader introduction to (co)iterative definitions arising from (co)algebras, see [10].

2.1 Iteration

The simplest recursion scheme that can be described by these structures is *iteration*: given a F-algebra $(C, \varphi : F(C) \to C)$, there is a unique F-algebra homomorphism $(\!|\varphi|\!)_F : (\mu F, \mathrm{in}_F) \to (C, \varphi)$. That $(\!|\_|\!)_F$ is in fact a recursion operator can be seen from the slighty transformed universal property:

$$(\!|\varphi|\!)_F = \varphi \circ F\big((\!|\varphi|\!)_F\big) \circ \mathrm{in}_F^{-1}$$

Uses of iteration in mathematically structured functional programming are ubiquitous. We have argued in [3] that iteration over functors of the form $\mathcal{A}(X) = A \times X + B$ is a systematic approach to the recursion scheme of typical *structure-oriented* scientific modeling questions, in particular *prediction* and *identification of initial conditions*.

2.2 Coiteration

The dual of iteration, (still?) distinctly less popular in computation theory, is *coiteration*: given a F-coalgebra $(C, \varphi : C \to F(C))$, there is a unique F-coalgebra homomorphism $[\![\varphi]\!]_F : (C, \varphi) \to (\nu F, \mathrm{out}_F)$. That $[\![\_]\!]_F$ is in fact a recursion operator can be seen from the slightly transformed universal property:

$$[\![\varphi]\!]_F = \mathrm{out}_F^{-1} \circ F\big([\![\varphi]\!]_F\big) \circ \varphi$$

Since coiteration produces infinite data, it is hardly ever considered outside the realms of process theory and lazy functional programming. We have argued in [3], based on an earlier discussion in [10], that coiteration over functors of the form $\mathcal{A}(X)$, dual to the above, is a systematic approach to the recursion scheme of typical *behavior-oriented* scientific modeling questions, in particular *symbolic dynamics* and the study of *irreversibility*.

2.3 Course-of-Value Iteration

Course-of-value (cov) iteration is a recursion scheme where the function value for a structured argument may depend on the function values for subarguments at any nesting depth, as opposed to ordinary iteration where dependency is limited to the function values for immediate, maximal subarguments. The key ingredient for the additional power is to augment the functor F by taking the product with some object C, intuitively a coloring. Henceforth, we assume C to be a *nonempty* set, avoiding uninteresting pathological cases. Since the product with a fixed object is a pervasive construct in the following, we abbreviate the functor $C \times (-)$ to C:

$$C(X) = C \times X \qquad\qquad C(h) = \mathrm{id}_C \times h$$

Then we can consider final CF-coalgebras and, most importantly, operations of type $\varphi : F(\nu CF) \to C$. Note that νCF parses as $\nu(CF)$. We define the cov

iteration of φ, written $\{\!|\varphi|\!\}_F : \mu F \to C$ as the following equation from [15]; the situation and type information is summarized in the diagram below:

$$\{\!|\varphi|\!\}_F = \underbrace{\pi_1 \circ \mathrm{out}_{CF}}_{\mathrm{top}_{CF}} \circ (\!|\underbrace{\mathrm{out}_{CF}^{-1} \circ \langle \varphi, \mathrm{id}_{F(\nu CF)} \rangle}_{\overline{\varphi}}|\!)_F \tag{6}$$

$$\begin{array}{ccccc} F(\mu F) & \xrightarrow{\mathrm{in}_F} & \mu F & & \\ {\scriptstyle F((\!|\overline{\varphi}|\!)_F)}\downarrow & & \downarrow{\scriptstyle (\!|\overline{\varphi}|\!)_F} & \searrow^{\{\!|\varphi|\!\}_F} & \\ F(\nu CF) & \xrightarrow[\overline{\varphi}]{} & \nu CF & \xrightarrow[\mathrm{top}_{CF}]{} & C \end{array}$$

That $\{\!|\_|\!\}_F$ is in fact a recursion operator can be seen from the, slightly convoluted, universal property

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

which is a key result of [15]. This recursion scheme is not only defined in terms of ordinary iteration, complicating the domain of evaluation from C to νCF, but vice versa also contains ordinary iteration over the domain C as a degenerate case: It is easy to see that $\{\!|\varphi \circ F(\mathrm{top}_{CF})|\!\}_F = (\!|\varphi|\!)_F$ for $\varphi : F(C) \to C$.

Note that in the data structure νCF of histories, the most recent results are on top.

2.4 Simple Examples

For the motivating examples of this paper and many others besides, consider the extremely simple functor $N(X) = 1 + X$. It has the following well-known initial algebra and final coalgebra:

$$\begin{array}{ll} \mu N = \mathbb{N} & \mathrm{in}_N = [0, \mathit{succ}] \\ \nu N = \mathbb{N} \cup \{\omega\} & \mathrm{out}_N = \mathit{pred} \end{array}$$

where *pred* is the partial predecessor function with $\mathit{pred}(\omega) = \omega$.

For cov iteration, we also need the C-colored final coalgebra, which consists of the nonempty finite and infinite streams over C together with the total *head* and partial *tail* operations.

$$\nu CN = C^{1..\omega} \qquad \mathrm{out}_{CN} = \langle \mathit{head}, \mathit{tail} \rangle$$

The domain of cov operations $N(\nu CN)$ is the set $C^{0..\omega}$ of finite and infinite streams over C, now additionally including the empty stream (). Because of the intrinsic left bias of the *head*/*tail* structure, history is encoded with the most recent element on the left.

