GENERIC TRACE SEMANTICS VIA COINDUCTION

ICHIRO HASUO, BART JACOBS, AND ANA SOKOLOVA

Institute for Computing and Information Sciences, Radboud University Nijmegen, the Netherlands
and Research Institute for Mathematical Sciences, Kyoto University, Japan

Institute for Computing and Information Sciences, Radboud University Nijmegen, the Netherlands

Department of Computer Sciences, University of Salzburg, Austria

ABSTRACT. Trace semantics has been defined for various kinds of state-based systems, notably with different forms of branching such as non-determinism vs. probability. In this paper we claim to identify one underlying mathematical structure behind these "trace semantics", namely coinduction in a Kleisli category. This claim is based on our technical result that, under a suitably order-enriched setting, a final coalgebra in a Kleisli category is given by an initial algebra in **Sets**. Formerly the theory of coalgebras has been employed mostly in **Sets** where coinduction yields a finer process semantics of bisimilarity. Therefore this paper extends the application field of coalgebras, providing a new instance of the principle "process semantics via coinduction".

1. INTRODUCTION

Trace semantics is a commonly used semantic relation for reasoning about state-based systems. Trace semantics for labeled transition systems is found on the coarsest edge of the linear time-branching time spectrum [57]. Moreover, trace semantics is defined for a variety of systems, among which are probabilistic systems [49].

In this paper we claim that these various forms of "trace semantics" are instances of a general construction, namely coinduction in a Kleisli category. Our point of view here is categorical, coalgebraic in particular. Hence this paper demonstrates the abstraction power of categorical/coalgebraic methods in computer science, uncovering basic mathematical structures underlying various concrete examples.
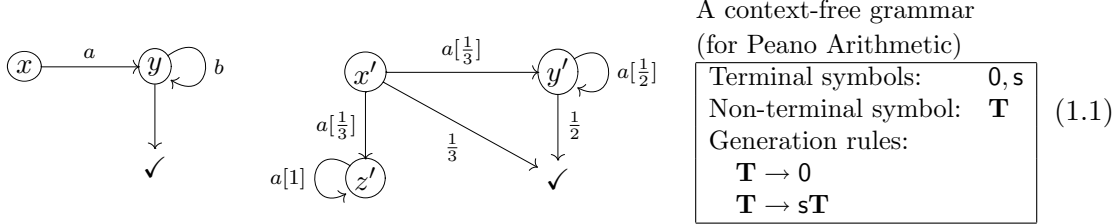
DOI:10.2168/LMCS-???

1.1. "Trace semantics" in various contexts.

1.1. "Trace semantics" in various contexts. First we motivate our contribution through examples of various forms of "trace semantics". Think of the following three state-based, branching systems.

A context-free grammar
(for Peano Arithmetic)

Terminal symbols:	$0, \mathsf{s}$
Non-terminal symbol:	$\mathbf{T}$
Generation rules:	
$\mathbf{T} \to 0$	
$\mathbf{T} \to \mathsf{s}\mathbf{T}$	

(1.1)

- The first one is a non-deterministic system with a special state ✓ denoting successful termination. To its state x we can assign its *trace set*:

$$\mathsf{tr}(x) = \{a, ab, abb, \dots\} = ab^* \ , \tag{1.2}$$

  that is, the *set* of the possible linear-time behavior (namely words) that can arise through an execution of the system.[1] In this case the trace set $\mathsf{tr}(x)$ is also called the *accepted language*; formally it is defined (co)recursively by the following equations. For an arbitrary state x,

$$\begin{aligned} \langle\rangle \in \mathsf{tr}(x) &\iff x \to \checkmark \\ a \cdot \sigma \in \mathsf{tr}(x) &\iff \exists y.\, (\, x \xrightarrow{a} y \,\wedge\, \sigma \in \mathsf{tr}(y)\,) \end{aligned} \tag{1.3}$$

  Here $\langle\rangle$ denotes the empty word; $\sigma = a_1 a_2 \dots a_n$ is a word in actions.
- The second system has a different type of branching, namely probabilistic branching. Here $x' \xrightarrow{a[1/3]} y'$ denotes: at the state x', a transition to y' outputting a occurs with probability $1/3$. Now, to the state x', we can assign its *trace distribution*:

$$\mathsf{tr}(x) = \left[\begin{array}{l} \langle\rangle \mapsto \frac{1}{3}, \quad a \mapsto \frac{1}{3}\cdot\frac{1}{2}, \quad a^2 \mapsto \frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{2}, \quad \cdots \\ a^n \mapsto \frac{1}{3}\cdot\left(\frac{1}{2}\right)^n, \quad \cdots \end{array}\right] \ ,$$

  that is, the probability distribution over the set of linear-time behavior.[2] Its formal (corecursive) definition is as follows.

$$\begin{aligned} \mathsf{tr}(x)(\langle\rangle) &= \mathsf{Pr}(x \to \checkmark) \ , \\ \mathsf{tr}(x)(a \cdot \sigma) &= \textstyle\sum_{y \in X} \mathsf{Pr}(x \xrightarrow{a} y) \cdot \mathsf{tr}(y)(\sigma) \ , \end{aligned} \tag{1.4}$$

  where $\mathsf{Pr}(\dots)$ denotes the probability of an transition.
- The third one can be thought of as state-based systems, with non-terminal symbols as states. It is non-deterministic because a state $\mathbf{T}$ has two possible transitions. It comes natural to call the following set of parse-trees its "trace semantics".

  It is also a set of linear-time behavior (as in the first example), but the notion of linear-time behavior is different here.

---

[1] The infinite trace ab^ω is out of our scope here: we will elaborate this point later in Section 4.2.

[2] Here again, we do not consider the infinite trace $a^\omega \mapsto 1/3$.

1.2. **Coalgebras and coinduction.** In recent years the theory of *coalgebras* has emerged as the "mathematics of state-based systems" [25, 47, 26]. In the categorical theory of coalgebras, an important definition/reasoning principle is *coinduction*: a system (identified with a coalgebra $c : X \to FX$) is assigned a unique morphism beh_c into the final coalgebra.

$$
\begin{array}{ccc}
FX & \dashrightarrow^{F(\mathsf{beh}_c)} & FZ \\
c\uparrow & & \cong\uparrow\text{final} \\
X & \dashrightarrow_{\mathsf{beh}_c} & Z
\end{array}
$$

The success of coalgebras is largely due to the fact that, when **Sets** is taken as the base category, the final coalgebra semantics is fully abstract with respect to the conventional notion of *bisimilarity*: for states x and y of coalgebras $X \xrightarrow{c} FX$ and $Y \xrightarrow{d} FY$,

$$\mathsf{beh}_c(x) = \mathsf{beh}_d(y) \qquad \Longleftrightarrow \qquad x \text{ and } y \text{ are bisimilar.}$$

This is the case for a wide variety of systems (i.e. for a variety of functors F), hence *coinduction in* **Sets** *captures bisimilarity.*

However, there is not so much work so far that captures other behavioral equivalences (coarser than bisimilarity) by the categorical principle of coinduction. The current work—capturing trace semantics by coinduction in a Kleisli category—therefore extends the application field of the theory of coalgebras.

1.3. **Our contributions.** Our technical contributions are summarized as follows. Assume that T is a monad on **Sets** which has a suitable order structure; we shall denote its Kleisli category by $\mathcal{K}\ell(T)$.

- *Trace semantics via coinduction in a Kleisli category.* Commutativity of the coinduction diagram

$$
\begin{array}{cccl}
\overline{F}X & \dashrightarrow^{\overline{F}(\mathsf{tr}_c)} & \overline{F}Z & \quad \text{in} \quad \mathcal{K}\ell(T), \\
c\uparrow & & \cong\uparrow\text{final} & \quad \text{the Kleisli category for } T \\
X & \dashrightarrow_{\mathsf{tr}_c} & Z &
\end{array}
\qquad (1.5)
$$

  is shown to be equivalent to the conventional recursive definition of trace semantics such as (1.3) and (1.4). This is true for both trace set semantics (for nondeterministic systems) and trace distribution semantics (for probabilistic systems). The induced arrow tr_c thus gives (conventional) trace semantics for a system c.
- *Identification of the final coalgebra in a Kleisli category.* We show that

  > an initial algebra in **Sets**
  > coincides with
  > a final coalgebra in $\mathcal{K}\ell(T)$.

  In particular, the final coalgebra in **Rel** is the initial algebra in **Sets**, because the category **Rel** of sets and relations is a Kleisli category for a suitable monad. This coincidence happens in the following two steps:
    - the initial algebra in **Sets** lifts to a Kleisli category, due to a suitable adjunction-lifting result;

  – in a Kleisli category we have *initial algebra-final coalgebra coincidence.* Here we use the classical result by Smyth and Plotkin [51], namely *limit-colimit coincidence* which is applicable in a suitably order-enriched category.

Note the presence of two parameters in (1.5): a monad T and an endofunctor F, both on **Sets**. The monad T specifies the *branching type* of systems. We have three leading examples:[3]

- the powerset monad $\mathcal{P}$ modeling *non-deterministic* or *possibilistic* branching;
- the subdistribution monad $\mathcal{D}$

$$\mathcal{D}X = \{d : X \to [0,1] \mid \sum_{x \in X} d(x) \leq 1\}$$

  modeling *probabilistic* branching; and
- the lift monad $\mathcal{L} = 1 + (\_)$ modeling system with *exception* (or *deadlock, non-termination*).

The functor F specifies the *transition type* of systems: our understanding of "transition type" shall be clarified by the following examples.

- In labeled transition systems (LTSs) with explicit termination—no matter if they are non-deterministic or even probabilistic—a state either
    – terminates $(x \to \checkmark)$, or
    – outputs one symbol and moves to another state $(x \xrightarrow{a} x')$,
  in one transition. This "transition type" is expressed by the functor $FX = 1 + \Sigma \times X$, where Σ is the output alphabet and $1 = \{\checkmark\}$.
- In context-free grammars (CFGs) as state-based systems, a state evolves into a sequence of terminal and non-terminal symbols in a transition. The functor

$$FX = (\Sigma + X)^*$$

  with Σ being the set of terminal symbols, expresses this transition type.

Clear separation of branching and transition types is important in our generic treatment of trace semantics. The transition type F determines the set of linear-time behavior (which is in fact given by the initial F-algebra in **Sets**). We model a system by a coalgebra $X \xrightarrow{c} \overline{F}X$ in the Kleisli category $\mathcal{K}\ell(T)$—see (1.5)—where $\overline{F}$ is a suitable lifting of F in $\mathcal{K}\ell(T)$. By the definition of a Kleisli category we will easily see the following bijective correspondence.

$$\frac{X \xrightarrow{\quad c \quad} \overline{F}X \ \ \text{in } \mathcal{K}\ell(T)}{X \xrightarrow{\quad c \quad} TFX \ \ \text{in } \textbf{Sets}}$$

Hence our system—a *function* of the type $X \to TFX$—first resolves a branching of type T and then makes a transition of type F. Many branching systems allow such representation so that our generic coalgebraic trace semantics applies to them.

---

[3]Other examples include the monad $X \mapsto (\mathbb{N} \cup \{\infty\})^X$ for multisets, the monad $X \mapsto [0,\infty]^X$ for real valuations, and the monad $X \mapsto \mathcal{P}(M \times \_)$ with a monoid M for timed systems [30]. These monads can be treated in a similar way as our leading examples. We leave out the details.

1.4. **Generic theory of traces and simulations.** In the study of coalgebras as 'categorical presentation of state-based systems', there are three ingredients playing crucial roles: *coalgebras* as systems; *coinduction* yielding process semantics; and *morphisms of coalgebras* as behavior-preserving maps. In this paper we study the first two in a Kleisli category. What about morphisms of coalgebras?

In [14] this question is answered by identifying *lax/oplax morphisms of coalgebras* in a Kleisli category as *forward/backward simulations*. Use of traces and simulations is a common technique in formal verification of systems (see e.g. [41]): a desirable property is expressed in terms of traces; and then a system is shown to satisfy the property by finding a suitable simulation. Therefore this paper, together with [14], forms an essential part of developing a "generic theory of traces and simulations" using coalgebras in a Kleisli category. The categorical genericity—especially the fact that we can treat non-deterministic and probabilistic branching in a uniform manner—is exploited in [19] to obtain a simulation-based proof method for a probabilistic notion of anonymity for network protocols. Currently we are investigating how much more applicational impact can be brought about by our generic theory of traces and simulations.

1.5. **Testing and trace semantics.** Since the emergence of the theory of coalgebras, the significance of *modal logics* as specification languages has been noticed by many authors. This is exemplified by the slogan in [36]: 'modal logic is to coalgebras what equational logic is to algebras'. Inspired by coalgebras on Stone spaces and the corresponding modal logic, recent developments [35, 5, 6, 37, 45, 33, 32] have identified the following situation as the essential mathematical structure underlying modal logics for coalgebras.

$$F^{\mathrm{op}} \left(\mathbb{C}^{\mathrm{op}} \underset{S^{\mathrm{op}}}{\overset{P}{\underset{\top}{\rightleftarrows}}} \mathbb{A} \right) M \quad \text{together with} \quad MP \overset{\delta}{\Longrightarrow} PF^{\mathrm{op}}$$

In fact, it is noticed in [45] that such a situation not only hosts a modal logic but also a more general notion of *testing* (in the sense of [57, 53], also called *testing scenarios*). Therefore we shall call the above situation a *testing situation*.

In the last technical section of the paper we investigate coalgebraic trace semantics for the special case $T = \mathcal{P}$ (modeling non-determinism) from this testing point of view. First, we present some basic facts on testing situations, especially on the relationship between the induced *testing equivalence* and the final coalgebra semantics. These two process equivalences are categorically presented as kernel pairs, which enables a fairly simple presentation of the theory of coalgebraic testing. In addition, we observe that the coinduction scheme in the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ gives rise to a canonical testing situation, in which the set of tests is given by an initial F-algebra.

The material on testing in the last section has not been presented in the earlier versions [16, 17] of this paper.

1.6. **Organization of the paper.** In Section 2 we observe that a coalgebra in a Kleisli category is an appropriate "denotation" of a branching system, when we focus on trace semantics. In Section 3 we present our main technical result that an initial algebra in **Sets** yields a final coalgebra in $\mathcal{K}\ell(T)$. The relationship to axiomatic domain theory—which employs similar mathematical arguments—is also discussed here. Section 4 presents some examples of the use of coinduction in $\mathcal{K}\ell(T)$ and argues that the coinduction principle is

a general form of trace semantics. In Section 5 we review the preceding material from the testing point of view.

2. Coalgebras in a Kleisli category

In the study of coalgebras as "categorical presentations of state-based systems", the category **Sets** of sets and functions has been traditionally taken as a base category (see e.g. [25, 47]). An important fact in such a setting is that bisimilarity is often captured by coinduction.[4]

However, bisimilarity is not the only process equivalence. In some applications one would like coarser equivalences, for example in order to abstract away internal branching structures. One of such coarser semantics, which has been extensively studied, is *trace equivalence*. For example, the process algebra CSP [21] has trace semantics as its operational model. Trace equivalence is coarser than bisimilarity, as the following classic example of "trace-equivalent but not bisimilar" systems illustrates.

