Path category for free
Open morphisms from coalgebras with non-deterministic branching

Thorsten Wißmann\textsuperscript{1}[0000–0001–8993–6486], Jérémy Dubut\textsuperscript{2,3}, Shin-ya Katsumata\textsuperscript{2}, and Ichiro Hasuo\textsuperscript{2,4}

\textsuperscript{1} Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
\url{thorsten.wissmann@fau.de}
\textsuperscript{2} National Institute of Informatics, Tokyo, Japan
\{dubut, s-katsumata, hasuo\}@nii.ac.jp
\textsuperscript{3} Japanese-French Laboratory for Informatics, Tokyo, Japan
\textsuperscript{4} SOKENDAI, Kanagawa, Japan

\textbf{Abstract.} There are different categorical approaches to variations of transition systems and their bisimulations. One is coalgebra for a functor $G$, where a bisimulation is defined as a span of $G$-coalgebra homomorphism. Another one is in terms of path categories and open morphisms, where a bisimulation is defined as a span of open morphisms. This similarity is no coincidence: given a functor $G$, fulfilling certain conditions, we derive a path-category for pointed $G$-coalgebras and lax homomorphisms, such that the open morphisms turn out to be precisely the $G$-coalgebra homomorphisms. The above construction provides path-categories and trace semantics for free for different flavours of transition systems: (1) non-deterministic tree automata (2) regular non-deterministic nominal automata (RNNA), an expressive automata notion living in nominal sets (3) multisorted transition systems. This last instance relates to Lasota’s construction, which is in the converse direction.

\textbf{Keywords:} Coalgebra · Open maps · Categories · Nominal Sets

1 Introduction

\textit{Coalgebras} [24] and \textit{open maps} [14] are two main categorical approaches to transition systems and bisimulations. The former describes the branching type of systems as an endofunctor, a system becoming a coalgebra and bisimulations being spans of coalgebra homomorphisms. Coalgebra theory makes it easy to consider state space types in different settings, e.g. nominal sets [15,16] or algebraic categories [4,10,19]. The latter, open maps, describes systems as objects

* This research was supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST. The first author was supported by the DFG project MI 717/5-1. He expresses his gratitude for having been invited to Tokyo, which initiated the present work.
of a category and the execution types as particular objects called paths. In this case, bisimulations are spans of open morphisms, namely morphisms reflecting extensions of executions. Open maps are particularly adapted to extend bisimilarity to history dependent behaviors, e.g., true concurrency [7,6], timed systems [21] and weak (bi)similarity [8]. Coalgebra homomorphisms and open maps are then key concepts to describe bisimilarity categorically. They intuitively correspond to functional bisimulations, that is, those maps between states whose graph is a bisimulation.

We are naturally interested in the relationship between those two categorical approaches to transition systems and bisimulations. A reduction of open maps situations to coalgebra was given by Lasota using multi-sorted transition systems [18]. In this paper, we give the reduction in the other direction: from the category \( \text{Coalg}_i(TF) \) of pointed \( TF \)-coalgebras and lax homomorphisms, we construct the path-category \( \text{Path} \) and a functor \( J : \text{Path} \rightarrow \text{Coalg}_i(TF) \) such that \( \text{Path} \)-open morphisms coincide with strict homomorphisms, hence functional bisimulations. This development is carried out with the case where \( T \) is a powerset-like functor, and covers transition systems allowing non-deterministic branching.

The key concept in the construction of \( \text{Path} \) is precise maps. Roughly speaking, an object \( P \in \text{Path} \) represents a generic shape of finite-depth deterministic transitions. The property required for such a shape is the unique path property: every non-initial state has exactly one previous state, and precise maps are designed to capture this property. Then we represent a finite-depth deterministic transition system by a finite sequence \( f_i : X_i \rightarrow FX_{i+1} + 1 \) of precise maps (we add 1 for termination). \( J \) converts such a data into a pointed \( TF \)-coalgebra.

Once we set up the situation \( J : \text{Path} \rightarrow \text{Coalg}_i(TF) \), we are on the framework of open map bisimulations. Our construction of \( \text{Path} \) using precise maps is justified by the characterization theorem: \( \text{Path} \)-open morphisms and strict coalgebra homomorphisms coincide (Theorem 3.20 and Theorem 3.24). This coincidence relies on the concept of path-reachable coalgebras, namely, coalgebras such that every state can be reached by a path. Under mild conditions, path-reachability is equivalent to an existing notion in coalgebra, defined as the non-existence of a proper sub-coalgebra (Section 3.5). Additionally, this characterization produces a canonical trace semantics for free, given in terms of paths (Section 3.6).

We illustrate our reduction with several concrete situations: different classes of non-deterministic top-down tree automata using analytic functors (Section 4.1), Regular Nondeterministic Nominal Automata (RNNA), an expressive automata notion living in nominal sets (Section 4.2), multisorted transition sys-
tems, used in Lasota’s work to construct a coalgebra situation from an open map situation (Section 4.3).

**Notation** We assume basic categorical knowledge and stick to standard notation (see e.g. [1,3]). In particular, the cotupling of morphisms \( f : A \to C \), \( g : B \to C \) is denoted by \([f, g] : A + B \to C\), and the unique morphism to the terminal object is \(! : X \to 1\) for every \(X\).

## 2 Two categorical approaches for bisimulations

We introduce the two formalisms involved in the present paper: the open maps (Section 2.1) and the coalgebras (Section 2.2). Those formalisms will be illustrated on the classic example of Labelled Transition Systems (LTSs).

**Definition 2.1.** Fix a set \(A\) called the alphabet. A Labelled Transition System is a triple \((S, i, \Delta)\) with \(S\) a set of states, \(i \in S\) the initial state, and \(\Delta \subseteq S \times A \times S\) the transition relation. When \(\Delta\) is obvious from the context, we write \(s \xrightarrow{a} s'\) to mean \((s, a, s') \in \Delta\).

To formalise LTSs as a category, one has to describe morphisms, that is, those functions that preserves the structure of LTSs:

**Definition 2.2.** A morphism of LTSs from \(T = (S, i, \Delta)\) to \(T' = (S', i', \Delta')\) is a function \(f : S \to S'\) such that for every \((s, a, s') \in \Delta\), \((f(s), a, f(s')) \in \Delta'\). LTSs and morphisms of LTSs form a category, which we denote by \(\text{LTS}_A\).

Some authors choose other notions of morphisms (e.g., [14]), allowing them to operate between LTSs with different alphabets for example. The usual way of comparing LTSs is by using simulations and bisimulations [22]. The former describes what it means for a system to have at least the behaviours of another, the latter describes that two systems have exactly the same behaviours. Concretely:

**Definition 2.3.** A simulation from \(T = (S, i, \Delta)\) to \(T' = (S', i', \Delta')\) is a relation \(R \subseteq S \times S'\) such that (1) initial states are related, that is, \((i, i') \in R\), and (2) transitions of \(T\) can be simulated by transitions of \(T'\), that is, for every \(s \xrightarrow{a} t\) and \((s, s') \in R\), there is \(t' \in S'\) such that \(s' \xrightarrow{a} t'\) and \((t, t') \in R\). Such a relation \(R\) is a bisimulation if \(R^{-1} = \{(s', s) \mid (s, s') \in R\}\) is also a simulation.

Morphisms of LTSs are functional simulations, i.e., functions between states whose graph is a simulation. So how to model (1) systems, (2) functional simulations and (3) functional bisimulations categorically? In the next two sections, we will describe known answers to this question, with open maps and coalgebra. In any cases, it is possible to capture similarity and bisimilarity of two LTSs \(T\) and \(T'\). A simulation is a (jointly monic) span of a functional bisimulation and a functional simulation, and a bisimulation is a simulation whose converse is also a simulation, as depicted in Table 1. Consequently, to understand similarity and bisimilarity, it is enough to understand functional simulations and bisimulations.
2.1 Open maps

The categorical framework of open maps [14] assumes functional simulations to be already modeled as a category. For example, for LTS\(_A\), objects are systems, and morphisms are functional simulations. The main part is to describe functional bisimulations in form of functional simulations. The main idea is to describe the notion of “run” in a system using morphisms. A run labelled by \(a = a_1 \cdots a_n \in A^*\) in a LTS \(T = (S, i, \Delta)\) consists of states \(s_1, \cdots, s_n \in S\) satisfying:

\[
i \xrightarrow{a_1} s_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} s_n\]  \(1\)

To represent such a run by a LTS-morphism, we use a particular class of LTSs called linear systems. A linear system \(P_a\) over \(a\) represents the “common shape” of runs along the string \(a\), and is defined to be \((\{0, \cdots, n\}, 0, \{(i - 1, a_i, i) \mid 1 \leq i \leq n\})\). We then easily see that runs labelled by \(a\) in the LTS \(T\) bijectively correspond to LTS-morphisms of type \(P_a \rightarrow T\). Moreover, we can represent transfers and extensions of runs:

1. The transfer of a run \(r : P_a \rightarrow T\) by a LTS-morphism \(f : T \rightarrow T'\) is the composite \(f \cdot r : P_a \rightarrow T'\), which is a run in \(T'\).

2. Consider the full subcategory \(\text{Lin}_A\) of LTS\(_A\) consisting of all linear systems. A morphism in \(\text{Lin}_A\) is naturally seen as an embedding of a run shape to a longer one. Thus by composing a run \(r' : P_b \rightarrow T\) and a \(\text{Lin}_A\)-morphism \(\Phi : P_a \rightarrow P_b\), we obtain the truncation \(r' \cdot \Phi : P_a \rightarrow T\) of \(r'\) by \(\Phi\). We say that a run \(r' : P_a \rightarrow T\) along \(\Phi : P_a \rightarrow P_b\) if \(r\) is a truncation of \(r'\) by \(\Phi\), that is, \(r' = r \cdot \Phi\).

These concepts are readily expressible in a more general situation of a category \(\mathcal{M}\) together with a specified category \(\mathcal{P}\) mapped into \(\mathcal{M}\) [14].

Definition 2.4. An open map situation is given by categories \(\mathcal{M}, \mathcal{P}\) (of systems and paths respectively) together with a functor \(J : \mathcal{P} \rightarrow \mathcal{M}\).

Evidently, \(\text{Lin}_A \hookrightarrow \text{LTS}_A\) is an open map situation. We note that \(\text{Lin}_A\) is isomorphic to the prefix order over \(A^*\).

We have already seen that morphisms between LTSs are functional simulations. Which ones are functional bisimulations? A sufficient condition is for the graph of a morphism \(f : T \rightarrow T'\) to satisfy the properties of a bisimulation:

\[
\forall s \in S, \forall f(s) \xrightarrow{a} t', \exists s' \in S, s \xrightarrow{a} s' \land f(s') = t'\]  \(2\)

It is not a necessary condition because bimisimilarity is a property only on the reachable part of the system, so it suffices to quantify over the \(s\) that are reachable by a run. Equivalently, we can quantify over runs:

\[
\forall \text{run } p = s_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n \quad \forall \text{run } q = f(s_0) \xrightarrow{a_1} \cdots \xrightarrow{a_n} f(s_n) \xrightarrow{b} t' \quad \exists \text{run } d = s_0 \xrightarrow{a_1} \cdots \xrightarrow{a_n} s_n \xrightarrow{b} s', f(s') = t'\]  \(3\)

In other words, for any run \(p\) and extension \(q\) of \(f \cdot p\) along \(\Phi : P_a \rightarrow P_{ab}\), there exists an extension \(d\) of \(p\) along \(\Phi\) such that \(f \cdot d = q\). By unfolding the definition
of extension in a general open map situation, and generalising Φ to arbitrary P-morphisms, we obtain the concept of open morphism [14]:

Definition 2.5. Let \( J : P \to \mathcal{M} \) be an open map situation. An \( \mathcal{M} \)-morphism \( f : T \to T' \) is said to be open if for every morphism \( p, q \in \mathcal{M} \) and \( \Phi \in P \) making the square on the right commute, there is \( d \) making the two triangles commute.

E.g. Open morphisms in reachable \( \text{LTS}_A \) are precisely functional bisimulations.

2.2 Coalgebras

The theory of G-coalgebras is another categorical framework to study bisimulations. The type of systems is modelled using an endofunctor \( G : \mathcal{C} \to \mathcal{C} \) and a system is then a coalgebra for this functor, that is, a pair of an object \( S \) of \( \mathcal{C} \) (modeling the state space), and of a morphism of type \( S \to GS \) (modeling the transitions). Let us describe this on LTSs. The main part of a LTS is the transition relation \( \Delta \subseteq S \times A \times S \). Equivalently, this can be defined as a function \( \Delta : S \to \mathcal{P}(A \times S) \), where \( \mathcal{P} \) is powerset. In other words, the transition relation is a coalgebra for the Set-functor \( \mathcal{P}(A \times -) \). Intuitively, this coalgebra gives the one-step behaviour of a LTS: \( S \) describes the state space of the system, \( \mathcal{P} \) describes the “branching type” as being non-deterministic, \( A \times S \) describe the “computation type” as being linear, and the function itself lists all possible futures after one-step of computation of the system. Now, changing the underlying category or the endofunctor, allows to model different types of systems. This is the usual framework of coalgebra, as described for example in [24].

In this paper, we will treat branching types and initial states as follows. As in [11], we decompose the functor \( G \), as in the case of LTSs, as \( TF \) where \( T \) is the branching type (non-deterministic, probabilistic, weighted), and \( F \) is the computation type (linear, tree-like, ...). Secondly, initial states are modelled as in [2, Sec. 3B], where we describe initial states as a pointing of the system. For example, an initial state of a LTS is the same as a function from the singleton set \( \{ \ast \} \) to the state space \( S \). More generally, a system will be modelled as a coalgebra \( \alpha : S \to GS \) as previously, together with a morphism \( i : I \to S \) from a fixed object \( I \) of \( \mathcal{C} \) describing the “type of initial states”. This object \( I \) will often be the final object of \( \mathcal{C} \), but we will see other examples later. We call them \( I \)-pointed \( G \)-coalgebras.

In this framework, it is actually simpler to model functional bisimulations, or more precisely, to describe functions that satisfy the equation (2), of the previous section. The intuition is that those functions are those that preserve the initial state, and preserve and reflect the one-step relation.

Definition 2.6. An \( I \)-pointed \( G \)-coalgebra homomorphism \( I \overset{i}{\to} S \overset{\alpha}{\to} GS \) from \( I \overset{i}{\to} S \overset{\alpha}{\to} GS \) to \( I \overset{i'}{\to} S' \overset{\alpha'}{\to} GS' \) is a morphism \( f : S \to S' \) making the right-hand diagram commute.

For instance, when \( G = \mathcal{P}(A \times -) \), one can easily see that a function \( f \) being a \( G \)-coalgebra morphism is stronger than being a functional simulation;
the preservation of transitions actually corresponds to the pointwise inequality $Gf(\alpha(s)) \subseteq \alpha'(f(s))$. To express this condition for general $G$-coalgebras, we introduce a partial order $\sqsubseteq_{X,Y}$ on each homset $\mathbb{C}(X,GY)$ in a functorial manner.

