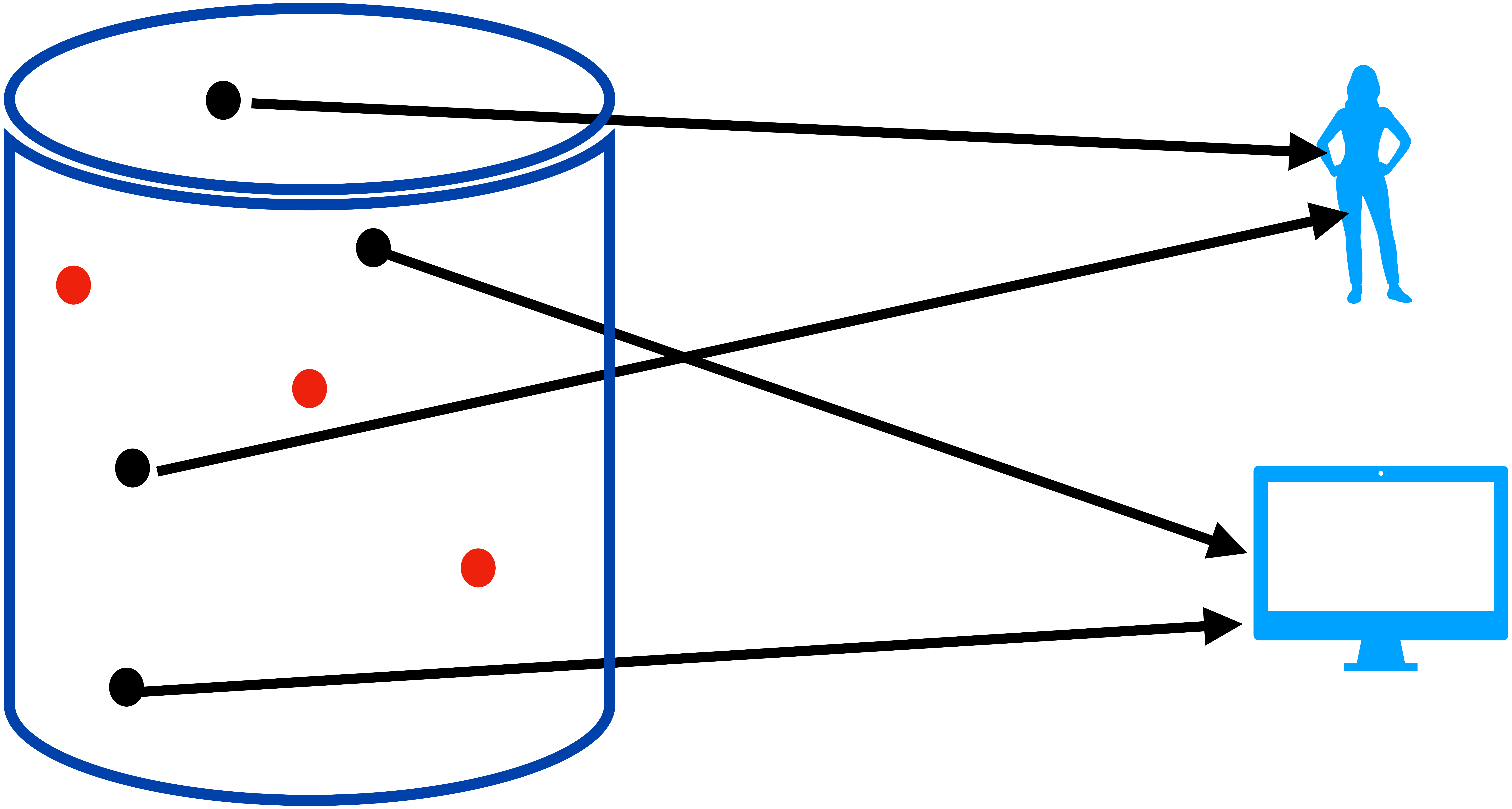
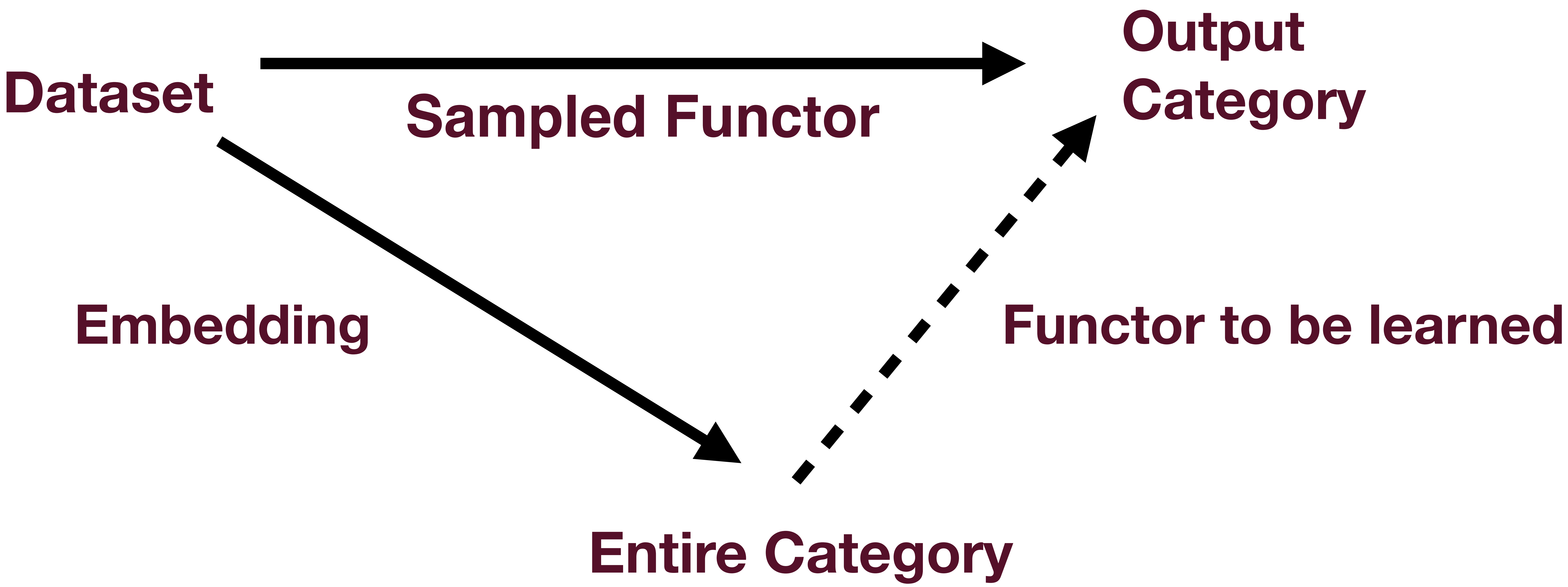
Universal Imitation Games: Generative AI as **Coalgebras**

Sridhar Mahadevan
Adobe Research and University of Massachusetts

Dynamic Imitation Games



Learning Functors



Dataset

Sampled Functor

Output Category

Embedding

Functor to be learned

Entire Category

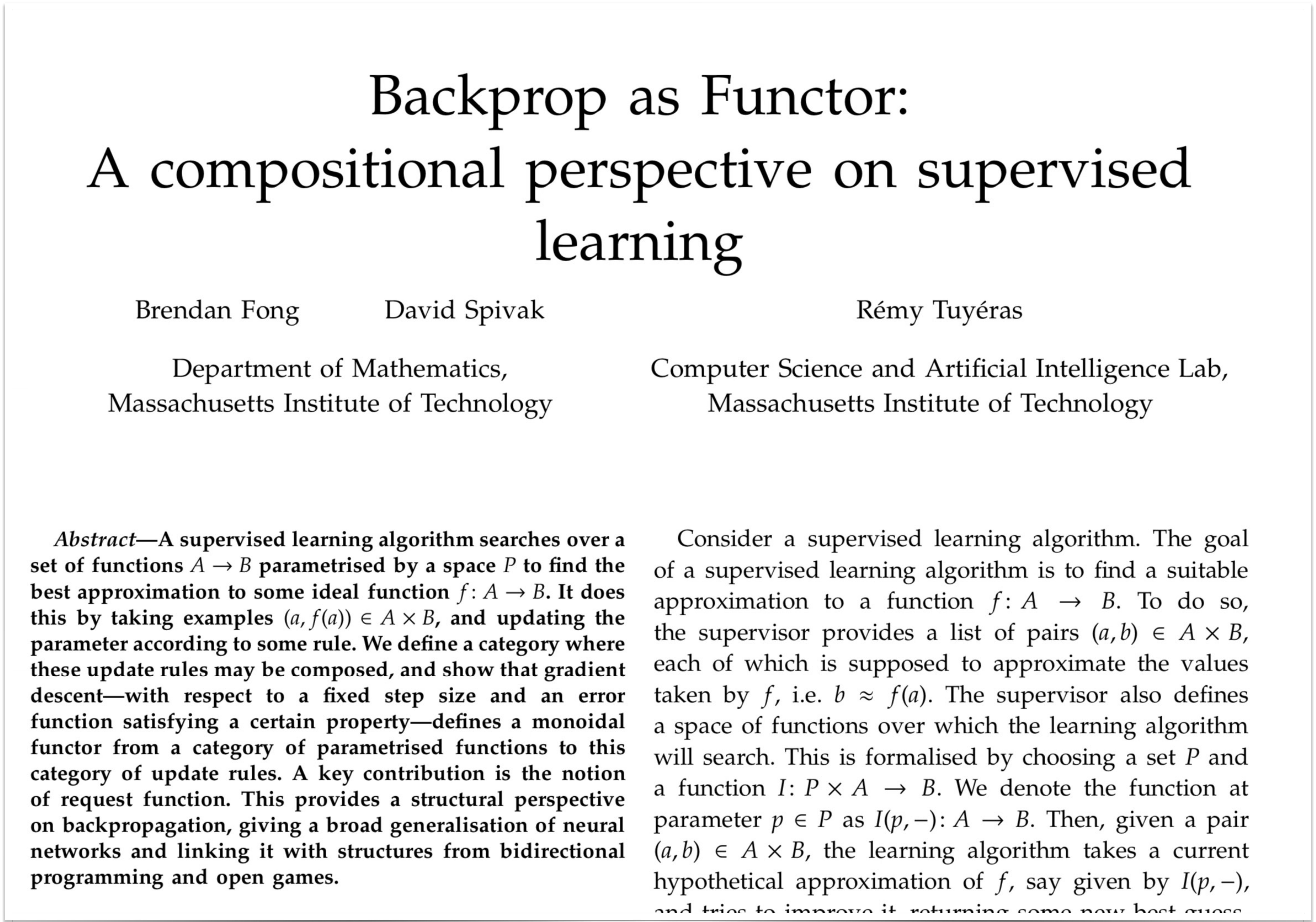
Find the universal property!

Kan Extensions of Functors

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "\mathcal{F}"] \arrow[dr, "\mathcal{K}"'] & \Downarrow \eta & \mathcal{E} \\
& \mathcal{D} \arrow[ur, dashed, "\operatorname{Lan}_{\mathcal{K}} \mathcal{F}"'] &
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{C} \arrow[rr, "\mathcal{F}"] \arrow[dr, "\mathcal{K}"'] & \eta \Uparrow & \mathcal{E} \\
& \mathcal{D} \arrow[ur, dashed, "\operatorname{Ran}_{\mathcal{K}} \mathcal{F}"'] &
\end{tikzcd}$$

Every concept in category theory can be expressed as a Kan extension!



Backprop as Functor: A compositional perspective on supervised learning

Brendan Fong David Spivak

Department of Mathematics,
Massachusetts Institute of Technology

Rémy Tuyéras

Computer Science and Artificial Intelligence Lab,
Massachusetts Institute of Technology

Abstract—**A supervised learning algorithm searches over a set of functions $A \to B$ parametrised by a space P to find the best approximation to some ideal function $f\colon A \to B$. It does this by taking examples $(a, f(a)) \in A \times B$, and updating the parameter according to some rule. We define a category where these update rules may be composed, and show that gradient descent—with respect to a fixed step size and an error function satisfying a certain property—defines a monoidal functor from a category of parametrised functions to this category of update rules. A key contribution is the notion of request function. This provides a structural perspective on backpropagation, giving a broad generalisation of neural networks and linking it with structures from bidirectional programming and open games.**

Consider a supervised learning algorithm. The goal of a supervised learning algorithm is to find a suitable approximation to a function $f\colon A \to B$. To do so, the supervisor provides a list of pairs $(a, b) \in A \times B$, each of which is supposed to approximate the values taken by f, i.e. $b \approx f(a)$. The supervisor also defines a space of functions over which the learning algorithm will search. This is formalised by choosing a set P and a function $I\colon P \times A \to B$. We denote the function at parameter $p \in P$ as $I(p, -)\colon A \to B$. Then, given a pair $(a, b) \in A \times B$, the learning algorithm takes a current hypothetical approximation of f, say given by $I(p, -)$, and tries to improve it, returning some new best guess

Monoidal Categories

- Equipped with a product internal bifunctor:
  - $\otimes : C \times C \to C$
  - Identity element 1: $1 \otimes c \simeq c \simeq c \otimes 1$
- Unit interval [0,1]: closed symmetric monoidal preorder
- $\mathscr{V}$ $-$enriched monoidal category: $a, b \in C \Rightarrow C(a, b) \in \mathscr{V}$