Iteration and coiteration over N obey simple rules. Given a N-algebra (C, φ) with $\varphi = [z, s]$, we have:

$$(\!|\varphi|\!)_N(n) = s^n(z)$$

For instance, define powers of two as iteration of doubling, starting from one:

$$(\!|[one, double]|\!)_N(n) = 2^n$$

Given a N-coalgebra (C, φ) with $\varphi : C \to N(C)$ understood as the partial function $\varphi : C \nrightarrow C$, we have

$$[\![\varphi]\!]_N(x) = \sup\{n \in \mathbb{N} \mid \varphi^n(x) \text{ defined}\}$$

where $\sup \mathbb{N} = \omega$ and iteration of partial functions is strict. For a useful example of coiteration over N, first define the *domain restriction* of a total function $f : X \to Y$ to a region $U \subseteq X$ as:

$$f|_U : X \nrightarrow Y \qquad\qquad (f|_U)(x) = \begin{cases} f(x) & x \in U \\ \text{undefined} & x \notin U \end{cases}$$

Then we can define the *iterated logarithm*, as it appears in the study of algorithmic complexity, concisely as the coiteration of the logarithm, domain-restricted to a certain open real interval:

$$[\![\log|_{(1,\infty)}]\!]_N = \log^*$$

For examples in the realm of scientific modeling, consider a discrete-time dynamical system with state space S and single-step transition function $f : S \to S$. For a given initial state $x_0 \in S$, the operation $t = [x_0, f]$ gives rise to a N-algebra (S, t). Iteration produces precisely the *trajectory* of x_0, that is the infinite sequence of states obtained by repeated application of f:

$$(\!|t|\!)_N(n) = f^n(x_0)$$

Now consider a region $U \subseteq S$ and the partial step function $f|_U$ where the domain is restricted to U. This gives rise to a N-coalgebra $(U, f|_U)$. Coiteration classifies states in S according to *escape time*, that is the length of the longest prefix of the trajectory completely contained in U:

$$[\![f|_U]\!]_N(x) = \sup\{n \in \mathbb{N} \mid f^n(x) \in U\}$$

In particular, U is a *stationary solution* of the system if and only if $[\![f|_U]\!]_N(x) = \omega$ for all $x \in U$.

The paradigmatic example for a cov iteration is the Fibonacci sequence. Define an auxiliary operation $\varphi : \mathbb{N}^{0..\omega} \to \mathbb{N}$ such that

$$\varphi() = 0 \qquad\qquad \varphi(a) = a + 1 \qquad\qquad \varphi(a, b, \dots) = a + b \tag{7}$$

and conclude $fib = \{\!|\varphi|\!\}_N$. Note that there is a generalization of the system from $\mathbb{N}$ to $\mathbb{Z}$ that has several advantages; see below.

The simple harmonic oscillator model (1a) has a straightforward reconstruction as a cov iteration, completely analogous to the Fibonacci sequence: Fix the model parameters ω and δ to obtain the loose specification

$$\varphi(a, b, \dots) = \big(2 - (\omega\delta)^2\big)a - b$$

Now fix a reference time t and the first two observations $f() = x_t$ and $f(a) = x_{t+\delta}$. This specifies f uniquely. Then we have $x_{t+n\delta} = \{\!|\varphi|\!\}_N(n)$. Models (1b) and (1c), being essentially independent of history, are more adequately reconstructed as ordinary iterations. We leave the generating maps as exercises to the reader.

ARMA and TFM models are more complicated due to having input. Their reconstruction as cov iterations over functor N is possible (with exponential carrier and some higher-order functional programming; see [13]), but not straightforward and out of scope here. A more direct approach is by the composite functor $M = DN$ for some input object D, which yields complete and incomplete traces:

$$\nu CM = (C \times D)^{1..\omega} \qquad\qquad M(\nu CM) = D \times (C \times D)^{0..\omega}$$

For any cov trace function φ, the syntactic trace space νCM splits into *valid* traces $\mathrm{Img}\{\!|\varphi|\!\}_M$ and the complementary *invalid* traces. It is easy to see by induction that the recursive function $\{\!|\varphi|\!\}_M$ is determined completely by its behavior on valid traces.

3 General Theory: State Systems

Our theoretical approach hinges on the observation that definition (6) as given in [15] is not at all the only ordinarily iterative representation of a given cov iterative function; in fact, the collection of such representations has a rich categorial structure, covering all possible degrees of nominal history-dependence.

In all of the following, consider an endofunctor F and object C fixed.

3.1 Abstract State Systems

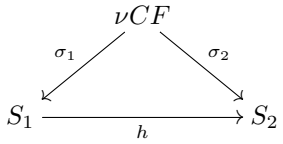
Definition 1 (Abstract State System). *A pair $(S, \sigma : \nu CF \to S)$ is called an **abstract state system**, if and only if top_{CF} factors through σ, that is, there is a complementary map $\psi : S \to C$ that makes the following diagram commute:*

$$\begin{array}{ccc} \nu CF & \xrightarrow{\sigma} & S \\ {\scriptstyle \mathrm{out}_{CF}}\downarrow & \searrow^{\mathrm{top}_{CF}} & \downarrow{\scriptstyle \psi} \\ CF(\nu CF) & \xrightarrow[\pi_1]{} & C \end{array} \tag{8}$$

*The object S is called **state space**, the maps σ/ψ are called **state abstraction/valuation**, respectively.*

For the rationale of not stating the valuation ψ explicitly in the definition of a state system, see Lemma 4 below and Corollary 14 at the end of this section.