**Definition 2.7.** A partial order on $G$-homsets is a functor $\sqsubseteq : \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbb{Pos}$ such that $U \cdot \sqsubseteq = \mathbb{C}(-,G-)$; here $U : \mathbb{Pos} \rightarrow \mathbb{Set}$ is the forgetful functor from the category $\mathbb{Pos}$ of posets and monotone functions.

The functoriality of $\sqsubseteq$ amounts to that $f_1 \sqsubseteq f_2$ implies $Gh \cdot f_1 \cdot g \sqsubseteq Gh \cdot f_2 \cdot g$.

**Definition 2.8.** A coalgebra situation is given by a category $\mathbb{C}$ (of state spaces), an object $I$ of $\mathbb{C}$ (type of initial states), an endofunctor $G : \mathbb{C} \rightarrow \mathbb{C}$ (transition type) and a partial order $\sqsubseteq : \mathbb{C}^{\text{op}} \times \mathbb{C} \rightarrow \mathbb{Pos}$ on $G$-homsets.

**Example 2.9.** Set, with $\{\ast\}$, $\mathcal{P}(A \times \_)$ and the order $f \sqsubseteq g$ in $\mathcal{P}(A \times X)$ iff for every $x \in X$, $f(x) \subseteq g(x)$ is a coalgebra situation, modelling LTS.

**Definition 2.10.** An $I$-pointed lax $G$-coalgebra homomorphism $f : (S,\alpha,i) \rightarrow (S',\alpha',i')$ is a morphism $f : S \rightarrow S'$ making the right-hand diagram commute. In a coalgebra situation, the $I$-pointed $G$-coalgebras and lax homomorphisms form a category, denoted by $\mathbf{Coalg}_I(I,G)$.

### 3 The open map situation for categories of coalgebras

Fix a category $\mathbb{C}$ with pullbacks, functors $T,F : \mathbb{C} \rightarrow \mathbb{C}$, an object $I \in \mathbb{C}$ and a partial order $\sqsubseteq^T$ on $T$-homsets. They determine a coalgebra situation $(\mathbb{C},I,TF,\sqsubseteq)$ where $\sqsubseteq$ is the partial order on $TF$-homsets defined by $\sqsubseteq_{X,Y} = \sqsubseteq^T_{X,FY}$. Here, $F$ handles the linear behaviour of coalgebras, e.g. $FX = A \times X$, and $T$ models the branching behaviour, e.g. powerset construction $\mathcal{P} : \mathbb{Set} \rightarrow \mathbb{Set}$.

Under some conditions on $T$ and $F$, we construct a path-category $\mathbf{Path}(I,F+1)$ and an open map situation $\mathbf{Path}(I,F+1) \hookrightarrow \mathbf{Coalg}_I(I,TF)$ where (strict) $TF$-coalgebra homomorphisms and $\mathbf{Path}(I,F+1)$-open morphisms coincide.

#### 3.1 Precise morphisms

While the path category is intuitively clear for $FX = A \times X$, it is not for inner functors $F$ that model tree languages. For example for $FX = A + X \times X$, a $\mathcal{P}F$-coalgebra accepts binary trees with leaves labelled in $A$ as the input.

**Definition 3.1.** For a functor $F : \mathbb{C} \rightarrow \mathbb{C}$, a morphism $s : S \rightarrow FR$ is called $F$-precise if for all $f,g,h$ the following implication holds:

$$
\begin{array}{cc}
S \xrightarrow{f} FC & S \xrightarrow{f} FC \\
\downarrow Fh & \downarrow Fh \\
FR \xrightarrow{Fg} FD & FR \xrightarrow{Fd} FR \\
\end{array}
\quad &
\begin{array}{c}
S \xrightarrow{h} C \\
\downarrow Fr & \downarrow Fr \\
FR \xrightarrow{g} D & FR \xrightarrow{g} D
\end{array}
$$
Remark 3.2. If $F$ preserves weak pullbacks, then we can assume $g = \text{id}$ wlog.

Example 3.3. Intuitively speaking, for a polynomial $\text{Set}$-functor $F$, a map $s: S \to FR$ is $F$-precise iff every element of $R$ is mentioned precisely once in the definition of the map $f$. For example, for $FX = A \times X + \{\bot\}$, the case needed later for LTSs, a map $f: X \to FY$ is precise iff for every $y \in Y$, there is a unique pair $(x, a) \in X \times A$ such that $f(x) = (a, y)$. For $FX = X \times X + \{\bot\}$ on $\text{Set}$, the map $f: X \to FY$ in Figure 1 is not $F$-precise, because one element of $Y$ is used three times, and two elements do not occur in $f$ at all. However, $f': X \to FY'$ is $F$-precise because every element of $Y'$ is used precisely once in $f'$, and we have that $Fh \cdot f' = f$. Also note that $f'$ defines a forest where $X$ is the set of roots, which is closely connected to the intuition that, in the $F$-precise map $f'$, from every element of $Y'$, there is precisely one edge up to a root in $X$.

So when transforming a non-precise map into a precise map, one duplicates elements that are used multiple times and drops elements that are not used. We will cover functors $F$ for which this pattern provides $F$-precise maps:

Definition 3.4. Fix a class of objects $S \subseteq \text{obj C}$ closed under isomorphism. We say that $F$ admits precise factorizations w.r.t. $S$ if for every morphism $f: S \to FY$ with $S \in S$, there exist $Y' \in S$, $h: Y' \to Y$ and $f': S \to FY'$ $F$-precise with $Fh \cdot f' = f$.

Intuitively $S$ contains objects that have a certain choice principle (see (Ax5) later). Hence for $\mathbb{C} = \text{Set}$, $S$ contains all sets. However for the category of nominal sets, $S$ will only contain the strong nominal sets (see details in subsection 4.2).

Remark 3.5. Precise morphisms are essentially unique. If $f_1: X \to FY_1$ and $f_2: X \to FY_2$ are $F$-precise and if there is some $h: Y_1 \to Y_2$ with $Fh \cdot f_1 = f_2$, then $h$ is an isomorphism. Consequently, if $f: S \to FY$ with $S \in S$ is $F$-precise and $F$-admits precise factorizations, then $Y \in S$.

Functors admitting precise factorizations are closed under basic constructions:

Proposition 3.6. The following functors admit precise factorizations w.r.t. $S$:
1. Constant functors, if $\mathbb{C}$ has an initial object $0$ and $0 \in S$.
2. $F \cdot F'$ if $F: \mathbb{C} \to \mathbb{C}$ and $F': \mathbb{C} \to \mathbb{C}$ do so.
3. \( \prod_{i \in I} F_i \), if all \((F_i)_{i \in I}\) do so and \( S \) is closed under \( I \)-coproducts.

4. \( \prod_{i \in I} F_i \), if all \((F_i)_{i \in I}\) do so, \( C \) is \( I \)-extensive and \( S \) is closed under \( I \)-coproducts.

5. Right-adjoint functors, if and only if its left-adjoint preserves \( S \)-objects.

Example 3.7. Every polynomial \( \text{Set} \)-functor admits precise factorizations. The bag functor \( B \), where \( B(X) \) is the set of finite multisets, also admits precise factorizations (as we will see later). In contrast, \( F = \mathcal{P} \) does not admit precise factorizations, and if \( f : X \to \mathcal{P}Y \) is \( \mathcal{P} \)-precise, then \( f(x) = \emptyset \) for all \( x \in X \).

### 3.2 Path categories in pointed coalgebras

We define a path for \( I \)-pointed \( TF \)-coalgebras as a tree according to \( F \). Following the observation in Example 3.3, one layer of the tree is modelled by a \( F \)-precise morphism and hence a path in a \( TF \)-coalgebra is defined to be a finite sequence of \( (F + 1) \)-precise maps, where the \( \bot + 1 \) comes from the dead states w.r.t. \( T \); the argument is given later in Remark 3.23 when reachability is discussed. Since the \( \bot + 1 \) is not relevant yet, we define \( \text{Path}(I, F) \) in the following and will use \( \text{Path}(I, F + 1) \) later. We simplify the notation when working with families:

Notation. We write \( X_n \) for finite families \( (X_k)_{0 \leq k < n} \).

Definition 3.8. The category \( \text{Path}(I, F) \) consists of the following. An object is \((p_{n+1}, p_n)\) for an \( n \in \mathbb{N} \) with \( P_0 = I \) and \( p_n \) a family of \( F \)-precise maps \( (p_k : P_k \to FP_{k+1})_{k < n} \). We say that \((p_{n+1}, p_n)\) is a path of length \( n \). A morphism \( \phi_{n+1} : (p_{n+1}, p_n) \to (q_{m+1}, q_m) \) is a family \((\phi_k : P_k \to Q_k)_{k \leq n} \) with \( \phi_0 = \text{id}_I \) and \( q_k \cdot \phi_k = F\phi_{k+1} \cdot p_k \) for all \( 0 \leq k \leq n \).

Example 3.9. Paths for \( FX = A \times X + 1 \) and \( I \) singleton are as follows. First, a map \( f : I \to FX \) is precise iff (up-to isomorphism) either \( X = I \) and \( f(*) = (a, \bot) \) for some \( a \in A \); or \( X = \emptyset \) and \( f(*) = \bot \). Then a path is isomorphic to an object of the form: \( p_i = I \) for \( i \leq k \), \( p_i = \emptyset \) for \( i > k \), \( p_i(*) = (a_i, \bot) \) for \( i < k \), and \( p_k(*) = \bot \). A path is the same as a word, plus some “junk”, concretely, a word in \( A^* \bot^* \). For LTSs, an object in \( \text{Path}(I, F) \) with \( FX = A \times X \) is simply a word in \( A^* \).

For a more complicated functor, Figure 2 depicts a path of length 4, which is a tree for the signature with one unary, one binary symbol, and a constant. The layers of the tree are the sets \( P_k \). Also note that since every \( p_i \) is \( F \)-precise, there is precisely one route to go from every element of a \( P_k \) to the root \( \bot \).
Remark 3.10. The inductive continuation of Remark 3.5 is as follows. Given a morphism $\phi_{n+1}$ in $\text{Path}(I,F)$, since $\phi_0$ is an isomorphism, then $\phi_k$ is an isomorphism for all $0 \leq k \leq n$. If $F$ admits precise factorizations and if $I \in S$, then for every path $(P_{n+1},p_n)$, all $P_k$, $0 \leq k \leq n$, are in $S$.

Remark 3.11. If in Definition 3.4, the connecting morphism $h: Y' \to Y$ uniquely exists, then it follows by induction that the hom-sets of $\text{Path}(I,F)$ are at most singleton. This is the case for all polynomial functors, but not the case for the bag functor on sets (discussed in subsection 4.1).

Definition 3.12. The path poset $\text{PathOrd}(I,F)$ is the set $\biguplus_{0 \leq n} \mathcal{C}(I,F^n1)$ equipped with the order: for $u: I \to F^n1$ and $v: I \to F^m1$, we define $u \leq v$ if $n \leq m$ and $F^n(!) \cdot v = u$.

So $u \leq v$ if $u$ is the truncation of $v$ to $n$ levels. This matches the morphisms in $\text{Path}(I,F)$ that witnesses that one path is prefix of another:

Proposition 3.13. 1. The functor $\text{Comp}: \text{Path}(I,F) \to \text{PathOrd}(I,F)$:

$$I = P_0 \overset{p_0}{\to} FP_1 \cdots \overset{F^n P_n}{\to} F^n1$$ on objects $(P_{n+1},p_n)$

is full, and reflects isomorphisms.

2. If $F$ admits precise factorizations w.r.t. $S$ and $I \in S$, then $\text{Comp}$ is surjective on objects.

3. If additionally the connecting morphism $h$ in Definition 3.4 is unique, then $\text{Comp}$ has a right-inverse.

In particular, $\text{PathOrd}(I,F)$ is $\text{Path}(I,F)$ up to isomorphism. In the instances, it is often easier to characterize $\text{PathOrd}(I,F)$. This also shows that $\text{Path}(I,F)$ contains the elements – understood as morphisms from $I$ – of the finite start of the final chain of $F$: $1 \to F1 \overset{F1}{\to} F^21 \overset{F^21}{\to} F^31 \cdots$.

Example 3.14. When $FX = A \times X + 1$, $F^n1$ is isomorphic to the set of words in $A^* \perp^*$ of length $n$. Consequently, $\text{PathOrd}(I,F)$ is the set of words in $A^* \perp^*$, equipped with the prefix order. In this case, $\text{Comp}$ is an equivalence of categories.

3.3 Embedding paths into pointed coalgebras

The paths $(P_{n+1},p_n)$ embed into $\text{Coalg}_l(I,TF)$ as one expects it for examples like Figure 2: one takes the disjoint union of the $P_k$, one has the pointing $I = P_0$ and the linear structure of $F$ is embedded into the branching type $T$.

During the presentation of the results, we require $T$, $F$, and $I$ to have certain properties, which will be introduced one after the other. The full list of assumptions is summarized in Table 2:

(Ax1) – The main theorem will show that coalgebra homomorphisms in $\text{Coalg}_l(I,TF)$ are the open maps for the path category $\text{Path}(I,F + 1)$. So from now on, we assume that $\mathcal{C}$ has finite coproducts and to use the results from the previous sections, we fix a class $S \subseteq \text{obj} \mathcal{C}$ such that $F + 1$ admits precise
factorizations w.r.t. \( S \) and that \( I \in S \).

(Ax2) – Recall, that a family of morphisms \( (e_i : X_i \rightarrow Y)_{i \in I} \) with common

codomain is called jointly epic if for \( f, g : Y \rightarrow Z \) we have that \( f \cdot e_i = g \cdot e_i \ \forall i \in I \)

implies \( f = g \). For \( \text{Set} \), this means, that every element \( y \in Y \) is in the image

of some \( e_i \). Since we work with partial order on \( T \)-homsets, we also need the
generalization of this property if \( f \sqsubseteq g \) are of the form \( Y \rightarrow TZ' \).

(Ax3) – In this section, we encode paths as a pointed coalgebra by construct-
ing a functor \( J : \text{Path}(I, F + 1) \leftrightarrow \text{Coalg}_I(I, TF) \). For that we need to embed the
linear behaviour \( FX + 1 \) into \( TFX \). This is done by a natural transformation

\( [\eta, \bot] : \text{Id} + 1 \rightarrow T \), and we require that \( \bot : 1 \rightarrow T \) is a bottom element for \( \sqsubseteq \).

Example 3.15. For the case where \( T \) is the powerset functor \( \mathcal{P} \), \( \eta \) is given by the
unit \( \eta_X(x) = \{ x \} \), and \( \bot \) is given by empty sets \( \bot_X(\ast) = \emptyset \).

Definition 3.16. We have an inclusion functor \( J : \text{Path}(I, F + 1) \leftrightarrow \text{Coalg}_I(I, TF) \)

that maps a path \( (P_{n+1}, p_n) \) to an \( I \)-pointed \( TF \)-coalgebra on \( \coprod_{0 \leq k \leq n} P_k \).