It is first noticed in [46] that the Kleisli category for the powerset monad is an appropriate base category for trace semantics for non-deterministic systems. This observation is pursued further in [24, 16, 17]. In [15] it is recognized that the same is true for the subdistribution monad for probabilistic systems. The current paper provides a unified framework which yields those preceding results, in terms of **Cppo**-enrichment of a Kleisli category; see Section 2.3. In this section we first aim to justify the use of coalgebras in a Kleisli category.

2.1. **Monads and Kleisli categories.** Here we recall the relevant facts about monads and Kleisli categories. For simplicity we exclusively consider monads on **Sets**.

A *monad* on **Sets** is a categorical construct. It consists of

- an endofunctor T on **Sets**;
- a *unit* natural transformation $\eta : \mathrm{id} \Rightarrow T$, that is, a function $X \xrightarrow{\eta_X} TX$ for each set X satisfying a suitable naturality condition; and
- a *multiplication* natural transformation $\mu : T^2 \Rightarrow T$, consisting of functions $T^2 X \xrightarrow{\mu_X} TX$ with X ranging over sets.

The unit and multiplication are required to satisfy the following compatibility conditions.

$$TX \xrightarrow{\eta_{TX}} T^2 X \xleftarrow{T\eta_X} TX \qquad\qquad T^3 X \xrightarrow{T\mu_X} T^2 X$$

See [42, 3] for the details.

---

[4]Non-examples include LTSs with unbounded branching degree. They are modeled as coalgebras for $FX = \mathcal{P}(\Sigma \times X)$. Lambek's Lemma readily shows that this choice of F does not have a final coalgebra in **Sets**, because it would imply an isomorphism $Z \xrightarrow{\cong} \mathcal{P}(\Sigma \times Z)$ which is impossible for cardinality reasons.

The monad structures play a crucial role in modeling "branching". Intuitively, the unit η embeds a non-branching behavior as a trivial branching (with only one possibility to choose). The multiplication μ "flattens" two successive branchings into one branching, abstracting away internal branchings:

 (2.1)

The following examples will illustrate how this flattening phenomenon is a crucial feature of trace semantics.

In this paper we concentrate on the three monads mentioned in the introduction: $\mathcal{L}$, $\mathcal{P}$ and $\mathcal{D}$.

- The lift monad $\mathcal{L} = 1 + (\_)$—where we denote $1 = \{\bot\}$ with $\bot$ meaning *deadlock*—has a standard monad structure induced by a coproduct. For example, the multiplication $\mu_X^{\mathcal{L}} : 1 + 1 + X \to 1 + X$ carries $x \in X$ to itself and both $\bot$'s to $\bot$.
- The powerset monad $\mathcal{P}$ has a unit given by singletons and a multiplication given by unions. The monad $\mathcal{P}$ models non-deterministic branching: the "flattening" in (2.1) corresponds to the following application of the multiplication of $\mathcal{P}$.

$$\mathcal{PP}X \xrightarrow{\quad \mu_X^{\mathcal{P}} \quad} \mathcal{P}X$$
$$\big\{\, \{x,y\}, \{z\} \,\big\} \longmapsto \{x,y,z\}$$

- The subdistribution monad $\mathcal{D}$ has a unit given by the *Dirac distributions*.

$$X \xrightarrow{\quad \eta_X^{\mathcal{D}} \quad} \mathcal{D}X$$
$$x \longmapsto \left[\begin{array}{l} x \mapsto 1 \\ x' \mapsto 0 \quad (\text{for } x' \neq x) \end{array}\right]$$

Its multiplication is given by multiplying the probabilities along the way. That is,

$$\mu_X^{\mathcal{D}}(\xi) = \lambda x. \sum_{d \in \mathcal{D}X} \xi(d) \cdot d(x) \ ,$$

which models "flattening" of the following kind.

that is,

$$\left[\begin{array}{ll} \left[\begin{array}{l} x \mapsto 1/2 \\ y \mapsto 1/2 \end{array}\right] & \mapsto \quad 1/3 \\ [z \mapsto 1] & \mapsto \quad 2/3 \end{array}\right] \quad \xmapsto{\mu} \quad \left[\begin{array}{l} x \mapsto 1/6 \\ y \mapsto 1/6 \\ z \mapsto 2/3 \end{array}\right] \ .$$

Given any monad T, its *Kleisli category* $\mathcal{K\ell}(T)$ is defined as follows. Its objects are the objects of the base category, hence sets in our consideration. An arrow $X \to Y$ in $\mathcal{K\ell}(T)$ is

the same thing as an arrow $X \to TY$ in the base category, here **Sets**.

$$\frac{X \longrightarrow Y \;\; \text{in } \mathcal{K}\ell(T)}{X \longrightarrow TY \;\; \text{in } \mathbf{Sets}}$$

Identities and composition of arrows are defined using the unit and the multiplication of T. Moreover, there is a canonical adjunction

$$\mathbf{Sets} \underset{K}{\overset{J}{\rightleftarrows}} \bot \; \mathcal{K}\ell(T) \qquad\qquad (2.2)$$

such that J carries $X \xrightarrow{f} Y$ in **Sets** to $X \xrightarrow{\eta_Y \circ f} Y$ in $\mathcal{K}\ell(T)$. See [42, 3] for details.

The relevance in this paper is that a Kleisli category can be thought of as a category where the branching is implicit. For example, an arrow $X \to Y$ in the Kleisli category $\mathcal{K}\ell(\mathcal{P})$ is a function $X \to \mathcal{P}Y$ hence a "non-deterministic function". When $T = \mathcal{D}$, then by writing $X \to Y$ in the Kleisli category we mean a function with probabilistic branching. Moreover, composition of arrows in $\mathcal{K}\ell(T)$ is given by

$$X \xrightarrow{f} Y \xrightarrow{g} Z \;\; \text{in } \mathcal{K}\ell(T) \quad = \quad X \xrightarrow{f} TY \xrightarrow{Tg} T^2 Z \xrightarrow{\mu_Z} TZ \;\; \text{in } \mathbf{Sets};$$

that is, making one transition (by g) after another (by f), and then flattening (by μ_Z). For example, this general definition instantiates as follows when $T = \mathcal{D}$. For $X \xrightarrow{f} Y \xrightarrow{g} Z$,

$$(g \circ f)(x)(z) = \sum_{y \in Y} f(x)(y) \cdot g(y)(z) \ .$$

2.2. **Lifting functors by distributive laws.** In this paper a state-based system is presented as a coalgebra $X \to \overline{F}X$ in $\mathcal{K}\ell(T)$, where $\overline{F} : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ is a lifting of $F : \mathbf{Sets} \to \mathbf{Sets}$. This lifting $F \mapsto \overline{F}$ is equivalent to a *distributive law* $FT \Rightarrow TF$. The rest of this section elaborates on this point.

Various kinds of state-based, branching systems are expressed as a function of the form $X \xrightarrow{c} TFX$ with T a monad (for branching type) and F a functor (for transition type). The following examples are already hinted at in the introduction.

- For $T = \mathcal{P}$ and $F = 1 + \Sigma \times \_$, a function $X \xrightarrow{c} TFX$ is an LTS with explicit termination. For example, consider the following system

$$
\begin{array}{ccl}
X & \xrightarrow{\quad\quad c \quad\quad} & \mathcal{P}(1 + \Sigma \times X) \\
x & \longmapsto & \{\checkmark, (a_1, x_1), (a_2, x_2)\}
\end{array}
$$

  where $\checkmark$ is the element of 1.[5] Then the state x can make three possible transitions, namely: $x \to \checkmark$ (successful termination), $x \xrightarrow{a_1} x_1$, and $x \xrightarrow{a_2} x_2$, when written in a conventional way.

_____

[5]Note that the singleton $1 = \{\checkmark\}$ here in $F = 1 + \Sigma \times \_$ has a different interpretation from $1 = \{\bot\}$ in $T = \mathcal{L} = 1 + \_$. The intuition is as follows. On the one hand, when an execution hits successful termination $\checkmark$, it yields its history of observations as its trace. On the other hand, when an execution hits deadlock $\bot$ then it yields no trace no matter what is the history before hitting $\bot$. This distinction will be made formal in Example 4.3.

- By replacing $T = \mathcal{P}$ by $\mathcal{D}$, but keeping F the same, we obtain a probabilistic system such as the one in the middle of (1.1). For example,

$$X \xrightarrow{\quad c \quad} \mathcal{D}(1 + \Sigma \times X)$$

$$x' \longmapsto \left[\begin{array}{l} (a, y') \mapsto 1/3 \\ (a, z') \mapsto 1/3 \\ \checkmark \mapsto 1/3 \end{array} \right] \quad .$$

- For $T = \mathcal{P}$ and $F = (\Sigma + \_)^*$, a function $X \xrightarrow{c} TFX$ is a CFG with Σ the terminal alphabet (but without finiteness conditions). See [16] for more details.

All these systems are modeled by a function $X \xrightarrow{c} TFX$, hence an arrow $X \xrightarrow{c} FX$ in $\mathcal{K}\ell(T)$. Our question here is: is c a coalgebra in $\mathcal{K}\ell(T)$? In other words: is the functor F on **Sets** also a functor on $\mathcal{K}\ell(T)$?

Hence, to develop a generic theory of traces in $\mathcal{K}\ell(T)$, we need to lift F to a functor $\overline{F}$ on $\mathcal{K}\ell(T)$. A functor $\overline{F}$ is said to be a *lifting* of F if the following diagram commutes. Here J is the left adjoint in (2.2).

$$\begin{array}{ccc} \mathcal{K}\ell(T) & \xrightarrow{\overline{F}} & \mathcal{K}\ell(T) \\ J\uparrow & & \uparrow J \\ \mathbf{Sets} & \xrightarrow{F} & \mathbf{Sets} \end{array} \qquad (2.3)$$

The following fact is presented in [43]; see also [39, 40]. Its proof is straightforward.

Lemma 2.1. *A lifting $\overline{F}$ of F is in bijective correspondence with a **distributive law** $\lambda : FT \Rightarrow TF$. A distributive law λ is a natural transformation which is compatible with T's monad structure, in the following way.*

$$\begin{array}{ccc} FX \xrightarrow{F\eta_X} FTX & & FT^2X \xrightarrow{\lambda_{TX}} TFTX \xrightarrow{T\lambda_X} T^2FX \\ \searrow_{\eta_{FX}} \quad \downarrow{\lambda_X} & & F\mu_X\downarrow \qquad\qquad\qquad\qquad \downarrow{\mu_{FX}} \\ \qquad TFX & & FTX \xrightarrow{\qquad\qquad \lambda_X \qquad\qquad} TFX \end{array} \qquad \square$$

A distributive law λ induces a lifting $\overline{F}$ as follows. On objects: $\overline{F}X = FX$. Given $f : X \to Y$ in $\mathcal{K}\ell(T)$, we need an arrow $\overline{F}f : FX \to FY$ in $\mathcal{K}\ell(T)$. Recall that f is a function $X \to TY$ in **Sets**; one takes $\overline{F}f$ to be the arrow which corresponds to the function

$$FX \xrightarrow{Ff} FTY \xrightarrow{\lambda_Y} TFY \quad \text{in } \mathbf{Sets} \ .$$

A distributive law specifies how a branching (of type T) "distributes" over a transition (of type F). Let us look at an example. For $T = \mathcal{P}$ and $F = 1 + \Sigma \times \_$ (the combination for LTSs with explicit termination), we have the following distributive law.

$$1 + \Sigma \times (\mathcal{P}X) \xrightarrow{\quad \lambda_X \quad} \mathcal{P}(1 + \Sigma \times X)$$

$$\checkmark \longmapsto \{\checkmark\}$$

$$(a, S) \longmapsto \big\{(a, x) \mid x \in S\big\}$$

For example,

$$\bullet \xrightarrow{a} \begin{smallmatrix} \nearrow x \\ \rightsquigarrow y \\ \searrow z \end{smallmatrix} \quad \xmapsto{\lambda} \quad \bullet \begin{smallmatrix} \xrightarrow{a} x \\ \xrightarrow{a} y \\ \xrightarrow{a} z \end{smallmatrix} \qquad \text{that is} \qquad \big(a, \{x, y, z\}\big) \xmapsto{\lambda} \big\{(a, x), (a, y), (a, z)\big\} \ ,$$

where waving arrows $\rightsquigarrow$ denote branchings.

Throughout the paper we need the global assumption that a functor F has a lifting $\overline{F}$ on $\mathcal{K}\ell(T)$, or equivalently, that there is a distributive law $\lambda : FT \Rightarrow TF$. Now we present some sufficient conditions for existence of λ. In most examples one of these conditions holds.

First, take $T = \mathcal{P}$, in which case we have $\mathcal{K}\ell(\mathcal{P}) \cong \mathbf{Rel}$, the category of sets and binary relations. We can provide the following condition that uses relation liftings.

Lemma 2.2 (From [24]). *Let $F \colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor that preserves weak pullbacks. Then there exists a distributive law $\lambda \colon F\mathcal{P} \Rightarrow \mathcal{P}F$ given by*

$$\lambda_X(u) = \big\{\, v \in FX \mid (v, u) \in \mathrm{Rel}_F(\in_X) \,\big\} \ ,$$

where $u \in F\mathcal{P}X$ and $\mathrm{Rel}_F(\in_X) \subseteq FX \times F\mathcal{P}X$ is the F-relation lifting of the membership relation $\in_X$. $\qquad\square$

In fact, the functor $\overline{F} : \mathbf{Rel} \to \mathbf{Rel}$ induced by this distributive law carries an arrow $R : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$—which is a binary relation between X and Y—to its F-relation lifting $\mathrm{Rel}_F(R)$. That is,

$$\overline{F}R = \mathrm{Rel}_F(R) \quad : \ FX \longrightarrow FY \tag{2.4}$$

in $\mathcal{K}\ell(\mathcal{P}) \cong \mathbf{Rel}$.

More generally, when a monad T is *commutative* and a functor F is *shapely*, we can provide a canonical distributive law. The class of such monads and functors is wide and all the examples in this paper are contained.

- A *commutative* monad [34] is intuitively a monad whose corresponding algebraic theory has only commutative operators. We exploit the fact that a commutative monad is equipped with an arrow called *double strength*

$$\mathsf{dst}_{X,Y} : TX \times TY \longrightarrow T(X \times Y)$$

  for any sets X and Y; the double strength must be compatible with the monad structure of T in an obvious way.

  Our three examples of monads are all commutative, with the following double strengths.

$$\begin{array}{rcl}
\mathsf{dst}^{\mathcal{L}}_{X,Y}(u, v) & = & \left\{ \begin{array}{ll} (u, v) & \text{if } u \in X \text{ and } v \in Y, \\ \bot & \text{if } u = \bot \text{ or } v = \bot, \end{array} \right. \\
\mathsf{dst}^{\mathcal{P}}_{X,Y}(u, v) & = & u \times v \ , \\
\mathsf{dst}^{\mathcal{D}}_{X,Y}(u, v) & = & \lambda(x, y).\ u(x) \cdot v(y) \ .
\end{array} \tag{2.5}$$

- The family of *shapely functors* [27][6] on $\mathbf{Sets}$ is defined inductively by the following BNF notation:

$$F ::= \mathrm{id} \mid \Sigma \mid F_1 \times F_2 \mid \coprod_{i \in I} F_i \ ,$$

  where Σ denotes the constant functor into an arbitrary set Σ. Notice that taking infinite product is not allowed, nor exponentiation to the power of an infinite set. This is in order to ensure that we find an initial F-algebra as a suitable ω-colimit— see Proposition A.1.