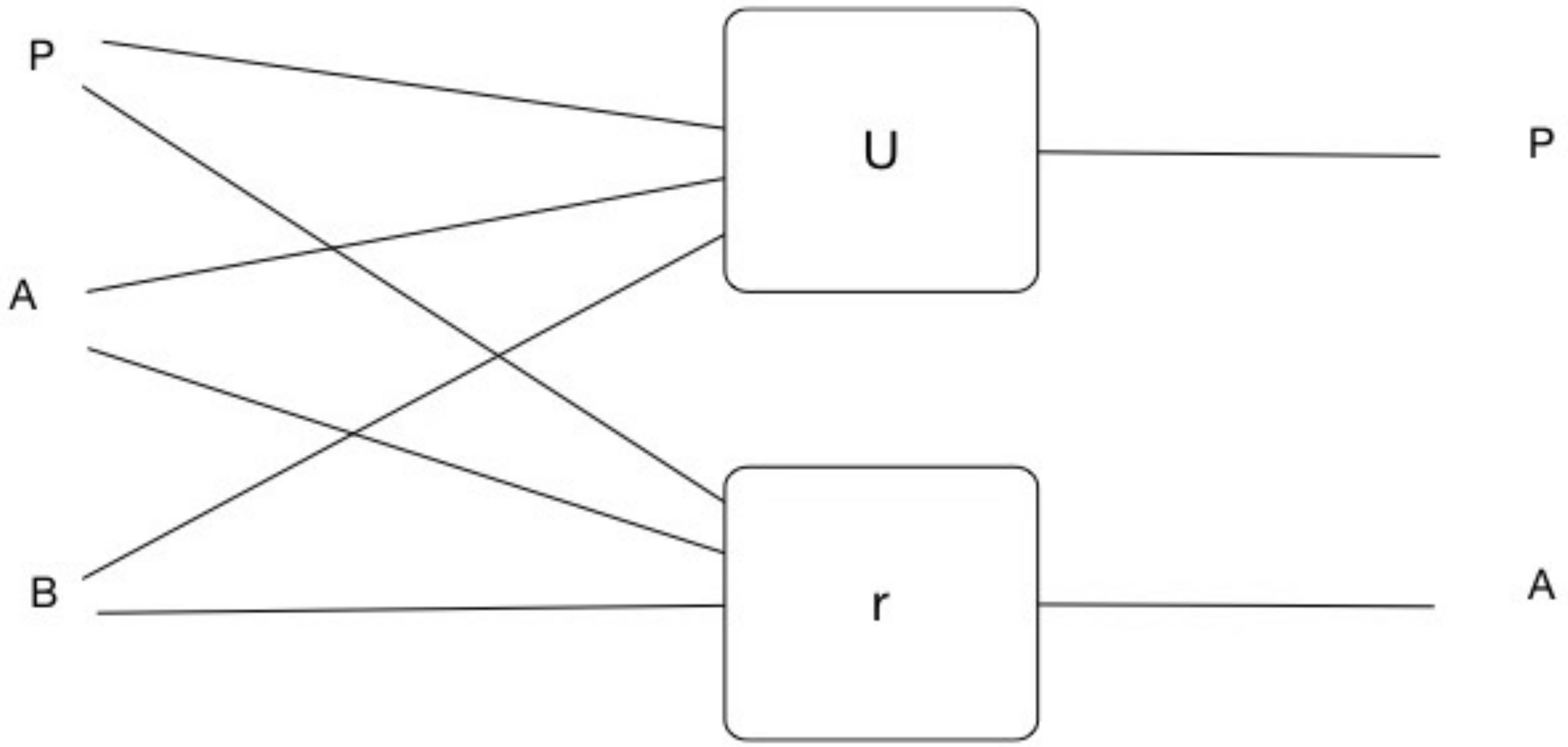
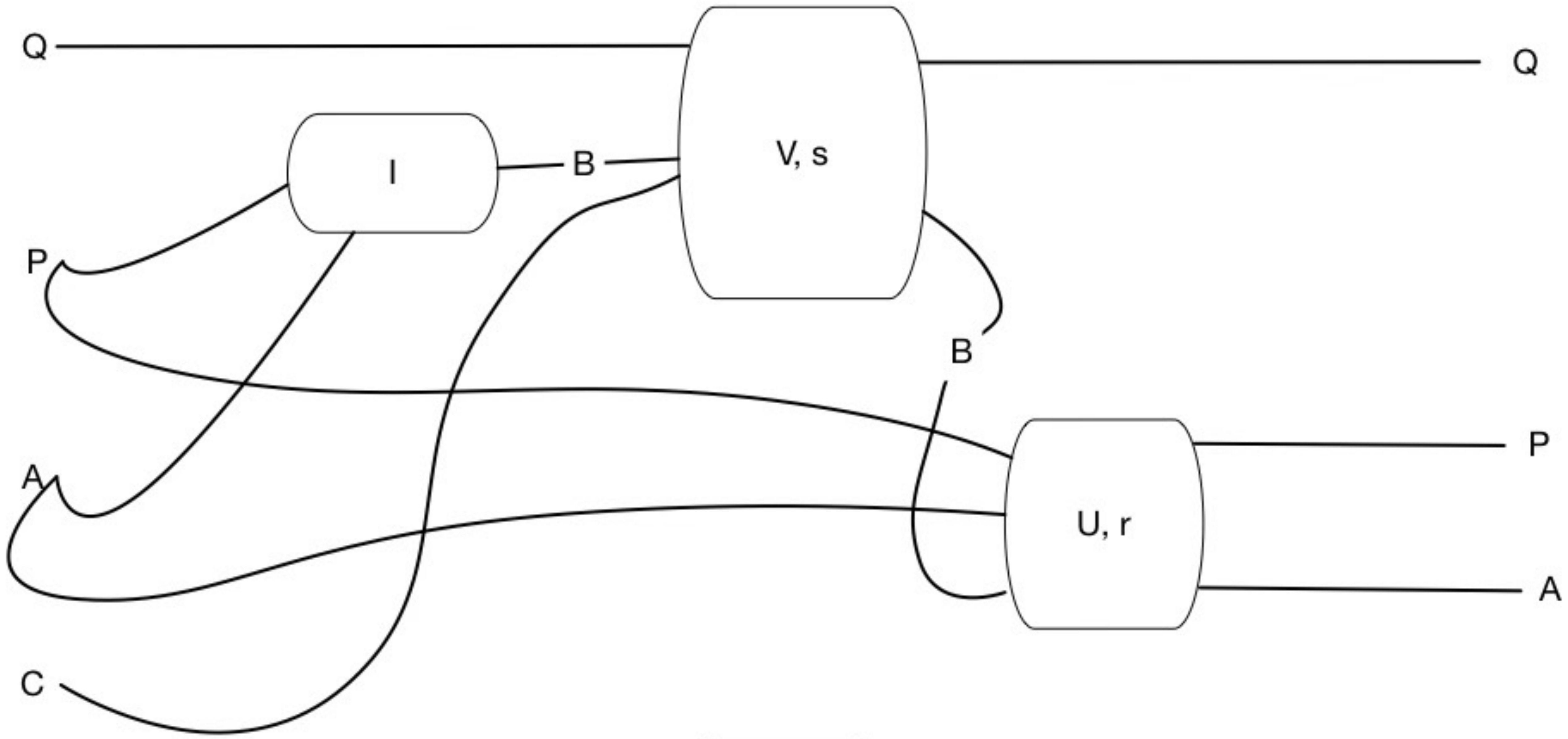


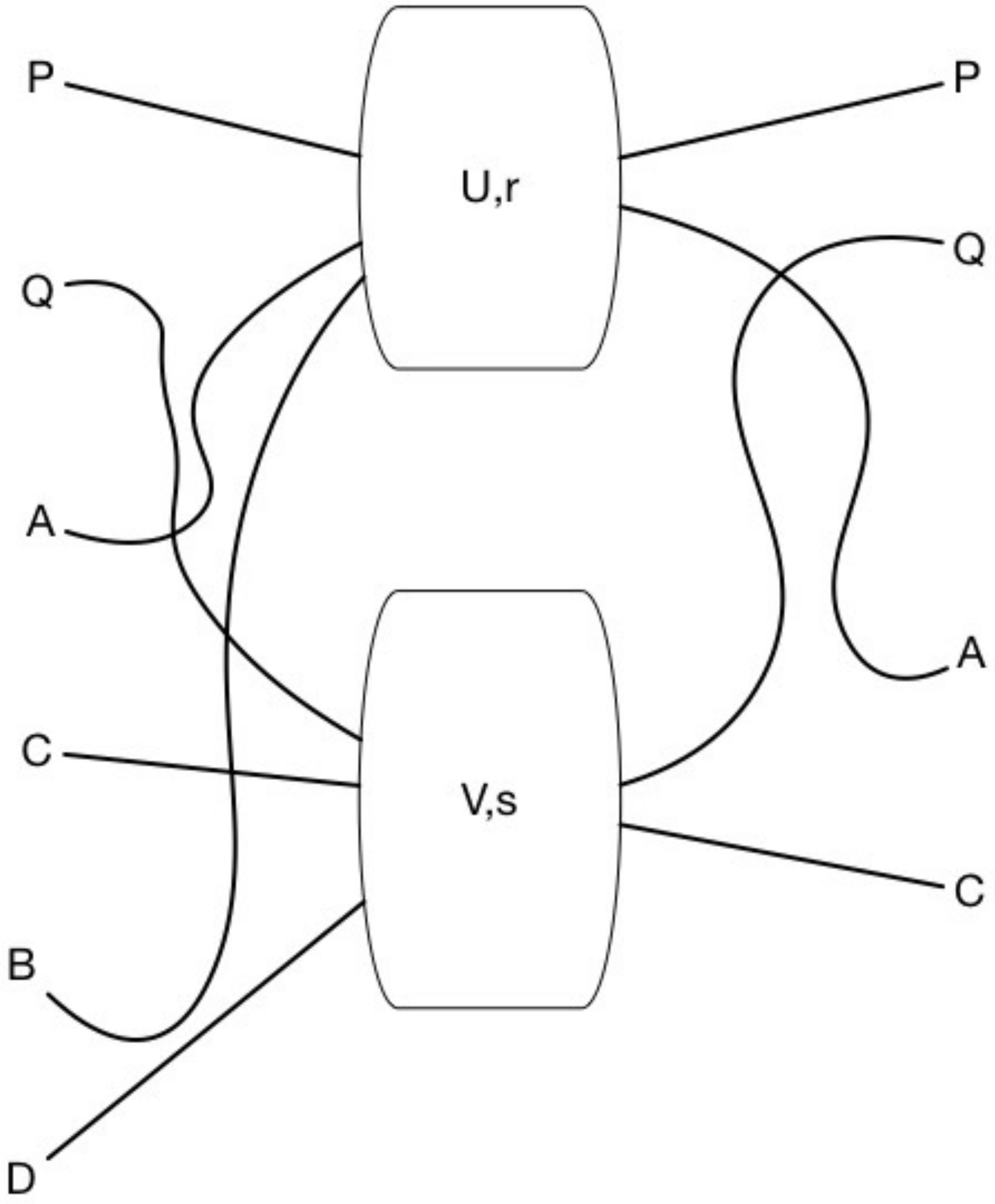
Figure 10: A learner in the symmetric monoidal category `Learn` is defined as a morphism. Later in Section 3, we will see how to define learners as coalgebras instead.

Definition 3. Fong et al. [2019] The symmetric monoidal category **Learn** is defined as a collection of objects that define sets, and a collection of an equivalence class of learners. Each learner is defined by the following 4-tuple (see Figure 10).

- A parameter space P
- An implementation function $I : P \times A \to B$
- An update function $U : P \times A \times B \to P$
- A request function $r : P \times A \times B \to A$



Sequential Composition



Parallel Composition

$$A \xrightarrow{(P,I,U,r)} B \xrightarrow{(Q,J,V,s)} C$$

The composite learner $A \to C$ is defined as $(P \times Q, I \cdot J, U \cdot V, r \cdot s)$, where the composite implementation function is

$$(I \cdot J)(p, q, a) \coloneqq J(q, I(p, a))$$

and the composite update function is

$$U \cdot V(p, q, a, c) \coloneqq (U(p, a, s(q, I(p, a), c))\, , V(q, I(p, a), c)$$

and the composite request function is

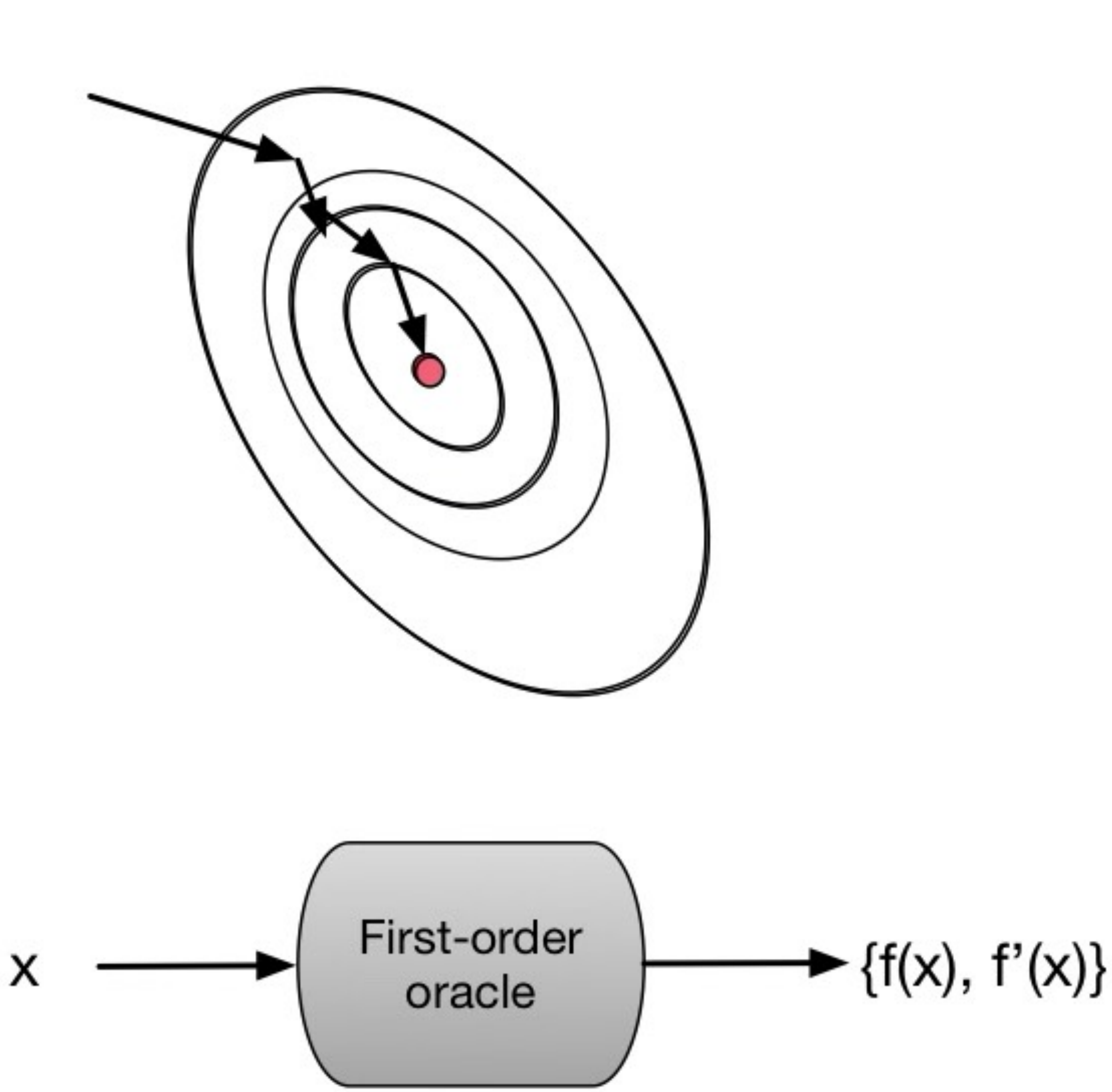
$$(r \cdot s)(p, q, a, c) \coloneqq r(p, a, s(q, I(p, a), c)).$$

$$\begin{tikzcd}
NNet \arrow[rr] \arrow[dr, "F"'] & & Learn \\
& Param \arrow[ur, "L_{\epsilon,e}"'] &
\end{tikzcd}$$

Backprop is a functor!

$$U_I(p,a,b) \coloneqq p - \epsilon \nabla_p E_I(p,a,b)$$

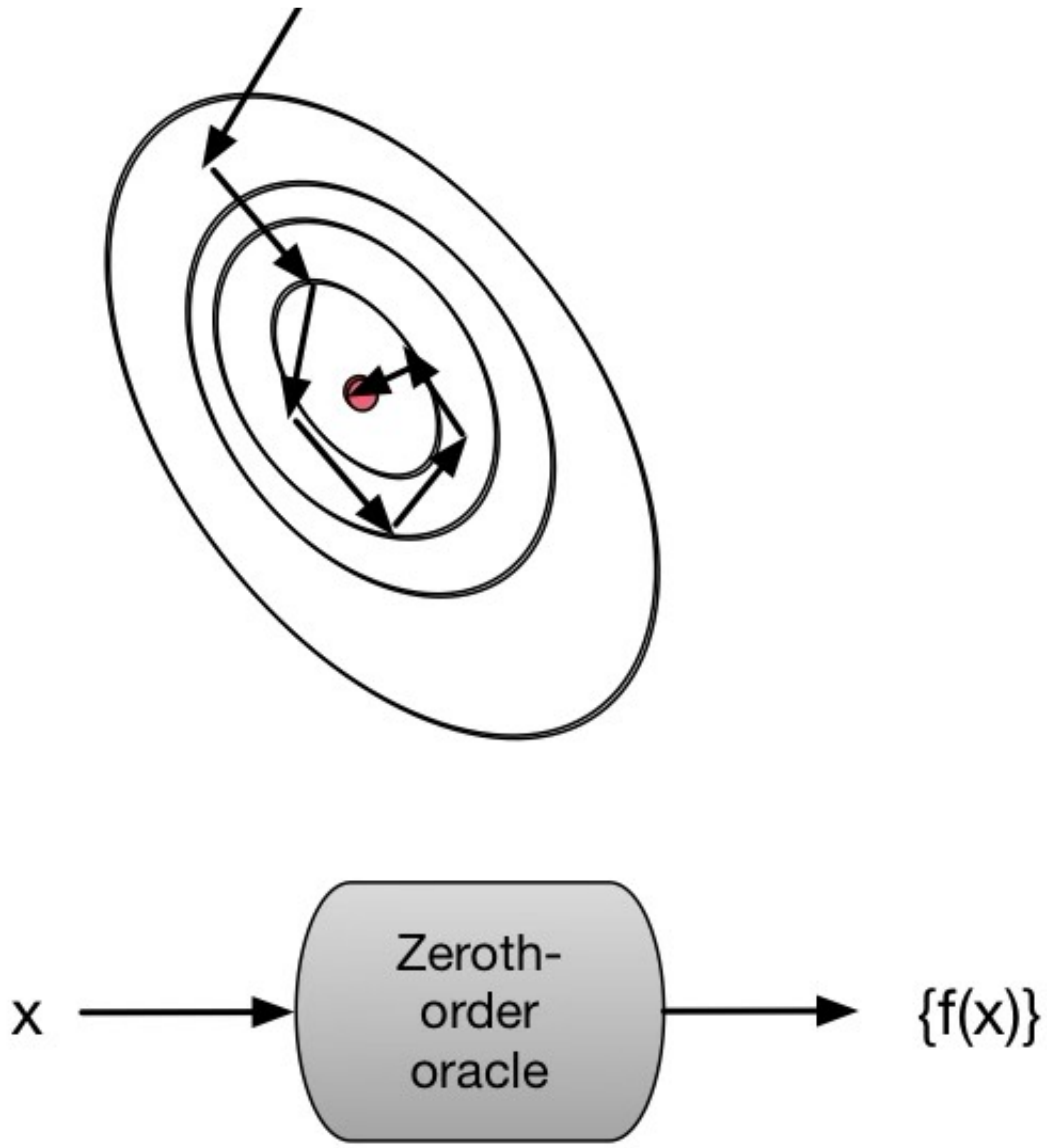
$$r_I(p,a,b) \coloneqq f_a(\nabla_a E_I(p,a,b))$$