The pointing is given by \( \eta_0 : I = P_0 \rightarrow \coprod P_{n+1} \) and the structure by:

\[
\coprod_{0 \leq k \leq n} P_k + P_n \rightarrow [\text{Fin}_{k+1} + 1 \cdot p_k]_{0 \leq k \leq n} + 1 \rightarrow F \coprod_{0 \leq k \leq n} P_k + 1 \rightarrow [\eta, \bot] TF \coprod_{0 \leq k \leq n} P_k + 1.
\]

Example 3.17. In the case of LTSs, a path, or equivalently a word \( a_1...a_k \bot...\bot \in A^* \bot^* \), is mapped to the finite linear system corresponding to \( a_1...a_k \) (see Section

2.1), seen as a coalgebra (see Section 2.2).

Proposition 3.18. Given a morphism \( [x_k]_{k \leq n} : \coprod P_{n+1} \rightarrow X \) for some system

\( (X, \xi, x_0) \) and a path \( (P_{n+1}, p_n) \), we have

\[
J(P_{n+1}, p_n) [x_k]_{k \leq n} \rightarrow (X, x_0) \iff \forall k < n : \quad P_k \xrightarrow{x_k} X
\]

a run in \( \text{Coalg}_I(I, TF) \)

\[
\text{FP}_{k+1} + 1 \rightarrow FX + 1 \rightarrow TFX.
\]

Also note that the pointing \( x_0 \) of the coalgebra is necessarily the first component

of any run in it. In a run \( [x_k]_{k \leq n} \), \( p_k \) corresponds to an edge from \( x_k \) to \( x_{k+1} \).

Example 3.19. For LTSs, since the \( P_k \) are singletons, \( x_k \) just picks the \( k \)th state

of the run. The right-hand side of this lemma describes that, this is a run iff

there is a transition from the \( k \)th state and the \( k + 1 \)th state.

3.4 Open morphisms are exactly coalgebra homomorphisms

In this section, we prove our main contribution, namely that \( \text{Path}(I, F + 1) \)-
open maps in \( \text{Coalg}_I(I, TF) \) are exactly coalgebra homomorphisms. For the first
direction of the main theorem, that is, that coalgebra homomorphisms are open,
we need two extra axioms:

(Ax4) – describing that the order on \( \mathcal{C}(X, TY) \) is essentially point-wise. This

holds for the powerset because every set is the union of its singleton subsets.

(Ax5) – describing that \( \mathcal{C}(X, TY) \) admits a choice-principle. This holds for

the powerset because whenever \( y \in h[x] \) for a map \( h : X \rightarrow Y \) and \( x \subseteq X \), then

there is some \( \{ x' \} \subseteq x \) with \( h(x') = y \).
Table 2. Main assumptions on $F, T: C \rightarrow C$, $\subseteq T$, $S \subseteq \text{obj } C$

|   |   |
|---|---|
| $F$ (Ax1) | $F + 1$ admits precise factorizations, w.r.t. $S$ and $I \in S$ |
| $T$ (Ax2) | If $(e_i: X_i \rightarrow Y)_{i \in I}$ jointly epic, then $f \cdot e_i \subseteq g \cdot e_i$ for all $i \in I \Rightarrow f \subseteq g$. |
| (Ax3) | If $(\eta, \bot): \text{Id} + 1 \rightarrow T$, with $\bot \cdot Y \cdot ! X \subseteq f$ for all $f: X \rightarrow TY$ |
| (Ax4) | For every $f: X \rightarrow TY$, $X \in S$, $f = \bigsqcup\{[\eta, \bot] \cdot f' \subseteq f \mid f': X \rightarrow Y + 1\}$ |
| (Ax5) | $\forall A \in S$ $\exists x' \xrightarrow{[\eta, \bot]_Y} (A + 1) \xrightarrow{\eta} A \xrightarrow{X} TX$ $\xrightarrow{T_{h'}} A \xrightarrow{x' \cdot [\eta, \bot]_Y} X + 1 \xrightarrow{h_{\bot}} Y + 1 \xrightarrow{\eta} TY$ |

**Theorem 3.20.** Under the assumptions of Table 2, a coalgebra homomorphism in $\text{Coalg}(I, TF)$ is $\text{Path}(I, F + 1)$-open.

The converse is not true in general, because intuitively, open maps reflect runs, and thus only reflect edges of reachable states, as we have seen in Section 2.1. The notion of a state being reached by a path is the following:

**Definition 3.21.** A system $(X, \xi, x_0)$ is path-reachable if the family of runs $[x_k]_{k \leq n}: J(P_{n+1}, p_n) \rightarrow (X, \xi, x_0)$ (of paths from $\text{Path}(I, F + 1)$) is jointly epic.

**Example 3.22.** For LTSs, this means that every state in $X$ is reached by a run, that is, there is a path from the initial state to every state of $X$.

**Remark 3.23.** In Definition 3.21, it is crucial that we consider $\text{Path}(I, F + 1)$ and not $\text{Path}(I, F)$ for functors incorporating ‘arities $\geq 2$’. This does not affect the example of LTSs, but for $I = 1$, $FX = X \times X$ and $T = \mathcal{P}$ in $\text{Set}$, the coalgebra $(X, \xi, x_0)$ on $X = \{x_0, y_1, y_2, z_1, z_2\}$ given by

$$\xi(x_0) = \{(y_1, y_2)\}, \quad \xi(y_1) = \{(z_1, z_2)\}, \quad \xi(y_2) = \{(z_1)\} = \xi(z_2) = \emptyset$$

is path-reachable for $\text{Path}(I, F + 1)$. There is no run of a length 2 path from $\text{Path}(I, F)$, because $y_2$ has no successors, and so there is no path to $z_1$ or to $z_2$.

**Theorem 3.24.** Under the assumptions of Table 2, if $(X, \xi, x_0)$ is path-reachable, then an open morphism $h: (X, \xi, x_0) \rightarrow (Y, \zeta, y_0)$ is a coalgebra homomorphism.

**3.5 Connection to other notions of reachability**

There is another concise notion for reachability in the coalgebraic literature [2].

**Definition 3.25.** A subcoalgebra of $(X, \xi, x_0)$ is a coalgebra homomorphism $h: (Y, \zeta, y_0) \rightarrow (X, \xi, x_0)$ that is carried by a monomorphism $h: X \rightarrow Y$. Furthermore $(X, \xi, x_0)$ is called reachable if it has no proper subcoalgebra, i.e. if any subcoalgebra $h$ is an isomorphism.
Under further assumptions, this notion coincides with the path-based definition of reachability (Definition 3.21). We assume for the present subsection 3.5.

**Assumption 3.26.** Let \( C \) be cocomplete, have (epi,mono)-factorizations and wide pullbacks of monomorphisms.

The first direction follows directly from Theorem 3.20:

**Proposition 3.27.** Every path-reachable \((X, \xi, x_0)\) has no proper subcoalgebra.

For the other direction it is needed that \( TF \) preserves arbitrary intersections, that is, wide pullbacks of monomorphisms. In \( \text{Set} \), this means that for a family \((X_i \subseteq Y)_{i \in I}\) of subsets we have that \( \bigcap_{i \in I} TX_i \) and \( TF \bigcap_{i \in I} X_i \) are isomorphic subsets of \( TFY \).

**Proposition 3.28.** If, furthermore, for every monomorphism \( m: Y \to Z \), the function \( C(\cdot, Tm): C(X, TY) \to C(X, TZ) \) reflects joins and if \( TF \) preserves arbitrary intersections, then a reachable coalgebra \((X, \xi, x_0)\) is also path-reachable.

All those technical assumptions are satisfied in the case of LTSs, and will also be satisfied in all our instances in the section 4.

### 3.6 Trace semantics for pointed coalgebras

The characterization from Theorem 3.20 and Theorem 3.24 points out a natural way of defining a trace semantics for pointed coalgebras. Indeed, the paths category \( \text{Path}(I, F + 1) \) provides a natural way of defining the runs of a system. A possible way to go from runs to trace semantics is to describe accepting runs as the subcategory \( J': \text{Path}(I, F) \hookrightarrow \text{Path}(I, F + 1) \). We can define the trace semantics of a system \((X, \xi, x_0)\) as the set:

\[
\text{tr}(X, \xi, x_0) = \{ \text{Comp}(P_{n+1}, p_n) \mid \exists \text{ run } [x_k]_k \leq n : JJ'(P_{n+1}, p_n) \to (X, \xi, x_0) \text{ with } (P_{n+1}, p_n) \in \text{Path}(I, F) \}
\]

Since \( \text{Path}(I, F) \)-open maps preserve and reflect runs, we have the following:

**Corollary 3.29.** \( \text{tr}: \text{Coalg}(I, TF) \to (\mathcal{P}(\text{PathOrd}(I, F)), \subseteq) \) is a functor and if \( f: (X, \xi, x_0) \to (Y, \zeta, y_0) \) is \( \text{Path}(I, F + 1) \)-open, then \( \text{tr}(X, \xi, x_0) = \text{tr}(Y, \zeta, y_0) \).

Let us look at two LTS-related examples (we will describe some others in the next section). First, for \( FX = A \times X \). The usual trace semantics is given by all the words in \( A^* \) that are labelled of a run of a system. This trace semantics is obtained because \( \text{PathOrd}(I, F) = \bigsqcup_{n \geq 0} A^n \) and because \( \text{Comp} \) maps every path to its underlying word. Another example is given for \( FX = A \times X + \{ \checkmark \} \), where \( \checkmark \) stands for a final state. In this case, a path in \( \text{Path}(I, F) \) of length \( n \) is either a path that can still be extended or encodes less than \( n \) steps to an accepting state \( \checkmark \). This obtains the trace semantics containing the set of accepted words, as in automata theory, plus the set of possibly infinite runs.
4 Instances

4.1 Analytic functors and variety of tree automata

In Section 3.1, Proposition 3.6, we have seen that when the category $C$ has enough well-behaved coproducts, then (Ax1) is automatically satisfied for all polynomial functors. That is the case when $C = S = \text{Set}$, which already allows us to consider many models like LTSs, or top-down non-deterministic tree automata (as we will see soon). However, in $\text{Set}$, we can do much better by proving that (Ax1) is satisfied for a larger class of functors, namely the analytic functors:

**Definition 4.1** [12,13]. An analytic $\text{Set}$-functor is a functor of the form $FX = \prod_{\sigma/n \in \Sigma} X^n / G_{\sigma}$ where $\Sigma$ is a signature, and for every $\sigma/n \in \Sigma$ we have a subgroup $G_{\sigma}$ of the permutation group $\mathfrak{S}_n$ on $\text{ar}(\sigma) = n$ elements.

**Example 4.2.** Every polynomial functor is analytic. The bag-functor is analytic, where $\Sigma = \mathbb{N}$ has one operation symbol per arity and $G_{\sigma} = \mathfrak{S}_{\text{ar}(\sigma)}$ is the full permutation group on $\text{ar}(\sigma)$ elements. It is the archetype of an analytic functor, in the sense that for every analytic functor $F: \text{Set} \to \text{Set}$, there is a natural transformation into the bag functor $\alpha: F \to B$. If $F$ is given by $\Sigma$ and $G_{\sigma}$ as above, then $\alpha_X$ is given by

$$FX = \prod_{\sigma/n \in \Sigma} X^n / G_{\sigma} \to \prod_{\sigma/n \in \Sigma} X^n / \mathfrak{S}_n \to \prod_{n \in \mathbb{N}} X^n / \mathfrak{S}_n = BX.$$  

**Proposition 4.3.** For an analytic $\text{Set}$-functor $F$, a map $f: X \to FY$ is $F$-precise iff $\alpha_Y \cdot f$ is $B$-precise iff every element of $Y$ appears precisely once in the definition of $f$. So every analytic functor has precise factorisations w.r.t. $\text{Set}$.

Much as the case of LTS (with possibly final states) from Section 3.6, we can model tree automata with pointed coalgebras. More precisely, we can model top-down non-deterministic bounded tree automata as in [5]. Given a signature $\Sigma$, such a tree automata is the same as a $\{\ast\}$-pointed $\mathcal{P}(\prod_{\sigma/n \in \Sigma} X^n)$-coalgebra. Since $FX = \prod_{\sigma/n \in \Sigma} X^n$ is a polynomial functor, we can use our machinery to study tree automata. In particular, we can recover the language of this automata as a trace semantics: elements of $\text{Path}(\{\ast\}, F+1)$ are essentially the same as multicontexes, that is trees with possibly several holes, plus some junk. Considering $\text{Path}(I, F)$ its full subcategory of such paths with holes only on the last level, and $\text{PathOrd}(I, F)$ the category of multicontexes ordered by prefix, we obtain that the trace semantics of a system is exactly finite truncations of its tree language. We can also model many kinds of symbols: (1) bounded symbols: the ones we have seen so far, given with an arity $n$, and their component in the functor $F$ is given by $X^n$, (2) unbounded symbols: those can have any number of children, so their component is given by $\prod_{n \in \mathbb{N}} X^n$, (3) commutative bounded symbols: they have a fixed number of children, but those are unordered. Their component is given by $X^n / \mathfrak{S}_n$, (4) commutative unbounded symbols: combination of the last two. Their component is given by the bag functor.

We can imagine many other types (cyclic, alternate, ...), using the proper subgroup of the the permutation group.
4.2 Nominal Sets: Regular Nondeterministic Nominal Automata

We derive an open map situation from the coalgebraic situation for regular nondeterministic nominal automata (RNNAs) [25]. They are an extension of automata to accept words with binders, consisting of literals $a \in \mathbb{A}$ and binders $|a|$ for $a \in \mathbb{A}$; the latter is counted as length 1. An example of such a word of length 4 is $a|c|b|c$, where the last $c$ is bound by $|c|$. The order of binders makes difference: $|a|c|b|a \neq |a|b|a$. RNNAs are coalgebraically represented in the category of nominal sets [9], a formalism about atoms (e.g. variables) that sit in more complex structures (e.g. lambda terms), and gives a notion of binding. Because the choice principle (Ax4) and (Ax5) are not satisfied by every nominal sets, we instead use the class of strong nominal sets for the precise factorization (Definition 3.4).

**Definition 4.4** [9,23]. Fix a countably infinite set $\mathbb{A}$, called the set of atoms. For the group $\mathcal{S}_f(\mathbb{A})$ of finite permutations on the set $\mathbb{A}$, a group action $(X, \cdot)$ is a set $X$ together with a group homomorphism $\cdot : \mathcal{S}_f(\mathbb{A}) \rightarrow \mathcal{S}_f(X)$, written in infix notation. An element $x \in X$ is supported by $S \subseteq \mathbb{A}$, if for all $\pi \in \mathcal{S}_f(\mathbb{A})$ with $\pi(a) = a \forall a \in S$ we have $\pi \cdot x = x$. A nominal set has the group action for $\mathcal{S}_f(\mathbb{A})$ such that every $x \in X$ is finitely supported, i.e. supported by a finite $S \subseteq \mathbb{A}$. A map $f : (X, \cdot) \rightarrow (Y, \ast)$ is equivariant if for all $x \in X$ and $\pi \in \mathcal{S}_f(\mathbb{A})$ we have $f(\pi \cdot x) = \pi \ast f(x)$. The category of nominal sets and equivariant maps is denoted by $\text{Nom}$. A nominal set $(X, \cdot)$ is called strong if for all $x \in X$ and $\pi \in \mathcal{S}_f(\mathbb{A})$ with $\pi \cdot x = x$ we have $\pi(a) = a$ for all $a \in \text{supp}(x)$.