The construction of a distributive law is done inductively on the construction of shapely F.

---

[6]Shapely functors here are called *polynomial* functors by some authors, although other authors allow infinite powers or the powerset construction.

Lemma 2.3. *Let $T\colon \mathbf{Sets} \to \mathbf{Sets}$ be a commutative monad, and $F\colon \mathbf{Sets} \to \mathbf{Sets}$ a shapely functor. Then there is a distributive law $\lambda\colon FT \Rightarrow TF$.*

Proof. By induction on the structure of the shapely functor F.

- If F is the identity functor, then the λ is the identity natural transformation $T \Rightarrow T$.
- If F is a constant functor, say $X \mapsto \Sigma$, then λ is the unit $\eta_\Sigma\colon \Sigma \to T\Sigma$ at $\Sigma \in \mathbf{Sets}$.
- If $F = F_1 \times F_2$ we use induction in the form of distributive laws $\lambda^{F_i}\colon F_iT \Rightarrow TF_i$ for $i \in \{1, 2\}$ to form the composite:

$$F_1TX \times F_2TX \xrightarrow{\lambda^{F_1} \times \lambda^{F_2}} TF_1X \times TF_2X \xrightarrow{\mathsf{dst}} T(F_1X \times F_2X) \ .$$

- If F is a coproduct $\coprod_{i \in I} F_i$ then we use laws $\lambda^{F_i}\colon F_iT \Rightarrow TF_i$ for $i \in I$ in:

$$\coprod_{i \in I} F_i(TX) \xrightarrow{[T(\kappa_i) \circ \lambda^{F_i}]_{i \in I}} T(\coprod_{i \in I} F_iX) \ .$$

It is straightforward to check that such λ is natural and compatible with the monad structure. $\qquad\square$

We have provided some *sufficient* conditions for a distributive law to exist, that is, for a functor F to be lifted to $\mathcal{K}\ell(T)$. This does not mean the results in the sequel hold exclusively for commutative monads and shapely functors.

2.3. **Order-enriched structures of Kleisli categories.** The notion of branching naturally involves a partial order: one branching is bigger than another if the former offers "more possibilities" than the latter. Formally, this order appears as the **Cppo**-*enriched structure* of a Kleisli category. It plays an important role in the initial algebra-final coalgebra coincidence in Section 3.1.

A **Cppo**-*enriched category* $\mathbb{C}$ is a category where:

- Each homset $\mathbb{C}(X, Y)$ carries a partial order $\sqsubseteq$ as in

$$X \underset{f}{\overset{g}{\rightrightarrows}} Y$$

  which makes $\mathbb{C}(X, Y)$ an ω-cpo with a bottom. This means:
  - for an increasing ω-chain of arrows from X to Y,

  $$f_0 \sqsubseteq f_1 \sqsubseteq \dots \quad : \quad X \longrightarrow Y \ ,$$

  there exists its join $\bigsqcup_{n<\omega} f_n : X \to Y$;
  - for any X and Y there exists a bottom arrow $\perp_{X,Y} : X \to Y$ which is the minimum in $\mathbb{C}(X, Y)$.
- Moreover, composition of arrows is continuous as a function $\mathbb{C}(X, Y) \times \mathbb{C}(Y, Z) \to \mathbb{C}(X, Z)$. This means that the following joins are preserved:[7]

$$g \circ \left(\bigsqcup_{n<\omega} f_n\right) = \bigsqcup_{n<\omega}(g \circ f_n) \quad \text{and} \quad \left(\bigsqcup_{n<\omega} f_n\right) \circ h = \bigsqcup_{n<\omega}(f_n \circ h) \ .$$

  Note that composition need not preserve bottoms (i.e. it is not necessarily *strict*).

---

[7]This component-wise preservation of joins is equivalent to the continuity of the composition function. See [1, Lemma 3.2.6].

This is in fact an instance of a more general notion of $\mathbb{V}$-*enriched categories* where $\mathbb{V}$ is the category **Cppo** of pointed (i.e. with $\bot$) cpo's and continuous (but not necessarily strict) functions. See [38, 29, 7] for more details on enriched category theory, and [1] on cpo's and domain theory.

Lemma 2.4. *For our three examples $\mathcal{L}$, $\mathcal{P}$ and $\mathcal{D}$ of a monad T, the Kleisli category $\mathcal{K}\ell(T)$ is **Cppo**-enriched. Moreover, composition of arrows is left-strict: $\bot \circ f = \bot$.*

The left-strictness of composition will be necessary later.

Proof. Notice first that a set TY for $T \in \{\mathcal{L}, \mathcal{P}, \mathcal{D}\}$ carries a cpo structure with $\bot$. The set $\mathcal{L}Y = \{\bot\} + Y$ carries the flat order with a bottom:

$$y \quad y' \quad y'' \quad \cdots$$
$$\bot$$

embodying the idea that $\bot$ denotes non-termination or *deadlock*—in contrast to $\checkmark$ for successful termination. The set $\mathcal{P}Y$ carries an inclusion order; in $\mathcal{D}Y$ we define $d \sqsubseteq e$ if $d(y) \leq e(y)$ for each $y \in Y$. The bottom element in $\mathcal{D}Y$ is the zero distribution $[y \mapsto 0]$: this belongs to the set $\mathcal{D}Y$ because $\mathcal{D}$ is the *sub*-distribution monad.

The cpo structure of a homset $\mathcal{K}\ell(T)(X, Y)$ comes from that of TY in a pointwise manner:

$$X \overset{g}{\underset{f}{\rightrightarrows}} Y \qquad \text{if and only if} \qquad \forall x \in X.\ f(x) \sqsubseteq_{TY} g(x) \ .$$

It is laborious but straightforward to show that composition in $\mathcal{K}\ell(T)$ is continuous and left-strict. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\Box$

We are concerned with coalgebras $X \to \overline{F}X$ in the category $\mathcal{K}\ell(T)$, which we assume is **Cppo**-enriched. Hence it comes natural to require that functor $\overline{F}$ is somehow compatible with the **Cppo**-enriched structure of $\mathcal{K}\ell(T)$. The obvious choice is to require that $\overline{F}$ is a **Cppo**-*enriched functor* (see e.g. [7]), i.e. $\overline{F}$ is *locally continuous*. It means that for an increasing ω-chain $f_n : X \to Y$, we have

$$\overline{F}(\bigsqcup_{n<\omega} f_n) = \bigsqcup_{n<\omega} (\overline{F}f_n) \ .$$

This is indeed the assumption chosen in axiomatic domain theory. We will come back to this point later in Section 3.3. However, for our later purpose, we only need the weaker condition of *local monotonicity*: $f \sqsubseteq g$ implies $\overline{F}f \sqsubseteq \overline{F}g$.

For a monad $T = \{\mathcal{L}, \mathcal{P}, \mathcal{D}\}$ and a shapely functor F (recall Lemma 2.3), the lifted $\overline{F}$ is indeed locally continuous. We emphasize again that this does not mean our results in Section 3 hold exclusively for shapely functors.

Lemma 2.5. *Let F be a shapely functor and $T \in \{\mathcal{L}, \mathcal{P}, \mathcal{D}\}$. The lifting $\overline{F} : \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ induced by Lemma 2.3 is locally continuous.*

Proof. By induction on the construction of shapely functors.

- $F = \mathrm{id}$, the identity functor. Then $\overline{F} = \mathrm{id}$ which satisfies the condition.
- $F = \Sigma$, a constant functor. Then $\overline{F}$ maps every arrow to the identity map on Σ in $\mathcal{K}\ell(T)$. This is obviously locally continuous.

- $F = F_1 \times F_2$. First notice that, for $f : X \to Y$ in $\mathcal{K}\ell(T)$, we obtain $\overline{F}f$ as the following composite in **Sets**.

$$F_1 X \times F_2 X \xrightarrow{\overline{F_1}f \times \overline{F_2}f} TF_1 Y \times TF_2 Y$$

$$\overline{F}f \searrow \quad \downarrow \mathsf{dst}_{F_1 Y, F_2 Y}$$

$$T(F_1 Y \times F_2 Y)$$

Because the order in $\mathcal{K}\ell(T)(FX, FY)$ is pointwise, it suffices to show the following: $\mathsf{dst} : TX \times TY \to T(X \times Y)$ is a continuous map between cpo's. It is easy to check that this is indeed the case. See (2.5).

- $F = \coprod_{j \in J} F_j$. For $f : X \to Y$ in $\mathcal{K}\ell(T)$, we obtain the map $\overline{F}f$ as the composite $[T\kappa_j]_{j \in J} \circ \coprod_{j \in J} \mathcal{K}\ell(F_j)(f)$ in **Sets**. Since the order on the homset is pointwise, it suffices to show that each $T\kappa_j : TF_j Y \to T(\coprod_{j \in J} F_j Y)$ is continuous. This is easy. □

3. Final coalgebra in a Kleisli category

In this section we shall prove our main technical result: the initial F-algebra in **Sets** yields the final $\overline{F}$-coalgebra in $\mathcal{K}\ell(T)$. It happens in the following two steps: first, the initial algebra in **Sets** is lifted to the initial algebra in $\mathcal{K}\ell(T)$; second we have the initial algebra-final coalgebra coincidence in $\mathcal{K}\ell(T)$. For the latter we use the classical result [51] of limit-colimit coincidence. This is where the **Cppo**-enriched structure of $\mathcal{K}\ell(T)$ plays a role.

In the proof we use two standard constructions: initial/final sequences [2] and limit-colimit coincidence [51]. The reader who is not familiar with these constructions is invited to look at Appendices A.1 and A.2 where we briefly recall them.

The proof of our main theorem can be drastically simplified for the special case $T = \mathcal{P}$ in which case $\mathcal{K}\ell(\mathcal{P}) \cong \mathbf{Rel}$. This is because of the duality $\mathbf{Rel} \cong \mathbf{Rel}^{\mathrm{op}}$; see Section 3.2.

3.1. **The initial algebra in Sets is the final coalgebra in $\mathcal{K}\ell(T)$.** First, it is standard that an initial algebra in **Sets** is lifted to an initial algebra in $\mathcal{K}\ell(T)$. Such a phenomenon is studied for instance in [11, 44] in the context of combining datatypes (modeled by an initial algebra) and effectful computations (modeled by a Kleisli category). For this result we do not need an order structure.

Proposition 3.1. *Let T be a monad and F be a endofunctor, both on a category $\mathbb{C}$. Assume that we have a distributive law $FT \Rightarrow TF$—or equivalently, we have a lifting $\overline{F}$ on $\mathcal{K}\ell(T)$. If F has an initial algebra $\alpha : FA \xrightarrow{\cong} A$ in $\mathbb{C}$, then*

$$J\alpha = \eta_A \circ \alpha : \quad \overline{F}A \longrightarrow A \qquad \text{in } \mathcal{K}\ell(T)$$

is an initial $\overline{F}$-algebra. Here J is the canonical Kleisli left adjoint as in (2.2).

We will use an instance of this result for $\mathbb{C} = \mathbf{Sets}$.

Proof. It follows from [20, Theorem 2.14] that a distributive law lifts the canonical Kleisli adjunction to an adjunction between the categories $\mathbf{Alg}(F)$ and $\mathbf{Alg}(\overline{F})$ of algebras.

$$
\begin{array}{ccc}
\mathbf{Alg}(F) & \underset{\perp}{\overset{J'}{\rightleftarrows}} & \mathbf{Alg}(\overline{F}) \\
\downarrow & & \downarrow \\
\mathbb{C} & \underset{K}{\overset{J}{\underset{\perp}{\rightleftarrows}}} & \mathcal{K}\ell(T)
\end{array}
$$

The left adjoint J' preserves the initial object (see e.g. [42]). $\square$

Second, we use the initial algebra-final coalgebra coincidence in $\mathcal{K}\ell(T)$—which holds in a suitable order-enriched setting—to identify the final coalgebra in $\mathcal{K}\ell(T)$. This is our main theorem.

Theorem 3.2 (Main theorem). *Assume the following:*

(1) *A monad T on $\mathbf{Sets}$ is such that its Kleisli category $\mathcal{K}\ell(T)$ is $\mathbf{Cppo}$-enriched and composition in $\mathcal{K}\ell(T)$ is left-strict.*

(2) *For an endofunctor F on $\mathbf{Sets}$, we have a distributive law $\lambda : FT \Rightarrow TF$. Equivalently, F has a lifting $\overline{F}$ on $\mathcal{K}\ell(T)$. Moreover, the lifting $\overline{F}$ is locally monotone.*

(3) *The functor F preserves ω-colimits in $\mathbf{Sets}$, hence has an initial algebra via the initial sequence (see Proposition A.1).*

Then the initial F-algebra $\alpha : FA \overset{\cong}{\Rightarrow} A$ yields a final $\overline{F}$-coalgebra in $\mathcal{K}\ell(T)$ by

$$(J\alpha)^{-1} = J(\alpha^{-1}) = \eta_{FA} \circ \alpha^{-1} : \quad A \longrightarrow \overline{F}A \qquad in \ \mathcal{K}\ell(T) \ .$$

We first present the main line of the proof. Some details are provided in the form of subsequent lemmas. Note that the assumptions are satisfied by $T \in \{\mathcal{L}, \mathcal{P}, \mathcal{D}\}$ and shapely F; see Lemmas 2.4 and 2.3.

Proof. By the assumption (3) we obtain the initial algebra via the initial sequence in $\mathbf{Sets}$.

$$
\text{In } \mathbf{Sets} \qquad\qquad (3.1)
$$

Here $0 = \emptyset \in \mathbf{Sets}$ is initial and $\mathbf{i} : 0 \to X$ is the unique arrow from 0 to an arbitrary X. We apply the functor $J : \mathbf{Sets} \to \mathcal{K}\ell(T)$ to the whole diagram. Since J is a left adjoint it preserves colimits: hence the two cocones in the following diagram are both colimits again.

$$
\text{In } \mathcal{K}\ell(T) \qquad\qquad (3.2)
$$

The ω-chain in this diagram is in fact the initial sequence for the functor $\overline{F}$ (Lemma 3.3) because, for example, a left adjoint J preserves initial objects. Moreover the lower cone is

the image of the upper cone under $\overline{F}$; see the diagram (2.3). Hence the diagram (3.2) is equal to the following one. Recall that $\overline{F}X = FX$ on objects.

$$
\text{(3.3)}
$$

Thus Proposition A.1 yields that $J\alpha : \overline{F}A \overset{\cong}{\Rightarrow} A$ is an initial $\overline{F}$-algebra. This can be seen as a more concrete proof of Proposition 3.1.

Now we show the initial algebra-final coalgebra coincidence in $\mathcal{K}\ell(T)$. This is done by reversing all the arrows in (3.3) and transforming the diagram into the one of the final sequence and its limits.

We notice (Lemma 3.5) that each arrow $\overline{F}^n\,¡$ in the initial sequence is an embedding (Definition A.4). Hence the limit-colimit coincidence Theorem A.8 says that every arrow in the diagram is an embedding. Note that $J\alpha$ and $J\alpha^{-1}$, inverse to each other, form an embedding-projection pair.