x

First-order oracle

{f(x), f’(x)}

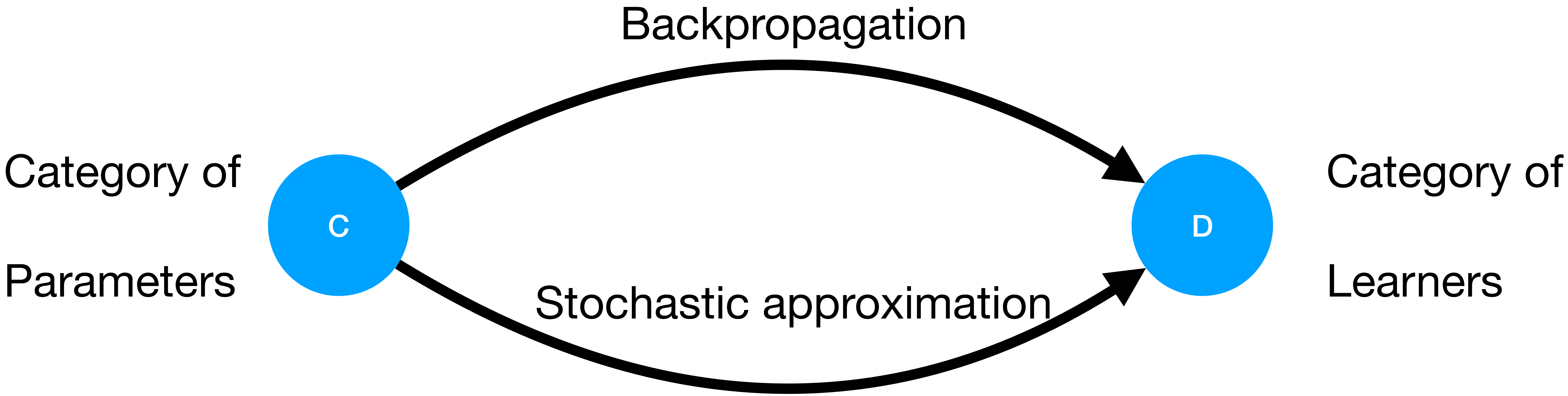


x

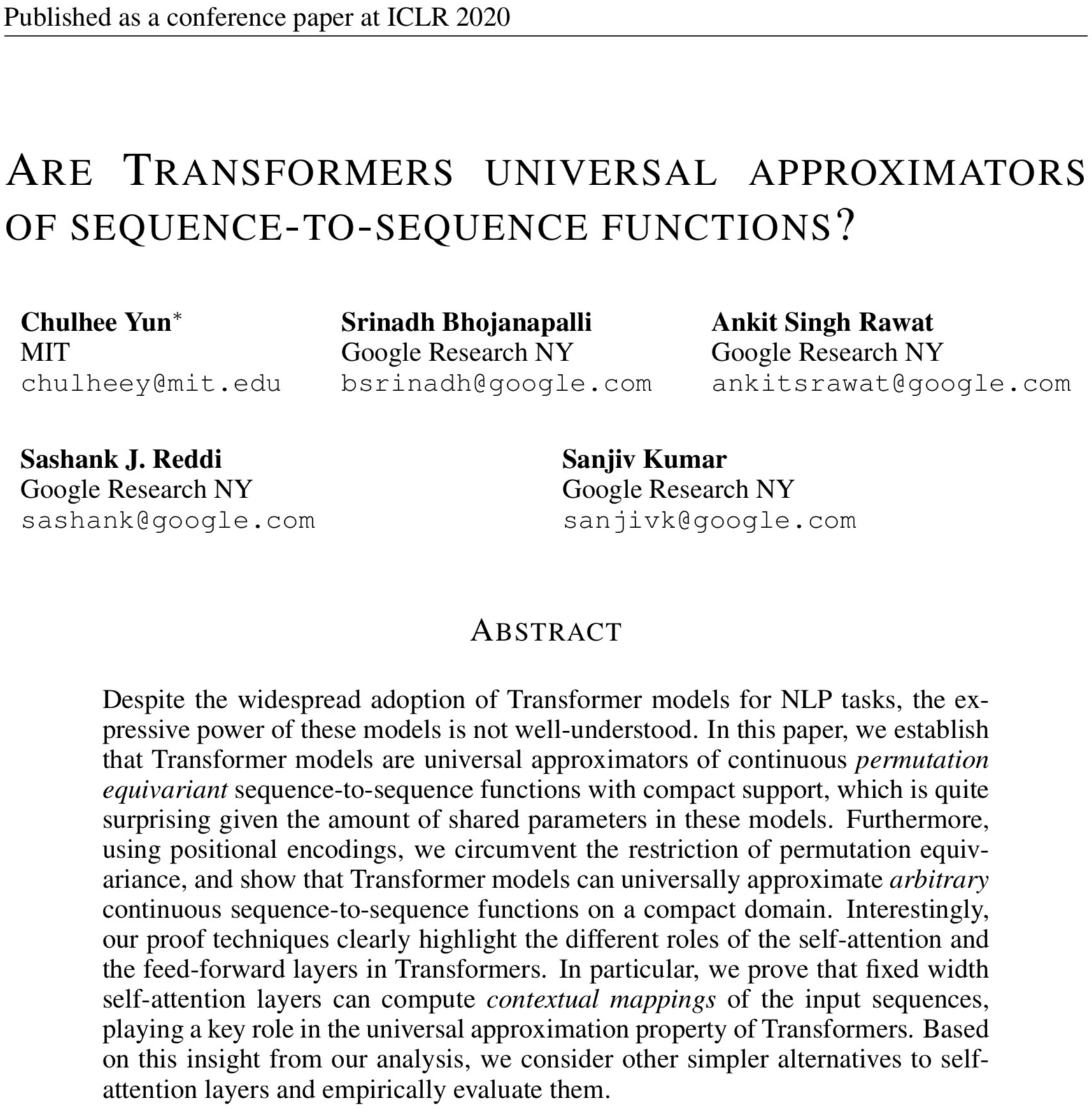
Zeroth-order oracle

{f(x)}

Natural Transformations for Deep Learning



$$\begin{tikzcd}
Fc \arrow[r, "Ff"] \arrow[d, "\mu_c"'] & Fc' \arrow[d, "\mu_{c'}"] \\
Gc \arrow[r, "Gf"] & Gc'
\end{tikzcd}$$



Published as a conference paper at ICLR 2020

ARE TRANSFORMERS UNIVERSAL APPROXIMATORS OF SEQUENCE-TO-SEQUENCE FUNCTIONS?

Chulhee Yun[*]
MIT
chulheey@mit.edu

Srinadh Bhojanapalli
Google Research NY
bsrinadh@google.com

Ankit Singh Rawat
Google Research NY
ankitsrawat@google.com

Sashank J. Reddi
Google Research NY
sashank@google.com

Sanjiv Kumar
Google Research NY
sanjivk@google.com

ABSTRACT

Despite the widespread adoption of Transformer models for NLP tasks, the expressive power of these models is not well-understood. In this paper, we establish that Transformer models are universal approximators of continuous *permutation equivariant* sequence-to-sequence functions with compact support, which is quite surprising given the amount of shared parameters in these models. Furthermore, using positional encodings, we circumvent the restriction of permutation equivariance, and show that Transformer models can universally approximate *arbitrary* continuous sequence-to-sequence functions on a compact domain. Interestingly, our proof techniques clearly highlight the different roles of the self-attention and the feed-forward layers in Transformers. In particular, we prove that fixed width self-attention layers can compute *contextual mappings* of the input sequences, playing a key role in the universal approximation property of Transformers. Based on this insight from our analysis, we consider other simpler alternatives to self-attention layers and empirically evaluate them.

Permutation-equivariant functions

f(XP) = f(X)P

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "P"'] & Y \arrow[r, "g"] \arrow[d, "P"] & Z \arrow[d, "P"] \\
XP \arrow[r, "f"'] & YP \arrow[r, "g"'] & ZP
\end{tikzcd}$$

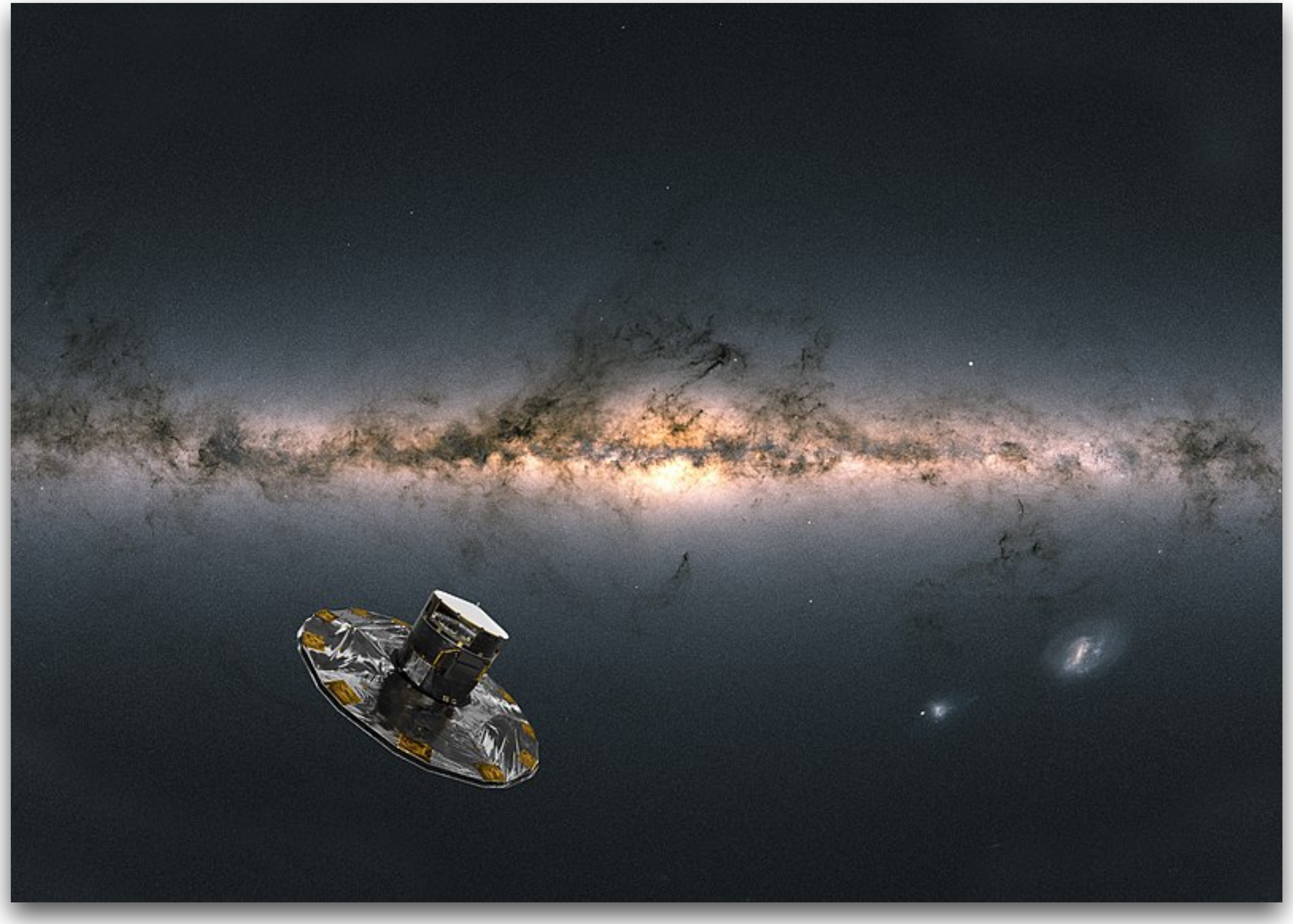
$$\begin{aligned}\mathrm{Attn}(X) &= X + \sum_{i=1}^{h} W_O^i W_V^i X \cdot \sigma[W_K^i X)^T W_Q^i X] \\ \mathrm{FF}(X) &= \mathrm{Attn}(X) + W_2 \cdot \mathrm{ReLU}(W_1 \cdot \mathrm{Attn}(X) + b_1 \mathbf{1}_n^T,\end{aligned}$$

Definition 32. The category $\mathcal{C}_T$ of Transformer models is defined as follows:

- The objects Obj(C) are defined as vectors $X \in \mathbb{R}^{d \times n}$ denoting n-length sequences of tokens of dimension d.
- The arrows or morphisms of the category $\mathcal{C}_T$ are defined as a family of sequence-to-sequence functions and defined as:

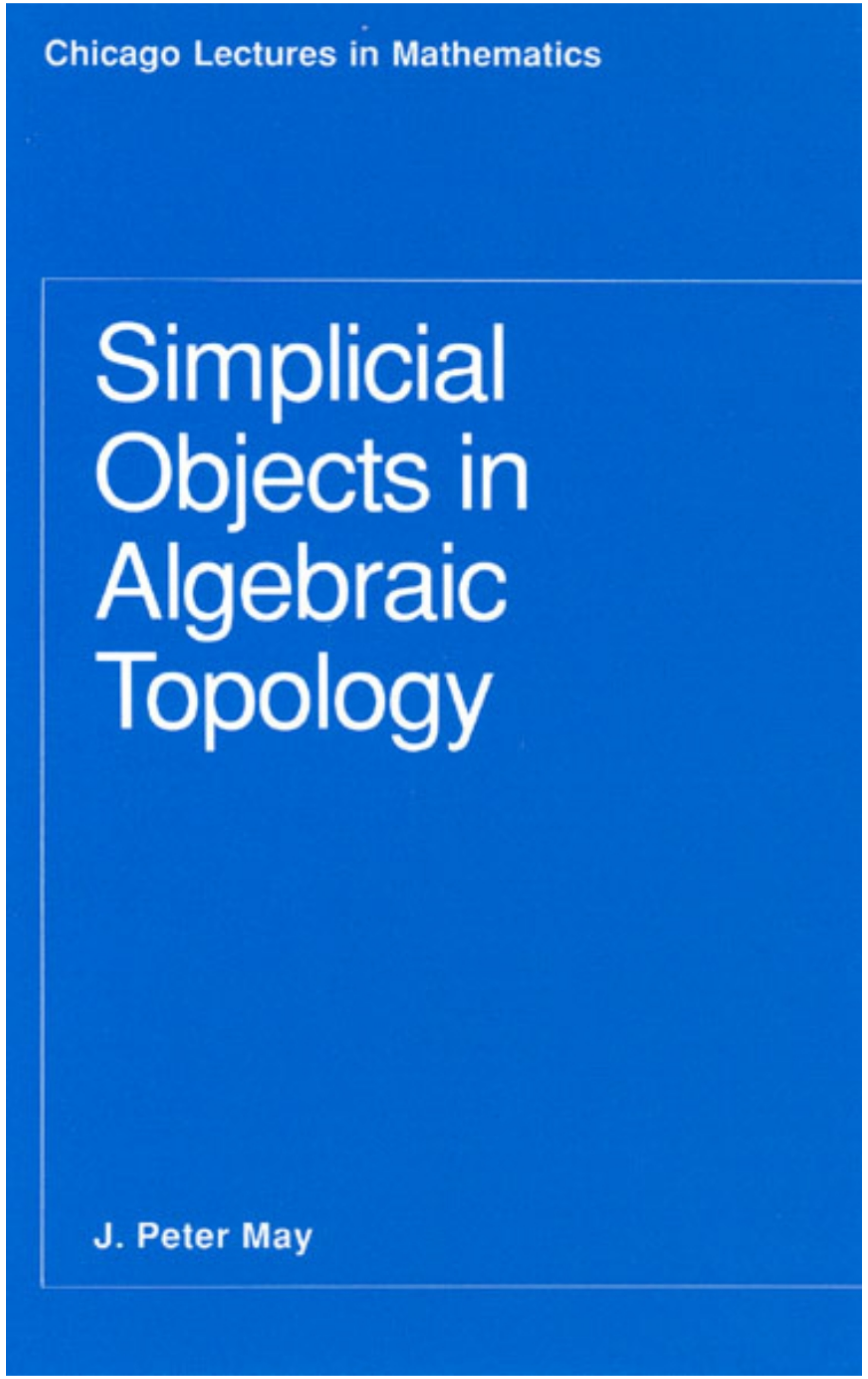
$$T^{h,m,r} := \{f : \mathbb{R}^{d \times n} \to \mathbb{R}^{d \times n} \mid \text{where } f(XP) = XP, \text{ for some permutation matrix } P\}$$

GAIA: Generative AI Architecture



Higher-order category theory for Deep Learning!