Intuitively, the support of an element is the set of free literals. An equivariant map can forget some of the support of an element, but can never introduce new atoms, i.e. $\text{supp}(f(x)) \subseteq \text{supp}(x)$. The intuition behind strong nominal sets is that all atoms appear in a fixed order, that is, $\mathbb{A}^n$ is strong, but $\mathcal{P}_f(\mathbb{A})$ (the finite powerset) is not. We set $S$ to be the class of strong nominal sets. In the application, we fix the set $I = \mathbb{A}^*^n$ of distinct $n$-tuples of atoms ($n \geq 0$) as the pointing. The hom-sets $\text{Nom}(X, \mathcal{P}_f Y)$ are ordered point-wise.

**Proposition 4.5.** Uniformly finitely supported powerset $\mathcal{P}_{ufs}(X) = \{Y \subseteq X \mid \bigcup_{y \in Y} \text{supp}(y) \text{ finite} \}$ satisfies (Ax2-5) w.r.t. $S$ the class of strong nominal sets.\(^5\)

As for $F$, we study a LTS-like functor, extended with the binding functor [9]:

**Definition 4.6.** For a nominal set $X$, define the $\alpha$-equivalence relation $\sim_{\alpha}$ on $\mathbb{A} \times X$ by: $(a, x) \sim_{\alpha} (b, y) \iff \exists c \in \mathbb{A} \backslash \text{supp}(x) \backslash \text{supp}(y)$ with $(a c) \cdot x = (b c) \cdot y$. Denote the quotient by $[\mathbb{A}]X := \mathbb{A} \times X / \sim_{\alpha}$. The assignment $X \mapsto [\mathbb{A}]X$ extends to a functor, called the binding functor $[\mathbb{A}]$: $\text{Nom} \rightarrow \text{Nom}$.

RNNAs are precisely $\mathcal{P}_{ufs}F$-coalgebras for $FX = \{\check{\bullet}\} + [\mathbb{A}]X + \mathbb{A} \times X$ [25]. In this paper we additionally consider initial states for RNNAs.

\(^5\)There are two variants of powersets discussed in [25]. The finite powerset $\mathcal{P}_f$ also fulfills the axioms. However, finitely supported powerset $\mathcal{P}_{ufs}$ does not fulfill (Ax5).
Proposition 4.7. The binding functor $[A]$ admits precise factorizations w.r.t. strong nominal sets and so does $FX = \{\checkmark\} + [A]X + A \times X$.

An element in $\text{PathOrd}(A^n, F)$ may be regarded as a word with binders under a context $a \vdash w$, where $a \in A^n$, all literals in $w$ are bound or in $a$, and $w$ may end with $\checkmark$. Moreover, two word-in-contexts $a \vdash w$ and $a' \vdash w'$ are identified if their closures are $\alpha$-equivalent, that is, $|a_1 \cdots a_n w = |a'_1 \cdots a'_n w'$. The trace semantics of a RNNA $T$ contains all the word-in-contexts corresponding to runs in $T$. This trace semantics distinguishes whether words are concluded by $\checkmark$.

4.3 Subsuming arbitrary open morphism situations

Lasota [18] provides a translation of a small path-category $P \hookrightarrow M$ into a functor $F: \text{Set}^{\text{obj} P} \rightarrow \text{Set}^{\text{obj} P}$ defined by $F(X_P)_P = (\prod_{Q \in P} (P(X_Q))^{P(P,Q)})_{P \in P}$. So the hom-sets $\text{Set}^{\text{obj} P}(X, FY)$ have a canonical order, namely the point-wise inclusion. This admits a functor $\text{Beh}$ from $M$ to $F$-coalgebras and lax coalgebra homomorphisms, and Lasota shows that $f \in M(X,Y)$ is $P$-open iff $\text{Beh}(f)$ is a coalgebra homomorphism. In the following, we show that we can apply our framework to $F$ by a suitable decomposition $F = TF$ and a suitable object $I$ for the initial state pointing. As usual in open map papers, we require that $P$ and $\mathbb{M}$ have a common initial object $0_P$. Observe that we have $F = T \cdot F$ where

$$T(X_P)_{P \in P} = (P(X_P))_{P \in P} \quad \text{and} \quad F(X_P)_{P \in P} = (\prod_{Q \in P} P(P,Q) \times X_Q)_{P \in P}.$$ 

Lasota considers coalgebras without pointing, but one indeed has a canonical pointing as follows. For $P \in P$, define the characteristic family $\chi^P \in \text{Set}^{\text{obj} P}$ by $\chi^P_Q = 1$ if $P = Q$ and $\chi^P_Q = \emptyset$ if $P \neq Q$. With this, we fix the pointing $I = \chi^0_P$.

Proposition 4.8. $T, F$ and $I$ satisfy the axioms from Table 2, with $S = \text{Set}^{\text{obj} P}$.

The path category in $\text{Coalg}(I, TF)$ from our theory can be described as follows.

Proposition 4.9. An object of $\text{Path}(I, F)$ is a sequence of composable $P$-morphisms $0_P \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots \rightarrow P_n$. 

5 Further work

We proved that coalgebra homomorphisms for systems with non-deterministic branching can be seen as open maps for a canonical path-category, constructed from the computation type $F$. This limitation to non-deterministic systems is unsurprising: as we have proved in Section 4.3 on Lasota’s work [18], every open map situation can be encoded as a coalgebra situation with a powerset-like functor, so with non-deterministic branching. As a future work, we would like to extend this theory of path-category for categories of coalgebras for further kinds of branchings, especially probabilistic and weighted. This will require (1) to adapt open maps to allow those kinds of branchings (2) adapt the axioms from Table 2, by replacing the “+1” part of (Ax1), which is the linear subfunctor of a powerset-like functor, to something depending on the branching type.
References

1. Ad´ amek, J., Herrlich, H., Strecker, G.E.: Abstract and concrete categories: The joy of cats. online and enhanced edition of the book published in 1990 by John Wiley and Sons (http://katmat.math.uni-bremen.de/acc/acc.pdf) (2004)
2. Ad´ amek, J., Milius, S., Moss, L.S., Sousa, L.: Well-pointed coalgebras. Logical Methods in Computer Science 9(3), 1–51 (2013)
3. Awodey, S.: Category Theory. Oxford University Press, Inc., New York, NY, USA, 2nd edn. (2010)
4. Bonchi, F., Silva, A., Sokolova, A.: The Power of Convex Algebras. In: Meyer, R., Nestmann, U. (eds.) 28th International Conference on Concurrency Theory (CONCUR 2017). vol. 85, pp. 23:1–23:18. Dagstuhl, Germany (2017). https://doi.org/10.4230/LIPIcs.CONCUR.2017.23
5. Comon, H., Dauchet, M., Gilleron, R., Jacqemard, F., Lugiez, D., Löding, C., Tison, S., Tommasi, M.: Tree Automata Techniques and Applications. Online http://tata.gforge.inria.fr (2007)
6. Dubut, J., Goubault, E., Goubault-Larrecq, J.: Natural homology. In: Proceedings of the 42nd International Colloquium on Automata, Languages and Programming (ICALP’15) – Part II. Lecture Notes in Computer Science, vol. 9135, pp. 171–183. Springer (July 2015)
7. Fahrenberg, U., Legay, A.: History-Preserving Bisimilarity for Higher-Dimensional Automata via Open Maps. Electronic Notes in Theoretical Computer Science 298, 165–178 (November 2013)
8. Fiore, M.P., Cattani, G.L., Winskel, G.: Weak bisimulation and open maps. In: 14th Annual IEEE Symposium on Logic in Computer Science (LICS’99). pp. 67–76 (1999)
9. Gabbay, M., Pitts, A.M.: A new approach to abstract syntax involving binders. In: Longo, G. (ed.) Proceedings of the Fourteenth Annual IEEE Symp. on Logic in Computer Science, LICS 1999. pp. 214–224. IEEE Computer Society Press (July 1999)
10. Hansen, H.H., Klin, B.: Pointwise extensions of GSOS-defined operations. Math. Struct. Comp. Sci. 21(1), 321–361 (2011)
11. Hasuo, I., Jacobs, B., Sokolova, A.: Generic trace semantics via coinduction. Logical Methods in Computer Science 3(4) (2007)
12. Joyal, A.: Une th´ eorie combinatoire des s´ eries formel les. Advances in Mathematics 42(1), 1–82 (1981)
13. Joyal, A.: Foncteurs analytiques et esp` eces de structures. In: Proceedings of the “Colloque de combinatoire ´ enum´ erative”. pp. 126–159. Lecture Notes in Mathematics, Springer-Verlag (1986)
14. Joyal, A., Nielsen, M., Winskel, G.: Bisimulation from Open Maps. Information and Computation 127, 164–185 (1996)
15. Kozen, D., Mamouras, K., Petrisan, D., Silva, A.: Nominal kleene coalgebra. In: Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II. pp. 286–298 (2015). https://doi.org/10.1007/978-3-662-47666-6_23, https://doi.org/10.1007/978-3-662-47666-6_23
16. Kurz, A., Petrisan, D., Severi, P., de Vries, F.: Nominal coalgebraic data types with applications to lambda calculus. Logical Methods in Computer Science 9(4) (2013). https://doi.org/10.2168/LMCS-9(4:20)2013, https://doi.org/10.2168/LMCS-9(4:20)2013
17. Kurz, A., Petrisan, D., Velebil, J.: Algebraic theories over nominal sets. CoRR abs/1006.3027 (2010)
18. Lasota, S.: Coalgebra morphisms subsume open maps. Theoretical Computer Science 280(1), 123 – 135 (2002)
19. Milius, S.: A sound and complete calculus for finite stream circuits. In: Proc. 25th Annual Symposium on Logic in Computer Science (LICS’10). pp. 449–458 (2010)
20. Milius, S., Schröder, L., Wißmann, T.: Regular behaviours with names. Applied Categorical Structures 24(5), 663–701 (2016)
21. Nielsen, M., Hune, T.: Bisimulation and Open Maps for Timed Transition Systems. Fundamenta Informaticae 38, 61–77 (1999)
22. Park, D.: Concurrency and Automata on Infinite Sequences. Theoretical Computer Science 104, 167–183 (1981)
23. Pitts, A.M.: Nominal Sets: Names and Symmetry in Computer Science, Cambridge Tracts in Theoretical Computer Science, vol. 57. Cambridge University Press (2013)
24. Rutten, J.: Universal coalgebra: a theory of systems. Theoretical Computer Science 249(1), 3 – 80 (2000)
25. Schröder, L., Kozen, D., Milius, S., Wißmann, T.: Nominal automata with name binding. In: Esparza, J., Murawski, A. (eds.) Proc. 20th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2017). Lecture Notes in Computer Science, vol. 10203, pp. 124–142. Springer (2017)
A Omitted Proofs

Proof of Remark 3.2

Lemma A.1. Assuming that $F$ preserves weak pullbacks, a morphism $s: S \to FR$ is $F$-precise iff for all $f, h$ the following implication holds:

$$
\begin{array}{c}
S \xrightarrow{f} FC \\
\downarrow_{s} & \exists d \\
FR & \xrightarrow{Fd} FR \\
\end{array}
\quad
\begin{array}{c}
S \xrightarrow{f} FC \\
\downarrow_{s} & \exists d \\
FR & \xrightarrow{Fd} FR \\
\end{array}
\text{ & } h \cdot d = \text{id}_R
$$

Proof. Sufficiency is clear, because one can fix $D = R$ and $g = \text{id}_R$ in the definition of $F$-precise. For necessity, consider $s, f, g, h$ as in the definition of $F$-precise. The pullback of $g$ along $h$ is weakly preserved by $F$, and so we have the following commuting diagram:

$$
\begin{array}{c}
S \xrightarrow{f} FP \\
\downarrow_{s} & \exists f' \\
FR & \xrightarrow{F\pi_1} FC \\
\end{array}
\quad
\begin{array}{c}
S \xrightarrow{f} FC \\
\downarrow_{s} & \exists d \\
FR & \xrightarrow{Fd} FR \\
\end{array}
\quad
\begin{array}{c}
FP \xrightarrow{F\pi_2} FC \\
\downarrow_{Fg} & \exists d' \\
FD \\
\end{array}
$$

Hence, $s$ induces some $d: R \to P$ with $Fd \cdot s = f'$ and $\pi_2 \cdot d = \text{id}_R$. The witness that $s$ is $F$-precise is $\pi_1 \cdot d: R \to C$, because $F(\pi_1 \cdot d) \cdot s = f$ and $h \cdot \pi_1 \cdot d = g$. \qed

Proof of Remark 3.5

Apply that $f_2$ is $F$-precise and one obtains a lifting $d: Y_2 \to Y_1$ with $Fd \cdot f_2 = f_1$ and $h \cdot d = \text{id}_{Y_2}$, i.e. $d$ has a left inverse, $h$. Additionally, since $f_1$ is $F$-precise, $Fd \cdot f_2 = f_1$ induces some $d': Y_1 \to Y_2$ with $f_2 = Fd' \cdot f_1$ and $d' \cdot d = \text{id}_{Y_1}$. Since $d$ has both a left and right inverse, it is an isomorphism, and so is its left-inverse $h$.

For the second part, we have a precise factorisation of $f$, that is, a $F$-precise morphism $f': S \to Y'$ with $Y' \in S$, and a morphism $h: Y' \to Y$ such that $f = Fh \cdot f'$. By the previous point, $h$ is an isomorphism, and since $S$ is closed under isomorphisms, $Y \in S$.

Proof of Proposition 3.6

1. Fix $Z$ an object of $C$, and assume given a morphism $f: S \to Z = C_Z Y$, with $S \in S$. Since a morphism of the form $f': S \to F0$ is always $F$-precise, then $f' = f: S \to Z = C_Z 0$ is $C_Z$-precise. Taking $h$ as the unique morphism from $0$ to $Y$, $Fh \cdot f' = \text{id}_Z \cdot f = f$.
2. Let us start with the following lemma:
Lemma A.2. If \( f : S \rightarrow FR \) is \( F \)-precise and \( f' : R \rightarrow F'Q \) is \( F' \)-precise, then \( Ff' \cdot f \) is \( F \cdot F' \)-precise.

Proof. Assume given the following situation:

\[
\begin{array}{ccc}
S & \xrightarrow{k} & FF'C \\
\downarrow{f} & & \downarrow{FF'u} \\
FR & & FF'Q \\
\downarrow{Ff'} & & \downarrow{FF'v} \\
FF'Q & \xrightarrow{FF'v} & FF'W
\end{array}
\]

In particular, we have \( (FF'v \cdot Ff') \cdot f = FF'u \cdot k \). Since \( f \) is \( F \)-precise, there is \( d : R \rightarrow F'C \) such that

\[
k = Fd \cdot f \]

\[
F'v \cdot f' = F'u \cdot d
\]

Since \( f' \) is \( F' \)-precise, there is \( d' : Q \rightarrow C \) such that

\[
d = F'd' \cdot f'
\]

\[
v = u \cdot d'
\]

This implies that \( k = FF'd' \cdot (Ff' \cdot f) \).

\[\square\]

Let us prove this point now. We start with a morphism \( f : S \rightarrow FF'Y \), with \( S \in S \). Since \( F \) admits precise factorisations w.r.t. \( S \), there are \( f' : S \rightarrow FY' \) \( F \)-precise and \( h : Y' \rightarrow F'Y \) with \( Y' \in S \), and \( Fh \cdot f' = f \). Now, since \( F' \) admits precise factorisations w.r.t. \( S \), there are \( f'' : Y' \rightarrow F''Y'' \) \( F' \)-precise and \( h' : Y'' \rightarrow Y \) such that \( F'h' \cdot f'' = h \). Consequently, \( f = FF'h' \cdot (Ff'' \cdot f') \) and \( Ff'' \cdot f' \) is \( FF' \)-precise by the previous lemma.