By taking the corresponding projections—they are uniquely determined (Lemma A.5) and are denoted by $(\_)^P$—we obtain the next diagram. The limit-colimit coincidence Theorem A.8 says that the two resulting cones are both limits. It is also obvious that the whole diagram commutes.

$$
\text{(3.4)}
$$

The ω^{op}-chain here is indeed a final sequence: Lemma 3.4 shows—using the assumption (1) on left-strictness—that 0 is also final in $\mathcal{K}\ell(T)$, and according to Lemma 3.5 we have $\left(\overline{F}^n\,¡\right)^P = \overline{F}^n\,!$. As to the lower cone we have $\left(\overline{F}J\alpha_n\right)^P = \overline{F}\left((J\alpha_n)^P\right)$ by Lemma 3.6.

Hence the diagram (3.4) is equal to the following one, showing the final sequence for $\overline{F}$, its limit (the upper one) and that limit mapped by $\overline{F}$ (the lower one) which is again a limit.

$$
\text{(3.5)}
$$

Here $! : X \to 0$ is the unique arrow to the final object 0 in $\mathcal{K}\ell(T)$. By Proposition A.2 we conclude that $J\alpha^{-1}$ is a final $\overline{F}$-coalgebra. $\qquad\square$

In the remainder of this section the lemmas used in the above proof are presented. We rely on the same assumptions as in Theorem 3.2.

Lemma 3.3. *The ω-chain in the diagram (3.2) is indeed the initial sequence for $\overline{F}$. That is, we have for each $n < \omega$,*

$$JF^n\big(\,\text{¡}\,{}_{F0}^{\mathbf{Sets}}\big) = \overline{F}^n\big(\,\text{¡}\,{}_{F0}^{\mathcal{K}\ell(T)}\big) \;:\; JF^n0 \longrightarrow JF^{n+1}0 \quad in\ \mathcal{K}\ell(T),$$

where $\text{¡}\,{}_{F0}^{\mathbf{Sets}} : 0 \to F0$ *in* $\mathbf{Sets}$ *and* $\text{¡}\,{}_{F0}^{\mathcal{K}\ell(T)} : 0 \to F0$ *in* $\mathcal{K}\ell(T)$ *denote the unique maps.*

Proof. By induction on n. For $n = 0$ the two maps are equal due to the initiality of $J0 = 0$ in $\mathcal{K}\ell(T)$. For the step case we use the commutativity $JF = \overline{F}J$ of (2.3). $\square$

Lemma 3.4. *The empty set 0 is both an initial and a final object in $\mathcal{K}\ell(T)$.*

In particular, this implies that the object $T0$ is final in $\mathbf{Sets}$.

Proof. The functor $J : \mathbf{Sets} \to \mathcal{K}\ell(T)$ preserves initial objects since it is a left adjoint. Therefore $0 = J0$ is initial in $\mathcal{K}\ell(T)$. Finality follows essentially from the left-strictness assumption: for each set X there exists at least one arrow $X \to 0$ in $\mathcal{K}\ell(T)$, for example $\bot_{X,0}$. To show the uniqueness of such an arrow, take an arbitrary arrow $f : X \to 0$ in $\mathcal{K}\ell(T)$. Recalling that the bottom map $\bot_{0,0} : 0 \to 0$ is also the identity arrow in $\mathcal{K}\ell(T)$ because of initiality, we obtain

$$f \;=\; \mathrm{id} \circ f \;=\; \bot_{0,0} \circ f \;\overset{(*)}{=}\; \bot_{X,0}\;,$$

where the compositions are taken in $\mathcal{K}\ell(T)$ and the equality marked by $(*)$ holds by left-strictness of composition. $\square$

Lemma 3.5. *Each arrow $\overline{F}^n\text{¡}$ in the initial sequence for $\overline{F}$, as in the diagram (3.3), is an embedding. Its corresponding projection is given by*

$$\big(\overline{F}^n\text{¡}\big)^P = \overline{F}^n\,!\qquad in\qquad F^n0 \underset{\overline{F}^n\,\text{¡}}{\overset{\overline{F}^n\,!}{\rightleftarrows}} F^{n+1}0\;,$$

where $!$ denotes the unique arrow from $F0$ to the final object 0 in $\mathcal{K}\ell(T)$ (cf. Lemma 3.4).

Proof. We show that $(\overline{F}^n\text{¡}, \overline{F}^n\,!)$ is an embedding-projection pair for all $n < \omega$. We have $\overline{F}^n\,! \circ \overline{F}^n\text{¡} = \mathrm{id}$ because $! \circ \text{¡} = \mathrm{id}$. For the other half we have

$$\begin{aligned}
\overline{F}^n\text{¡} \circ \overline{F}^n\,! &= \overline{F}^n(\text{¡} \circ !) \\
&= \overline{F}^n(\bot_{0,F0} \circ !) && \text{initiality of } 0 \text{ in } \mathcal{K}\ell(T)\\
&= \overline{F}^n(\bot_{F0,F0}) && \text{composition is left-strict}\\
&\le \overline{F}^n(\mathrm{id}) = \mathrm{id} && \overline{F} \text{ is locally monotone.} \qquad \square
\end{aligned}$$

Lemma 3.6. *We have $\big(\overline{F}J\alpha_n\big)^P = \overline{F}\big((J\alpha_n)^P\big)$. Hence the lower cone in the diagram (3.4) is the image of the upper cone under $\overline{F}$.*

Proof. It is easy to check that $\big(\overline{F}J\alpha_n, \overline{F}\big((J\alpha_n)^P\big)\big)$ indeed form an embedding-projection pair. Therein we use the monotonicity of $\overline{F}$'s action on arrows. $\square$

3.2. **Simpler proof in $\mathcal{K}\ell(\mathcal{P}) \cong \mathbf{Rel}$.** When $T = \mathcal{P}$ we have the self-duality

$$Op \quad : \quad \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} \xrightarrow{\cong} \mathcal{K}\ell(\mathcal{P}) \ .$$

This is because of the following bijective correspondence between functions

$$\frac{X \xrightarrow{\quad f \quad} \mathcal{P}Y \ \text{ in } \mathbf{Sets}}{Y \xrightarrow{\quad f^{\vee} \quad} \mathcal{P}X \ \text{ in } \mathbf{Sets}}$$

given by $f^{\vee}(y) = \{x \in X \mid y \in f(x)\}$. Recalling $\mathcal{K}\ell(\mathcal{P}) \cong \mathbf{Rel}$, this mapping $f \mapsto f^{\vee}$ corresponds to taking the opposite relation.

Due to this "global" duality $\mathcal{K}\ell(\mathcal{P}) \cong \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}}$, the proof of Theorem 3.2 is drastically simplified for $T = \mathcal{P}$. It essentially relies on the lifted self duality $\mathbf{Alg}(\overline{F}) \cong \mathbf{Alg}(\overline{F}^{\mathrm{op}})$, where the latter is isomorphic to $(\mathbf{Coalg}(\overline{F}))^{\mathrm{op}}$. We do not need here an order structure of $\mathcal{K}\ell(\mathcal{P})$ nor local monotonicity of $\overline{F}$.

Theorem 3.7. *Let $F : \mathbf{Sets} \to \mathbf{Sets}$ be a functor which preserves weak pullbacks, and $\overline{F} : \mathcal{K}\ell(\mathcal{P}) \to \mathcal{K}\ell(\mathcal{P})$ be its lifting induced by relation lifting (Lemma 2.2). Then the initial F-algebra in $\mathbf{Sets}$ yields the final $\overline{F}$-coalgebra in $\mathcal{K}\ell(\mathcal{P})$.*

Proof. We have the following situation because of the self-duality of $\mathcal{K}\ell(\mathcal{P})$.

$$\mathbf{Sets} \underset{K}{\overset{J}{\rightleftarrows}} \mathcal{K}\ell(\mathcal{P}) \xrightarrow[\cong]{Op^{\mathrm{op}}} \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}}$$

The adjunction $J \dashv K$ and the isomorphism $Op : \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} \xrightarrow{\cong} \mathcal{K}\ell(\mathcal{P})$ lift to those between the categories of algebras.

$$
\begin{array}{ccccccc}
\mathbf{Alg}(F) & \underset{\perp}{\overset{J'}{\rightleftarrows}} & \mathbf{Alg}(\overline{F}) & \xrightarrow[\cong]{(Op')^{\mathrm{op}}} & \mathbf{Alg}(\overline{F}^{\mathrm{op}}) & \xrightarrow{\cong} & (\mathbf{Coalg}(\overline{F}))^{\mathrm{op}} \\
\downarrow & & \downarrow & & \downarrow & & \\
\mathbf{Sets} & \underset{K}{\overset{J}{\rightleftarrows}} & \mathcal{K}\ell(\mathcal{P}) & \xrightarrow[\cong]{Op^{\mathrm{op}}} & \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} & &
\end{array}
$$

Indeed, $J \dashv K$ lifts due to Proposition 3.1; the lifted isomorphism $Op' : \mathbf{Alg}(\overline{F}) \xrightarrow{\cong} \mathbf{Alg}(\overline{F}^{\mathrm{op}})$ is because of the following commutativity:

$$
\begin{array}{ccc}
\mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} & \xrightarrow{\ Op\ } & \mathcal{K}\ell(\mathcal{P}) \\
{\scriptstyle \overline{F}^{\mathrm{op}}}\downarrow & & \downarrow{\scriptstyle \overline{F}} \\
\mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} & \xrightarrow[\ Op\ ]{} & \mathcal{K}\ell(\mathcal{P})
\end{array}
\qquad (3.6)
$$

which is because: $\overline{F}R = \mathrm{Rel}_F(R)$ (see (2.4)); and taking relation liftings is compatible with opposite relations (i.e. $\mathrm{Rel}_F(R^{\mathrm{op}}) = (\mathrm{Rel}_F R)^{\mathrm{op}}$, see [22]). Moreover the category $\mathbf{Alg}(\overline{F}^{\mathrm{op}})$ is obviously isomorphic to $(\mathbf{Coalg}(\overline{F}))^{\mathrm{op}}$.

Therefore the initial object in $\mathbf{Alg}(F)$ is carried to that in $(\mathbf{Coalg}(\overline{F}))^{\mathrm{op}}$, hence the final object in $\mathbf{Coalg}(\overline{F})$. $\qquad\square$

For monads such as $T = \mathcal{D}$ a "global" self-duality $\mathcal{K}\ell(T) \cong \mathcal{K}\ell(T)^{\mathrm{op}}$ is not available. Instead, in the proof of Theorem 3.2, we exploit the "partial" duality which holds between the colimit/limit of the initial/final sequence.

3.3. **Related work: axiomatic domain theory.** The initial algebra-final coalgebra coincidence is heavily exploited in the field of *axiomatic domain theory*, e.g. in [13, 12, 9, 50]. There, categories which have coinciding initial algebra and final coalgebra for each endofunctor are called *algebraically compact categories*. They draw special attention as suitable "categories of domains" for denotational semantics of datatype construction. The relevance comes as follows.

Let $\mathbb{C}$ be a "category of domains". We think of an object of the category $\mathbb{C}$ as a type. A "recursive" datatype constructor—a prototypical example is $(X, Y) \mapsto Y^X$—is presented as a bifunctor $G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$. Note the presence of both covariance and contravariance. We expect that such a category $\mathbb{C}$ has a canonical fixed point $\mathrm{Fix}\, G$ such that

$$G(\mathrm{Fix}\, G, \mathrm{Fix}\, G) \overset{\cong}{\Rightarrow} \mathrm{Fix}\, G \ ,$$

which models the recursive type determined by the datatype constructor G. Freyd [13] showed that if $\mathbb{C}$ is algebraically compact, then we can construct such a fixed point as a suitable initial algebra; moreover this fixed point is shown by Fiore [9] to be a canonical one in a suitable sense. The rough idea here is that the covariant part of G is taken care of by an initial algebra; the contravariant part is by a final coalgebra; the initial algebra-final coalgebra coincidence yields a fixed point of overall G.

Typical examples of algebraically compact categories are enriched over **Cppo** or one of its variants. This conforms the traditional use of the word "domain" for certain cpo's (e.g. in [1]).

Although we utilize the initial algebra-final coalgebra coincidence result in $\mathcal{K}\ell(T)$, we are not so much interested in algebraic compactness of $\mathcal{K}\ell(T)$. This is because our motivation is different from that of axiomatic domain theory. In studying trace semantics for coalgebras, we need not deal with *every* endofunctor on $\mathcal{K}\ell(T)$, but only such an endofunctor $\overline{F}$ which is a lifting of $F : \mathbf{Sets} \to \mathbf{Sets}$.

In a different context of functional programming, the work [44] also studies initial algebras and final coalgebras in a Kleisli category. The motivation there is to combine *data types* and *effects*. More specifically, an initial algebra and a final coalgebra support the *fold* and the *unfold* operators, respectively, used in recursive programs over datatypes. A computational effect is presented as a monad, and its Kleisli category is the category of effectful computations.

The difference between [44] and the current work is as follows. In [44], the original category of pure functions is already algebraically compact; the paper studies the conditions for the algebraic compactness to be carried over to Kleisli categories. In contrast, in the current work, it is a monad—with a suitable order structure, embodying the essence of "branching"—which yields the initial algebra-final coalgebra coincidence on a Kleisli category; the coincidence is not present in the original category **Sets**.

3.3.1. *Local continuity vs. local monotonicity.* In axiomatic domain theory, **Cppo**-enriched categories are said to be algebraically compact because, "in a 2-category setting" [12], every endofunctor has an initial algebra and a final coalgebra. Concretely this means: "every locally continuous functor".

In this spirit, we could have made a stronger assumption of $\overline{F}$'s local continuity in Theorem 3.2 instead of local monotonicity. If we do so, in fact, the proof of Theorem 3.2 becomes much simpler: the following proposition (Lemma in [12, p.98]) immediately yields the initial algebra-final coalgebra coincidence for a locally continuous $\overline{F}$.

Proposition 3.8 ([12]). *Let $\mathbb{D}$ be a* **Cppo***-enriched category whose composition is left-strict, and $G : \mathbb{D} \to \mathbb{D}$ be a locally continuous endofunctor. An initial algebra $\beta : GB \stackrel{\cong}{\Rightarrow} B$, if it exists, yields a final coalgebra $\beta^{-1} : B \stackrel{\cong}{\Rightarrow} GB$.*

Proof. Given a coalgebra $d : Y \to GY$, the function
$$\Phi \;:\; \mathbb{D}(Y, B) \longrightarrow \mathbb{D}(Y, B) \;, \qquad f \longmapsto \beta \circ Gf \circ d$$
is continuous due to the local continuity of G. Hence it has the least fixed point $\bigsqcup_{n < \omega} \Phi^n(\bot)$; this proves existence of a morphism from d to β^{-1}.

$$
\begin{array}{ccc}
GY & \longrightarrow & GB \\
{\scriptstyle d}\uparrow & & {\scriptstyle \cong}\uparrow{\scriptstyle \beta^{-1}} \\
Y & \longrightarrow & B
\end{array}
$$

Now we shall show its uniqueness. Assume that $g : Y \to B$ is a morphism of coalgebras as above, that is, $\Phi(g) = g$. Similarly to Φ, we define a function $\Psi : \mathbb{D}(B, B) \to \mathbb{D}(B, B)$ as the one which carries $h : B \to B$ to $\beta \circ Gh \circ \beta^{-1}$. We have

$$
\begin{aligned}
\bigsqcup_n \Phi^n(\bot) &= \bigsqcup_n \Phi^n(B \xrightarrow{g} B \xrightarrow{\bot} Y) & &\text{composition is left-strict, so } \bot \circ g = \bot \\
&= \bigsqcup_n \big(\Psi^n(\bot) \circ \Phi^n(g)\big) & &\Phi^n(\bot \circ g) = \Psi^n(\bot) \circ \Phi^n(g), \text{ by induction} \\
&= \big(\bigsqcup_n \Psi^n(\bot)\big) \circ \big(\bigsqcup_n \Phi^n(g)\big) & &\text{composition is continuous} \\
&= \bigsqcup_n \Phi^n(g) & &\bigsqcup_n \Psi^n(\bot) = \mathrm{id},\ (\ast) \\
&= g & &\Phi(g) = g \text{ by assumption.}
\end{aligned}
$$

Here $(\ast)$ holds because $\bigsqcup_n \Psi^n(\bot)$, being a fixed point for Ψ, is the unique morphism of algebras from β to β. This shows that the morphism g must be the least fixed point of Φ. $\square$

Nevertheless, for our main Theorem 3.2 we can do with only $\overline{F}$'s local monotonicity, by taking a closer look at the initial/final sequences.