Definition 2 (Abstract State System Homomorphism). *A homomorphism between abstract state systems (S_1, σ_1) and (S_2, σ_2) is defined in the obvious way as a map $h : S_1 \to S_2$ such that the following diagram commutes:*



(9)

Abstract state systems and their homomorphisms give rise to a category $\mathbf{State}(F, C)$ (of coslices under νCF) with the obvious identity and composition.

Definition 3 (Epic Abstract State System). *An abstract state system (S, σ) is called **epic** if and only if σ is an epimorphism in the underlying category.*

We shall argue in the following that epic state systems are the "right" ones for various mathematical and epistemological reasons. Note that there are several subclasses of special epimorphisms, such as *regular*, *strong*, and *split* epimorphisms, all of which notoriously coincide in the base category **Set**, obscuring the subtle differences. Because there is no experience with relevant examples in other categories, we reserve judgement which exactly is the appropriate class for epic state systems in full generality. Most of the following definitions use only the characteristic property of ordinary epimorphisms e, namely

$$f \circ e = g \circ e \implies f = g$$

for all possible maps f, g but Lemma 10 also assumes that F preserves the class of epimorphisms in question, which is granted only for split epimorphisms in arbitrary categories, but generally in **Set** [14].

Lemma 4. *An epic state system determines the state valuation ψ uniquely. Homomorphisms between epic state system are unique epimorphisms.*

Hence the restriction of $\mathbf{State}(F, C)$ to epic systems gives a full, thin subcategory $\mathbf{EpicState}(F, C)$.

Lemma 5. *The categories* (**Epic**)$\mathbf{State}(F, C)$ *have (the same) initial objects, namely the state system $(\nu CF, \mathrm{id}_{\nu CF})$, with the unique valuation top_{CF}, and the unique homomorphism to any abstract state system (S, σ) equalling σ.*

Lemma 6. *At least over* **Set**, *the category* $\mathbf{State}(F, C)$ *has generally no final objects.*

Proof. Consider the initial abstract state system $\mathcal{I} = (\nu CF, \mathrm{id}_{\nu CF})$. Add a distinguished element to obtain the system $\mathcal{I}' = (\nu CF + 1, \iota_1)$ which clearly satisfies (8). Now consider any abstract state system $\mathcal{S} = (S, \sigma)$. The morphisms $\mathrm{Hom}(\mathcal{I}', \mathcal{S})$ in $\mathbf{State}(F, C)$ are easily seen to correspond one-to-one to $\mathrm{Hom}(1, S)$ in **Set** and hence to S itself, namely by

$$\mathrm{Hom}(\mathcal{I}', \mathcal{S}) = \{[\sigma, i] \mid i \in \mathrm{Hom}(1, S)\}$$

where the first component σ is fixed by Lemma 5. Hence if a final abstract state system existed, it would need to have a one-element state space, so (8) would imply $\mathrm{top}_{CF}(x) = \mathrm{top}_{CF}(x')$ for all $x, x' \in \nu CF$, which is generally false. □

Lemma 7. *At least over* **Set**, *the category* **EpicState**(F, C) *has a final object, namely the state system* (C, top_{CF}), *with the valuation* id_C, *and the unique homomorphism from any abstract state system* (S, σ) *equalling its valuation* ψ.

This last result, while not very useful in itself, gives hints for the construction of more richly structured final systems; see section 4.2 below.

3.2 Concrete State Systems

Definition 8 (Concrete State System). *A triple* $(S, \sigma, \tau : F(S) \to S)$, *where* (S, σ) *is an abstract state system, is called a* ***concrete state system*** *for a cov operation* $\varphi : F(\nu CF) \to C$, *or* φ*-system for short, with* ***state transition*** τ, *if and only if the following* (F*-algebra homomorphism) diagram commutes:*

$$\begin{array}{ccc} F(\nu CF) & \xrightarrow{F(\sigma)} & F(S) \\ {\scriptstyle \overline{\varphi}}\downarrow & & \downarrow{\scriptstyle \tau} \\ \nu CF & \xrightarrow[\sigma]{} & S \end{array} \tag{10}$$

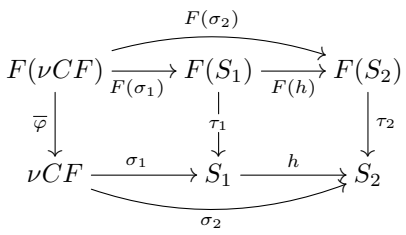
Definition 9 (Concrete State System Homomorphism). *A homomorphism between concrete state systems* (S_1, σ_1, τ_1) *and* (S_2, σ_2, τ_2) *is defined, in the obvious way, as a map* $h : S_1 \to S_2$ *that is both an abstract state homomorphism between* (S_1, σ_1) *and* (S_2, σ_2) *and a* F*-algebra homomorphism between* (S_1, τ_1) *and* (S_2, τ_2)*; that is, the following diagram additionally commutes:*

$$\begin{array}{ccc} F(S_1) & \xrightarrow{F(h)} & F(S_2) \\ {\scriptstyle \tau_1}\downarrow & & \downarrow{\scriptstyle \tau_2} \\ S_1 & \xrightarrow[h]{} & S_2 \end{array} \tag{11}$$

These definitions give rise to a (not necessarily full) subcategory **State**(φ) of **State**(F, C). The following lemma shows that the corresponding subcategory **EpicState**(φ) of **EpicState**(F, C) is full.