3. Let us start with the following lemma:

Lemma A.3. For a family of functors \( F_i : C \rightarrow C \), \( i \in I \), and \( F_i \)-precise morphisms \( f_i : X \rightarrow F_iY_i \), \( g : X \rightarrow GY_G \), then we have a \( \prod_{i \in I} F_i \)-precise morphism

\[
(F_i \text{in}_i \cdot f_i) : X \rightarrow \prod_{i \in I} F_i(\prod_{j \in I} Y_j).
\]

Proof. Consider a square

\[
\begin{array}{ccc}
X & \xrightarrow{(v_i)_{i \in I}} & \prod_{i \in I} F_iW \\
\downarrow{(f_i)_{i \in I}} & & \downarrow{[F_iu]_{i \in I}} \\
\prod_{i \in I} F_iY_i & \xrightarrow{[F_ih]_{i \in I}} & \prod_{i \in I} F_iZ \\
\downarrow{[F_i\text{in}_i]_{i \in I}} & & \downarrow{F_ih} \\
\prod_{i \in I} F_i(\prod_{j \in I} Y_j) & \xrightarrow{F_i \text{in}_i} & F_iZ
\end{array}
\]
Since \( f_i \) is \( F_i \)-precise, we obtain some \( d_i : Y_i \to W \) with \( F_id_i \cdot f_i = v_i \) and \( u \cdot d_i = h \cdot \text{in}_i \). We have \( [d_j]_{j \in I} : \prod_{j \in I} Y_j \to W \) with
\[
(\prod_{i \in I} F_i[d_j]_{j \in I}) \cdot (F_i\text{in}_i \cdot f_i)_{i \in I} = (Fd_i \cdot f_i)_{i \in I} = (v_i)_{i \in I}
\]
and
\[
u \cdot [d_i]_{i \in I} = [u \cdot d_i]_{i \in I} = [h \cdot \text{in}_i]_{i \in I} = h.
\]

Let us prove this point now. Consider \( \langle f_i \rangle_{i \in I} : X \to \prod_{i \in I} F_i Y \) and consider the \( F_i \)-precise factorizations:

\[
\begin{array}{ccc}
X & \xrightarrow{f_i} & F_i Y \\
\downarrow & & \uparrow_{F_i h_i} \\
F_i Y_i & & \\
\end{array}
\]

By Lemma A.3 we have that \( (F_i\text{in}_i \cdot f_i') : X \to \prod_{i \in I} F_i(\prod_{j \in I} Y_j) \) is \( \prod_{i \in I} F_i \)-precise and it is the \( \prod_{i \in I} F_i \)-precise factorization of \( \langle f_i \rangle_{i \in I} \):

\[
(\prod_{i \in I} F_i)[h_i]_{i \in I} \cdot (F_i\text{in}_i \cdot f_i')_{i \in I} = (F_i h_i \cdot f_i')_{i \in I} = (f_i)_{i \in I}.
\]

4. By \( I \)-extensive we mean ‘extensive’ if \( I \) is finite and ‘infinitary extensive’ if \( I \) is infinite. In any case, \( \mathbb{C} \) is \( I \)-extensive if for for all families \( (X_i)_{i \in I}, (Y_i)_{i \in I}, (g_i)_{i \in I}, (x_i)_{i \in I} \) and \( h \) with

\[
\begin{array}{ccc}
X_i & \xrightarrow{x_i} & P \\
\downarrow g_i & & \downarrow h \\
Y_i & \xrightarrow{\text{in}_i} & \prod_{j \in I} Y_j \\
\end{array}
\]

the following equivalence holds:

\[
(x_i : X_i \to P)_{i \in I} \text{ is a coproduct} \iff \forall i \in I \text{ the above square is a pullback}
\]

Let us start with the following lemma:

**Lemma A.4.** Given functors \( F_i : \mathbb{C} \to \mathbb{C} \), \( i \in I \), on an \( I \)-extensive category, and morphisms \( f_i : A_i \to F_i X_i \), \( i \in I \). Then \( \prod_{i \in I} (F_i\text{in}_i \cdot f_i) : \prod_{i \in I} A_i \to (\prod_{i \in I} F_i)(\prod_{i \in I} X_i) \) is \( \prod_{i \in I} F_i \)-precise if \( f_i \) is \( F_i \)-precise for every \( i \in I \).
Proof. Consider a commutative square

\[
\begin{array}{c}
\prod_{i \in I} A_i \\ \Downarrow \prod_{i \in I} f_i \\
\prod_{i \in I}(F_i X_i) \\ \Downarrow \prod_{i \in I} F_i \text{in}_i \\
(\prod_{i \in I} F_i)(\prod_{i \in I} X_i) \\
\Downarrow \\
\prod_{i \in I} F_i(\prod_{j \in I} X_j)
\end{array}
\]

Note that by the extensivity of \( \mathcal{C} \) we can assume that the top morphism \( \prod_{i \in I} A_i \rightarrow \prod_{i \in I} F_i C \) is indeed a coproduct of morphisms. Hence we have \( F_i g_i \cdot f_i = F_i c \cdot a_i \), for every \( i \in I \). Since \( f_i \) is \( F_i \)-precise, this induces some \( d_i : X_i \rightarrow C \) with \( F_i d_i \cdot f_i = a_i \) and \( c \cdot g_i = d_i \). In total, we have \( [d_i]_{i \in I} : \prod_{i \in I} X_i \rightarrow C \) with \( c \cdot [d_i]_{i \in I} = [g_i]_{i \in I} \) and

\[
(\prod_{i \in I} F_i)[d_i]_{i \in I} \cdot (\prod_{i \in I} (F_i \text{in}_i \cdot f_i)) = (\prod_{i \in I} (F_i d_i \cdot f_i)) = \prod_{i \in I} a_i
\]

Let us prove this point now. Given a morphism \( h : X \rightarrow (\prod_{i \in I} F_i)(Y) \), construct the pullbacks in the \( I \)-extensive category \( \mathcal{C} \):

\[
\begin{array}{ccc}
X_i & \rightarrow & X \\
\Downarrow f_i & & \Downarrow h \\
F_i Y & \rightarrow & (\prod_{i \in I} F_i)(Y)
\end{array}
\]

So we have \( X = \prod_{i \in I} X_i \) with the coproduct injections as in the top row of the above pullback diagram, and in particular \( h = [\text{in}_i \cdot f_i]_{i \in I} = \prod_{i \in I} f_i \). Let \( f_i' \) be the \( F_i \)-precise morphisms through which \( f_i \) factors for every \( i \in I \):

\[
\forall i \in I \quad \begin{array}{c}
X_i \rightarrow F Y_i \\
\Downarrow f_i \\
\Downarrow F y_i
\end{array}
\]

\[
\prod_{i \in I} F \text{in}_i : f_i' \Rightarrow [\text{in}_i \cdot f_i]_{i \in I} = \prod_{i \in I} f_i
\]

By Lemma A.4, \( \prod_{i \in I} F \text{in}_i \cdot f_i' \) is \( \prod_{i \in I} F_i \)-precise.

5. We first show that \( \eta_X : X \rightarrow R(LX) \) is \( R \)-precise for every \( X \in \mathcal{C} \). Since right adjoints preserve limits, \( R \) preserves (weak) pullbacks and it suffices
to check Lemma A.1/Remark 3.2. For any suitable commutative triangle we have

\[
\begin{array}{c}
X \xrightarrow{g} RZ \\
\eta_X \downarrow \nearrow Rh \\
RLX
\end{array}
\]

Since \( L \dashv R \), there exists a unique \( g' \) with

\[
\begin{array}{c}
X \xrightarrow{g} RZ \\
\eta_X \downarrow \nearrow Rg' \\
RLX
\end{array}
\quad \text{and} \quad
\begin{array}{c}
LX \xrightarrow{g'} Z \\
\text{id}_{LX} \downarrow \nearrow h \\
LX
\end{array}
\]

by the universal mapping property of \( L \dashv R \) and by the naturality of the isomorphism \( \mathbb{C}(LX, Z) \cong \mathbb{C}(X, RZ) \) respectively. Now, we can prove this point:

- For necessity and \( X \in \mathcal{S} \), we have that \( \eta_X : X \to RLX \) is \( R \)-precise by the previous argument, and hence \( LX \in \mathcal{S} \) by Remark 3.5.

- For sufficiency, consider a morphism \( f : X \to RY \). The adjunction induces a unique \( f' : LX \to Y \) with \( Rf' \cdot \eta_X = f \), where \( \eta_X : X \to RLX \) is \( R \)-precise as shown above. Since \( L \) preserves objects in \( \mathcal{S} \), we have that \( LX \in \mathcal{S} \).

**Proof of Example 3.7**

Given an \( \mathcal{P} \)-precise \( f : X \to \mathcal{P}Y \), define \( f' : X \to \mathcal{P}(Y + Y) \) by \( f'(x) = \{\text{inl}(y), \text{inr}(y) \mid y \in f(x)\} \), and so \( \mathcal{P}[\text{id}_Y, \text{id}_Y] \cdot f' = f \). Hence, we have some \( d : Y \to Y + Y \) with \( [\text{id}_Y, \text{id}_Y] \cdot d = \text{id}_Y \) and \( f' = \mathcal{P}d \cdot f \). The first equation implies that for every \( y \in Y \), \( d(y) \) is \( \text{inl}(y) \) or \( \text{inr}(y) \). So for \( x \in X \) and for every \( y \in Y \), either \( \text{inl}(y) \) or \( \text{inr}(y) \) is in \( d(f(x)) = (\mathcal{P}d \cdot f)(x) = f'(x) \), which is a contradiction unless \( X \) or \( Y \) is empty. Hence, \( f(x) = \emptyset \) for all \( x \in X \).

**Proof of Remark 3.10**

The proof is by induction on the length. Since \( P_0 = I = Q_0, \phi_0 : P_0 \to Q_0 \) is an isomorphism. Assume that \( \phi_k : P_k \to Q_k \) is isomorphic for \( k \leq n \). Since \( q_k \cdot \phi_k \) and \( p_k \) are \( F + 1 \)-precise, we have that \( \phi_{k+1} \) is isomorphic by Remark 3.5.

The second part of the remark is a consequence of the second part of Remark 3.5.

**Proof of Proposition 3.13**

1. **Functoriality**, consider a path morphism \( \phi_{n+1} : (P_{n+1}, p_n) \to (Q_{m+1}, q_m) \) and use that all components are isos (Remark 3.5)

\[
\begin{array}{c}
I = Q_0 \xrightarrow{q_0} FQ_1 \to \cdots \to F^nQ_n \to \cdots \to F^nQ_m \xrightarrow{F^{n+1}} F^{n+1} \\
\| \Downarrow F_{\phi_{n+1}}^{-1} \| \Downarrow F^n_{\phi_{n+1}}^{-1} \\
I = P_0 \xrightarrow{p_0} FP_1 \to \cdots \to F^nP_n \xrightarrow{F^n_{\phi_{n+1}}^{-1}} F^n
\end{array}
\]
Hence $\text{Comp}(P_{n+1}, p_n) \leq \text{Comp}(Q_{m+1}, q_m)$.

- **Fullness.** Since $\text{PathOrd}(I, F)$ is partially ordered, it suffices to show that whenever $\text{PathOrd}(I, F)(\text{Comp}(P_{n+1}, p_n), \text{Comp}(Q_{m+1}, q_m))$ is non-empty, then so is $\text{Path}(I, F)(P_{n+1}, p_n); (Q_{m+1}, q_m)$. We have $n \leq m$ and construct a morphism $\phi_{n+1}: (P_{n+1}, p_n) \rightarrow (Q_{m+1}, q_m)$ by induction. There is nothing to do in the base case $\phi_0 := \text{id}_I$. For the step, assume some $\phi_k: P_k \rightarrow Q_k$ making the following diagram commute:

$$
\begin{array}{ccc}
P_k & \xrightarrow{\phi_k} & Q_k \\
p_k \downarrow & & \downarrow q_k \\
FP_{k+1} & \xrightarrow{\phi_{k+1}} & FQ_{k+1}
\end{array}
\Rightarrow
\begin{array}{ccc}
P_k & \xrightarrow{\phi_k} & Q_k \\
FP_{k+1} & \xrightarrow{\phi_{k+1}} & FQ_{k+1}
\end{array}
\Rightarrow
\begin{array}{ccc}
P_{n-k} & \xrightarrow{\phi_{n-k}} & Q_{n-k} \\
P_{n-k} & \xrightarrow{\phi_{n-k}} & Q_{n-k}
\end{array}
\Rightarrow
\begin{array}{ccc}
P_{n-k+1} & \xrightarrow{\phi_{n-k+1}} & Q_{n-k+1}
\end{array}
$$

Since $p_k$ is $F$-precise, we obtain a morphism $\phi_{k+1}: P_{k+1} \rightarrow Q_{k+1}$ with

$$
\begin{array}{ccc}
P_k & \xrightarrow{\phi_k} & Q_k \\
FP_{k+1} & \xrightarrow{\phi_{k+1}} & FQ_{k+1}
\end{array}
\Rightarrow
\begin{array}{ccc}
P_{n-k+1} & \xrightarrow{\phi_{n-k+1}} & Q_{n-k+1}
\end{array}
$$

So, $\phi_{k+1}$ is indeed a morphism in $\text{Path}(I, F)$.

2. For every $u: I \rightarrow F^m1$ we can define a path $(P_{n+1}, p_n)$ inductively, starting with $P_0 := I \in S$. Given some $\mathbb{C}$-morphism $u': P_k \rightarrow F^{n-k}1$, $k \leq n$, $P_k \in S$, we are done with $r = \text{id}_1$ if $k = n$. If $k < n$ consider the $F$-precise factorization

$$
P_k \xrightarrow{p_k} FP_{k+1} \xrightarrow{\rho} F^{n-k-1}1
$$

providing $P_{k+1} \in S$ and $p_k: P_k \rightarrow FP_{k+1}$ and some $u'': P_{k+1} \rightarrow F^{n-k-1}1$. This defines the families $P_{n+1}$ and $p_n$ with $\text{Comp}(P_{n+1}, p_n) = u$.