4. Finite trace semantics via coinduction

In this section we shall further illustrate the observation that the principle of coinduction, when employed in $\mathcal{K}\ell(T)$, captures trace semantics of state-based systems. As we have shown in the previous section, an initial algebra in **Sets** constitutes the semantic domain, i.e. is a final coalgebra in $\mathcal{K}\ell(T)$. Viewing an initial algebra as the set of well-founded terms (such as finite words or finite-depth parse trees), this fact means that the "trace semantics" induced by coinduction is inevitably *finite*, in the sense that it captures only finite behavior. Here we will elaborate on this finiteness issue as well.

4.1. **Trace semantics by coinduction.** As we have seen in Section 2.2 various types of state-based systems allow their presentation as coalgebras $X \to \overline{F}X$ in a Kleisli category $\mathcal{K}\ell(T)$. For example,

- LTSs with explicit termination, with $T = \mathcal{P}$ and $F = 1 + \Sigma \times \_$;
- probabilistic LTSs (also called *generative probabilistic transition systems* in [58, 52]) with explicit termination, with $T = \mathcal{D}$ and $F = 1 + \Sigma \times \_$;
- context-free grammars with $T = \mathcal{P}$ and $F = (\Sigma + \_)^*$.

The main observation underlying this work is the following. If we instantiate the parameters

$$T \text{ for branching type} \qquad \text{and} \qquad F \text{ for transition type}$$

in the coinduction diagram

$$
\begin{array}{ccc}
\overline{F}X & \xdashrightarrow{\overline{F}(\mathsf{tr}_c)} & \overline{F}A \\
c\uparrow & & \cong\uparrow J\alpha^{-1} \qquad \text{in } \mathcal{K}\ell(T) \\
X & \xdashrightarrow[\mathsf{tr}_c]{} & A
\end{array}
\tag{4.1}
$$

with one of the above choices, then the commutativity of the diagram is equivalent to the corresponding (conventional) definition of trace semantics in Section 1.1. Therefore we claim that the diagram (4.1) is the mathematical principle underlying various "trace semantics", no matter if it is "trace set" (non-deterministic) or "trace distribution" (probabilistic).

Corollary 4.1 (Trace semantics for coalgebras)**.** *Assume that T and F are such as in Theorem 3.2, and $\alpha : FA \overset{\cong}{\Rightarrow} A$ is an initial F-algebra in* **Sets***. Given a coalgebra $c : X \to TFX$ in* **Sets***, we can assign a function*

$$\mathsf{tr}_c \; : \; X \longrightarrow TA \qquad \text{in } \mathbf{Sets}$$

which is, as an arrow $X \to A$ in $\mathcal{K}\ell(T)$, the unique one making the diagram (4.1) commute. We shall call this function tr_c the (finite) trace semantics for the coalgebra c. $\qquad\square$

Example 4.2. As further illustration we give details for the choice of parameters $T = \mathcal{P}$ and $F = 1 + \Sigma \times \_$. This is the suitable choice to deal with the first system in (1.1).

Now the coinduction diagram looks as follows. Recall that an initial F-algebra is carried by the set Σ^* of finite words.

$$
\begin{array}{ccc}
1 + \Sigma \times X & \xdashrightarrow{1 + \Sigma \times \mathsf{tr}_c} & 1 + \Sigma \times \Sigma^* \\
c\uparrow & & \cong\uparrow J([\mathsf{nil},\mathsf{cons}])^{-1} \qquad \text{in } \mathcal{K}\ell(\mathcal{P}) \\
X & \xdashrightarrow[\mathsf{tr}_c]{} & \Sigma^*
\end{array}
\tag{4.2}
$$

It assigns, to a system c, a function $\mathsf{tr}_c : X \to \mathcal{P}(\Sigma^*)$ which carries a state $x \in X$ to the set of finite words on Σ which can possibly arise as an execution "trace" of c starting from x. The commutativity states equality of two arrows $X \rightrightarrows 1 + \Sigma \times \Sigma^*$ in $\mathcal{K}\ell(\mathcal{P})$, that is, functions $X \rightrightarrows \mathcal{P}(1 + \Sigma \times \Sigma^*)$. Let us denote these functions by

$$u = (1 + \Sigma \times \mathsf{tr}_c) \circ c \quad \text{(up, then right)}, \qquad v = J([\mathsf{nil}, \mathsf{cons}])^{-1} \circ \mathsf{tr}_c \quad \text{(right, then up)}.$$

For each $x \in X$, the following conditions—derived straightforwardly by definition composition of $\mathcal{K}\ell(\mathcal{P})$, lifting of the functor $1 + \Sigma \times \_$, etc.—specify u and v's value at x, as a

subset of $1 + \Sigma \times \Sigma^*$.

$$
\begin{aligned}
\checkmark \in u(x) &\iff \checkmark \in c(x) \\
(a, \sigma) \in u(x) &\iff \exists x' \in X. \left((a, x') \in c(x) \wedge \sigma \in \mathsf{tr}_c(x') \right) \\
\checkmark \in v(x) &\iff \langle \rangle \in \mathsf{tr}_c(x) \\
(a, \sigma) \in v(x) &\iff a \cdot \sigma \in \mathsf{tr}_c(x)
\end{aligned}
$$

Commutativity of (4.2) amounts to $u = v$; this gives the condition (1.3).

From a different point of view we can also express that as follows: finality of the coalgebra $\Sigma^* \stackrel{\cong}{\Rightarrow} 1 + \Sigma \times \Sigma^*$ in (4.2) ensures that the conventional recursive definition (1.3) uniquely determines a function $\mathsf{tr}_c : X \to \mathcal{P}(\Sigma^*)$. Hence tr_c is *well-defined*.

Example 4.3. Let us take $T = \mathcal{L}$ (the lift monad) and $F = 1 + \Sigma \times \_$. In this case a coalgebra $X \stackrel{c}{\to} \mathcal{L}(1 + \Sigma \times X)$ in **Sets** is a system which can

- get into a deadlock ($c(x) = \bot$ where $\mathcal{L} = \{\bot\} + \_$),
- successfully terminate ($c(x) = \checkmark$ where $F = \{\checkmark\} + \Sigma \times \_$), or
- output a letter from Σ and move to the next state ($c(x) = (a, x')$).

By examining trace semantics for such systems, we shall formally put the difference between the computational meanings of the two elements, $\bot$ and $\checkmark$.

The coinduction diagram (4.1) instantiates to the same diagram as (4.2), but now in the category $\mathcal{K}\ell(\mathcal{L})$. Easy calculation shows that its commutativity amounts to the following condition. The function

$$
X \xrightarrow{\quad \mathsf{tr}_c \quad} \mathcal{L}(\Sigma^*) = \{\bot\} + \Sigma^* \qquad \text{in } \mathbf{Sets}
$$

satisfies, for each $x \in X$,

$$
\begin{aligned}
\mathsf{tr}_c(x) = \langle \rangle &\iff c(x) = \checkmark \ , \\
\mathsf{tr}_c(x) = a \cdot \sigma &\iff \exists x' \in X. \left(c(x) = (a, x') \wedge \mathsf{tr}_c(x') = \sigma \right) \ , \\
\mathsf{tr}_c(x) = \bot &\iff c(x) = \bot \quad \text{or} \quad \exists x' \in X. \left(c(x) = (a, x') \wedge \mathsf{tr}_c(x') = \bot \right) \ .
\end{aligned} \tag{4.3}
$$

Here $\sigma \in \Sigma^*$ is a word in Σ.

For the systems under consideration, we can think of three different kinds of possible executions.

- An execution eventually hitting $\checkmark$, that is, $x \stackrel{a_1}{\to} \cdots \stackrel{a_n}{\to} x_n \to \checkmark$. By the condition (4.3) it yields a word $\mathsf{tr}_c(x) = a_1 \ldots a_n$ as its trace.
- An execution eventually hitting $\bot$, that is, $x \stackrel{a_1}{\to} \cdots \stackrel{a_n}{\to} x_n \to \bot$. By the third line of (4.3) we see that $\mathsf{tr}_c(x_n) = \bot$; moreover $\mathsf{tr}_c(x_{n-1}) = \cdots = \mathsf{tr}_c(x) = \bot$. It properly reflects our intuition that a state x that eventually goes into deadlock does not yield a finite (or terminating) trace.
- An execution not hitting $\checkmark$ nor $\bot$, that is, $x \stackrel{a_1}{\to} x_1 \stackrel{a_2}{\to} \cdots$. In this case, the only possible solution of the "recursive equation" (4.3) is $\mathsf{tr}_c(x) = \mathsf{tr}_c(x_1) = \cdots = \bot$. The intuition here is: a state leading to *livelock* does not yield a finite trace.

4.2. **Infinite traces.** The trace semantics obtained via coinduction (Corollary 4.1) assigns, to each state $x \in X$, "a set of" (if $T = \mathcal{P}$) or "a distribution over" (if $T = \mathcal{D}$) elements of the initial algebra A. Elements of A are thought of as possible linear behavior of the system determined by the transition type (i.e. the functor F).

Now the intuition is that an initial F-algebra A consists of the well-founded (or finite-depth) terms and a final F-coalgebra Z consists of the possibly non-well-founded (or infinite-depth) terms. For example,

- for $F = 1 + \Sigma \times \_$, $A = \Sigma^*$ consists of all the finite words, and $Z = \Sigma^\infty = \Sigma^* + \Sigma^\omega$ is augmented with *streams*, i.e. infinite words;
- for $F = (\Sigma + \_)^*$, A is the set of finite-depth *skeletal parse trees* (see [16]), and Z additionally contains infinite-depth ones;
- for $F = \Sigma \times \_$ which models LTSs *without* explicit termination, $A = 0$ and $Z = \Sigma^\omega$.

Therefore our trace semantics $X \to TA$ only takes account of finite, well-founded linear-time behavior but not infinite ones. This is why the trace set (1.2) does not contain ab^ω; and also why we have been talking about LTSs *with* explicit termination—otherwise the finite trace semantics is always empty.

Designing a coalgebraic framework to capture possibly infinite trace semantics is the main aim of [24]. The work is done exclusively in a non-deterministic setting and the main result reads as follows.

Theorem 4.4 (Possibly infinite trace semantics for coalgebras, [24])**.** *Let F be a shapely functor on* **Sets***, and $\zeta : Z \overset{\cong}{\Rightarrow} FZ$ be a final coalgebra in* **Sets***. The coalgebra*

$$J\zeta \;:\; Z \longrightarrow \overline{F}Z \qquad in \; \mathcal{K}\ell(\mathcal{P})$$

is weakly final: that is, given a coalgebra $c : X \to \overline{F}X$, there is a morphism from c to $J\zeta$ but the morphism is not necessarily unique.

$$
\begin{array}{ccc}
\overline{F}X & \overset{\overline{F}(\mathsf{tr}_c^\infty)}{\rightsquigarrow} & \overline{F}Z \\
c\uparrow & & \cong\uparrow J\zeta \qquad in \; \mathcal{K}\ell(\mathcal{P}) \\
X & \underset{\mathsf{tr}_c^\infty}{\rightsquigarrow} & Z
\end{array}
\qquad (4.4)
$$

Still there is a canonical choice tr_c^∞ among such morphisms, namely the one which is maximal with respect to the inclusion order. We shall call the function $\mathsf{tr}_c^\infty : X \to \mathcal{P}Z$ the **possibly-infinite trace semantics** *for c.* □

Note here that, when we take $F = 1 + \Sigma \times \_$ and $T = \mathcal{P}$ (the choice for LTSs with termination), commutativity of (4.4) boils down to exactly the same conditions as (1.3):

$$\langle\rangle \in \mathsf{tr}_c^\infty(x) \iff x \to \checkmark, \qquad a \cdot \sigma \in \mathsf{tr}_c^\infty(x) \iff \exists y. \, (x \overset{a}{\to} y \wedge \sigma \in \mathsf{tr}_c^\infty(y) \,).$$
$$(4.5)$$

Weak finality of $\Sigma^\infty \overset{\cong}{\Rightarrow} 1 + \Sigma \times \Sigma^\infty$ (corresponding to $Z \overset{\cong}{\Rightarrow} \overline{F}Z$ in (4.4)) means the following. The recursive definition (4.5)—although it looks valid at the first sight—does *not* uniquely determine the infinite trace map $\mathsf{tr}_c^\infty : X \to \mathcal{P}(\Sigma^\infty)$. Instead, the map tr_c^∞ is the maximal one among those which satisfy (4.5).

As an example take the first system in (1.1). We expect its possibly-infinite trace map $X \to \mathcal{P}(\Sigma^\infty)$ to be such that $x \mapsto ab^* + ab^\omega$ and $y \mapsto b^* + b^\omega$. Indeed this satisfies (4.5) and is moreover the maximal. However, the function $x \mapsto ab^*$ and $y \mapsto b^*$—this is actually the finite trace $X \to \mathcal{P}(\Sigma^*)$ embedded along $\Sigma^* \hookrightarrow \Sigma^\infty$—also satisfies (4.5). In fact, [16, Section 5] shows a general fact that such an embedding of the finite trace map is the minimal one among those morphisms which make the diagram (4.4) commute.

The coalgebraic characterization (Theorem 4.4) of possibly-infinite trace semantics is not yet fully developed: for example we do not know whether the result holds in a probabilistic setting.

5. Trace semantics as testing equivalence

In this section we will observe that, in a non-deterministic setting, the coalgebraic finite trace semantics (i.e. coinduction in $\mathcal{K}\ell(\mathcal{P})$) gives rise to a canonical *testing situation* in which a test is an element of the initial F-algebra A in **Sets**. Here F specifies the transition type, just as before. The notion of testing situations (Definition 5.1) and its variants have attracted many authors' attention in the context of coalgebraic modal logic; our aim here is to demonstrate genericity and pervasiveness of the notion of testing situations by presenting an example which is not much like modal logic (that is, propositional logic plus modality).