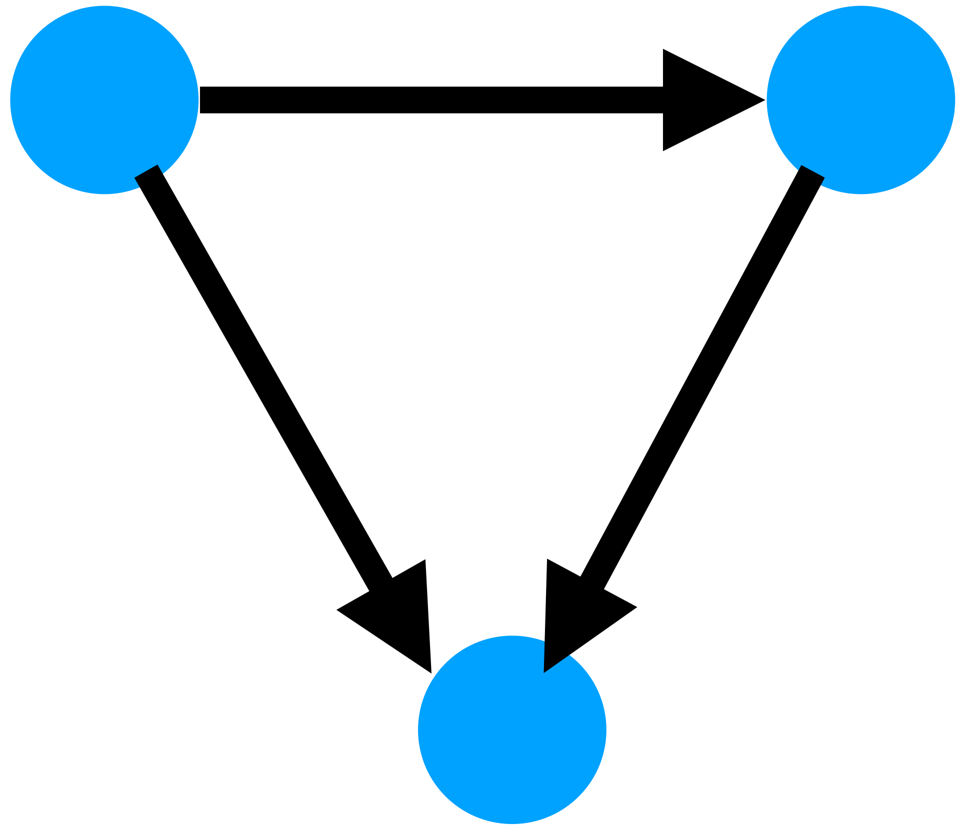
Simplicial Objects: One stop ML shopping center

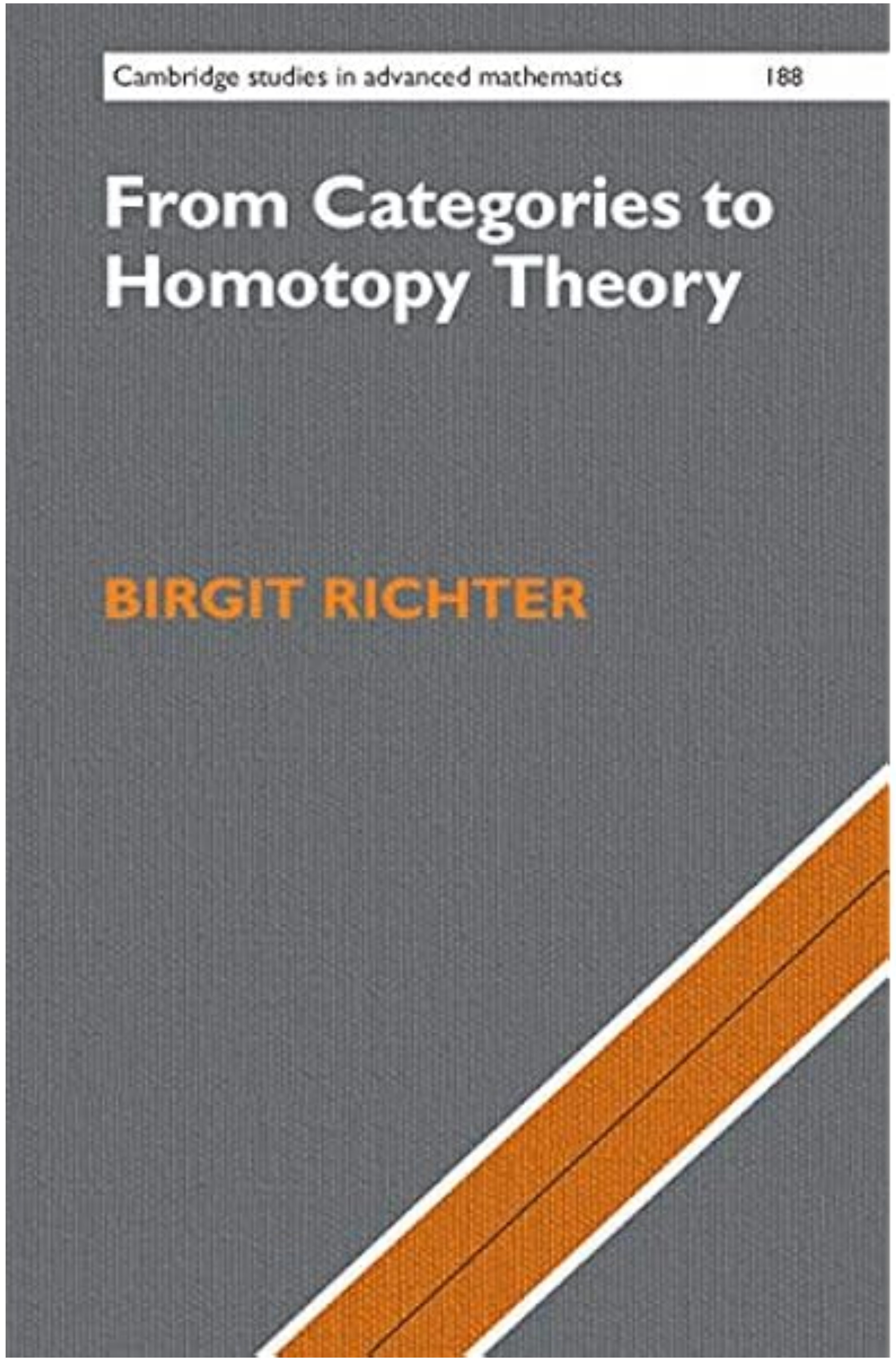


Chicago Lectures in Mathematics

Simplicial Objects in Algebraic Topology

J. Peter May





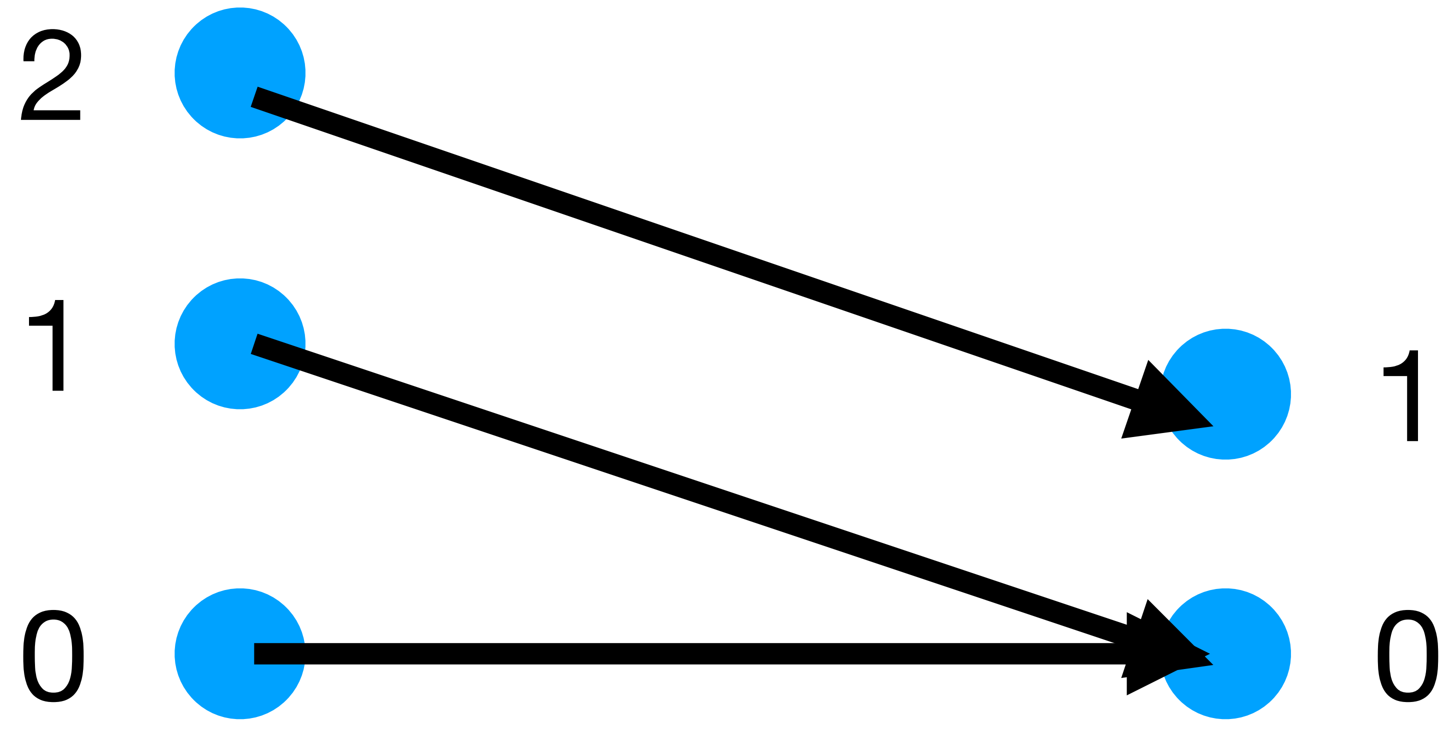
Cambridge studies in advanced mathematics 188

From Categories to Homotopy Theory

BIRGIT RICHTER

Simplicial Category Δ

- **Objects**: ordinal numbers
  - $[n] = \{0,1,\ldots,n-1\}$
- **Arrows**:
  - $f : [m] \to [n]$
  - If $i \leq j$, then $f(i) \leq f(j)$
  - All morphisms can be built out of primitive injections/surjections
    - $\delta_i : [n] \to [n+1]$: injection skipping i
    - $\sigma_i : [n] \to [n-1]$, surjection repeating i



Simplicial Sets: Contravariant Functors

$$\begin{tikzcd}
0 \arrow[r] & 1 \arrow[r, shift left] \arrow[r, shift right] & 2 \arrow[r, shift left=2] \arrow[r] \arrow[r, shift right=2] & 3
\end{tikzcd}$$

$\delta_i^n : [n] \to [n+1]$

$$\begin{tikzcd}
0 & 1 \arrow[l] & 2 \arrow[l] & 3 \arrow[l, shift left] \arrow[l, shift right]
\end{tikzcd}$$

$\sigma_i^n : [n+1] \to [n]$

$$X_n : [n] \to X : \Delta^{op} \to X$$

Nerve of a Category

- Recall a category is defined as a collection of objects, and a collection of arrows between any pair of objects

- A simplicial set is a contravariant functor mapping the simplicial category to the category of sets

- Any category can be mapped onto a simplicial set by constructing its nerve

- Intuitively, consider all sequences of composable morphisms of length n!

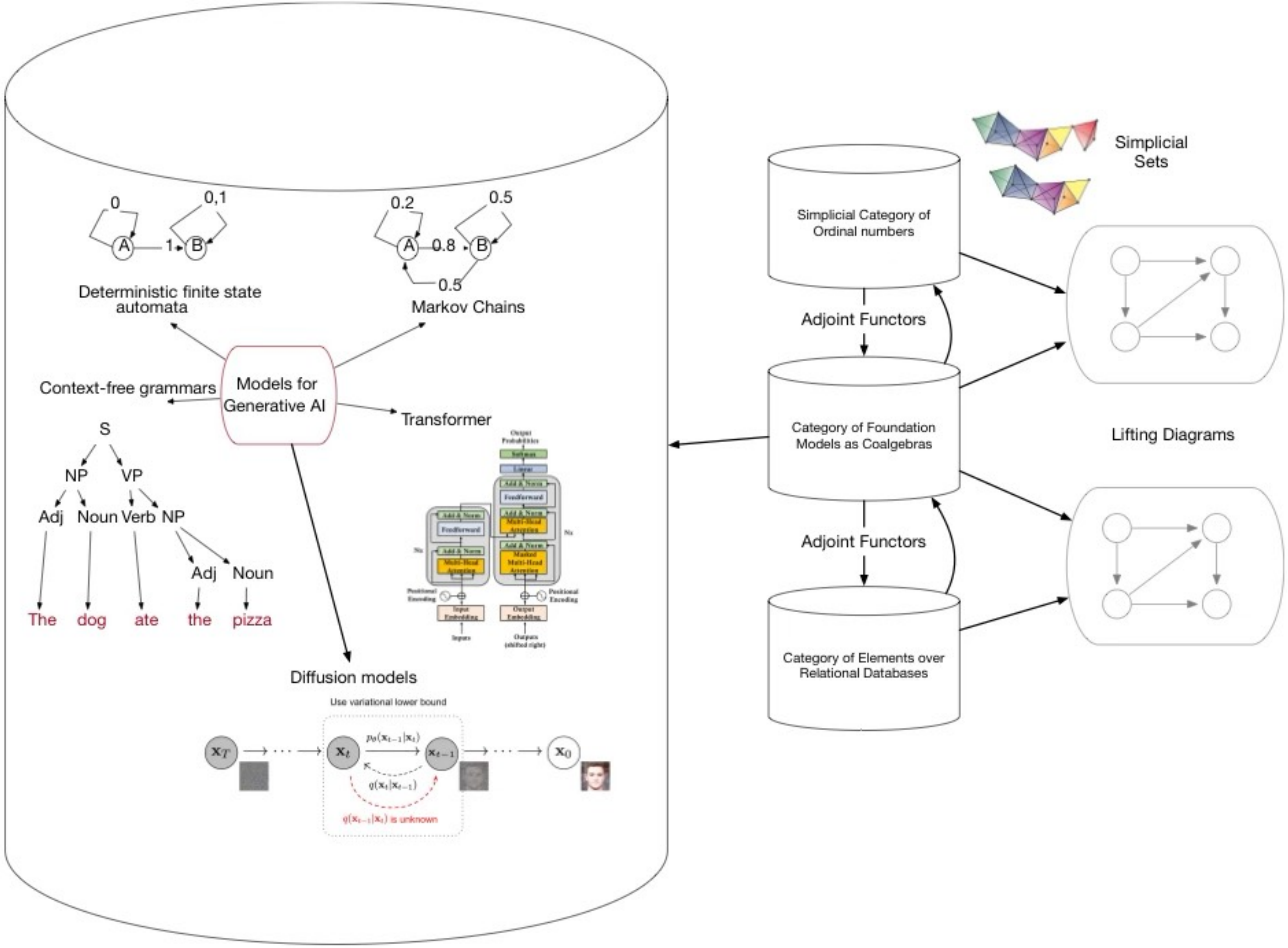
Nerve of the Category of Transformers

- Since Transformers define a category over Euclidean spaces of permutation-equivariant functions, we can construct its nerve
- Consider all compositions of Transformers building blocks of length n
- This construction maps the category of Transformers into a simplicial set
- It is a full and faithful embedding of Transformers as simplicial sets
- However, simplicial sets cannot be faithfully mapped back to ordinary categories