3. By Remark 3.11.

**Proof of Proposition 3.18**

For all $k < n$, the following diagram commutes:

$$
\begin{array}{ccc}
P_k & \xrightarrow{\text{in}_k} & \bigsqcup_{j \leq n} P_j \\
p_k \downarrow & & \downarrow \text{inr} \\
FP_{k+1} + 1 & \xrightarrow{\text{Fin}_{k+1} + 1} & F\bigsqcup_{j \leq n} P_j + 1 \xrightarrow{[\eta, \bot]} TF \bigsqcup_{j \leq n} P_{j+1} \\
\downarrow F[x_{k+1} + 1] & & \downarrow TF[x_{j}]_{j \leq n} \\
FX + 1 & \xrightarrow{[\eta, \bot]} & TX + 1 \\
\end{array}
$$
For sufficiency, assume that \([x_k]_{k \leq n}\) is a morphism in \(\text{Coalg}_I(I, TF)\). Hence for all \(k < n\):

\[
[\eta, \bot] \cdot (Fx_{k+1} + 1) \cdot p_k = TF[x_j]_{j \leq n} \cdot \rho \cdot \text{in}_k \subseteq \xi \cdot [x_j]_{j \leq n} \cdot \text{in}_k = \xi \cdot x_k.
\]

For necessity, we have for all \(k < n\):

\[
\begin{array}{ccc}
P_k & \xrightarrow{\text{in}_k} & \prod_{j \leq n} P_j \xrightarrow{[x_j]_{j \leq n}} X \\
p_k & \downarrow & \downarrow \subseteq \\
FP_{k+1} + 1 & \xrightarrow{F[p_{k+1} + 1]} & FX + 1 \xrightarrow{[\eta, \bot]} TXF \\
F[x_j]_{j \leq n} \cdot \rho \cdot \text{in}_k & \downarrow & \\
\end{array}
\]

and

\[
\begin{array}{ccc}
P_n & \xrightarrow{\text{in}_n} & \prod_{j \leq n} P_j \xrightarrow{[x_j]_{j \leq n}} X \\
\rho \cdot \text{in}_n & \downarrow & \downarrow \subseteq \\
F \prod_{j \leq n} P_j + 1 & \xrightarrow{\text{inr}!} & 1 \xrightarrow{!} \subseteq \\
TF \prod_{j \leq n} P_j & \xrightarrow{[\eta, \bot]} & TXF \\
\end{array}
\]

Since the family of coproduct injections \(\text{in}_k\) is jointly epic we have by (Ax2) that \([x_k]_{k \leq n}\) is indeed a weak homomorphism.

**Proof of Theorem 3.20**

As usual in open map proofs, this direction of the characterization theorem is shown by considering only Path-morphisms of length difference 1. Indeed every Path-morphism \(\phi\) from a path of length \(n\) to a path of length \(m \geq n + 2\) can be expressed as a composition of Path-morphisms of smaller length difference \(\phi = \psi \cdot \theta\). When applying the openness of a morphism \(h\) to \(\theta\), we obtain a diagonal \(d: \text{dom}(\psi) \longrightarrow \text{dom}(h)\) which allows applying the openness of \(h\) to \(\psi\), yielding the desired lifting for \(\phi\).
Consider a \( \text{Path}(I, F + 1) \)-morphism \( \phi_{n+1}: (P_{n+1}, p_n) \to (Q_{n+2}, q_{n+1}) \) and a commuting square in \( \text{Coalg}_I(I, TF) \) where \( h \) is a homomorphism:

\[
\begin{array}{cccc}
J(P_{n+1}, p_n) & \xrightarrow{[x_k]_{k \leq n}} & (X, \xi, x_0) \\
J\phi_{n+1} & \downarrow h & \\
J(Q_{n+2}, q_{n+1}) & \xrightarrow{[y_k]_{k \leq n+1}} & (Y, \zeta, y_0)
\end{array}
\]

By precomposition with \( i_n \) we obtain the following commuting diagram:

\[
\begin{array}{cccc}
P_n & \xrightarrow{x_n} & X & \xrightarrow{\xi} & TFX \\
& \downarrow \phi_n & & \downarrow h & \\
Q_n & \xrightarrow{y_n} & Y & \xrightarrow{\zeta} & TFX \\
& \downarrow q_n & \subseteq & \downarrow \zeta & \\
FQ_{n+1}+1 & \xrightarrow{Fy_{n+1}+1} & FY+1 & \xrightarrow{\eta, \perp} & TFY
\end{array}
\]

Applying (Ax5) to the outer part of the diagram yields some \( f: P_n \to FX + 1 \) and the commuting diagram:

\[
\begin{array}{cccc}
P_n & \xrightarrow{x_n} & X & \xrightarrow{\xi} & TFX \\
& \downarrow f & \subseteq & \downarrow [\eta_{FX, \perp}] & \\
FQ_{n+1}+1 & \xrightarrow{Fy_{n+1}+1} & FY+1 & \xrightarrow{[\eta_{FY, \perp}]} & TFY
\end{array}
\]

Since \( q_n \) is \( F + 1 \)-precise, we obtain the following:

\[
\begin{array}{cccc}
P_n & \xrightarrow{f} & FX + 1 & \xrightarrow{X} \\
& \downarrow q_n \cdot \phi_n & \subseteq & \downarrow h \\
FQ_{n+1}+1 & \xrightarrow{Fy_{n+1}+1} & FY+1 & \xrightarrow{[\eta_{FY, \perp}]} TFY
\end{array}
\]

Define \( d_k: Q_k \to X \) by \( x_k \cdot \phi_k^{-1} \) for \( k \leq n \). It remains to show that

\[
[d_k]_{k \leq n+1}: J(Q_{n+1}, Q_n) \to (X, \xi, x_0)
\]

is indeed a morphism in \( \text{Coalg}_I(I, TF) \) and that it makes the desired diagram in \( \text{Coalg}_I(I, TF) \) commute.

1. That \( [d_k]_{k \leq n+1} \) is a weak homomorphism follows from Proposition 3.18 because we have:

\[
\begin{array}{cccc}
Q_n & \xrightarrow{\phi_n^{-1}} & P_n & \xrightarrow{x_n} & X \\
& \downarrow q_n & \subseteq & \downarrow \xi \\
FQ_{n+1}+1 & \xrightarrow{Fd_{n+1}+1} & FX+1 & \xrightarrow{[\eta, \perp]} & TFX
\end{array}
\]
and for all $k < n$:

$$
\begin{array}{c}
Q_k \\
\downarrow_{q_n}
\end{array} \xrightarrow{\phi_k^{-1}} \begin{array}{c}
P_k \\
\downarrow_{f}
\end{array} \xrightarrow{x_k} X
\end{array}
$$

$$
FQ_{k+1} + 1 \xrightarrow{F\phi_{n+1}^{-1}} FP_{k+1} + 1 \xrightarrow{Fx_{k+1}} FX + 1 \xrightarrow{\eta, \bot} TFX
$$

2. The first desired commutativity $[d_k]_{k \leq n+1} \cdot J\phi_{n+1} = [x_k]_{k \leq n}$ is clear by the definition of $d_k$. The second commutativity $[y_k]_{k \leq n+1} = h \cdot [d_k]_{k \leq n+1}$ is proven because the coproduct injections are jointly epic. For $k \leq n$ we clearly have $y_k = h \cdot x_k \cdot \phi_k^{-1} = h \cdot d_k$. For $k = n+1$ we have $y_{n+1} = h \cdot d_{n+1}$ by the definition of $d_{n+1}$.

**Proof of Theorem 3.24**

Having $\zeta \cdot h \sqsupseteq TFh \cdot \xi$ already, we only need to show:

$$
\begin{array}{c}
X \xrightarrow{\xi} TFX \\
\downarrow_{h} \sqsubseteq \downarrow_{TFh}
\end{array}
$$

$$
\begin{array}{c}
Y \xrightarrow{\zeta} TFY
\end{array}
$$

By path-reachability, it suffices to show that for all runs $[x_k]_{k \leq n} \cdot J(P_{n+1}, P_n) \rightarrow (X, \xi, x_0)$ we have:

$$
\bigcup_{k \leq n} P_k \xrightarrow{[x_k]_{k \leq n}} X \xrightarrow{\xi} TFX
$$

$$
\begin{array}{c}
X \xrightarrow{h} Y \xrightarrow{\zeta} TFY
\end{array}
$$

By (Ax2), coproduct injections are jointly epic in an ordered-enriched sense, so by induction, it suffices to prove that:

$$
X \xrightarrow{h} Y \xrightarrow{\zeta} TFY
$$

By (Ax4), we can prove this by showing that for all $p'_n : P_n \rightarrow TFY$ we have.

We will prove the implication

$$
\begin{array}{c}
P_n \xrightarrow{x_n} X \xrightarrow{\xi} TFX
\end{array}
$$

$$
\begin{array}{c}
FY + 1 \xrightarrow{[\eta_{FY}, \bot\text{FY}]} TFY
\end{array}
$$

$$
\begin{array}{c}
P_n \xrightarrow{x_n} X \xrightarrow{\xi} TFX
\end{array}
$$

$$
\begin{array}{c}
FY + 1 \xrightarrow{[\eta_{FY}, \bot\text{FY}]} TFY
\end{array}
$$
In the following, this implication is proved for all $p'_n$. For such a $p'_n: P_n \rightarrow FY + 1$, since $P_n \in S$, (Ax1) yields an object $P_{n+1} \in S$ and morphisms $p_n: P_n \rightarrow FP_{n+1} + 1$ and $y_{n+1}: P_{n+1} \rightarrow Y$ with $(FY_{n+1} + 1) \cdot p_n = p'_n$. So we have a new path $(P_{n+2}, p_{n+1})$ and a morphism defined by

$$[y_k]_{k \leq n+1}: \prod_{k \leq n+1} P_k \rightarrow Y \quad \text{with } y_k = x_k \text{ for } k \leq n, \text{ and } y_{n+1} \text{ as above.}$$

This is indeed a lax homomorphism $J(P_{n+2}, p_{n+1}) \rightarrow (Y, \zeta, y_0)$ because of Proposition 3.18 and because for all $k < n$:

\[
\begin{array}{cccccc}
P_k & \xrightarrow{x_k} & X & \xrightarrow{h} & Y \\
p_k & \xrightarrow{F \xi} & FX + 1 & \xrightarrow{TF \xi} & TFY \\
 & \xrightarrow{F_{x_{k+1}+1} + 1} & FP_{k+1} + 1 & \xrightarrow{FY + 1} & Y \\
\end{array}
\]

and

\[
\begin{array}{cccccc}
P_n & \xrightarrow{x_n} & X & \xrightarrow{h} & Y \\
p_n & \xrightarrow{P_{P_{n+1} + 1}} & FP_{n+1} + 1 & \xrightarrow{FY + 1} & TFY \\
\end{array}
\]

Since $y_k = x_k$, for $k \leq n$, the constructed lax homomorphism makes the following square commute

$$J(P_{n+1}, p_n) \xrightarrow{[x_k]_{k \leq n}} (X, \xi, x_0) \xrightarrow{[\eta, \perp]} \text{in Coalg}_I(I, TF).$$

Since $h$ is open, we obtain a diagonal lifting, that is, a lax homomorphism $[d_k]_{k \leq n+1}: J P_{n+1} \rightarrow (X, \xi, x_0)$ with $h \cdot d_k = y_k$ for $k \leq n + 1$ and $d_k = x_k$ for $k \leq n$. This finally proves the desired implication:

\[
\begin{array}{cccccc}
P_n & \xrightarrow{d_n = x_n} & X \\
p_n & \xrightarrow{P_{P_{n+1} + 1}} & FP_{n+1} + 1 & \xrightarrow{FY + 1} & TFY \\
\end{array}
\]
Remarks on Assumption 3.26

The category of sets and the category of nominal sets have (epi, mono)-factorizations. However, the assumption is rather unusual, because in most categories, a morphism that is both epi and mono is not necessarily an isomorphism. By the assumption of (epi, mono)-factorizations together with the cocompleteness of \( C \), the following are equivalent for any family \( (e_i : X_i \to Y)_{i \in I} \):

1. \( (e_i)_{i \in I} \) are jointly epic.
2. \( \{e_i\}_{i \in I} : \prod_{i \in I} X_i \to Y \) is an epimorphism.
3. \( \{e_i\}_{i \in I} : \bigsqcup_{i \in I} X_i \to Y \) is a strong epimorphism.
4. \( (e_i)_{i \in I} \) are jointly strong epic.

All the results in subsection 3.5 work when replacing ‘jointly epic’ by ‘jointly strong epic’ and by replacing ‘(epi, mono)-factorizations’ by the very common assumption of ‘(strong epi, mono)-factorizations’. Since in all our instances (possibly sorted Sets, Nominal Sets), all epimorphisms are strong, we phrase the results and proofs in terms of (jointly) epic families for the sake of simplicity.

Proof of Proposition 3.27

Consider a subcoalgebra \( h : (Y, \zeta, y_0) \to (X, \xi, x_0) \). For every run \( [x_k]_{k \leq n} : J(P_{n+1}, p_n) \to (X, \xi, x_0) \), we have the commuting square:

\[
\begin{array}{ccc}
J0 = (I, \bot F_I! , \text{id}_I) & \xrightarrow{i} & (Y, \zeta, y_0) \\
\downarrow J! & & \downarrow h \\
J(P_{n+1}, p_n) & \xrightarrow{[x_k]_{k \leq n}} & (X, \xi, x_0)
\end{array}
\]

and since \( h \) is open by Theorem 3.20, we have a run \( [d_k]_{k \leq n} \) of \( (P_{n+1}, p_n) \) in \( (Y, \zeta, y_0) \). Since \( [x_k]_{k \leq n} \) is epic and \( h \cdot d_k = x_k \) for all \( 0 \leq k \leq n \), we have that the mono \( h \) is an epimorphism and hence an isomorphism.

Proof of Proposition 3.28

For functors \( H : C \to C \) preserving intersections, we have breadth-first-search:

**Lemma A.5.** Let \( H : C \to C \) preserve arbitrary intersections and let \( C \) have \((\mathcal{E}, \text{mono})\)-factorizations, countable coproducts, and arbitrary intersections. For an \( I \)-pointed \( H \)-coalgebra \( I \xrightarrow{i} X \xrightarrow{\xi} HX \), there are monomorphisms \( (m_k : X_k \to X) \) with the property that:

1. \( m_0 \) is the mono-part of the \((\mathcal{E}, \text{mono})\)-factorization of \( i \).
2. \( m_{k+1} \) is the least subobject of \( X \) such that \( \xi \cdot m_k \) factors through \( Hm_{k+1} \).
3. The union of the \( m_k \), i.e. the image of \( [m_k]_{k \geq 0} : \prod_{k \geq 0} X_k \to X \) is the carrier of a reachable subcoalgebra of \( (X, \xi, x_0) \). In particular, if \( (X, \xi, x_0) \) is reachable, then \( [m_k]_{k \geq 0} \in \mathcal{E} \).
Intuitively, $X_k \to X$ contains precisely the states that are $k$ steps away from the initial state $X_0$, and so the union $\bigcup_{k \geq 0} X_k$ contains all reachable states.