In Section 5.1 we introduce the notion of testing situations and investigate some of their general properties. Our main concern there is the comparison between two process equivalences, namely *testing equivalence* and *equivalence modulo final coalgebra semantics*. We present the equivalences categorically as suitable kernel pairs; this makes the arguments simple and clean. In Section 5.2 we present the canonical testing situation for trace semantics. Moreover we show that it is *expressive*: the testing captures final coalgebra semantics, which is now trace semantics.

5.1. **Testing situations.** Recent studies [5, 6, 37, 45, 33] on coalgebra and modal logic have identified (variants of) the following categorical situation as the essential underlying structure. Following [45], we prefer using a more general term "testing": it subsumes "modal logic" in the following sense. We learn properties of a system through pass or failure of *tests*; modal logic constitutes a special case where tests are modal formulas.

Definition 5.1. A *testing situation* is the following situation of a contravariant adjunction $S^{\mathrm{op}} \dashv P$ and two endofunctors F, M

$$F^{\mathrm{op}} \left(\; \mathbb{C}^{\mathrm{op}} \underset{S^{\mathrm{op}}}{\overset{P}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathbb{A} \; \right) M \tag{5.1}$$

plus a "denotation" natural transformation $\delta : MP \Rightarrow PF^{\mathrm{op}} : \mathbb{C}^{\mathrm{op}} \to \mathbb{A}$, which consists of arrows $MPX \xrightarrow{\delta_X} PFX$ in $\mathbb{A}$.

Note that the denotation δ is a parameter: the same "syntax for tests" $M : \mathbb{A} \to \mathbb{A}$ can have different interpretations with different δ.

The requirements in Definition 5.1 are the same as in [45, 33]. They are what we need to compare two process semantics, namely *testing equivalence*—which arises naturally from the concept of testing—and final coalgebra semantics.[8] We shall explain each ingredient's role, using the well-established terminology of modal logic.

- The endofunctor $F : \mathbb{C} \to \mathbb{C}$ makes **Coalg**(F) the category of "systems", or "Kripke models" in modal logic.
- The category $\mathbb{A}$—typical examples being **Bool** of Boolean algebras or **Heyt** of Heyting algebras—is that of "propositional logic". The functor M specifies "modality": modal operators and axioms. Then **Alg**(M) is the category of "modal algebras";

---

[8]In fact we can be even more liberal: existence of a denotation δ can be replaced by existence of a lifting $\hat{P} : \mathbf{Coalg}(F)^{\mathrm{op}} \to \mathbf{Alg}(M)$ of P. The results in this section nevertheless hold in that case. The latter condition (there is a lifting $\hat{P}$) is strictly weaker than the former (there is a natural transformation δ): obviously δ induces $\hat{P}$ but not the other way round. Let $\mathbb{C} = \omega^{\mathrm{op}}, \mathbb{A} = \omega, P = \mathrm{id}, F = (1 + \_)^{\mathrm{op}}$ and $M = 2 + \_$. Then both $\mathbf{Coalg}(F)$ and $\mathbf{Alg}(M)$ are the empty category hence P has the trivial lifting. However there is no natural transformation $MPX \to PF^{\mathrm{op}}X$.

the initial M-algebra $ML \overset{\cong}{\Rightarrow} L$ is a "modal logic" consisting of modal formulas, modulo logical equivalence.

- The denotation δ specifies how the modality M is interpreted via transitions of type F. This allows to give "Kripke semantics" for the modal logic: given a coalgebra (or a "Kripke model") $c : X \to FX$, interpretation $\llbracket \_ \rrbracket_c$ of modal formulas therein is given by the following induction.

$$
\begin{array}{ccc}
ML & \dashrightarrow & MPX \\
& & \downarrow \delta_X \\
\text{initial} \downarrow \cong & & PFX \\
& & \downarrow Pc \\
L & \dashrightarrow & PX \\
& \llbracket \_ \rrbracket_c &
\end{array}
\tag{5.2}
$$

- Why a right adjoint S of P^{op}? It allows us, via transposition, to assign a modal "theory" to each state of a Kripke model.

$$
\cfrac{L \xrightarrow{\;\llbracket \_ \rrbracket_c\;} PX \quad \text{in } \mathbb{A}}{X \xrightarrow{\;\mathsf{th}_c\;} SL \quad \text{in } \mathbb{C}} \; (S^{\mathrm{op}} \dashv P)
\tag{5.3}
$$

The theory $\mathsf{th}_c(x)$ associated with a state x contains precisely the modal formulas that hold at x.

Following the above intuition, we define the categorical notion of testing equivalence—two states are testing-equivalent if they have the same modal theory.

Definition 5.2. Assume that we have a testing situation (5.1), and that $\mathbb{C}$ has finite limits. On a coalgebra $X \overset{c}{\to} FX$, the *testing equivalence* TestEq_c is the kernel pair of the theory map th_c defined by (5.2) and (5.3). Equivalently,

$$
\mathsf{TestEq}_c \overset{\langle p_1, p_2 \rangle}{\rightarrowtail} X \times X \overset{\mathsf{th}_c \circ \pi_1}{\underset{\mathsf{th}_c \circ \pi_2}{\rightrightarrows}} SL
\tag{5.4}
$$

is an equalizer.

Similarly, we introduce the categorical notion of "equivalence modulo final coalgebra semantics"; we shall call it *FCS-equivalence* for short.

Definition 5.3. Assume that there is a final F-coalgebra $\zeta : Z \overset{\cong}{\Rightarrow} FZ$, and that $\mathbb{C}$ has finite limits. On a coalgebra $X \overset{c}{\to} FX$, the *FCS-equivalence* FCSEq_c is the kernel pair of the unique map $\mathsf{beh}_c : X \to Z$ induced by finality. Equivalently,

$$
\mathsf{FCSEq}_c \overset{\langle q_1, q_2 \rangle}{\rightarrowtail} X \times X \overset{\mathsf{beh}_c \circ \pi_1}{\underset{\mathsf{beh}_c \circ \pi_2}{\rightrightarrows}} Z
\tag{5.5}
$$

is an equalizer.

It is easily seen that the two "relations" TestEq_c and FCSEq_c on X are *equivalence relations* in the sense of [23, Section 1.3]. That is, they satisfy the reflexivity, symmetry, and transitivity conditions when the conditions are suitably formulated in categorical terms.

Remark 5.4. We present a process equivalence as an equalizer $R \rightarrowtail X \times X$ in $\mathbb{C}$. In [45], instead, a factorization structure on $\mathbb{C}$ is assumed and an equivalence R on X is presented as its projection map $X \twoheadrightarrow X/R$. In fact these approaches are equivalent if $\mathbb{C}$ is *regular*.

Intuitively, a category is regular if it has finite limits and a factorization structure which is compatible with limits in a suitable sense. Let $f : X \to Y$ be an arrow in regular $\mathbb{C}$, $f = m \circ e$ be the image factorization and $r_1, r_2 : R \rightrightarrows X$ be the kernel pair of f.

$$
\begin{array}{ccc}
R & \xrightarrow[r_2]{r_1} X & \xrightarrow{f} Y \\
 & e \downdownarrows & \nearrow m \\
 & \mathrm{Im}(f) &
\end{array}
$$

Then the pair of r_1 and r_2 is the kernel pair of e; conversely e is the coequalizer of r_1 and r_2. For the proof of this fact (which is not trivial), see [23, Lemma 4.4.6.viii], which is essentially from [28, Theorem 1.52].

Now our concern is the comparison between two process semantics TestEq_c and FCSEq_c, as subobjects of $X \times X$. The following lemma is crucial in our investigation.

Lemma 5.5. *A morphism of F-coalgebras preserves theory maps. That is,*

$$
\begin{array}{ccc}
FX & \xrightarrow{Ff} & FY \\
c\uparrow & & \uparrow d \\
X & \xrightarrow{f} & Y
\end{array}
\quad implies \quad
\begin{array}{ccc}
X & \xrightarrow{\mathsf{th}_c} & \\
f\downarrow & \searrow & \\
Y & \xrightarrow{\mathsf{th}_d} & SL
\end{array} \ .
$$

Proof. The following induction diagram proves $Pf \circ \llbracket \_ \rrbracket_d = \llbracket \_ \rrbracket_c$. Naturality of δ plays an important role there.

$$
\begin{array}{ccccc}
ML & - - \overset{Ff}{-} - \to & MPY & \xrightarrow{MPf} & MPX \\
 & & \downarrow{\delta_Y} & & \downarrow{\delta_X} \\
\text{initial} \downarrow \cong & & PFY & \xrightarrow{PFf} & PFX \\
 & & \downarrow{Pd} & & \downarrow{Pc} \\
L & - - - \underset{\llbracket\_\rrbracket_d}{- -} - \to & PY & \xrightarrow{Pf} & PX
\end{array} \ .
$$

Then the claim follows from naturality of the transposition (5.3). $\qquad\square$

We show that in a testing situation like (5.1), tests respect final coalgebra semantics. That is, testing does not distinguish two FCS-equivalent states.

Proposition 5.6. *Consider such a testing situation and equivalence relations as in Definitions 5.2 and 5.3. For any coalgebra $X \xrightarrow{c} FX$ we have an inclusion*

$$
\mathsf{FCSEq}_c \leq \mathsf{TestEq}_c
$$

of subobjects of $X \times X$.

Proof. It suffices to show that the arrow $\langle q_1, q_2 \rangle$ in (5.5) equates the parallel arrows in (5.4); then the claim follows from universality of an equalizer.

$$
\begin{aligned}
\mathsf{th}_c \circ \pi_1 \circ \langle q_1, q_2 \rangle &= \mathsf{th}_c \circ q_1 & \\
&= \mathsf{th}_\zeta \circ \mathsf{beh}_c \circ q_1 & (*) \\
&= \mathsf{th}_\zeta \circ \mathsf{beh}_c \circ q_2 & \text{due to (5.5)} \\
&= \mathsf{th}_c \circ q_2 & (*) \\
&= \mathsf{th}_c \circ \pi_2 \circ \langle q_1, q_2 \rangle \ . &
\end{aligned}
$$

Here $(*)$ is an instance of Lemma 5.5: beh_c is a morphism of coalgebras from c to the final ζ. $\qquad\square$

The converse $\mathsf{TestEq}_c \leq \mathsf{FCSEq}_c$ does not hold in general. For a fixed type of systems (i.e. for fixed $F : \mathbb{C} \to \mathbb{C}$), we can think of logics with varying degree of expressive power; this results in process equivalences with varying granularity. This view is systematically presented by van Glabbeek in [57] as the *linear time-branching time spectrum*—a categorical version of which we consider as an important direction of future work.

It is when we have $\mathsf{FCSEq}_c \overset{\cong}{\Rightarrow} \mathsf{TestEq}_c$ that a modal logic (considered as a testing situation) is said to be *expressive*. Recall that FCSEq_c usually coincides with *bisimilarity* if $\mathbb{C}$ is **Sets**: in this case an expressive logic *captures* bisimilarity.

The following proposition states a (rather trivial) equivalent condition for a testing situation to be expressive. For more ingenious sufficient conditions—which essentially rely on the transpose of δ being monic—see e.g. [33].

Proposition 5.7. *Consider a testing situation as in Definitions 5.2 and 5.3. The testing is expressive, that is, for any coalgebra $X \overset{c}{\to} FX$ we have*

$$\mathsf{TestEq}_c \overset{\cong}{\Rightarrow} \mathsf{FCSEq}_c$$

as subobjects of $X \times X$, if and only if the theory map $\mathsf{th}_\zeta : Z \to SL$ for the final coalgebra is a mono.

Proof. We first prove the "if" direction. In view of Proposition 5.6, it suffices to show that $\langle p_1, p_2 \rangle$ in (5.4) equalizes $\mathsf{beh}_c \circ \pi_1$ and $\mathsf{beh}_c \circ \pi_2$ (which proves $\mathsf{TestEq}_c \leq \mathsf{FCSEq}_c$).

$$
\begin{aligned}
\mathsf{th}_\zeta \circ \mathsf{beh}_c \circ p_1 &= \mathsf{th}_c \circ p_1 && \text{by Lemma 5.5} \\
&= \mathsf{th}_c \circ p_2 && \text{due to (5.4)} \\
&= \mathsf{th}_\zeta \circ \mathsf{beh}_c \circ p_2 && \text{by Lemma 5.5}
\end{aligned}
$$

We have $\mathsf{beh}_c \circ p_1 = \mathsf{beh}_c \circ p_2$ since th_ζ is a mono.

To prove the "only if" direction, first we observe that the FCS-equivalence on the final coalgebra $\zeta : Z \overset{\cong}{\Rightarrow} FZ$ is the diagonal relation: that is,

This is because $\mathsf{beh}_\zeta = \mathrm{id} : Z \to Z$. Now assume that $\mathsf{th}_\zeta \circ k = \mathsf{th}_\zeta \circ l$ for $k, l : Y \rightrightarrows Z$. Universality of an equalizer TestEq_ζ induces a mediating arrow m in the following diagram.

The whole diagram commutes since $\mathsf{TestEq}_\zeta \cong \mathsf{FCSEq}_\zeta$ (by assumption) and $\mathsf{FCSEq}_\zeta \cong Z$ (by the above observation), both as subobjects of $Z \times Z$. This proves $k = l$. $\square$

Remark 5.8. The literature [5, 6] considers more restricted settings than the testing situations in Definition 5.1. There an adjunction $S^{\mathrm{op}} \dashv P$ is replaced by a dual equivalence of categories, and a denotation δ is required to be a natural isomorphism. These additional restrictions allow one to say more about the situations: logics are always expressive; the

main concern of [6] is how to present an abstract modal logic $M : \mathbb{A} \to \mathbb{A}$ by concrete syntax. However, for our purpose in Section 5.2 the greater generality of our notion of testing situations is needed.

5.2. **Canonical testing for trace semantics in $\mathcal{K}\ell(\mathcal{P})$.** In this section we shall present a canonical testing situation for coalgebras in $\mathcal{K}\ell(\mathcal{P})$. We shall also show that the testing is "expressive", in the sense that the testing captures final coalgebra semantics. The intuition is as follows.

Trace semantics for non-deterministic systems assigns to each system c its "(finite) trace set" map $\mathsf{tr}_c : X \to \mathcal{P}A$, where A carries an initial algebra in **Sets**. This suggests a natural testing framework where: an element t of A is a test; a state $x \in X$ of a system passes a test t (i.e. $x \models t$) if and only if the trace set of x includes t (i.e. $t \in \mathsf{tr}_c(x)$). An important point here is that A, carrying an initial algebra in **Sets**, usually gives a *well-founded syntax* for tests.[9]

We focus on a non-deterministic setting (i.e. $T = \mathcal{P}$) in this section and leave a probabilistic one as future work. Although the above intuition is true in probabilistic settings as well—where the 2-valued (pass/failure) observation scheme is replaced by the refined $[0, 1]$-valued one—we do not know yet how to extend the current material to probabilistic settings. The difficulty is that the category $\mathcal{K}\ell(\mathcal{D})$ is not self-dual, as opposed to $\mathcal{K}\ell(\mathcal{P})$; see (5.6) below.