Simplicial Sets vs. Categories

- Any category can be embedded faithfully into a simplicial set using its nerve
- The embedding is full and faithful (perfect reconstruction)
- Unfortunately, the converse is not possible
- Given a simplicial set, the left adjoint functor that maps it into a category is lossy!
- GAIA (in theory!) is more powerful than existing generative AI formalisms

GAIA: Categorical Foundations of Generative AI



Models for Generative AI

Deterministic finite state automata

Markov Chains

Context-free grammars

Transformer

Diffusion models

Simplicial Category of Ordinal numbers

Category of Foundation Models as Coalgebras

Category of Elements over Relational Databases

Adjoint Functors

Adjoint Functors

Simplicial Sets

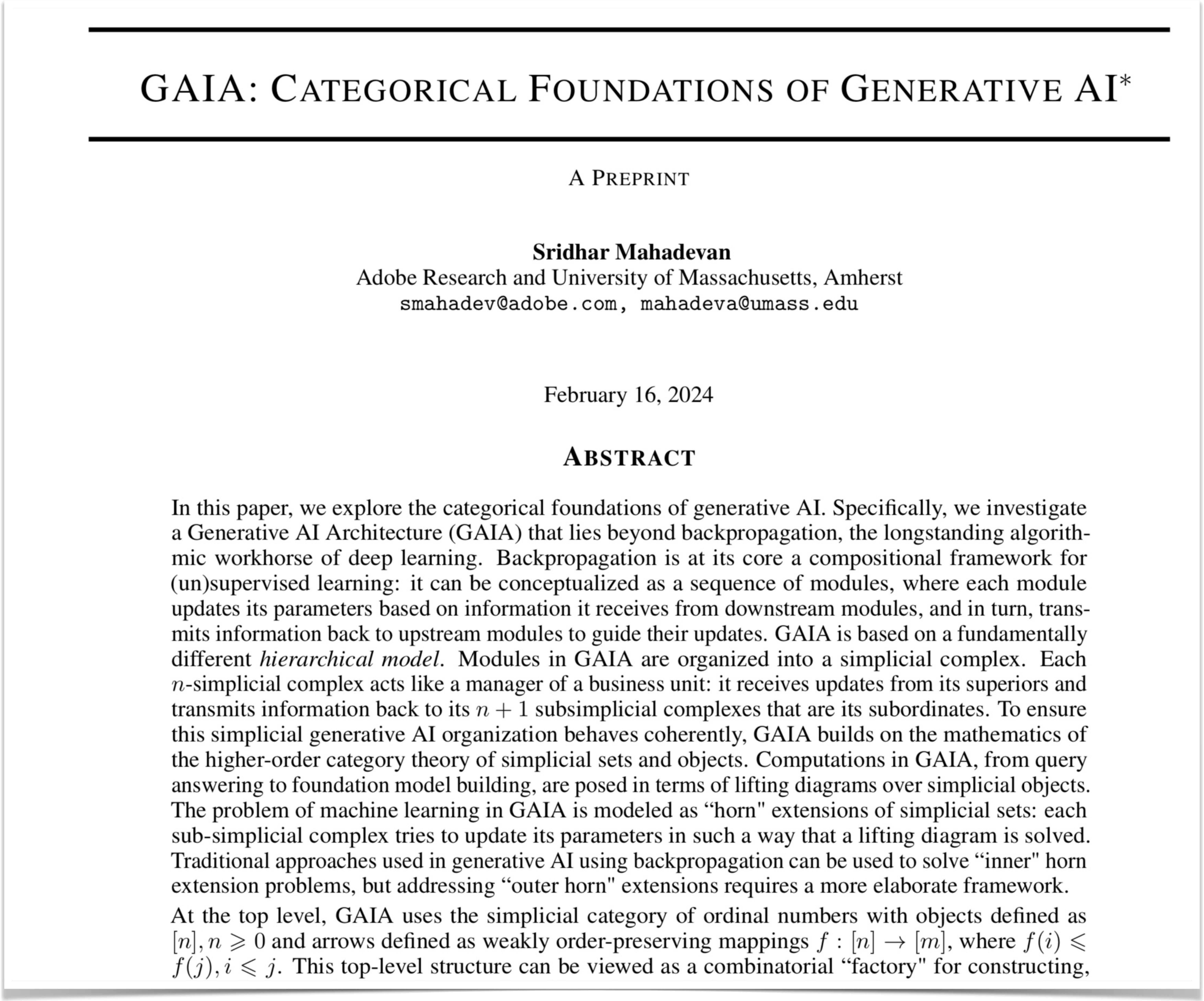
Lifting Diagrams

0
0,1
A
1
B
0.2
0.5
A
0.8
B
0.5
S
NP
VP
Adj
Noun
Verb
NP
Adj
Noun
The
dog
ate
the
pizza
Use variational lower bound
$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_{t-1}$
$\mathbf{x}_0$
$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown

Paper online at my

UMass home page

Forthcoming book!



GAIA: Categorical Foundations of Generative AI*

A Preprint

Sridhar Mahadevan
Adobe Research and University of Massachusetts, Amherst
smahadev@adobe.com, mahadeva@umass.edu

February 16, 2024

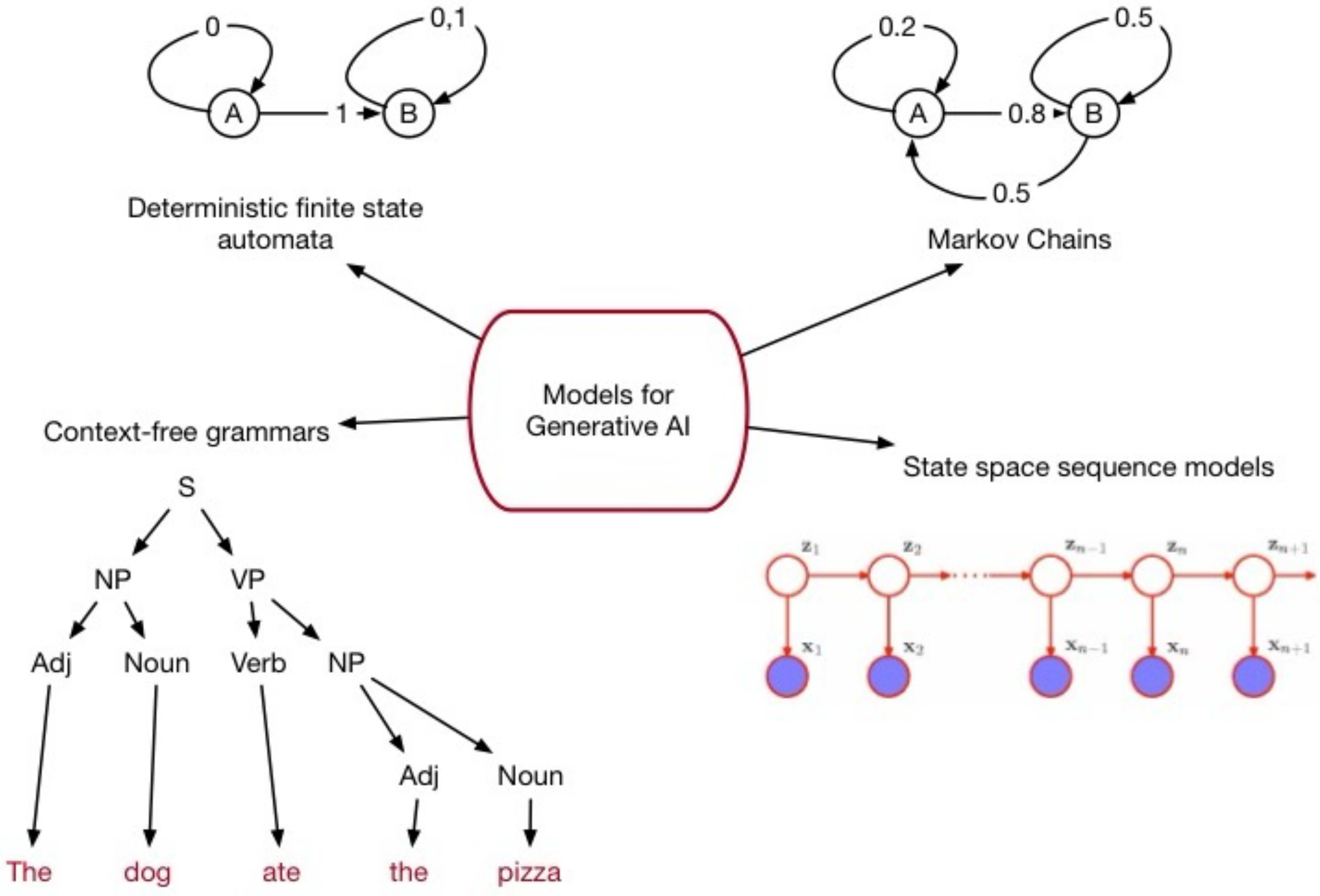
Abstract

In this paper, we explore the categorical foundations of generative AI. Specifically, we investigate a Generative AI Architecture (GAIA) that lies beyond backpropagation, the longstanding algorithmic workhorse of deep learning. Backpropagation is at its core a compositional framework for (un)supervised learning: it can be conceptualized as a sequence of modules, where each module updates its parameters based on information it receives from downstream modules, and in turn, transmits information back to upstream modules to guide their updates. GAIA is based on a fundamentally different *hierarchical model*. Modules in GAIA are organized into a simplicial complex. Each n-simplicial complex acts like a manager of a business unit: it receives updates from its superiors and transmits information back to its $n + 1$ subsimplicial complexes that are its subordinates. To ensure this simplicial generative AI organization behaves coherently, GAIA builds on the mathematics of the higher-order category theory of simplicial sets and objects. Computations in GAIA, from query answering to foundation model building, are posed in terms of lifting diagrams over simplicial objects. The problem of machine learning in GAIA is modeled as "horn" extensions of simplicial sets: each sub-simplicial complex tries to update its parameters in such a way that a lifting diagram is solved. Traditional approaches used in generative AI using backpropagation can be used to solve "inner" horn extension problems, but addressing "outer horn" extensions requires a more elaborate framework.

At the top level, GAIA uses the simplicial category of ordinal numbers with objects defined as $[n], n \geqslant 0$ and arrows defined as weakly order-preserving mappings $f : [n] \rightarrow [m]$, where $f(i) \leqslant f(j), i \leqslant j$. This top-level structure can be viewed as a combinatorial "factory" for constructing,

Examples of

Universal coalgebras



Models for Generative AI

Deterministic finite state automata

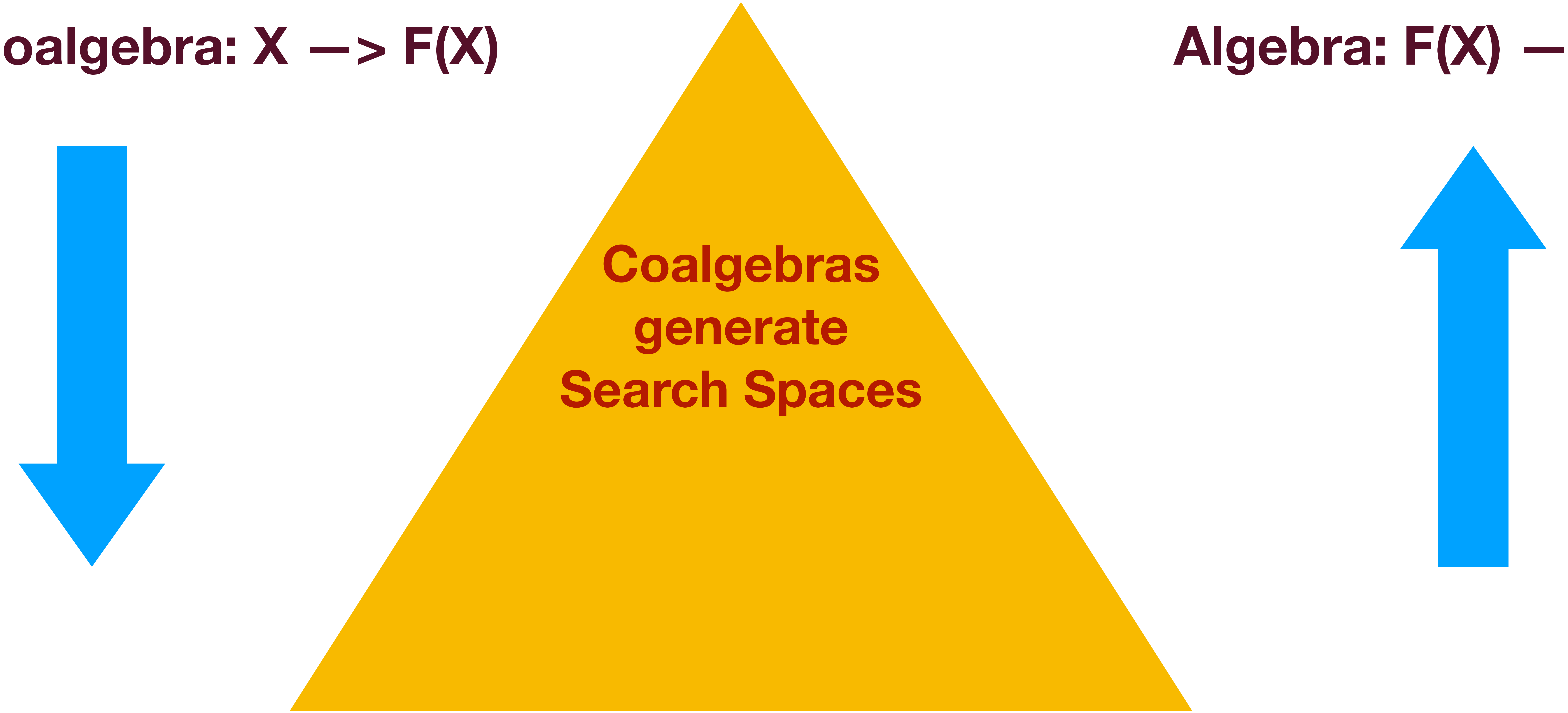
Markov Chains

Context-free grammars

State space sequence models

Coalgebra: X —> F(X)

Algebra: F(X) —> X



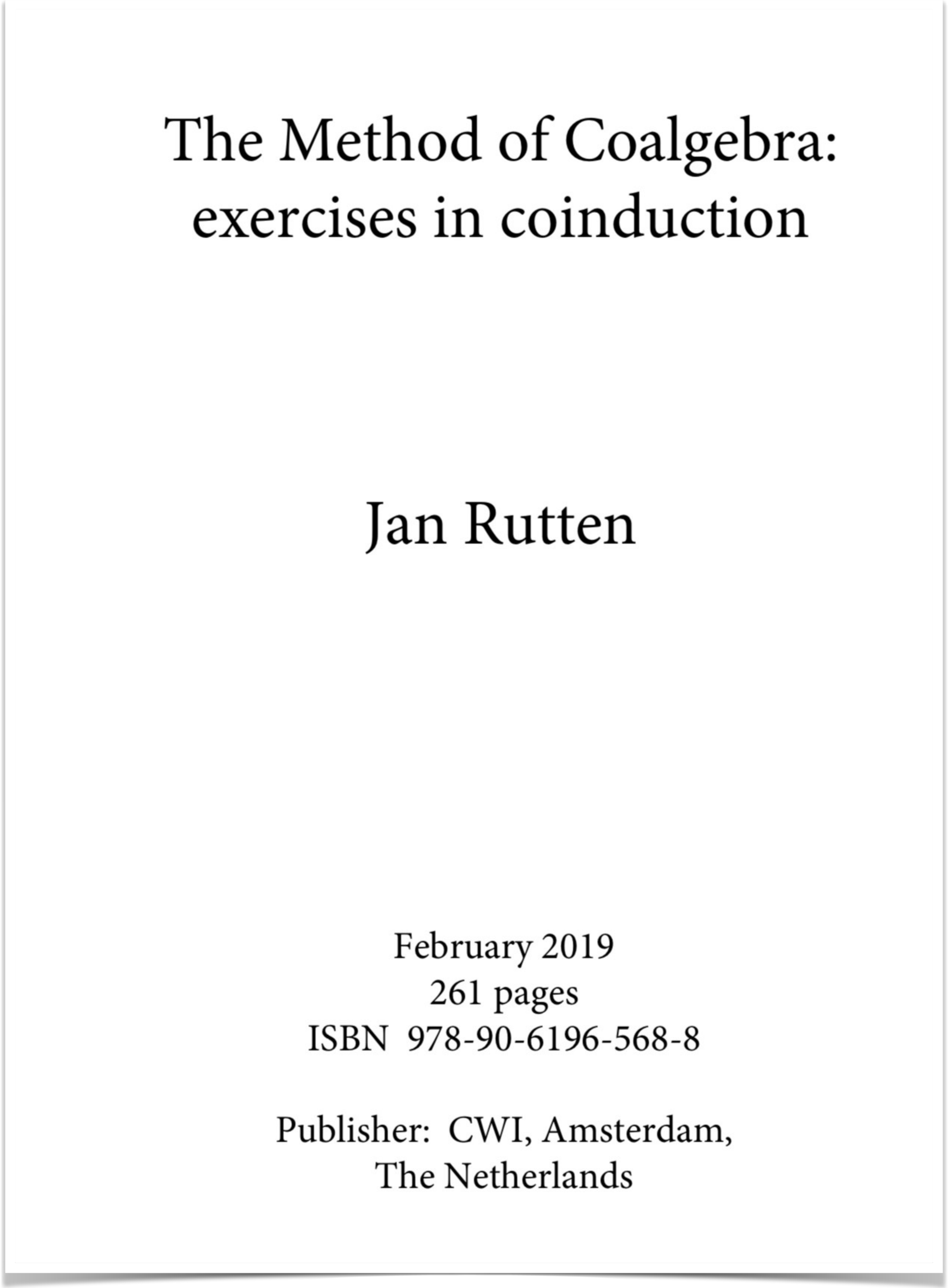
Coalgebras generate Search Spaces

From Induction to Coinduction

- **Machine learning has traditionally been modeled as induction**
- **Identification in the limit: Gold, Solomonoff**
- **PAC Learning: Valiant, Vapnik**
- **Algorithmic Information Theory: Chaitin, Kolmogorov**
- **Occam's Razor, Minimum Description Length**