Lemma 10. *The additional consistency condition on concrete state system homomorphisms is essentially redundant for epic state systems; assuming* F *preserves enough epimorphisms, any map between state spaces of epic systems satisfying* (9) *automatically satisfies* (11).

Proof. Consider the diagram



where the inner left and outer quadrangle commute by (10), the triangles commute by (9), and the inner right quadrangle is the goal (11). A short diagram chase yields $h \circ \tau_1 \circ F(\sigma_1) = \tau_2 \circ F(h) \circ F(\sigma_1)$. Since σ_1 is an epimorphism, so is $F(\sigma_1)$ under mild assumptions as discussed above; conclude $h \circ \tau_1 = \tau_2 \circ F(h)$. □

Lemma 11. *The initial abstract state system in* (**Epic**)**State**(F, C) *extends to an initial concrete state system in in* (**Epic**)**State**(φ)*, namely* $(\nu CF, \mathrm{id}_{\nu CF}, \overline{\varphi})$.

By contrast, the dual problem of final concrete state systems is relatively complex. Results are few and mostly negative, and many questions remain open.

Lemma 12. *The final epic abstract state system does not generally extend to concrete state system, let alone a final one.*

Proof. Assume there is a φ-system $(C, \mathrm{top}_{CF}, \tau)$. Then (10) implies that $\tau \circ F(\mathrm{top}_{CF}) = \mathrm{top}_{CF} \circ \overline{\varphi} = \varphi$. That is, φ factors through $F(\mathrm{top}_{CF})$. This is not only generally false, but defies the very purpose of cov iteration, covering only the degenerate case of ordinary iteration over C, as discussed in section 2.3. □

The constructive disproof of final objects for the abstract case **State**(F, C) in Lemma 7 does not carry over to the concrete case **State**(φ). A family of nontrivial and well-known examples for final concrete epic state systems will be given below in section 4.2. No such example for the non-epic case is known, leaving the problem open.

3.3 Simulation by State Systems

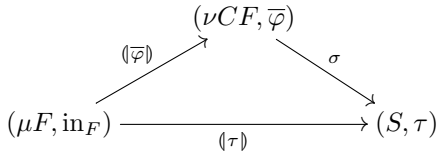
Now we can clarify the notion of representation introduced at the start of this section, namely in terms of simulation.

Theorem 13 (Simulation). *Let $\mathcal{S} = (S, \sigma, \tau)$ be a φ-system for a fixed cov operation φ. Let ψ be any suitable state valuation. Then the cov iteration of φ is simulated by ordinary iteration in terms of $\mathcal{S}$.*

$$\{\!|\varphi|\!\}_F = \psi \circ (\!|\tau|\!)_F$$

Note that for the initial φ-system from Lemma 11 this equation reduces to (6).

Proof. Consider the following diagram in the category of F-algebras, which commutes by initiality:



Hence back in the underlying category:

$$\psi \circ (\!|\tau|\!)_F = \psi \circ \sigma \circ (\!|\overline{\varphi}|\!)_F = \mathrm{top}_{CF} \circ (\!|\overline{\varphi}|\!)_F = \{\!|\varphi|\!\}_F$$ □

```
var x : S
x := ε
forever
  output ψ(x)
  x := ▷(x)
```

Fig. 1. Enumeration in concrete state systems over N, pseudocode

Corollary 14. *The choice of a particular state valuation ψ for a given cov operation φ is irrelevant for the purpose of simulation.*

This concludes the "universal" part of the theory. For particular choices of functor F and classes of operations φ, results are greatly more specific and practically useful.

4 Specific Theory: Autonomous Systems

4.1 Stream-Like State Spaces

For the special case $F = N$ already discussed in section 2.4, a closer look reveals details hinting at a more concrete design strategy for state-based models (automata). Up to natural isomorphisms, the operations of a φ-system are of the following types:

$$\varphi : C^{0..\omega} \to C \qquad \sigma : C^{0..\omega} \to S$$
$$\overline{\varphi} : C^{a..b} \to C^{a'..b'} \qquad \tau : 1 + S \to S$$

The F-algebra operation $\overline{\varphi}$ acts predictably on stream lengths, always adding one element. Hence it can be soundly given many different types, with arbitrarily wide integer bounds $a' \leq a{+}1$ and $b' \geq b{+}1$. Usage will be clear from the context.

The state transition is conveniently decomposed as $\tau = [\varepsilon, \triangleright]$ with $\varepsilon \in S$ and $\triangleright : S \to S$. Thus an algorithm that enumerates the values of $\{\!|\varphi|\!\}_N$ takes the form of a stream generator loop, as depicted in Fig. 1.

Lemma 15. *For a concrete φ-state system over N we have for all $n \geq 0$:*

$$\triangleright^n \circ \sigma = \sigma \circ \overline{\varphi}^n \qquad \{\!|\varphi|\!\}_N(n) = (\psi \circ \triangleright^n)(\varepsilon) = (\varphi \circ \overline{\varphi}^n)()$$

Concrete state systems are of particular interest, as theoretical objects rather than as algorithmic specifications, when the structure of their state space S is significantly simpler than $C^{1..\omega}$. For interpretation as physical systems, state spaces with simple product structure, which can be read as a small collection of independently observable variables, are preferred; see the oscillator example in section 1.1. By contrast, interpretation as an automaton prefers a simple coproduct structure, read as a finite enumeration of alternative states.