The canonical testing situation which captures finite trace semantics is the following one.

$$\mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} \underset{Op^{\mathrm{op}}}{\overset{Op}{\underset{\cong}{\rightleftarrows}}} \mathcal{K}\ell(\mathcal{P}) \underset{J}{\overset{K}{\underset{\top}{\rightleftarrows}}} \mathbf{Sets} \quad \overline{F}^{\mathrm{op}} \circlearrowright \qquad F \circlearrowright \tag{5.6}$$

Here $J \dashv K$ is the canonical Kleisli adjunction. Recall the self duality $Op : \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} \overset{\cong}{\Rightarrow} \mathcal{K}\ell(\mathcal{P})$ from Section 3.2. The denotation is given by (the components of) the distributive law $\lambda : F\mathcal{P} \Rightarrow \mathcal{P}F$. The following lemma establishes naturality of the denotation.

Lemma 5.9. *Let $F : \mathbf{Sets} \to \mathbf{Sets}$ be a functor which preserves weak pullbacks, and $\overline{F}$ be its lifting induced by the relation lifting (Lemma 2.2). Then the components $F\mathcal{P}X \overset{\lambda_X}{\to} \mathcal{P}FX$ of the corresponding distributive law λ also form a natural transformation*

$$F \circ K \circ Op \Longrightarrow K \circ Op \circ \overline{F}^{\mathrm{op}} \; : \; \mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} \longrightarrow \mathbf{Sets} \; .$$

Proof. The desired natural transformation is obtained from another natural transformation

$$\lambda' \; : \; FK \Longrightarrow K\overline{F} \; : \; \mathcal{K}\ell(\mathcal{P}) \longrightarrow \mathbf{Sets}$$

which we describe in a moment, by post-composing the functor Op. That is, the desired one is the composite

$$FKOp \overset{\lambda' \circ Op}{\Longrightarrow} K\overline{F}Op \overset{(*)}{=} KOp\overline{F}^{\mathrm{op}} \; ,$$

---

[9]Recall the construction of an initial algebra in **Sets** via the initial sequence (Proposition A.1). The set A is the colimit (*union* in **Sets**) of the initial sequence $0 \to F0 \to F^2 0 \to \cdots$. Each $F^n 0$ can be thought of as the set of terms with depth $\leq n$.

or equivalently, in a 2-categorical presentation,

$$
\begin{array}{ccccc}
\mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} & \xrightarrow{\ \ Op\ \ } & \mathcal{K}\ell(\mathcal{P}) & \xrightarrow{\ \ K\ \ } & \mathbf{Sets} \\
\overline{F}^{\mathrm{op}}\downarrow & (*)\ \| & \overline{F}\downarrow & \Downarrow_{\lambda'} & \downarrow F \\
\mathcal{K}\ell(\mathcal{P})^{\mathrm{op}} & \xrightarrow[\ \ Op\ \ ]{} & \mathcal{K}\ell(\mathcal{P}) & \xrightarrow[\ \ K\ \ ]{} & \mathbf{Sets}
\end{array} \quad .
$$

Here the equality $(*)$ is the one in (3.6).

Now we describe the natural transformation λ'. Its components are given by those of λ; naturality of λ' is an easy consequence of λ's being a distributive law. Indeed, given an arrow $f : X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$, the following shows that the naturality square commutes.

$$
\begin{aligned}
K\overline{F}f \circ \lambda_X = \mu^{\mathcal{P}}_{FY} \circ \mathcal{P}\overline{F}f \circ \lambda_X && \text{definition of } K \\
= \mu^{\mathcal{P}}_{FY} \circ \mathcal{P}\lambda_Y \circ \mathcal{P}Ff \circ \lambda_X && \text{definition of } \overline{F} \\
= \mu^{\mathcal{P}}_{FY} \circ \mathcal{P}\lambda_Y \circ \lambda_{\mathcal{P}Y} \circ F\mathcal{P}f && \text{naturality of } \lambda \\
= \lambda_Y \circ F\mu^{\mathcal{P}}_Y \circ F\mathcal{P}f && \lambda \text{ is compatible with the multiplication } \mu^{\mathcal{P}} \text{ of } \mathcal{P} \\
= \lambda_Y \circ FKf && \text{definition of } K \qquad\qquad \square
\end{aligned}
$$

The previous lemma establishes that the situation (5.6) is indeed a testing situation as defined in Definition 5.1.

In the previous Section 5.1, the use of testing situations is demonstrated through comparing testing equivalence and final coalgebra semantics, both described as suitable kernel pairs. Unfortunately this argument is not valid in the current situation (5.6), since the category $\mathcal{K}\ell(\mathcal{P})$ does not have kernel pairs.

Still, we shall claim that the situation (5.6) is "expressive", in the sense that final coalgebra semantics is captured by testing. This claim is supported by the following fact: in the current situation the two arrows tr_c and th_c simply coincide. Therefore their kernel relations—in any reasonable formalization—should coincide as well.

Proposition 5.10. *Let $X \xrightarrow{c} \overline{F}X$ be a coalgebra in $\mathcal{K}\ell(\mathcal{P})$. In the testing situation (5.6), the following arrows in $\mathcal{K}\ell(\mathcal{P})$ coincide.*

- $\mathsf{tr}_c : X \to A$, *giving the final coalgebra (trace) semantics for c.*
- $\mathsf{th}_c : X \to A$, *giving the testing semantics, i.e. the set of passed tests.*

Therefore the testing is "expressive": tests from an initial F-algebra captures trace semantics (which is via a final $\overline{F}$-coalgebra).

Note that, in the general setting in Section 5.1, the codomains of tr_c and th_c need not coincide.

Proof. We shall show that the transpose

$$
\mathsf{tr}_c^{\vee} \ : \ A \longrightarrow \mathcal{P}X \qquad \text{in } \mathbf{Sets}
$$

of tr_c under the adjunction in (5.6) makes the diagram (5.2)—which defines $[\![\_]\!]_c$—commute. This proves $\mathsf{tr}_c^{\vee} = [\![\_]\!]_c$, hence $\mathsf{tr}_c = [\![\_]\!]_c^{\vee} = \mathsf{th}_c$.

First note that the transpose $\mathsf{tr}_c^{\vee} : A \to \mathcal{P}X$ is given by the arrow $Op(\mathsf{tr}_c) : A \to X$ in $\mathcal{K}\ell(\mathcal{P})$ thought of as an arrow in **Sets**. In the sequel we shall write $Op(\mathsf{tr}_c)$ for $\mathsf{tr}_c^{\vee}$.

Commutativity of the diagram (4.1)—defining tr_c—yields the following equality.

$$
Op(\mathsf{tr}_c) \circ Op(J\alpha^{-1}) = Op(c) \circ Op(\overline{F}^{\mathrm{op}}\mathsf{tr}_c) \qquad \text{in } \mathcal{K}\ell(\mathcal{P}).
$$

By the definition of composition in $\mathcal{K}\ell(\mathcal{P})$, it reads as follows in **Sets**.

$$\mu_X \circ \mathcal{P}(Op(\mathsf{tr}_c)) \circ Op(J\alpha^{-1}) = \mu_X \circ \mathcal{P}(Op(c)) \circ Op(\overline{F}^{\mathrm{op}}\mathsf{tr}_c) \qquad (5.7)$$

We use this equality in showing that $Op(\mathsf{tr}_c)$ makes the diagram (5.2) commute.

$$\begin{aligned}
Op(\mathsf{tr}_c) \circ \alpha &= \mu_X \circ \eta_X \circ Op(\mathsf{tr}_c) \circ \alpha && \text{unit law} \\
&= \mu_X \circ \mathcal{P}(Op(\mathsf{tr}_c)) \circ \eta_A \circ \alpha && \text{naturality of } \eta \\
&= \mu_X \circ \mathcal{P}(Op(\mathsf{tr}_c)) \circ Op(J\alpha^{-1}) && Op(J\alpha^{-1}) = J\alpha = \eta_A \circ \alpha \\
&= \mu_X \circ \mathcal{P}(Op(c)) \circ Op(\overline{F}^{\mathrm{op}}\mathsf{tr}_c) && \text{by (5.7)} \\
&= \mu_X \circ \mathcal{P}(Op(c)) \circ \overline{F}\, Op(\mathsf{tr}_c) && Op\overline{F}^{\mathrm{op}} = \overline{F}\, Op, \text{ (3.6)} \\
&= \mu_X \circ \mathcal{P}(Op(c)) \circ \lambda_X \circ F\, Op(\mathsf{tr}_c) && \text{definition of } \overline{F} \\
&= K\, Op(c) \circ \lambda_X \circ F\, Op(\mathsf{tr}_c) \ .
\end{aligned}$$

Recall that M in (5.2) is now F; P in (5.2) is now $K\, Op$. This concludes the proof. $\qquad \square$

6. Conclusions and future work

We have developed a mathematical principle underlying "trace semantics" for various kinds of branching systems, namely coinduction in a Kleisli category. This general view is supported by a technical result that a final coalgebra in a Kleisli category is induced by an initial algebra in **Sets**.

The possible instantiations of our generic framework include non-deterministic systems and probabilistic systems, but do *not* yet include systems with both non-deterministic and probabilistic branching. The importance of having both of these branchings in system verification has been claimed by many authors e.g. [60, 48], with an intuition that probabilistic branching models the choices "made by the system, i.e. on *our* side", while (coarser) non-deterministic choices are "made by the (unknown) environment of the system, i.e. on the *adversary's* side". A typical example of such systems is given by *probabilistic automata* introduced by Segala [48].

In fact this combination of non-deterministic and probabilistic branching is a notoriously difficult one from a theoretical point of view [8, 59, 54]: many mathematical tools that are useful in a purely non-deterministic or probabilistic setting cease to work in the presence of both. For our framework of generic trace semantics, the problem is that we could not find a suitable monad T with an order structure.

We have used the order-enriched structure of a Kleisli category (expressing "more possibilities") to obtain the initial algebra-final coalgebra coincidence result. However, an order structure is not the only one that can yield such coincidence: other examples include metric, quasi-metric and quantale-enriched structures (in increasing generality). See e.g. [56, 10] for the potential use of such enriched structures in a coalgebraic setting. The relation of the current work to such structures is yet to be investigated.

In the discipline of process algebra, a system is represented by an *algebraic* term (such as $a.P \parallel a.Q$) and a structural operational semantics (SOS) rule determines its dynamics, that is, its *coalgebraic* structure. This is where "algebra meets coalgebra" and the interaction is studied e.g. in [55, 4, 31]. In our recent work [18] we claim the importance of the *microcosm principle* in this context and provide a "general compositionality theorem": under suitable assumptions, the final coalgebra semantics is compatible with the algebraic structure. The

results of the current paper say that the final coalgebra semantics can be interpreted as finite trace semantics, hence the result in [18] also yields a general compositionality result for trace semantics.

In this paper we have included some material—on possibly-infinite traces and testing situations—which, unfortunately, we have worked out only in a non-deterministic setting. A fully general account on these topics is left as future work.

Finally, there are so many different process semantics for branching systems, between two edges of bisimilarity and trace equivalence in the linear time-branching time spectrum [57]. How to capture them in a coalgebraic setting is, we believe, an important and challenging question.

Acknowledgment

References

[1] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D.M. Gabbai, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 3, pages 1–168. Oxford Univ. Press, 1994.

[2] J. Adámek and V. Koubek. Least fixed point of a functor. *Journ. Comp. Syst. Sci*, 19(2):163–178, 1979.

[3] M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer, Berlin, 1985.

[4] F. Bartels. *On generalised coinduction and probabilistic specification formats. Distributive laws in coalgebraic modelling*. PhD thesis, Free Univ. Amsterdam, 2004.

[5] M.M. Bonsangue and A. Kurz. Duality for logics of transition systems. In V. Sassone, editor, *FoSSaCS*, volume 3441 of *Lect. Notes Comp. Sci.*, pages 455–469. Springer, 2005.

[6] M.M. Bonsangue and A. Kurz. Presenting functors by operations and equations. In L. Aceto and A. Ingólfsdóttir, editors, *FoSSaCS*, volume 3921 of *Lect. Notes Comp. Sci.*, pages 172–186. Springer, 2006.

[7] F. Borceux. *Handbook of Categorical Algebra*, volume 50, 51 and 52 of *Encyclopedia of Mathematics*. Cambridge Univ. Press, 1994.

[8] L. Cheung. *Reconciling Nondeterministic and Probabilistic Choices*. PhD thesis, Radboud Univ. Nijmegen, 2006.

[9] M. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps*. Distinguished Dissertations in Computer Science. Cambridge Univ. Press, 1996.

[10] M. Fiore. A coinduction principle for recursive data types based on bisimulation. *Inf. & Comp.*, 127(2):186–198, 1996.

[11] M. Fokkinga. Monadic maps and folds for arbitrary datatypes. *Memoranda Informatica, University of Twente*, 94–28, 1994.

[12] P.J. Freyd. Remarks on algebraically compact categories. In M. Fourman, P. Johnstone, and A. Pitts, editors, *Applications of Categories in Computer Science*, number 177 in LMS, pages 95–106. Cambridge Univ. Press, 1992.

[13] P. Freyd. Algebraically complete categories. In A. Carboni, M. Pedicchio, and G. Rosolini, editors, *Como Conference on Category Theory*, number 1488 in Lect. Notes Math., pages 95–104. Springer, Berlin, 1991.

[14] I. Hasuo. Generic forward and backward simulations. In C. Baier and H. Hermanns, editors, *International Conference on Concurrency Theory (CONCUR 2006)*, volume 4137 of *Lect. Notes Comp. Sci.*, pages 406–420. Springer, Berlin, 2006.

[15] I. Hasuo and B. Jacobs. Coalgebraic trace semantics for probabilistic systems. In P. Mosses, J. Power, and M. Seisenberger, editors, *CALCO-jnr Workshop*, 2005.

[16] I. Hasuo and B. Jacobs. Context-free languages via coalgebraic trace semantics. In J.L. Fiadeiro, N. Harman, M. Roggenbach, and J. Rutten, editors, *International Conference on Algebra and Coalgebra in Computer Science (CALCO'05)*, volume 3629 of *Lect. Notes Comp. Sci.*, pages 213–231. Springer, Berlin, 2005.

[17] I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace theory. In N. Ghani and A.J. Power, editors, *International Workshop on Coalgebraic Methods in Computer Science (CMCS 2006)*, volume 164 of *Elect. Notes in Theor. Comp. Sci.*, pages 47–65. Elsevier, Amsterdam, 2006.

[18] I. Hasuo, B. Jacobs, and A. Sokolova. The microcosm principle and concurrency in coalgebras. In M. Haveraaen, J. Power, and M. Seisenberger, editors, *CALCO-jnr Workshop*, 2007. Extended version available from `http://www.cs.ru.nl/~ichiro`.

[19] I. Hasuo and Y. Kawabe. Probabilistic anonymity via coalgebraic simulations. In R. De Nicola, editor, *European Symposium on Programming (ESOP 2007)*, volume 4421 of *Lect. Notes Comp. Sci.*, pages 379–394. Springer, 2007.

[20] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. & Comp.*, 145:107–152, 1998.

[21] C. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[22] J. Hughes and B. Jacobs. Simulations in coalgebra. *Theor. Comp. Sci.*, 327(1-2):71–108, 2004.

[23] B. Jacobs. *Categorical Logic and Type Theory*. North Holland, Amsterdam, 1999.

[24] B. Jacobs. Trace semantics for coalgebras. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, volume 106 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, Amsterdam, 2004.

[25] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:222–259, 1997.