Coinduction: A New Paradigm for ML

- **Generative AI is all about modeling infinite data streams**
  - **Automata, Grammars, Markov processes, LLMs, diffusion models**
- **Infinite data streams define non-well-founded sets**
- **Final coalgebras generalize (greatest) fixed points**
- **Reinforcement learning is an example of coinduction in a coalgebra**
- **Causal inference is also usefully modeled in coalgebras**

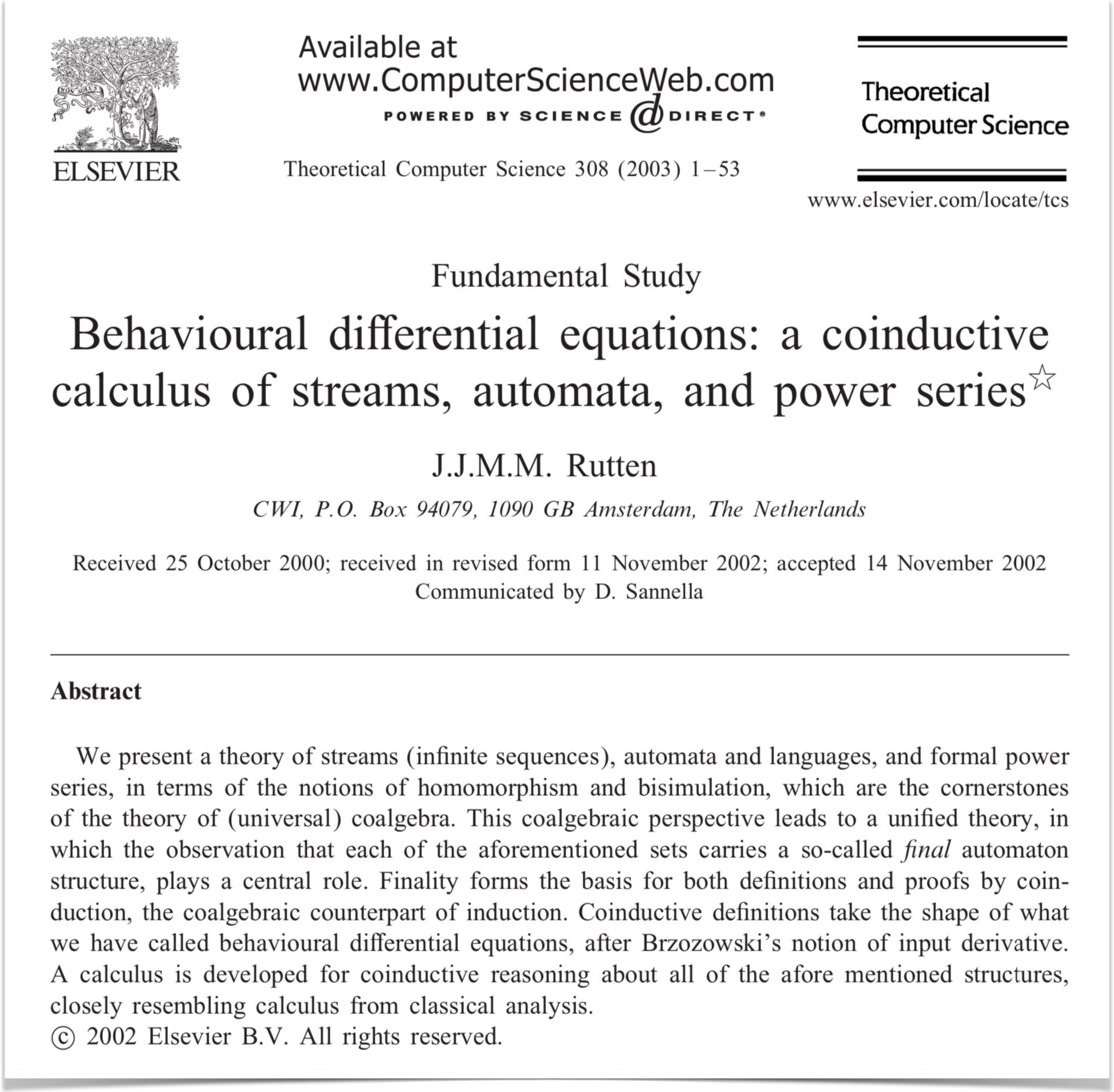


The Method of Coalgebra: exercises in coinduction

Jan Rutten

February 2019
261 pages
ISBN 978-90-6196-568-8

Publisher: CWI, Amsterdam,
The Netherlands



ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 308 (2003) 1–53

Theoretical Computer Science

www.elsevier.com/locate/tcs

Fundamental Study

Behavioural differential equations: a coinductive calculus of streams, automata, and power series☆

J.J.M.M. Rutten

CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands

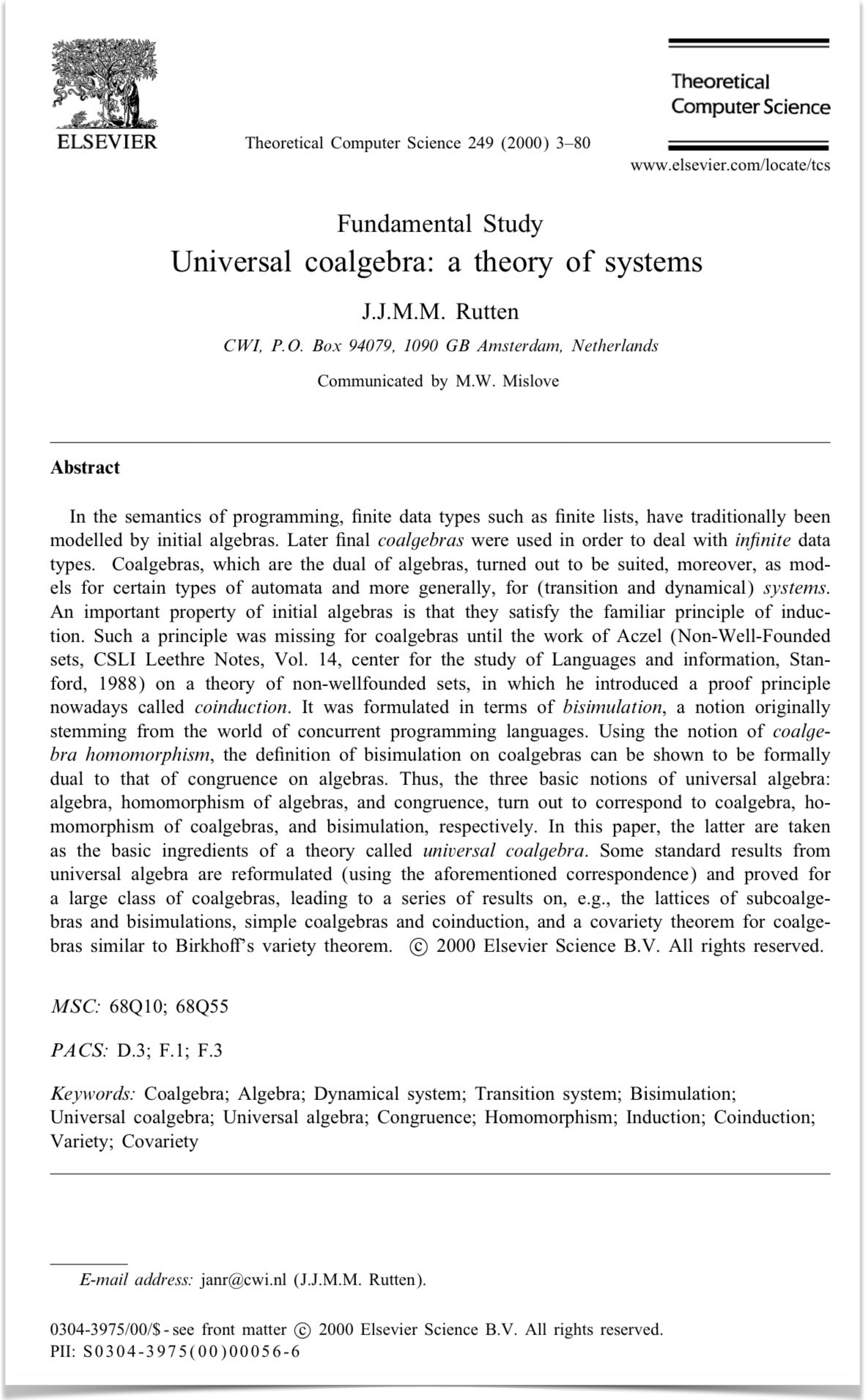
Received 25 October 2000; received in revised form 11 November 2002; accepted 14 November 2002
Communicated by D. Sannella

Abstract

We present a theory of streams (infinite sequences), automata and languages, and formal power series, in terms of the notions of homomorphism and bisimulation, which are the cornerstones of the theory of (universal) coalgebra. This coalgebraic perspective leads to a unified theory, in which the observation that each of the aforementioned sets carries a so-called *final* automaton structure, plays a central role. Finality forms the basis for both definitions and proofs by coinduction, the coalgebraic counterpart of induction. Coinductive definitions take the shape of what we have called behavioural differential equations, after Brzozowski's notion of input derivative. A calculus is developed for coinductive reasoning about all of the afore mentioned structures, closely resembling calculus from classical analysis.

Conductive Inference

- Based on non-well-founded sets
- Uses the category-theoretic framework of universal coalgebras
- Coinduction generalizes (greatest) fixed point analysis
- Reinforcement learning: metric coinduction in stochastic coalgebras



ELSEVIER

Theoretical Computer Science 249 (2000) 3–80

Theoretical Computer Science

www.elsevier.com/locate/tcs

Fundamental Study

Universal coalgebra: a theory of systems

J.J.M.M. Rutten

CWI, P.O. Box 94079, 1090 GB Amsterdam, Netherlands

Communicated by M.W. Mislove

Abstract

In the semantics of programming, finite data types such as finite lists, have traditionally been modelled by initial algebras. Later final *coalgebras* were used in order to deal with *infinite* data types. Coalgebras, which are the dual of algebras, turned out to be suited, moreover, as models for certain types of automata and more generally, for (transition and dynamical) *systems*. An important property of initial algebras is that they satisfy the familiar principle of induction. Such a principle was missing for coalgebras until the work of Aczel (Non-Well-Founded sets, CSLI Leethre Notes, Vol. 14, center for the study of Languages and information, Stanford, 1988) on a theory of non-wellfounded sets, in which he introduced a proof principle nowadays called *coinduction*. It was formulated in terms of *bisimulation*, a notion originally stemming from the world of concurrent programming languages. Using the notion of *coalgebra homomorphism*, the definition of bisimulation on coalgebras can be shown to be formally dual to that of congruence on algebras. Thus, the three basic notions of universal algebra: algebra, homomorphism of algebras, and congruence, turn out to correspond to coalgebra, homomorphism of coalgebras, and bisimulation, respectively. In this paper, the latter are taken as the basic ingredients of a theory called *universal coalgebra*. Some standard results from universal algebra are reformulated (using the aforementioned correspondence) and proved for a large class of coalgebras, leading to a series of results on, e.g., the lattices of subcoalgebras and bisimulations, simple coalgebras and coinduction, and a covariety theorem for coalgebras similar to Birkhoff's variety theorem. © 2000 Elsevier Science B.V. All rights reserved.

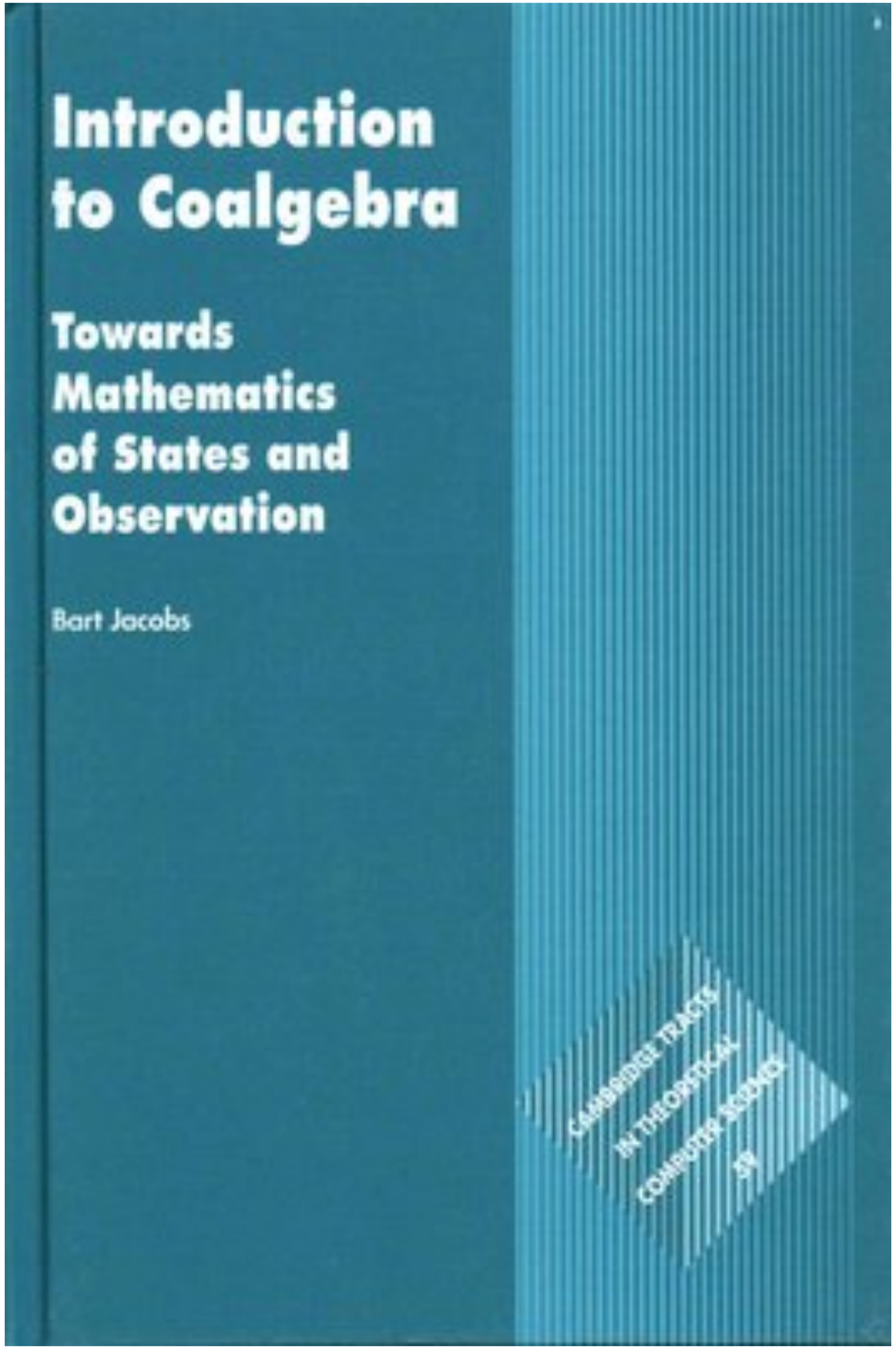
MSC: 68Q10; 68Q55

PACS: D.3; F.1; F.3

Keywords: Coalgebra; Algebra; Dynamical system; Transition system; Bisimulation; Universal coalgebra; Universal algebra; Congruence; Homomorphism; Induction; Coinduction; Variety; Covariety

E-mail address: janr@cwi.nl (J.J.M.M. Rutten).