**Proof.** 1. and 2. Let $I \to X_0 \xrightarrow{m_0} X$ be the $(\mathcal{E},\text{mono})$-factorization of $i: I \to X$. For the inductive step, assume $m_k: X_k \to X$. Consider the family of subobjects $m': X' \to X$ with the property that there exists some $\xi': X_k \to HX'$ with $Hm' \cdot \xi' = \xi \cdot m_k$. Denote the intersection of these subobjects by $m_{k+1}: X_{k+1} \to X$, i.e. we have the wide pullback:

Note that since $H$ preserves intersections, i.e. pullbacks of monomorphisms, it preserves monomorphisms, because a morphism $m$ is mono. So the above wide pullback of monos is mapped again to a pullback of monos, and we have the commutative diagram:

By the universal property of the wide pullback, there is a unique map $\xi_k: X_k \to HX_{k+1}$ with $Hm_{k+1} \cdot \xi_k = \xi \cdot m_k$. Indeed, $m_{k+1}: X_{k+1} \to X$ is the least subobject of $X$ with this property, because any other $m': X' \to X$ with this property is itself included in the diagram for the intersection and hence $X_{k+1} \to X'$.
3. Consider the \((E,\text{mono})\)-factorization \(\coprod_{k \geq 0} X_k \xrightarrow{e} Y \xrightarrow{h} X\) of \([m_k]_{k \geq 0}\), then with the \(\xi_k\) as above, we have the commutative diagram:

\[
\begin{array}{ccccccccc}
I & \xrightarrow{i} & X & \xrightarrow{\xi} & HX & \xleftarrow{\text{commutative}} \\
I \xrightarrow{e \cdot i' \cdot \text{in}_0} & Y & \xleftarrow{\text{commutative}} & & \xrightarrow{\text{commutative}} & \\
\prod_{k \geq 0} X_k & \xrightarrow{\coprod_{k \geq 0} \xi_k} & \prod_{k \geq 0} HX_{k+1} & \xrightarrow{[H(\text{in}_{k+1})]_{k \geq 0}} & H \prod_{k \geq 0} X_k \\
& \uparrow{\text{commutative}} & & \uparrow{\text{commutative}} & \\
& & H[m_k]_{k \geq 0} & & \\
\end{array}
\]

and since \(e \in E\) is left-orthogonal to the mono \(Hh\), there is a unique morphism \(y: Y \rightarrow HY\) making \(e\) and \(h\) coalgebra homomorphisms. It remains to show that \((Y, y, e \cdot \text{in}_0 \cdot i')\) does not have a proper subcoalgebra. Assume a pointed subcoalgebra \(f: (Z, \zeta, z_0) \rightarrow (Y, y, e \cdot i')\), with \(f: Z \rightarrow Y\). We show that \(e\) factors through \(f\), by constructing maps \(d_k: X_k \rightarrow Z\) with \(e \cdot \text{in}_k = f \cdot d_k\) inductively. For \(k = 0\), we have the following commuting square which uniquely induces \(d_0\):

\[
\begin{array}{ccc}
I & \xrightarrow{i'} & X_0 \\
\downarrow{z_0} & \exists!{d_0} & \downarrow{e \cdot \text{in}_0} \\
Z & \xrightarrow{e \cdot f} & Y \\
\end{array}
\]

Given \(d_k: X_0 \rightarrow Z\) with \(f \cdot d_k = e \cdot \text{in}_k\), we have the commutative diagram:

\[
\begin{array}{cccc}
X & \xrightarrow{\xi} & HX & \\
\downarrow{\xi_k} & \downarrow{Hm_{k+1}} & \downarrow{Hh} & \\
X_k & \xrightarrow{e \cdot \text{in}_k} & HX_{k+1} & \xleftarrow{Hf} \\
\downarrow{d_k} & \downarrow{H(e \cdot \text{in}_{k+1})} & \downarrow{Hf} & \\
Y & \xrightarrow{Hf} & HY & \\
\downarrow{f} & \downarrow{\zeta} & \downarrow{HZ} & \\
Z & \xrightarrow{\zeta} & HZ & \\
\end{array}
\]

So \(h \cdot f\) is a subobject with the property that \(\xi \cdot m_k\) factors through \(H(h \cdot f)\). Since by item (2.), \(X_{k+1}\) is the least subobject with this property, there is some \(d_{k+1}: X_{k+1} \rightarrow Z\) with \((h \cdot f) \cdot d_{k+1} = m_{k+1}\) and in particular \(f \cdot d_{k+1} = e \cdot \text{in}_{k+1}\) since \(h\) is monic.

In total, we have \(f \cdot [d_k]_{k \geq 0} = [e \cdot \text{in}_k]_{k \geq 0} = e \in E\), and so \(f \in E\) (by \(E\)-laws in factorizations) and hence the mono \(f\) must be an isomorphism.

We now can continue with the proof of the main statement, and in the following we instantiate Lemma A.5 with \(H = TF, E = \text{epi}\).

Proof (of Proposition 3.28). Let \(\xi_k: X_k \rightarrow TFX_{k+1}\) the witness of item 2 of Lemma A.5, and so \(TFm_{k+1} \cdot \xi_k = \xi \cdot m_k\) and \(m_{k+1}\) is the least subobject with
this property. In the following, we choose sets $\mathcal{F}_k$, $k \geq 0$, containing morphisms $(E, e) \in \mathcal{F}_k$, $e: E \rightarrow X_k$, $E \in \mathcal{S}$, inductively:

1. $\mathcal{F}_0$ contains only $x'_0: I \rightarrow X_0$, where $x_0 = m_0 \cdot x'_0$ is the pointing $I \rightarrow X$, as provided by item 1 of Lemma A.5.
2. For every $(E, e) \in \mathcal{F}_k$, let $G_{(E, e)} \subseteq C(E, FX_{k+1} + 1)$ be the set of morphisms $g: E \rightarrow FX_{k+1} + 1$ with:

\[
\begin{array}{c}
E \\
\downarrow^g
\end{array}
\quad e
\quad X_k
\quad \xi_k
\quad TFX_{k+1}
\quad \downarrow^{[\eta, \bot]_{FX_{k+1}}}
\quad FX_{k+1} + 1
\quad \downarrow^{[\eta, \bot]_{FX_{k+1}}}
\quad E X_k
\end{array}
\]

For each $g \in G_{(E, e)}$, choose some object $Y_g \in \mathcal{S}$, and morphisms $y_g: Y_g \rightarrow X_{k+1}$ and a $F + 1$-precise $p_g: E \rightarrow FY_g + 1$ with $g = (Fy_g + 1) \cdot p_g$, according to (Ax1). Define $\mathcal{F}_{k+1} := \{(Y_g, y_g) | (E, e) \in \mathcal{F}_k, g \in G_{(E, e)}\}$.

We need to prove some properties about the $\mathcal{F}_k$

- By construction, for every $(E', e') \in \mathcal{F}_{k+1}$, there is some $(E, e) \in \mathcal{F}_k$ and an $F + 1$-precise map $p_g: E \rightarrow FE' + 1$ with

\[
\begin{array}{c}
E \\
\downarrow^{p_g}
\end{array}
\quad e
\quad X_k
\quad \xi_k
\quad TFX_{k+1}
\quad \downarrow^{[\eta, \bot]_{FX_{k+1}}}
\quad FX_{k+1} + 1
\quad \downarrow^{[\eta, \bot]_{FX_{k+1}}}
\quad E X_k
\end{array}
\]

Consequently by Proposition 3.18, for every $(E, e) \in \mathcal{F}_k$, $k \geq 0$, there is a run $[x_j]_{j \leq k}: J(P_{k+1}, p_k) \rightarrow (X, \xi, x_0)$ with $P_k = E$, $x_k = in_k \cdot e$.

- For every family $\mathcal{F}_k$, $k \geq 0$, the morphism $[e]_{(E, e) \in \mathcal{F}_k}$ is an isomorphism; this means that for the factorization:

\[
[e]_{(E, e) \in \mathcal{F}_k} \equiv (\bigsqcup_{(E, e) \in \mathcal{F}_k} E \xrightarrow{q_k} s_k \xrightarrow{p_k} X_k)
\]

we have that $s_k$ is an isomorphism for every $k \geq 0$.

\[
\xi_k \cdot e \overset{(Ax4)}{=} \bigsqcup_{g \in G_{(E, e)}} [\eta_{FX_{k+1}}, \bot_{FX_{k+1}}] \cdot g
\]

\[
\overset{\text{Def.}}{=} \bigsqcup_{g \in G_{(E, e)}} [\eta_{FX_{k+1}}, \bot_{FX_{k+1}}] \cdot (Fy_g + 1) \cdot p_g
\]

\[
\overset{(Y_g, y_g) \in \mathcal{F}_{k+1}}{=} \bigsqcup_{g \in G_{(E, e)}} [\eta_{FX_{k+1}}, \bot_{FX_{k+1}}] \cdot (F(s_{k+1} \cdot q_{k+1} \cdot in_{y_g}) + 1) \cdot p_g
\]
Naturality = $\bigcup_{g \in G(E,e)} TFS_{k+1} \cdot [\eta FX'_{k+1}, \perp FX_{k+1}'] \cdot (F(q_{k+1} \cdot \text{in}_g) + 1) \cdot p_g$

Reflection = $TFS \cdot \bigcup_{g \in G(E,e)} [\eta FX'_{k+1}, \perp FX_{k+1}'] \cdot (F(q_{k+1} \cdot \text{in}_g) + 1) \cdot p_g$

short hand $v(E,e) : E \to TF_X^{k+1}$

Hence, $TFS \cdot v(E,e) = \xi_k \cdot e$ for all $(E,e) \in \mathcal{F}_k$. Since the $e \in \mathcal{F}_k$ are jointly epic by the induction hypothesis, we have a unique diagonal $\xi'_k$ in the following square:

\[ \begin{array}{ccc}
\prod_{(E,e) \in \mathcal{F}_k} E & \xrightarrow{[e]_{(E,e) \in \mathcal{F}_k}} & X_k \\
\downarrow v_{(E,e)} & & \downarrow \xi_k \\
TF_X^{k+1} & \xrightarrow{TFS_{k+1}} & TF_X^{k+1}
\end{array} \]

Since $\xi_k$ was constructed to be the least morphism, $s_{k+1}$ is necessarily an isomorphism.

Now we have that $f_k := [e]_{(E,e) \in \mathcal{F}_k} \cdot \prod_{(E,e) \in \mathcal{F}_k} E \to X_k$ is epic for every $k \geq 0$. Since $(X,\xi,x_0)$ is reachable, $[m_k]_{k \geq 0} : \prod_{k \geq 0} X_k \to X$ (cf. item 3 of Lemma A.5) is an epimorphism. Hence, the family $(m_k \cdot f_k)_{k \geq 0}$ is jointly epic; this family is contained in the family of runs in $(X,\xi,x_0)$, and so the family of runs is jointly epic. □

**Proof of Proposition 4.3**

The first step is to describe when a map is $B$-precise:

**Lemma A.6.** A map $s : X \to BY$ is $B$-precise iff for all $y \in Y$,

\[ \sum_{x \in S} s(x)(y) = 1. \]

**Proof.** Sufficiency is proven by the two inequalities:

1. Let $t_{X,Y} : BY \times X \to B(Y \times X)$ be the functorial strength defined as for all Set-functors by $t_{X,Y}(b,x) = B(y \mapsto (y,x))(b)$. We thus have the commuting diagram

\[ \begin{array}{ccc}
X & \xrightarrow{(s,\text{id}_X)} & BY \times X \\
\downarrow s & & \downarrow t_{X,Y} \\
& & B(Y \times X) \\
& \xrightarrow{B\pi_1} & BY
\end{array} \]

Since $s$ is $B$-precise, we obtain a map $d : Y \to Y \times X$ with $\pi_1 \cdot d = \text{id}_Y$ and $t_{X,Y} \cdot (s,\text{id}_X) = Bd \cdot s$, that means $B(y \mapsto (y,x))(s(x)) = B(y \mapsto (y,d'(y)))(s(x))$, with $d' = \pi_2 \cdot d$. Hence, every $y \in Y$ appears in at most one $s(x)$, namely only in $s(d'(y))$ or not at all; this means for every $y \in Y$, only one summand of $\sum_{x \in S} s(x)(y)$ is non-zero.
To see that every summand of $\sum_{x \in S} s(x)(y)$ is at most 1, define $f : X \to \mathcal{B}(\mathbb{N} \times Y)$ by
\[
 f(x)(n, y) = \begin{cases} 
 1 & \text{if } n < s(x)(y) \\
 0 & \text{otherwise}
\end{cases}
\]
and so
\[
\mathcal{B}(\pi_2)(f(x))(y) = \sum_{(n, y) \in \mathbb{N} \times Y} f(x)(n, y) = \sum_{n < s(x)(y)} 1 = s(x)(y)
\]
Hence, we have some map $d : Y \to \mathbb{N} \times Y$ with $f = \mathcal{B}(d) \cdot s$ and $d(y) = (d'(y), y)$ for some $d' : Y \to \mathbb{N}$. The first equality expands to
\[
f(x)(n, y) = \mathcal{B}(d)(s(x))(n, y) = \sum_{y' \in Y} s(x)(y') = \begin{cases} 
 s(x)(y) & \text{if } n = d'(y) \\
 0 & \text{otherwise}
\end{cases}
\]
This implies that for every $x$ and $y$ there is at most one $n \in \mathbb{N}$ such that $f(x)(n, y) > 0$, hence $s(x)(y) \leq 1$.

$(\geq 1)$ Assume that $\sum_{x \in X} s(x)(y) = 0$ for some $y \in Y$. Define $s' : X \to \mathcal{B}(Y \setminus \{y\})$ by $s'(x)(y') = s(x)(y')$. With the obvious inclusion $i : Y \setminus \{y\} \hookrightarrow Y$ we have $s = \mathcal{B}i \cdot s'$ and so the $\mathcal{B}$-precise $s$ induces a map $d : Y \to Y \setminus \{y\}$ with $i \cdot d = \text{id}_Y$, a contradiction.

For necessity, consider $f$ and $h$ with
\[
\begin{array}{ccc}
X & \xrightarrow{f} & BC \\
\downarrow s & & \downarrow \mathcal{B}h \\
BY & & \\
\end{array}
\tag{4}
\]
By assumption on $s$ we have for all $y \in Y$:
\[
1 = \sum_{x \in X} s(x)(y) \overset{(4)}{=} \sum_{x \in X} \sum_{c \in C} f(x)(c) = \sum_{x \in X, c \in C} f(x)(c) = \sum_{x \in X, c \in C} f(x)(c)
\]
Since all summands of the right-hand sum are non-negative integers, there exists precisely one $x_y \in X, c_y \in C$ with $h(c_y) = y$ such that $f(x)(c) \neq 0$. Hence, define $d : Y \to C$ by this witness: $d(y) = c_y$. This implies directly that $h(d(y)) = y$ and that $s(x)(y) = f(x)(d(y))$ for all $x \in X, y \in Y$.
\[
\mathcal{B}d(s(x))(c) = \sum_{y \in Y, d(y) = c} s(x)(y) = \sum_{y \in Y, d(y) = c} f(x)(d(y)) = f(x)(c)
\]
where the last equality holds because $d$ is injective.

Now, we want to transfer preciseness from the bag functor to any analytic functor using the natural transformation $\alpha$. This is done using the following:
Lemma A.7. Let $F$ preserve weak pullbacks and let $\alpha: F \to G$ be a natural transformation whose naturality squares are weak pullbacks. Then a morphism $f: X \to FY$ is $F$-precise iff $\alpha_Y \cdot f$ is $G$-precise. If $G$ admits precise factorizations w.r.t. $S \subseteq \text{obj} \, C$, then so does $F$.