[26] B. Jacobs. Introduction to coalgebra. Towards mathematics of states and observations. Draft of a book, `www.cs.ru.nl/B.Jacobs/PAPERS/index.html`, 2007.

[27] C. Jay. A semantics for shape. *Science of Comput. Progr.*, 25:251–283, 1995.

[28] P. Johnstone. *Topos Theory*. Academic Press, London, 1977.

[29] G. Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in LMS. Cambridge Univ. Press, 1982.

[30] M. Kick, J. Power, and A. Simpson. Coalgebraic semantics for timed processes. *Inf. & Comp.*, 204(4):588–609, 2006.

[31] B. Klin. From bialgebraic semantics to congruence formats. In *Workshop on Structural Operational Semantics (SOS 2004)*, volume 128, pages 3–37, 2005.

[32] B. Klin. Bialgebraic operational semantics and modal logic. In *Logic in Computer Science*, 2007. To appear.

[33] B. Klin. Coalgebraic modal logic beyond **Sets**. In *MFPS XXIII*, volume 173, pages 177–201. Elsevier, Amsterdam, 2007.

[34] A. Kock. Monads on symmetric monoidal closed categories. *Arch. Math.*, XXI:1–10, 1970.

[35] C. Kupke, A. Kurz, and Y. Venema. Stone coalgebras. *Theor. Comp. Sci.*, 327(1-2):109–134, 2004.

[36] A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Universität München, April 2000.

[37] A. Kurz. Coalgebras and their logics. *SIGACT News*, 37(2):57–77, 2006.

[38] F. Lawvere. Metric spaces, generalized logic, and closed categories. *Seminario Matematico e Fisico. Rendiconti di Milano*, 43:135–166, 1973. Reprinted in *Theory and Applications of Categories*, 1:1–37, 2002.

[39] M. Lenisa, J. Power, and H. Watanabe. Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, number 33 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2000. `www.elsevier.nl/locate/entcs/volume33.html`.

[40] M. Lenisa, J. Power, and H. Watanabe. Category theory for operational semantics. *Theor. Comp. Sci.*, 327(1–2):135–154, 2004.

[41] N. Lynch and F. Vaandrager. Forward and backward simulations. I. Untimed systems. *Inf. & Comp.*, 121(2):214–233, 1995.

[42] S. Mac Lane. *Categories for the Working Mathematician*. Springer, Berlin, 1971.

[43] P.S. Mulry. Lifting theorems for Kleisli categories. In *Mathematical Foundations of Programming Semantics (MFPS IX)*, pages 304–319, London, UK, 1994. Springer-Verlag.

[44] A. Pardo. Fusion of recursive programs with computational effects. *Theor. Comp. Sci.*, 260(1–2):165–207, 2001.

[45] D. Pavlović, M. Mislove, and J.B. Worrell. Testing semantics: connecting processes and process logics. In M. Johnson and V. Vene, editors, *Algebraic Methodology and Software Technology (AMAST 2006)*, volume 4019 of *Lect. Notes Comp. Sci.* Springer, 2006.

[46] J. Power and D. Turi. A coalgebraic foundation for linear time semantics. In *Category Theory and Computer Science*, number 29 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 1999.

[47] J. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000.

[48] R. Segala. *Modeling and verification of randomized distributed real-time systems.* PhD thesis, MIT, 1995.

[49] R. Segala. A compositional trace-based semantics for probabilistic automata. In *International Conference on Concurrency Theory (CONCUR '95)*, pages 234–248. Springer-Verlag, 1995.

[50] A.K. Simpson. Recursive types in Kleisli categories. Unpublished paper, available at `http://homepages.inf.ed.ac.uk/als/Research/`, 1992.

[51] M. Smyth and G. Plotkin. The category theoretic solution of recursive domain equations. *SIAM Journ. Comput.*, 11:761–783, 1982.

[52] A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems.* PhD thesis, TU Eindhoven, 2005.

[53] M. Stoelinga and F.W. Vaandrager. A testing scenario for probabilistic automata. In J.C.M. Baeten, J.K. Lenstra, J. Parrow, and G.J. Woeginger, editors, *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 464–477. Springer, 2003.

[54] R. Tix, K. Keimel, and G.D. Plotkin. Semantic domains for combining probability and non-determinism. *Elect. Notes in Theor. Comp. Sci.*, 129:1–104, 2005.

[55] D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Logic in Computer Science*, pages 280–291. IEEE, Computer Science Press, 1997.

[56] D. Turi and J. Rutten. On the foundations of final semantics: non-standard sets, metric spaces and partial orders. *Math. Struct. in Comp. Sci.*, 8(5):481–540, 1998.

[57] R. van Glabbeek. The linear time–branching time spectrum I; the semantics of concrete, sequential processes. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001. Available at `http://boole.stanford.edu/pub/spectrum1.ps.gz`.

[58] R. van Glabbeek, S. Smolka, and B. Steffen. Reactive, generative, and stratified models of probabilistic processes. *Inf. & Comp.*, 121:59–80, 1995.

[59] D. Varacca and G. Winskel. Distributing probabililty over nondeterminism. *Math. Struct. in Comp. Sci.*, 16(1):87–113, 2006.

[60] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *FOCS '85*, pages 327–338, 1985.

Appendix A. Preliminaries

A.1. **Initial/final sequences.** Here we recall the standard construction [2] of the initial algebra (or the final coalgebra) via the initial (or final) sequence. Notice that the base category need not be **Sets**.

Let $\mathbb{C}$ be a category with initial object 0, and $F : \mathbb{C} \to \mathbb{C}$ be an endofunctor. The *initial sequence*[10] of F is a diagram

$$0 \xrightarrow{\ \text{!}\ } F0 \xrightarrow{\ F\,\text{!}\ } \ \cdots \ \xrightarrow{\ F^{n-1}\,\text{!}\ } F^n 0 \xrightarrow{\ F^n\,\text{!}\ } \ \cdots$$

where $\text{!} : 0 \to X$ is the unique arrow.

Now assume that:

- the initial sequence has an ω-colimit[11] $(F^n 0 \xrightarrow{\alpha_n} A)_{n<\omega}$;
- the functor F preserves that ω-colimit.

---

[10] In this paper we consider only initial/final sequences of length ω.

[11] An ω-colimit is a colimit of a diagram whose shape is the ordinal ω.

Then we have two cocones $(\alpha_n)_{n<\omega}$ and $(F\alpha_{n-1})_{n<\omega}$ over the initial sequence. Moreover, the latter is again a colimit: hence we have mediating isomorphisms between these cones.

Proposition A.1. *The F-algebra $\alpha : FA \stackrel{\cong}{\Rightarrow} A$ is initial.*

Proof. For future reference we prove the dual result: see Proposition A.2. $\square$

The dual of this construction yields a final F-coalgebra. Assume that the base category $\mathbb{C}$ has a terminal object 1. The *final sequence* of F is

$$1 \xleftarrow{\quad ! \quad} F1 \xleftarrow{\quad F! \quad} \cdots \xleftarrow{\quad F^{n-1}! \quad} F^n1 \xleftarrow{\quad F^n! \quad} \cdots \;,$$

where $! : X \to 1$ is the unique arrow. Assume that it has an ω^{op}-limit $(Z \xrightarrow{\zeta_n} F^n1)_{n<\omega}$, and also that F preserves that ω^{op}-limit. We have the following situation.

Proposition A.2. *The coalgebra $\zeta : Z \stackrel{\cong}{\Rightarrow} FZ$ is final.*

Proof. Any F-coalgebra $c : X \to FX$ induces a cone $(X \xrightarrow{\beta_n} F^n1)_{n<\omega}$ over the final sequence in the following way.

$$\beta_0 = \; ! : X \longrightarrow 1 \; , \qquad \beta_{n+1} = F\beta_n \circ c \; .$$

Now we can prove the following: for an arrow $f : X \to Z$, f is a morphism of coalgebras from c to ζ if and only if f is a mediating arrow from the cone $(\beta_n)_{n<\omega}$ to the limit $(\zeta_n)_{n<\omega}$. Hence such a morphism of coalgebras uniquely exists. $\square$

It is easy to see that every shapely functor in **Sets** preserves ω-colimits and ω^{op}-limits. Hence we have the following.

Lemma A.3. *A shapely functor F has both an initial algebra and a final coalgebra in **Sets**.*
$\square$

A.2. **limit-colimit coincidence.** We recall some relevant notions and results from [51]. The idea is that in a suitable order-enriched setting, (co)limits are equivalently described as an order-theoretic notion of **O**-*(co)limits*. Due to the inherent coincidence between **O**-limits and **O**-colimits, we also obtain the so-called *limit-colimit coincidence*.

$$
\begin{array}{ccc}
\text{limit} & & \text{colimit} \\
\| & & \| \\
\textbf{O}\text{-limit} & \rule{2cm}{0.4pt} & \textbf{O}\text{-colimit} \\
& \text{\scriptsize obvious coincidence} &
\end{array}
$$

The notions of **O**-(co)limits are stated in terms of *embedding-projection pairs* which we can define in an order-enriched category. In the sequel we assume the **Cppo**-enriched structure.

Definition A.4 (Embedding-projection pairs). Let $\mathbb{C}$ be a **Cppo**-enriched category. A pair of arrows

$$X \underset{p}{\overset{e}{\rightleftarrows}} Y$$

in $\mathbb{C}$ is said to be an *embedding-projection pair* if we have $p \circ e = \mathrm{id}$ and $e \circ p \sqsubseteq \mathrm{id}$. Diagrammatically presented,

By $p \circ e = \mathrm{id}$ we automatically have that e is a mono and p is an epi. Both split.

Proposition A.5. *Let $(e, p), (e', p') : X \rightleftarrows Y$ be two embedding-projection pairs with the same (co)domains. Then $e \sqsubseteq e'$ holds if and only if $p' \sqsubseteq p$. As a consequence, one component of an embedding-projection pair determines the other.* $\square$

This proposition justifies the notation e^P for the projection corresponding to a given embedding e, and p^E for the embedding corresponding to a given projection p. It is easy to check that

$$(e \circ f)^P = f^P \circ e^P \qquad \text{and} \qquad (p \circ q)^E = q^E \circ p^E \ .$$

Definition A.6 (**O**-(co)limits). Let $X_0 \xrightarrow{f_0} X_1 \xrightarrow{f_1} \cdots$ be an ω-chain in a **Cppo**-enriched $\mathbb{C}$. A cocone $(X_n \xrightarrow{\sigma_n} C)_{n < \omega}$ over this chain is said to be an **O**-*colimit* if:

- each σ_n is an embedding;

- the sequence of arrows $(C \xrightarrow{\sigma_n^P} X_n \xrightarrow{\sigma_n} C)_{n < \omega}$ is increasing. Moreover its join taken in the cpo $\mathbb{C}(C, C)$ is id_C.

Dually, a cone $(C \xrightarrow{\gamma_n} Y_n)_{n < \omega}$ over an ω^{op}-chain $Y_0 \xleftarrow{g_0} Y_1 \xleftarrow{g_1} \cdots$ is an **O**-*limit* if: each γ_n is a projection, and the sequence $(\gamma_n^E \circ \gamma_n : C \to C)_{n < \omega}$ is increasing and its join is id_C.

The following proposition establishes the equivalence between (co)limits and **O**-(co)limits. For its full proof the reader is referred to [51].

Proposition A.7 (Propositions A, B, C, D in [51]). *Let $X_0 \xrightarrow{e_0} X_1 \xrightarrow{e_1} \cdots$ be an ω-chain where each e_n is an embedding.*

(1) *Let $(X_n \xrightarrow{\sigma_n} C)_{n < \omega}$ be the colimit over the chain. Then each σ_n is also an embedding. Moreover, $(\sigma_n)_{n < \omega}$ is an **O**-colimit.*

(2) *Conversely, an **O**-colimit $(X_n \xrightarrow{\sigma_n} C)_{n < \omega}$ over the chain is a colimit.*

Dually, let $X_0 \xleftarrow{p_0} X_1 \xleftarrow{p_1} \cdots$ be an ω^{op}-chain where each p_n is a projection.

(3) *Let* $(D \xrightarrow{\tau_n} X_n)_{n<\omega}$ *be a limit over the chain. Then each* τ_n *is also a projection. Moreover* $(\tau_n)_{n<\omega}$ *is an* **O**-*limit.*

(4) *Conversely, an* **O**-*limit* $(D \xrightarrow{\tau_n} X_n)_{n<\omega}$ *over the chain is a limit.*

Proof. For later reference we present the proof of (4). Let $(B \xrightarrow{\beta_n} X_n)_{n<\omega}$ be an arbitrary cone over the chain $X_0 \xleftarrow{p_0} X_1 \xleftarrow{p_1} \cdots$. First we prove the uniqueness of a mediating map $f : B \to D$.

$$
\begin{aligned}
f = \mathrm{id}_D \circ f &= \left(\bigsqcup_{n<\omega}(\tau_n^E \circ \tau_n)\right) \circ f & &(\tau_n)_{n<\omega} \text{ is an } \mathbf{O}\text{-limit} \\
&= \bigsqcup_{n<\omega}(\tau_n^E \circ \tau_n \circ f) & &\text{composition is continuous} \\
&= \bigsqcup_{n<\omega}(\tau_n^E \circ \beta_n) & &f \text{ is mediating .}
\end{aligned}
$$

We conclude the proof by showing that the sequence $(\tau_n^E \circ \beta_n)_{n<\omega}$ is increasing, hence such f indeed exists.

$$
\tau_n^E \circ \beta_n \;=\; \tau_n^E \circ p_n \circ \beta_{n+1} \;=\; \tau_{n+1}^E \circ p_n^E \circ p_n \circ \beta_{n+1} \;\sqsubseteq\; \tau_{n+1}^E \circ \beta_{n+1}
$$

The last inequality holds because $p_n^E \circ p_n \sqsubseteq \mathrm{id}$ from the definition of embedding-projection pairs. $\qquad\square$

Theorem A.8 (Limit-colimit coincidence). *Let* $X_0 \xrightarrow{e_0} X_1 \xrightarrow{e_1} \cdots$ *be an* ω-*chain where each* e_n *is an embedding, and* $(X_n \xrightarrow{\sigma_n} C)_{n<\omega}$ *be the colimit over the chain. Then each* σ_n *is an embedding, and the cone* $(C \xrightarrow{\sigma_n^P} X_n)_{n<\omega}$ *is a limit over the* ω^{op}-*chain* $X_0 \xleftarrow{e_0^P} X_1 \xleftarrow{e_1^P} \cdots$.

Dually, the limit of an ω^{op}-*chain of projections consists of projections. By taking the corresponding embeddings we obtain a colimit of an* ω-*chain of embeddings.*

Proof. We prove the first statement. By Proposition A.7 each σ_n is an embedding, and moreover $(\sigma_n)_{n<\omega}$ is an **O**-colimit. Now obviously $(\sigma_n^P)_{n<\omega}$ is a cone over $X_0 \xleftarrow{e_0^P} X_1 \xleftarrow{e_1^P} \cdots$. Here we use the inherent coincidence of **O**-(co)limits: namely, the condition that $(\sigma_n)_{n<\omega}$ is an **O**-colimit is exactly the same as that $(\sigma_n^P)_{n<\omega}$ is an **O**-limit. We use Proposition A.7 to conclude the proof. $\qquad\square$