0304-3975/00/$ - see front matter © 2000 Elsevier Science B.V. All rights reserved.
PII: S0304-3975(00)00056-6

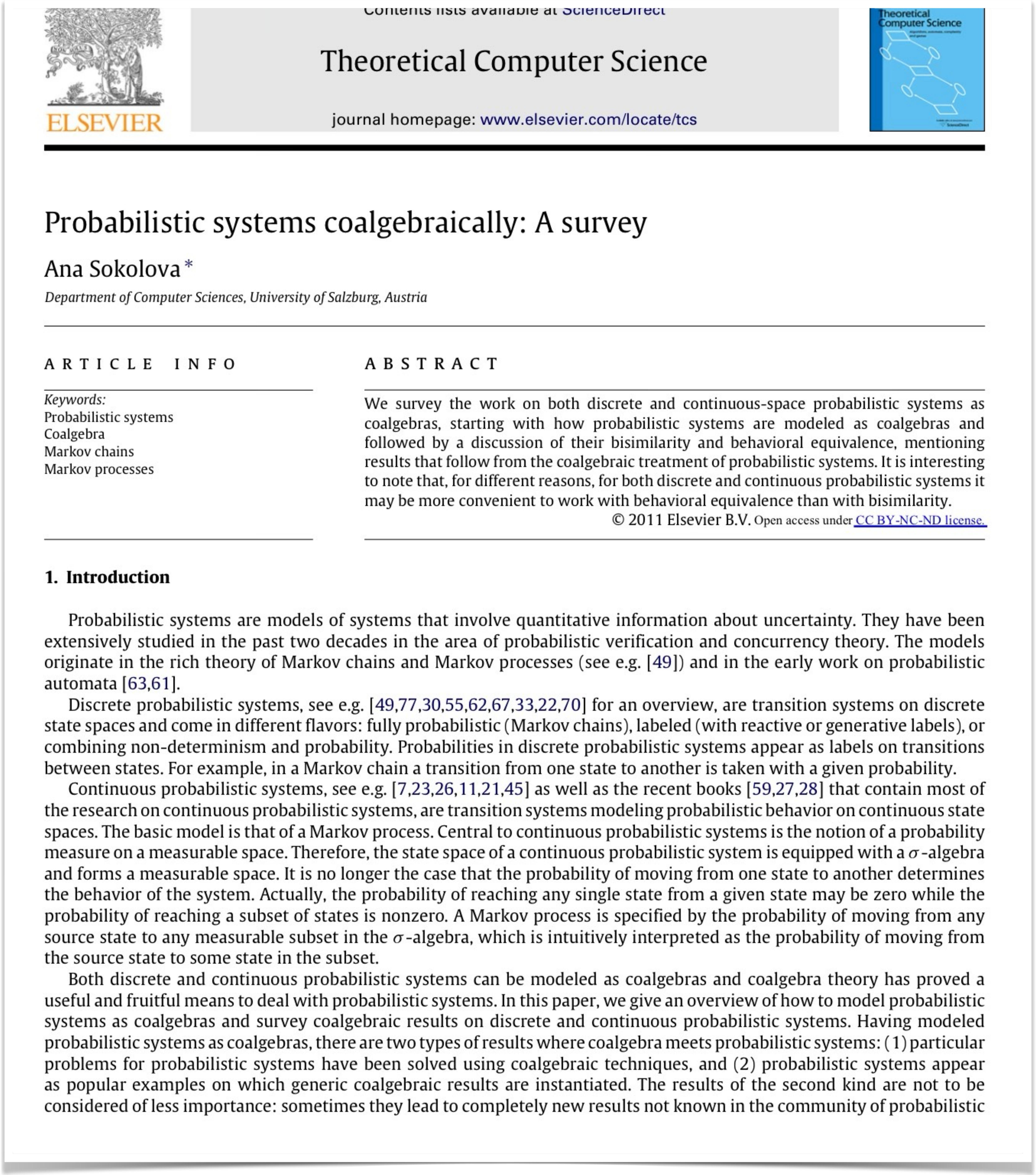


Introduction to Coalgebra

Towards Mathematics of States and Observation

Bart Jacobs

CAMBRIDGE TRACTS IN THEORETICAL COMPUTER SCIENCE 59



Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Probabilistic systems coalgebraically: A survey

Ana Sokolova *

Department of Computer Sciences, University of Salzburg, Austria

ARTICLE INFO

Keywords:
Probabilistic systems
Coalgebra
Markov chains
Markov processes

ABSTRACT

We survey the work on both discrete and continuous-space probabilistic systems as coalgebras, starting with how probabilistic systems are modeled as coalgebras and followed by a discussion of their bisimilarity and behavioral equivalence, mentioning results that follow from the coalgebraic treatment of probabilistic systems. It is interesting to note that, for different reasons, for both discrete and continuous probabilistic systems it may be more convenient to work with behavioral equivalence than with bisimilarity.

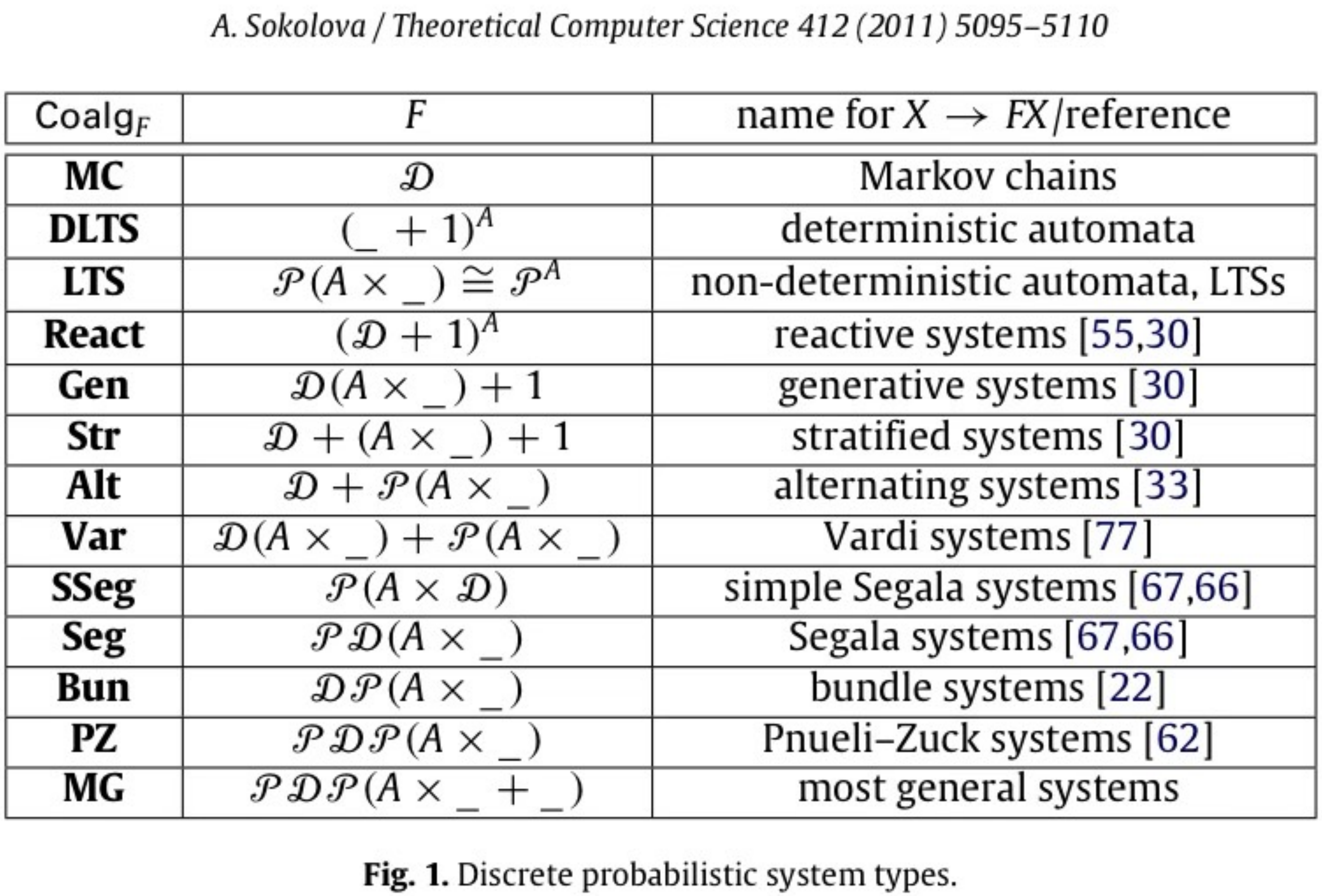
1. Introduction

Probabilistic systems are models of systems that involve quantitative information about uncertainty. They have been extensively studied in the past two decades in the area of probabilistic verification and concurrency theory. The models originate in the rich theory of Markov chains and Markov processes (see e.g. [49]) and in the early work on probabilistic automata [63,61].

Discrete probabilistic systems, see e.g. [49,77,30,55,62,67,33,22,70] for an overview, are transition systems on discrete state spaces and come in different flavors: fully probabilistic (Markov chains), labeled (with reactive or generative labels), or combining non-determinism and probability. Probabilities in discrete probabilistic systems appear as labels on transitions between states. For example, in a Markov chain a transition from one state to another is taken with a given probability.

Continuous probabilistic systems, see e.g. [7,23,26,11,21,45] as well as the recent books [59,27,28] that contain most of the research on continuous probabilistic systems, are transition systems modeling probabilistic behavior on continuous state spaces. The basic model is that of a Markov process. Central to continuous probabilistic systems is the notion of a probability measure on a measurable space. Therefore, the state space of a continuous probabilistic system is equipped with a σ-algebra and forms a measurable space. It is no longer the case that the probability of moving from one state to another determines the behavior of the system. Actually, the probability of reaching any single state from a given state may be zero while the probability of reaching a subset of states is nonzero. A Markov process is specified by the probability of moving from any source state to any measurable subset in the σ-algebra, which is intuitively interpreted as the probability of moving from the source state to some state in the subset.

Both discrete and continuous probabilistic systems can be modeled as coalgebras and coalgebra theory has proved a useful and fruitful means to deal with probabilistic systems. In this paper, we give an overview of how to model probabilistic systems as coalgebras and survey coalgebraic results on discrete and continuous probabilistic systems. Having modeled probabilistic systems as coalgebras, there are two types of results where coalgebra meets probabilistic systems: (1) particular problems for probabilistic systems have been solved using coalgebraic techniques, and (2) probabilistic systems appear as popular examples on which generic coalgebraic results are instantiated. The results of the second kind are not to be considered of less importance: sometimes they lead to completely new results not known in the community of probabilistic



A. Sokolova / Theoretical Computer Science 412 (2011) 5095–5110

Coalg_F	F	name for $X \to FX$/reference
MC	$\mathcal{D}$	Markov chains
DLTS	$(\_ + 1)^A$	deterministic automata
LTS	$\mathcal{P}(A \times \_) \cong \mathcal{P}^A$	non-deterministic automata, LTSs
React	$(\mathcal{D} + 1)^A$	reactive systems [55,30]
Gen	$\mathcal{D}(A \times \_) + 1$	generative systems [30]
Str	$\mathcal{D} + (A \times \_) + 1$	stratified systems [30]
Alt	$\mathcal{D} + \mathcal{P}(A \times \_)$	alternating systems [33]
Var	$\mathcal{D}(A \times \_) + \mathcal{P}(A \times \_)$	Vardi systems [77]
SSeg	$\mathcal{P}(A \times \mathcal{D})$	simple Segala systems [67,66]
Seg	$\mathcal{P}\mathcal{D}(A \times \_)$	Segala systems [67,66]
Bun	$\mathcal{D}\mathcal{P}(A \times \_)$	bundle systems [22]
PZ	$\mathcal{P}\mathcal{D}\mathcal{P}(A \times \_)$	Pnueli–Zuck systems [62]
MG	$\mathcal{P}\mathcal{D}\mathcal{P}(A \times \_ + \_)$	most general systems

Fig. 1. Discrete probabilistic system types.

RL algorithms can be explored for these stochastic coalgebras!

Final Coalgebras

- **In a category of coalgebras, where each object is X -> F(X), a final coalgebra is an isomorphism X ~ F(X)**

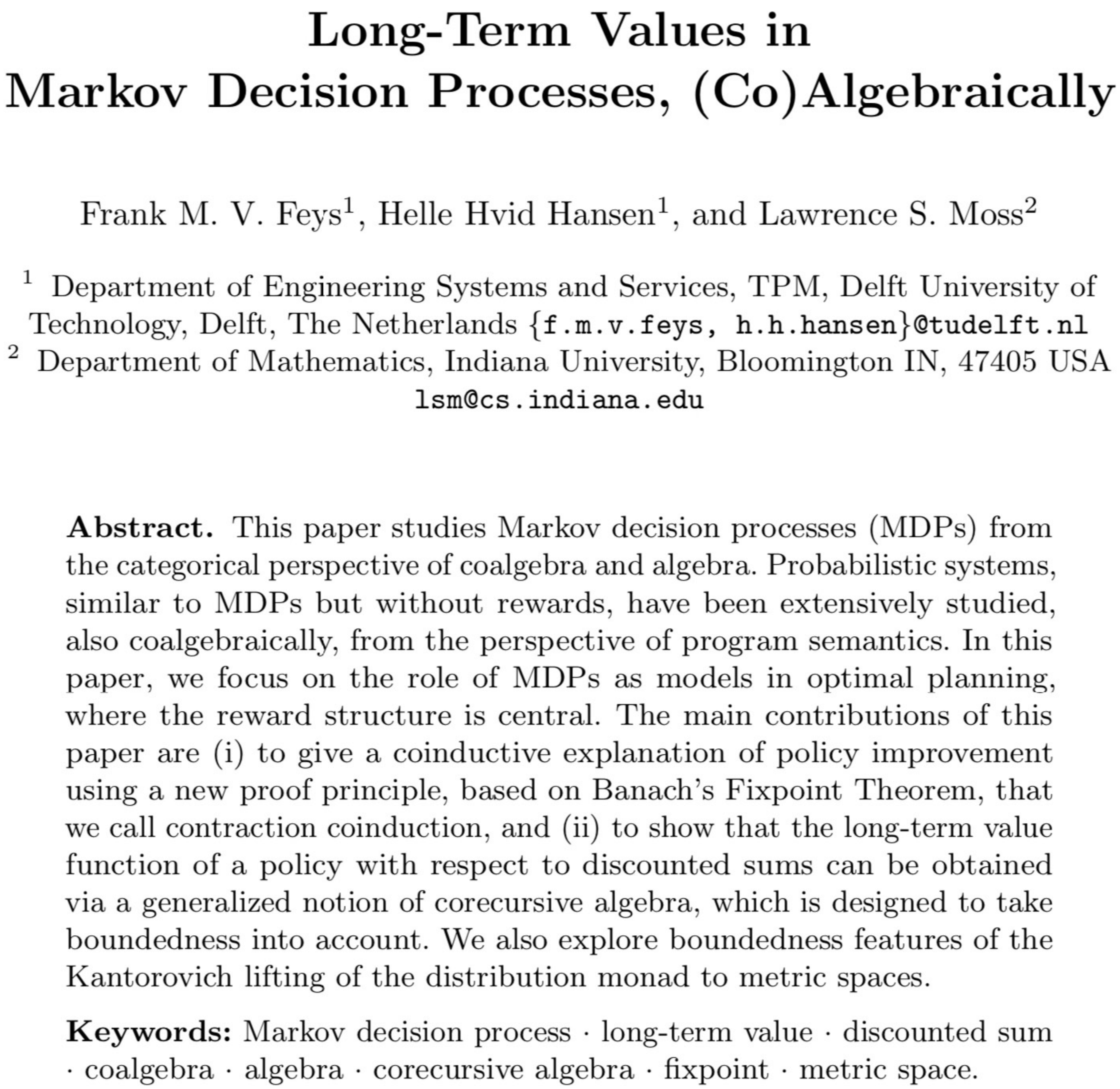
- **Final coalgebra theorem (Aczel, Mendler): for a wide class of endofunctors, final coalgebras exist (weak pullbacks)**

- **RL is essentially coinduction in a coalgebra**

$$V^{\pi} = R^{\pi} + \gamma P^{\pi} V^{\pi} = T^{\pi}(V)$$

MDP Coalgebras

- **Any MDP is defined as a tuple M = (S,A,R,P)**
- **Given any action a, it induces a distribution on next states**
- **Any fixed policy defines an induced Markov chain**
- **Markov chains are coalgebras of the distribution functor D**
  - $\alpha_S^M : S \to^M \mathscr{D}(S)$



Long-Term Values in Markov Decision Processes, (Co)Algebraically

Frank M. V. Feys[1], Helle Hvid Hansen[1], and Lawrence S. Moss[2]

[1] Department of Engineering Systems and Services, TPM, Delft University of Technology, Delft, The Netherlands {f.m.v.feys, h.h.hansen}@tudelft.nl
[2] Department of Mathematics, Indiana University, Bloomington IN, 47405 USA
lsm@cs.indiana.edu

Abstract. This paper studies Markov decision processes (MDPs) from the categorical perspective of coalgebra and algebra. Probabilistic systems, similar to MDPs but without rewards, have been extensively studied, also coalgebraically, from the perspective of program semantics. In this paper, we focus on the role of MDPs as models in optimal planning, where the reward structure is central. The main contributions of this paper are (i) to give a coinductive explanation of policy improvement using a new proof principle, based on Banach's Fixpoint Theorem, that we call contraction coinduction, and (ii) to show that the long-term value function of a policy with respect to discounted sums can be obtained via a generalized notion of corecursive algebra, which is designed to take boundedness into account. We also explore boundedness features of the Kantorovich lifting of the distribution monad to metric spaces.