4.2 Dependency Patterns

Definition 16 (Bounded/Regular Iteration). *A cov operation $\varphi : C^{0..\omega} \to C$ is called k-**bounded**, for $k > 0$, if and only if*

$$\varphi = \varphi \circ take(k)$$

*where $take(k)$ takes the k first elements of its argument stream, or less if insufficient. It is called k-**regular** if and only if there is a surrogate history $\mathfrak{h} \in C^k$ such that*

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

where $append(s)(t)$ appends the stream s to t.

1-bounded cov iteration coincides with ordinary iteration. We write $\widehat{\varphi} : C^{0..k} \to C$ and $\widehat{\varphi} : C^k \to C$ for the domain restrictions of φ in bounded and regular systems, respectively. The auxiliary operation $take(k) : C^{a..b} \to C^{a'..b'}$ is polymorphic with $a' \leq \min(a, k)$ and $b' \geq \min(b, k)$.

Lemma 17. *Any k-bounded operation is $(k+1)$-bounded. Any k-regular operation is k-bounded.*

The TFM example (5) is 3-bounded, as evident from the ellipsis $(\dots)$ in the defining equation. The Fibonacci operation is a prototypic example of a 2-regular operation, see below. A cov operation may depend on point-wise finitely *many* elements without being k-bounded for any k (analogous to continuity versus uniform continuity); for instance, consider the well-known recursive definition of Bell numbers [11],

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$$

where each value depends on *all* of the preceding. Note that this example is *potentially infinitary*, as opposed to the actually infinitary FARIMA example.

The theory of φ-systems for bounded φ is particularly fruitful. In order to exploit it, we need to introduce some machinery for stream computation by bounded C-coiteration.

Definition 18 (Bounded Lookahead). *The family $(\searrow^n)$ of **lookahead** operators for all $n \geq 0$, that take a pair of maps $f : A \to A$ and $g : A \to B$ to a map $(g \searrow^n f) : A \to B^n$, is defined inductively as:*

$$g \searrow^0 f = \langle\rangle \qquad\qquad g \searrow^{n+1} f = cons \circ \langle g \circ f^n, g \searrow^n f \rangle$$

That is, in closed form, $g \searrow^n f = \langle g \circ f^{n-1}, \dots, g \circ f^0 \rangle$. Note the right-to-left "historical" enumeration order; the corresponding left-to-right enumeration would be given by $take(n) \circ [\![\langle g, f \rangle]\!]_C$.

Lemma 19. *Lookahead operators obey the following laws:*

$$(g \searrow^n f) \circ f = take(n) \circ (g \searrow^{n+1} f)$$
$$(\forall\, m < n.\, g \circ f^m \circ e = k \circ i^m \circ h) \implies (g \searrow^n f) \circ e = (k \searrow^n i) \circ h$$

Corollary 20. *As special cases of the latter we obtain:*

$$(g \searrow^n f) \circ f = (g \circ f) \searrow^n f$$
$$(\forall\, m < n.\, f^m \circ e = k \circ i^m) \implies (g \searrow^n f) \circ e = (g \circ k) \searrow^n i$$

Definition 21 (FIFO System). *A first-in-first-out buffer of $k > 0$ elements of C gives rise to a canonical concrete state system candidate, the k-**fifo** system, for any k-bounded cov operation φ:*

$$S_{[k]} = C^k \qquad \sigma_{[k]} = \mathit{head} \searrow^k \overline{\varphi}$$
$$\varepsilon_{[k]} = (\widehat{\varphi} \searrow^k \overline{\varphi})() \qquad \triangleright_{[k]} = \mathit{cons} \circ \langle \widehat{\varphi}, \mathit{take}(k-1) \rangle$$

Note that $\overline{\varphi}$ is used polymorphically as a map on $C^{0..\omega}$ and $C^{1..\omega}$ in the definitions of $\epsilon_{[k]}$ and $\sigma_{[k]}$, respectively. These can be made precise as $\overline{\varphi} \circ \iota_2$ and $\iota_2 \circ \overline{\varphi}$, respectively, where $\iota_2 : C^{1..\omega} \to C^{0..\omega}$ is the natural inclusion.

Lemma 22. *The fifo operations have alternative forms:*

$$\sigma_{[k]} = \mathit{take}(k) \circ \overline{\varphi}^{k-1} \qquad \varepsilon_{[k]} = \overline{\varphi}^k() \qquad \tau_{[k]} = \mathit{take}(k) \circ \overline{\varphi}$$

This seems to suggest that a "lookbehind" abstraction $\mathit{take}(k)$ might be more straightforward than the "lookahead" $\sigma_{[k]}$. However, that would unnecessarily complicate the construction of homomorphisms; see Theorem 25 below.

Corollary 23.

$$\widehat{\varphi} \circ \sigma_{[k]} = \mathit{head} \circ \overline{\varphi}^k$$

Now we can verify our educated guess.