Keywords: Markov decision process · long-term value · discounted sum · coalgebra · algebra · corecursive algebra · fixpoint · metric space.

This paper can be extended to the RL setting

Non-well-founded sets

- Non-well-founded sets violate the ZFC+ axioms of set theory
- In particular, the axiom of well-foundedness states that there cannot be any infinite membership chains
- Many sets in computer science are not well-founded
- Infinite data structures: lists, trees, recursion, stacks
- Many AI problems involve non-well-founded sets
    - Common knowledge, causality with feedback, natural language

The Powerset Functor

- **One of the simplest and most general coalgebras is from the powerset functor**
  - **X —> Pow(X)**
  - **X can be any (well-founded, non-well-founded) set**

Labeled Transition Systems as Coalgebras

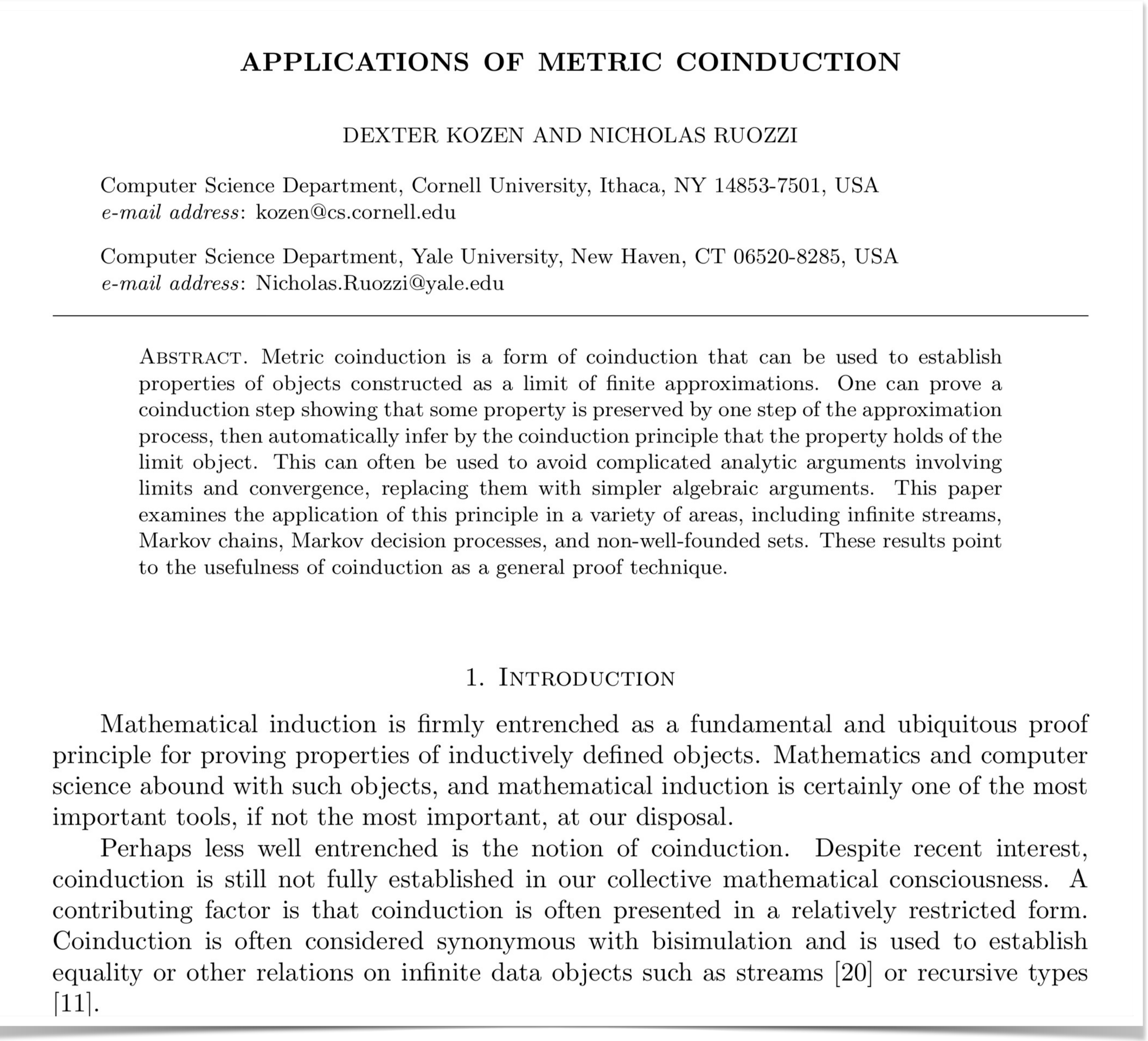
- **Any automata (deterministic or stochastic) is a coalgebra**
  - **Set of states S**
  - **Transition relation** $\rightarrow_S \subseteq S \times A \times S$
  - **Here,** $s \rightarrow^a t$ **is the same as** $(s, a, t) \in \rightarrow_S$
  - **Coalgebra of LTS defined by powerset functor L**
    - $\alpha_S : S \rightarrow L(S), s \mapsto \{(a, s') \,|\, s \rightarrow^a s'\}$

Homomorphisms of Coalgebras

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\alpha_X"'] & Y \arrow[d, "\alpha_Y"] \\
F(X) \arrow[r, "F(f)"'] & F(Y)
\end{tikzcd}$$

MDP homomorphisms are a special case of this framework

RL as Metric Coinduction



APPLICATIONS OF METRIC COINDUCTION

DEXTER KOZEN AND NICHOLAS RUOZZI

Computer Science Department, Cornell University, Ithaca, NY 14853-7501, USA
e-mail address: kozen@cs.cornell.edu

Computer Science Department, Yale University, New Haven, CT 06520-8285, USA
e-mail address: Nicholas.Ruozzi@yale.edu

Abstract. Metric coinduction is a form of coinduction that can be used to establish properties of objects constructed as a limit of finite approximations. One can prove a coinduction step showing that some property is preserved by one step of the approximation process, then automatically infer by the coinduction principle that the property holds of the limit object. This can often be used to avoid complicated analytic arguments involving limits and convergence, replacing them with simpler algebraic arguments. This paper examines the application of this principle in a variety of areas, including infinite streams, Markov chains, Markov decision processes, and non-well-founded sets. These results point to the usefulness of coinduction as a general proof technique.

1. Introduction

Mathematical induction is firmly entrenched as a fundamental and ubiquitous proof principle for proving properties of inductively defined objects. Mathematics and computer science abound with such objects, and mathematical induction is certainly one of the most important tools, if not the most important, at our disposal.

Perhaps less well entrenched is the notion of coinduction. Despite recent interest, coinduction is still not fully established in our collective mathematical consciousness. A contributing factor is that coinduction is often presented in a relatively restricted form. Coinduction is often considered synonymous with bisimulation and is used to establish equality or other relations on infinite data objects such as streams [20] or recursive types [11].

$$\frac{\exists u\ \varphi(u) \qquad \forall u\ \varphi(u) \Rightarrow \varphi(H(u))}{\varphi(u^*)}$$

Contraction mapping convergence in MDPs

is a special case of metric coinduction

Induction vs Coinduction

- **Given the class of all (non)well-founded sets**
  - **X —> F(X) is the powerset coalgebra**
  - **F(X) —> X is the powerset algebra**
- **The initial object in the category of algebras is well-founded sets**
- **The final object in the category of coalgebras is non-well-founded sets**

Final Coalgebras

- **A final object in a category is defined as one for which there is a unique morphism into it from any other object**

- **In the category of coalgebras, the final object is called a final coalgebra**

- **Example: in the coalgebra of finite state automata, the final coalgebra is the smallest automaton accepting a language**

- **Example: in the coalgebra of MDPs, the final coalgebra is the smallest MDP that defines the optimal value function**

Lambek's Lemma

Definition 83. An F-coalgebra (A, α) is a *fixed point* for F, written as $A \simeq F(A)$ if α is an isomorphism between A and $F(A)$. That is, not only does there exist an arrow $A \rightarrow F(A)$ by virtue of the coalgebra α, but there also exists its inverse $\alpha^{-1} : F(A) \rightarrow A$ such that

$$\alpha \circ \alpha^{-1} = \mathbf{id}_{F(A)} \text{ and } \alpha^{-1} \circ \alpha = \mathbf{id}_A$$

The following lemma was shown by Lambek, and implies that the transition structure of a final coalgebra is an isomorphism.

Theorem 23. **Lambek:** A final F-coalgebra is a fixed point of the endofunctor F.

Math. Struct. in Comp. Science (2005), *vol.* 15, *pp.* 409–432. © 2005 Cambridge University Press
doi:10.1017/S0960129505004731 Printed in the United Kingdom

A general final coalgebra theorem

JIŘÍ ADÁMEK[†], STEFAN MILIUS[‡] and JIŘÍ VELEBIL[§]

[†‡]*Institute of Theoretical Computer Science, Technical University of Braunschweig, Germany*
E-mail: {adamek,milius}@iti.cs.tu-bs.de
[§]*Faculty of Electrical Engineering, Czech Technical University, Prague*

Received 27 March 2003

By the Final Coalgebra Theorem of Aczel and Mendler, every endofunctor of the category of sets has a final coalgebra, which, however, may be a proper class. We generalise this to all 'well-behaved' categories $\mathscr{K}$. The role of the category of classes is played by a free cocompletion $\mathscr{K}^\infty$ of $\mathscr{K}$ under transfinite colimits, that is, colimits of ordinal-indexed chains. Every endofunctor F of $\mathscr{K}$ has a canonical extension to an endofunctor F^∞ of $\mathscr{K}^\infty$ which is proved to have a final coalgebra (and an initial algebra). Based on this, we prove a general solution theorem: for every endofunctor of a locally presentable category $\mathscr{K}$ all guarded equation-morphisms have unique solutions. The last result does not need the extension $\mathscr{K}^\infty$: the solutions are always found within the category $\mathscr{K}$.

Occam's Razor Coalgebraically

- **We can now define a coalgebraic version of Occam's Razor**

- **Given any category of coalgebras, where there is a final coalgebra**

- **Any other coalgebra must define a unique morphism into the final coalgebra**

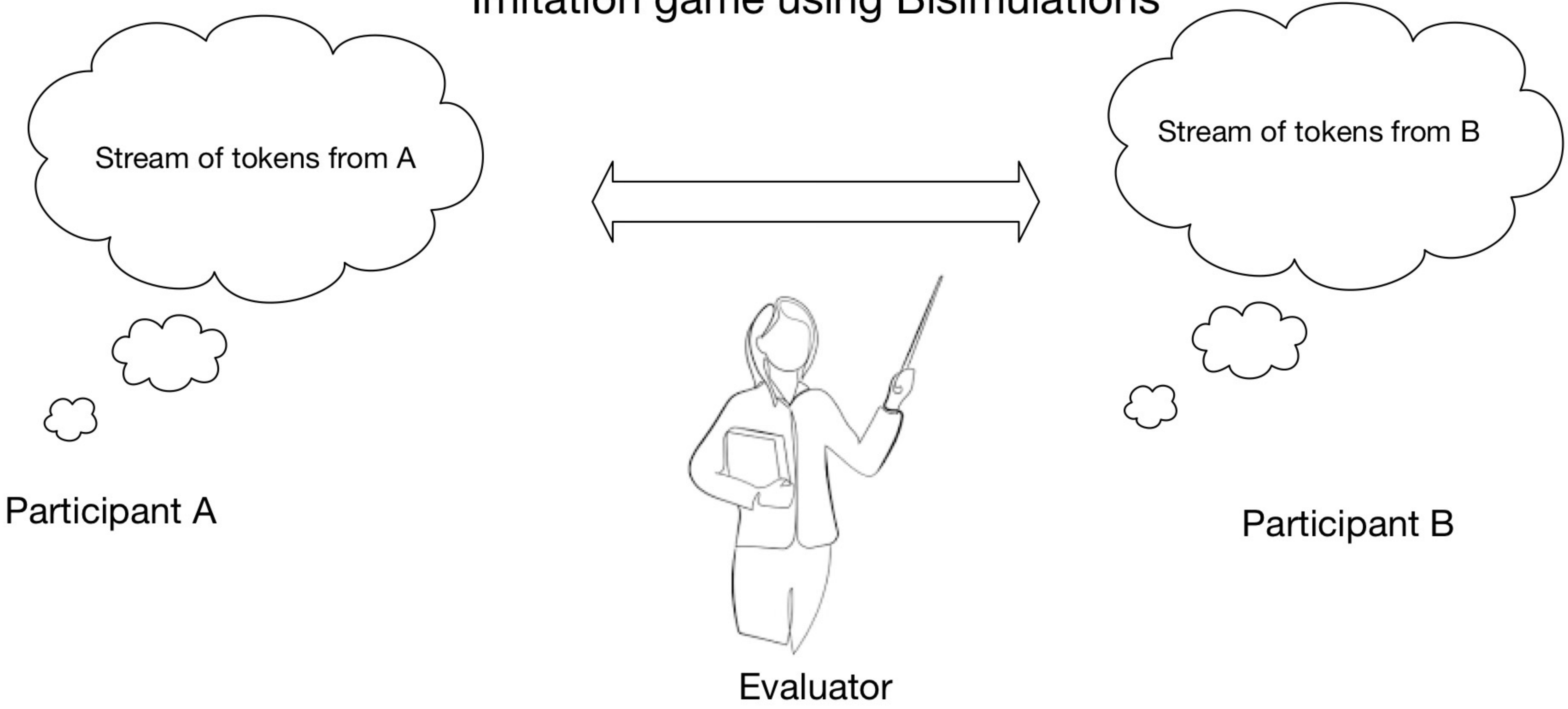
- **If this unique morphism is injective (or a monomorphism), the given coalgebra must be minimal**

- **States of the final coalgebra define ``behaviors" (see Jacobs book)**

Bisimulation for Imitation Games

$$\begin{tikzcd}
S \arrow[d, "\alpha_S"'] & R \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, dashed, "\alpha_R", "\exists"'] & T \arrow[d, "\alpha_T"] \\
F(S) & F(R) \arrow[l, "F(\pi_1)"] \arrow[r, "F(\pi_2)"'] & F(T).
\end{tikzcd}$$

Imitation game using Bisimulations



Stream of tokens from A

Stream of tokens from B

Participant A

Participant B

Evaluator

Summary

- Coalgebras provide a fundamental framework for modeling generative AI
- Each coalgebra is defined by a functor F: X —> F(X)
- Coinduction is the principle of finding a final coalgebra
- Reinforcement learning is the problem of finding final coalgebras in the category of MDP coalgebras