Theorem 24 (FIFO Simulation). *If φ is k-bounded, then the k-fifo system is a valid φ-system with valuation $\psi_{[k]} = \pi_k$.*

Proof. It is easy to see that (8) is satisfied:

$$\psi_{[k]} \circ \sigma_{[k]} = \pi_k \circ (\mathit{head} \searrow^k \overline{\varphi}) = \mathit{head} \circ \overline{\varphi}^0 = \mathrm{top}_{CN}$$

In the following we use the simplified fifo operations of Lemma 22. A low-level proof in terms of lookahead laws is also possible, and will be required for the more advacned Theorem 25 below. To prove that (10) is satisfied, distinguish cases of F. On one hand:

$$(\sigma_{[k]} \circ \overline{\varphi})() = (\mathit{take}(k) \circ \overline{\varphi}^k)() = \varepsilon_{[k]}$$

On the other hand, exploiting polymorphism as discussed:

$$\begin{aligned}
\sigma_{[k]} \circ \overline{\varphi} \circ \iota_2 &= \sigma_{[k]} \circ \overline{\varphi} \\
&= \mathit{take}(k) \circ \overline{\varphi} \circ \overline{\varphi}^{k-1} \\
&= \mathit{take}(k) \circ \mathit{cons} \circ \langle \widehat{\varphi} \circ \mathit{take}(k), \mathrm{id}_{C^{k..\omega}} \rangle \circ \overline{\varphi}^{k-1} \\
&= \mathit{cons} \circ \langle \widehat{\varphi} \circ \mathit{take}(k), \mathit{take}(k-1) \rangle \circ \overline{\varphi}^{k-1} \\
&= \mathit{cons} \circ \langle \widehat{\varphi}, \mathit{take}(k-1) \rangle \circ \mathit{take}(k) \circ \overline{\varphi}^{k-1} \\
&= \triangleright_{[k]} \circ \sigma_{[k]}
\end{aligned}$$

□

```
var x : C[1 .. k]
for i in k to 1
  x[i] := φ̂(x[i + 1], ..., x[k])
forever
  output x[k]
  x := φ̂(x[1], ..., x[k]), x[1], ..., x[k − 1]
```

Fig. 2. Enumeration in fifo systems over N, pseudocode

```
var x : ℤ[1 .. 2]
x := 1, 0
forever
  output x[2]
  x := (x[1] + x[2]), x[1]
```

Fig. 3. Enumeration in Fibonacci fifo system, pseudocode

The resulting, specialized enumeration algorithm is depicted in Fig. 2. In the case of the Fibonacci operation with $k = 2$ and $\widehat{\varphi}$ defined as in (7), we obtain the familiar iterative algorithm by straightforward code specialization, depicted in Fig. 3.

The existence of fifo systems is merely a new formalization of a well-known algorithmic technique. The added value of the theory explored here is demonstrated by the following novel, surprisingly strong characterization of semantic adequacy.

Theorem 25 (FIFO Finality). *If a fifo φ-system is epic, then it is final in* **EpicState**(φ). *The unique homomorphism from any other φ-system $(S, \sigma, [\varepsilon, \triangleright])$ with valuation ψ is $h = \psi \searrow^k \triangleright$.*

Note that there is no obvious *take*-based expression for h in analogy to Lemma 22.

Proof. By Lemma 4, it suffices to show that h is a homomorphism (†). We have

$$\begin{aligned} h \circ \sigma &= (\psi \searrow^k \triangleright) \circ \sigma \\ &= (\psi \circ \sigma) \searrow^k \overline{\varphi} && \text{(Lemma 15, Corollary 20)} \\ &= \mathit{head} \searrow^k \overline{\varphi} = \sigma_{[k]} && (8) \end{aligned}$$

thus verifying that h satisfies (9). Regarding (11) we have on one hand:

$$\begin{aligned} h(\varepsilon) &= (\psi \searrow^k \triangleright)(\varepsilon) \\ &= (\varphi \searrow^k \overline{\varphi})() = \varepsilon_{[k]} && \text{(Lemma 15 \& 19)} \end{aligned}$$

By Lemma 15, observe that

$$\psi \circ \triangleright^k \circ \sigma = \psi \circ \sigma \circ \overline{\varphi}^k$$

$$\begin{aligned}
&(8) && = \mathit{head} \circ \overline{\varphi}^k \\
&(\text{Corollary 23}) && = \widehat{\varphi} \circ \sigma_{[k]} = \widehat{\varphi} \circ h \circ \sigma
\end{aligned}$$

With σ epi (†) conclude $\psi \circ \triangleright^k = \widehat{\varphi} \circ h$. Then on the other hand:

$$\begin{aligned}
h \circ \triangleright &= (\psi \searrow^k \triangleright) \circ \triangleright \\
(\text{Lemma 19}) \qquad &= \mathit{take}(k) \circ (\psi \searrow^{k+1} \triangleright) \\
&= \mathit{take}(k) \circ \mathit{cons} \circ \left\langle \psi \circ \triangleright^k, (\psi \searrow^k \triangleright) \right\rangle \\
&= \mathit{take}(k) \circ \mathit{cons} \circ \langle \psi \circ \triangleright^k, h \rangle \\
(\text{above}) \qquad &= \mathit{take}(k) \circ \mathit{cons} \circ \langle \widehat{\varphi} \circ h, h \rangle \\
&= \mathit{take}(k) \circ \mathit{cons} \circ \langle \widehat{\varphi}, \mathrm{id}_{C^k} \rangle \circ h \\
&= \mathit{cons} \circ \left\langle \widehat{\varphi}, \mathit{take}(k-1) \right\rangle \circ h \\
&= \triangleright_{(k)} \circ h
\end{aligned}$$

Together conclude that h satisfies (11). □

As a concrete example, we verify that the Fibonacci fifo system over $C = \mathbb{Z}$ is final: for every pair $(a, b) \in \mathbb{Z}^2$, we find that $\sigma_{[2]}(b, a - b, \dots) = (a, b)$.

The fairly elegant proof makes use of the epimorphism property of σ in crucial ways, marked with (†); thus the restriction to $\mathbf{EpicState}(\varphi)$ is essential. There is no obvious strategy how to generalize the result to final objects in $\mathbf{State}(\varphi)$. Additionally, a "moral converse" of the theorem holds also.

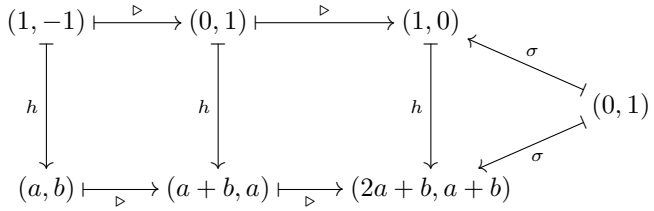
Lemma 26. *Non-epic fifo φ-systems are not generally final, in* $\mathbf{State}(\varphi)$.

Proof. As a counterexample, consider the Fibonacci sequence over $C = \mathbb{N}$ instead of $\mathbb{Z}$. We can give a hierarchy of four distinct, nested valid φ-systems with state $S_0 \supset S_1 \supset S_2 \supset S_3$, namely

$$\begin{aligned}
S_0 &= \mathbb{Z}^2 \\
S_1 &= \{(a, b) \in S_1 \mid a \geq 0; a + b \geq 0\} \\
S_2 &= \{(a, b) \in S_1 \mid a \geq 0; b \geq 0\} = \mathbb{N}^2 \\
S_3 &= \{(a, b) \in S_1 \mid a \geq b; b \geq 0\} = \operatorname{Img} \sigma_{[2]}
\end{aligned}$$

arising from the fifo system over $\mathbb{Z}$ discussed above. For S_2 and S_3, the valuation $\psi_{[2]} = \pi_2$ is already $\mathbb{N}$-valued and determined uniquely. For S_0 and S_1, we can choose arbitrary valuations for elements outside S_2 without violating (8).

The system S_3 is epic, but the canonical $\mathbb{N}$-valued fifo system S_2 is not. Neither S_2 nor S_3 is final: Assume, for contradiction, a homomorphism $h : S_{0/1} \to S_{2/3}$. Now observe that the following point diagram, where the squares and the triangle are instantiations of (11) and (9), respectively, must commute.



So $h(1,-1) = (a,b)$ implies $2a + b = 1$ and $a + b = 0$, which has no solution in $a, b \in \mathbb{N}$. □

Lemma 27. *If the k-fifo φ-system is epic, then the corresponding $(k+1)$-fifo φ-system is generally not.*

This straightforward observation clarifies that the parameter k of a final epic fifo φ-system is an objective measure of the order of history dependence in φ.

4.3 Regularization

Regularity is a stronger condition on cov operations than boundedness. We give some simple characterizations of regular operations.

Theorem 28 (Regularization). *If, for a k-bounded cov operation φ, the state transition $\triangleright_{[k]}$ is invertible, then there is an equivalent k-regular cov operation φ' such that $\{\!|\varphi|\!\}_N = \{\!|\varphi'|\!\}_N$.*

Proof. Choose $\varphi' = \varphi \circ \mathit{take}(k) \circ \mathit{append}(\mathfrak{h})$ with $\mathfrak{h} = \triangleright_{[k]}^{-k}(\varepsilon_{[k]})$. It suffices to show that the respective fifo systems coincide. Since $\mathit{take}(k) \circ \mathit{append}(\mathfrak{h})$ has no effect on the fifo state space C^k, the clause $\triangleright_{[k]} = \triangleright'_{[k]}$ is trivial. For the $\varepsilon_{[k]}$ clause, we have by Lemma 22:

$$\begin{aligned}
\varepsilon'_{[k]} &= \overline{\varphi'}^k() \\
&= (\mathit{take}(k) \circ \overline{\varphi}^k)(\mathfrak{h}) \\
&= (\mathit{take}(k) \circ \overline{\varphi}^k \circ \triangleright_{[k]}^{-k})(\varepsilon_{[k]}) \\
\text{(Lemma 22)} \qquad &= (\triangleright_{[k]}^{k} \circ \triangleright_{[k]}^{-k})(\varepsilon_{[k]}) = \varepsilon_{[k]}
\end{aligned}$$

□

For the concrete example of the Fibonacci operation, we find that $\triangleright_{[2]}$ is specified by the invertible matrix $\left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)$. The operation as defined above is its own regularization with $\mathfrak{h} = (1, -1)$. Note that there are equivalent definitions which are distinct from their regularization, for instance with the more commonly found (such as in [15]) second clause $\varphi(a) = 1$ instead of $a + 1$.

More algebraic structure on C gives more powerful results.

Corollary 29. *If $\widehat{\varphi} : C^k \to C$ is a linear form over a scalar field C, with coefficients $\alpha_1, \ldots, \alpha_k$, then $\triangleright_{[k]}$ is linear, and φ regularizes if and only if $\alpha_k \neq 0$.*

This result subsumes all finitary ARMA-like models.

5 Conclusion

The role of time in science has been hotly debated since antiquity, and quite remains so today [12]. The cov iteration approach to discrete-time history-dependent models goes a long way in retelling the meta-level discourse in a down-to-earth, formally precise and useful style. The categories (**Epic**)**State**(φ) of concrete systems span a solution space of hypothetical state-based realizations of a fixed black-box model with observable function $\{\!|\varphi|\!\}_F$.

At one extreme, the initial system is a purely syntactic solution which takes history at face value. It is always given trivially, but yields little scientific insight, and is also deficient as an algorithmic specification, because it prescribes a *space leak*: iterated invocation of the history constructor out_F^{-1} makes state representations grow boundlessly, even if the computation could also easily be performed on bounded space, such as by a fifo system. The sheer size and coalgebraic structure of νCF as a datatype may also be distressing for the working modeler unfamiliar with coalgebraic techniques.

At the other extreme, a final system, if such a thing exists, is a purely semantic, fully abstract solution, which makes only the empirically necessary distinctions, and is hence the "holy grail" of model semantics and epistemology. Final systems can reconstruct and justify established algorithmic design techniques of computational modeling.

Since the quotient structure of a final system may be complex and hard to find and work with, the practice of modeling often deals with intermediate forms, which have a more compact and regular state space than the initial rendering, yet tractable transition and valuation rules. In particular, product-shaped state spaces welcome the interpretation of their projections as independent, objectively real variables, both analytically in science and synthetically in engineering.

Epic state systems are of particular interest. From the formal perspective, they come with vastly more benign properties. From the epistemological perspective, they can be justified by an invocation of Occam's Razor: states are not to be multiplied without necessity (here for simulation). On the other hand, epic homomorphisms, that is unique surjective structure-preserving maps $h : S_1 \to S_2$, can be understood as precise semantic relationships between alternatives of hypothetical state, by turning state space S_1 into a possibly redundant but complete system of representatives for the more abstract but elusive state space S_2. This view renders philosophical arguments against the structure of S_1 as a datatype obsolete. It is foreseeable that advanced aspects of model analysis can be integrated into the picture as equational specifications of such homomorphisms. Furthermore, we point out other (co)limit constructions than initial/final objects in the thin category of epic state systems as an interesting open problem.

The recursion scheme of the elementary functor N is sufficient for many textbook examples, as well as for straightforward modeling of discrete-time systems without input. It comes with an extensive theory of canonical, even final systems, with well-understood algorithmic properties. It remains to be seen whether behaviorally more complex applications are better dealt with by reduction to this base case, or by extension of the theory to more advanced functors.

Bounded and regular operations are of particular algorithmic interest, because they do away with space leaks and base cases, respectively. We expect that algorithmic treatment of more complicated patterns of history dependence is also possible, and can lead to formally derived implementations of TFM. On the other hand, dependency on actually infinite histories is a feature of the cov operation only, not of the generated recursive function. This should serve as a warning for the philosophical discourse to properly distinguish the theoretical role of the former from the empirical role of the latter, such as in inverse FARIMA model estimation.

And finally, we are happy to have given yet another meaning to the phrase "universal coalgebra: a theory of systems."

References

1. Box, G.E., Jenkins, G.M.: Time series analysis: Forecasting and control. Holden–Day, San Francisco (1970)
2. Granger, C.W.J., Joyeux, R.: An introduction to long-memory time series models and fractional differencing. Journal of Time Series Analysis 1, 15–30 (1980)
3. Hauhs, M., Trancón y Widemann, B.: Applications of algebra and coalgebra in scientific modelling, illustrated with the logistic map. Electronic Notes in Theoretical Computer Science 264(2), 105–123 (2010)
4. Hipel, K.W., McLeod, A.I.: Time Series Modelling of Water Resources and Environmental Systems. Elsevier (1994)
5. Montanari, A., Rosso, R., Taqqu, M.S.: Fractionally differenced ARIMA models applied to hydrologic time series: Identification, estimation, and simulation. Water Resources Research 33(5), 1035–1044 (1997)
6. Montanari, A., Rosso, R., Taqqu, M.S.: A seasonal fractional ARIMA model applied to the Nile river monthly flows at Aswan. Water Resources Research 36(5), 1249–1259 (2000)
7. Peters, R.H.: A critique for ecology. Cambridge University Press (1991)
8. Quinn, C., Vilkomir, S.A., Parnas, D.L., Kostic, S.: Specification of software component requirements using the trace function method. In: Proc. ICSEA. p. 50. IEEE Computer Society (2006)
9. Rosen, R.: Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life. Columbia University Press, New York (1991)
10. Rutten, J.: Universal coalgebra: a theory of systems. Theoretical Computer Science 249(1), 3–80 (2000)
11. The on-line encyclopedia of integer sequences: A000110. http://oeis.org/A000110 (2013)
12. Smolin, L.: Time Reborn: From the Crisis in Physics to the Future of the Universe. Houghton Mifflin Harcourt, Boston (2013)
13. Trancón y Widemann, B.: The recursion scheme of the trace function method. In: Filipe, J., Maciaszek, L.A. (eds.) Proc. ENASE. pp. 146–155 (2012)
14. Trnková, V.: Some properties of Set functors. Comment. Math. Univ. Carol. 10(2), 323–352 (1969)
15. Uustalu, T., Vene, V.: Primitive (co)recursion and course-of-value (co)iteration, categorically. Informatica 10(1), 5–26 (1999)