Proof. For sufficiency, let $f: X \to FY$ be $F$-precise and consider a commuting diagram

$$
\begin{array}{ccc}
X & \xrightarrow{g} & GW \\
\downarrow{\alpha_Y \cdot f} & & \downarrow{G_w} \\
GY & \xrightarrow{G_y} & GZ
\end{array}
$$

Since the naturality square for $w$ is a weak pullback, a morphism $g': X \to FZ$ exists, making the following diagram commute:

$$
\begin{array}{ccc}
X & \xrightarrow{g} & GW \\
\downarrow{f} & \xrightarrow{\alpha_W} & \downarrow{G_w} \\
FY & \xrightarrow{F_y} & GFY \\
\downarrow{\alpha_Y} & \xrightarrow{G_y} & GZ
\end{array}
$$

Since $f$ is $F$-precise, we obtain a morphism $d: Y \to W$ with $w \cdot d = y$ and $Fd \cdot f = g'$, and thus also $g = \alpha_W \cdot g' = \alpha_W \cdot Fd \cdot f = Gd \cdot \alpha_Y \cdot f$ as desired.

For necessity, we use the simplified version of $F$-precise because $F$ preserves weak pullbacks. So let $\alpha_Y \cdot f$ be $G$-precise and consider a commuting diagram

$$
\begin{array}{ccc}
X & \xrightarrow{g} & FZ \\
\downarrow{f} & \xrightarrow{Fz} & FY
\end{array}
$$

Hence, $Gz \cdot \alpha_Z \cdot g = \alpha_Y \cdot f$ and since $\alpha_Y \cdot f$ is $G$-precise we obtain a morphism $d: Y \to Z$ with

$$
\begin{array}{ccc}
X & \xrightarrow{g} & FZ & \xrightarrow{\alpha_Z} & GZ \\
\downarrow{f} & \xrightarrow{Gd} & FY & \xrightarrow{\alpha_Y} & GY \\
\end{array}
$$

& $z \cdot d = \text{id}_Y$
The naturality square for \( d \) is a weak pullback, and so we have some \( f' : X \to FY \) making the following diagrams commute:

\[
\begin{array}{c}
X \\
\downarrow f
\end{array}
\begin{array}{c}
FY \\
\downarrow \alpha_Y
\end{array}
\xymatrix{\exists f' \ar[r] & \ar[r]_{Fd} & FZ \\
FY \downarrow \alpha_Y \ar[u]^g & GY \downarrow \alpha_Z \ar[u]^{Gd} & GZ}
\Rightarrow

\begin{array}{c}
X \\
\downarrow f
\end{array}
\begin{array}{c}
FY \\
\downarrow \alpha_Y
\end{array}
\xymatrix{FY \ar[r]_{Fd} & FZ \ar[r]_{Fz} & FY \\
FY \downarrow \alpha_Y \ar[u]^g & Fid \ar[u]}
\]

So \( d \) is indeed a diagonal lifting: \( Fd \cdot f = Fd \cdot f' = g \).

For **precise factorizations**, consider \( f : X \to FY \) with \( X \in S \). For the \( G \)-precise factorization \( Gy' \cdot g' \) of \( \alpha_Y \cdot f \) we have:

\[
\begin{array}{c}
X \\
\downarrow f
\end{array}
\begin{array}{c}
FY' \\
\downarrow \alpha_Y
\end{array}
\xymatrix{Y' \in S \\
GY' \ar[u]_{Gy'} \ar[u]}
\]

Since the naturality square is a weak pullback, we obtain some \( f' : X \to FY' \) with \( FY' \cdot f' = f \) and \( Y' \in S \) as desired. \( \Box \)

To conclude, since analytic functors preserve weak pullbacks, it is enough to prove the following:

**Lemma A.8.** The naturality squares of \( \alpha : F \to B \) are weak pullback squares.

**Proof.** Consider a map \( f : X \to Y \) and elements \( t \in FY, s \in BX \) with \( \alpha_Y(t) = Bf(s) \). We can write \( t \) as an equivalence class \( t = [\sigma(y_1, \ldots, y_n)] \). Every \( y_i \) is in the image of \( f \), and so any \( t' = [\sigma(x_1, \ldots, x_n)] \in FX \) with \( f(x_i) = y_i \) for all \( 1 \leq i \leq n \) has the property that \( \alpha_X(t') = s \) and \( Ff(t') = t \). \( \Box \)

So we obtain Proposition 4.3 as a corollary of Lemma A.7.

**Proof of Proposition 4.5**

In this section, we will use the following standard lemma on strong nominal sets (see e.g. [20, Prop. 5.10] or [17, p. 3]):

**Lemma A.9.** Given a strong nominal set \( X \), a subset \( O \subseteq X \) such that for every \( x' \in X \) there is precisely one \( x \in X \) with \( \pi \in \pi \{x\} \) fulfilling \( x' = \pi \cdot x \), and a map \( f : O \to Y \) into a nominal set \( Y \) with \( \text{supp}(f(x)) \subseteq \text{supp}(x) \forall x \in O \). Then there is a unique equivariant map \( f' : X \to Y \) that agrees with \( f \) on \( O \). Such a subset \( O \subseteq X \) exists for every nominal set \( X \).

Let us check the axioms:
(Ax2): obtain as in Set.
(Ax3): We indeed have natural transformations \( \eta: X \to \mathcal{P}_{ufs}(X) \) for singleton sets and \( \bot: 1 \to \mathcal{P}_{ufs}(X) \) for the empty set.
(Ax4): consider an equivariant \( A \to \mathcal{P}_{ufs}B \) with \( A \) strong. Since \( f \) is by definition an upper bound for the \( \mathcal{F} = \{[\eta_B, \bot] \cdot f' \subseteq f \mid f': A \to B + 1\} \), we need to prove that it is the least upper bound. Let \( g \) be an upper bound for \( \mathcal{F} \). In order to have \( f \subseteq g \) we need to show for every \( a \in A \) and \( b \in f(a) \) that \( b \in g(a) \). Since \( b \in f(a) \) and \( f(a) \) is ufs, we have \( \text{supp}(b) \subseteq \text{supp}(f(a)) \subseteq \text{supp}(a) \). Define \( f': A \to B + 1 \) via Lemma A.9, where \( O \subseteq A \) is chosen with \( a \in O \) and \( f'' : O \to B + 1 \) is defined by \( f''(a) = \text{inl}(b) \) and \( f''(x) = \text{inr}(*) \) for \( x \neq a \). Now, \( f' \) is the unique equivariant map extending \( f'' \). Since \( [\eta_B, \bot] \cdot f''(x) \subseteq f(x) \) for all \( x \in O \), we have that \( [\eta_B, \bot] \cdot f' \subseteq f \), and so \( [\eta_B, \bot] \cdot f' \in \mathcal{F} \), and thus \( [\eta_B, \bot] \cdot f' \subseteq g \), hence \( b \in g(a) \).
(Ax5): consider a weakly commuting square:

\[
\begin{array}{ccc}
A & \xrightarrow{x} & \mathcal{P}_{ufs}X \\
\downarrow{y} & \nearrow{\eta} & \downarrow{\mathcal{P}_{ufs}h} \\
Y & \xrightarrow{\eta_Y} & \mathcal{P}_{ufs}Y
\end{array}
\]

where \( A \) is strong. Pick any subset \( O \subseteq A \) fulfilling the assumption of Lemma A.9. For each \( a \in A \) we have that \( y(a) \in h[x(a)] \), so for each \( a \in O \), there exists some \( a' \in x(a) \) with \( h(a') = y(a) \). For each \( a \in O \), denote this witness by \( x'(a) := a' \). By Lemma A.9, \( x' \) extends to an equivariant map \( c : A \to X \) with \( c(a) = x'(a) \in x(a) \) for all \( a \in O \). For every \( b \in A \), there is some \( \pi \in \mathcal{S}_f(A) \) and \( a \in A \) with \( b = \pi \cdot a \), and hence \( x'(b) = x'(\pi \cdot a) = \pi \cdot x'(a) \in \pi \cdot x(a) = x(b) \), i.e. \( \eta_X \cdot x' \subseteq x \). By construction of \( c \), we have that \( h \cdot x' = y \).

**Proof of Proposition 4.7**

To prove this, by Proposition 3.6.4, it is enough to prove that the binding functor \( [\mathbb{A}] \) admits precise factorizations w.r.t. strong nominal sets.

Recall from [23] that \( [\mathbb{A}] \) has the left adjoint \( \mathbb{A}\#_- \), sending \( X \) to the nominal set \( \mathbb{A}\#_- X := \{(a, x) \in \mathbb{A} \times X \mid a \notin \text{supp}(x)\} \). If \( X \) is a strong nominal set, then so is \( \mathbb{A}\#_- X \), hence \( [\mathbb{A}] \) admits precise factorizations w.r.t. strong nominal sets by Proposition 3.6.5. \( \square \)

**Proof of Proposition 4.8**

All the axioms are proved as in Set, component-wise. The only interesting axiom is to prove that \( F\{X_P\}_{P \in \mathbb{P}} = \bigsqcup_{Q \in \mathbb{P}} \mathbb{P}(P, Q) \times X_Q \) admits precise factorisations. Since \( F = \bigsqcup_{Q \in \mathbb{P}} H_Q \times \pi_Q \) with:

- \( H_Q \) being the constant functor on \( (\mathbb{P}(P, Q))_{P \in \mathbb{P}} \),
- \( \pi_Q \) being the functor mapping \( \{X_P\}_{P \in \mathbb{P}} \) to \( \{X_Q\}_{P \in \mathbb{P}} \),
by Proposition 3.6.(1,3,4), it is enough to prove that \( \pi_Q \) admits precise factorisations.

For that purpose, given a morphism, i.e. family of maps, \((f_P)_{P \in P} : \{X_P\}_{P \in P} \to \pi_Q(\{Y_P\}_{P \in P}) = \{Y_Q\}_{P \in P}\), define

\[
(Z_P)_{P \in P} \quad \text{with } Z_Q = \prod_{P \in P} X_P \text{ and } Z_P = \emptyset \text{ for } P \neq Q.
\]

and \((\text{in}_P)_{P \in P} : \{X_P\}_{P \in P} \to \pi_Q(Z_P)_{P \in P}\)

\[
\text{in}_P : X_P \to (\pi_Q(Z_P)_{P \in P})_P = Z_Q = \prod_{P \in P} X_P
\]

and \((h_P)_{P \in P} : (Z_P)_{P \in P} \to (Y_P)_{P \in P}\) with

\[
h_Q = [f_P]_{P \in P} : \prod_{P \in P} X_P \to Y_Q, \quad h_P = ! : \emptyset \to Y_Q \text{ for } P \neq Q.
\]

Obviously, \((\pi_Q h)_P \cdot \text{in}_P = f_P\) for all \(P \in P\). It remains to show that \((\text{in}_P)_{P \in P}\) is \(\pi_Q\)-precise. Consider a commutative square:

\[
\begin{array}{ccc}
(X_P)_{P \in P} & \xrightarrow{(v_P)_{P \in P}} & \pi_Q((V)_{P \in P}) \\
& \downarrow (\text{in}_P)_{P \in P} & \downarrow \pi_Q((w_P)_{P \in P}) = w_Q \\
\pi_Q((Z)_{P \in P}) & \xrightarrow{(z_P)_{P \in P}} & \pi_Q((W)_{P \in P}) \quad \text{(*)}
\end{array}
\]

Define a diagonal \((d_P)_{P \in P} : (Z_P)_{P \in P} \to (V_P)_{P \in P}\) by

\[
d_Q = [v_P]_{P \in P} : \prod_{P \in P} X_P \to V_Q, \quad d_P = ! : Z_P \to V_P \text{ for } P \neq Q.
\]

Obviously, we have the first triangle \(v_P = d_Q \cdot \text{in}_P\) for all \(P \in P\). The other triangle follows by case distinction: for \(P \neq Q\), \(z_P = w_P \cdot d_P\) holds by initiality of \(Z_P = \emptyset\); we have

\[
z_Q \cdot \text{in}_P = (\ast) \quad w_Q \cdot v_P = w_Q \cdot d_Q \cdot \text{in}_P \quad \forall P \in P
\]

and since the \(\text{in}_P : X_P \to Z_Q\) are jointly epic, \(z_Q = w_Q \cdot d_Q\). So \((d_P)_{P \in P}\) is a diagonal lifting and so \((\text{in}_P)_{P \in P}\) is \(\pi_Q\)-precise.

**Proof of Proposition 4.9**

To prove this proposition, it is enough to prove the following:

**Lemma A.10.** A morphism \(f : \chi^P \to FY\) is \(F\)-precise iff \(Y = \chi^Q\) for some \(Q \in P\).
Proof. **Necessity.** Consider \( f_P : 1 \rightarrow \bigsqcup_{Q \in P} \mathbb{P}(P, Q) \times Y_Q \) and let \( f_P = \text{in}_Q(m, x) \). Define the injective \( h: \chi^Q \rightarrow (Y_R)_{R \in P} \) by \( h_R = !: \emptyset \rightarrow Y_R \) for \( R \neq Q \) and \( h_Q = x: 1 \rightarrow X_Q \). Furthermore, define by \( g: \chi^P \rightarrow F\chi^Q \ g_P = \text{in}_Q(m, *) \), and so \( Fh \cdot g = f \). Since \( f \) is \( F \)-precise, there is a \( d: Y \rightarrow \chi^Q \) with \( h \cdot d = \text{id}_Y \), so in total \( h \) is an isomorphism.

**Sufficiency.** For \( f: \chi^P \rightarrow F\chi^Q \) let \( m: P \rightarrow Q \) with \( f_P = \text{in}_Q(m, *) \) and consider a commuting diagram:

\[
\begin{array}{ccc}
\chi^P & \xrightarrow{w} & FW \\
\downarrow f & & \downarrow \text{in}_P \circ \text{obj}^P \\
F\chi^Q & \xrightarrow{Fz} & FZ
\end{array}
\]

And so in component \( P \) we have:

\[
\begin{array}{ccc}
1 & \xrightarrow{w_P} & \bigsqcup_{R \in P} \mathbb{P}(P, R) \times W_R \\
\downarrow \langle m, \text{id} \rangle & & \downarrow \text{in}_P \\
\mathbb{P}(P, Q) \times \chi^Q & \xrightarrow{\text{in}_Q} & \bigsqcup_{R \in P} \mathbb{P}(P, R) \times \chi^R \\
& & \downarrow \text{in}_P \\
& & \bigsqcup_{R \in P} \mathbb{P}(P, R) \times Z_R \\
& & \downarrow \text{in}_P
\end{array}
\]

Hence, (formally by the extensivity of \( \text{Set} \)) \( w_P \) is necessarily \( w_P = \text{in}_Q(m, x) \) for some \( x \in W_Q \) with \( h_Q(x) = z_Q(*) \). Define \( d: \chi^Q \rightarrow W \) by \( d_Q: 1 \rightarrow W_Q, d_Q = x \), and \( d_R = !: \emptyset \rightarrow W_R \) for \( R \neq Q \). Obviously \( d \) fulfils both \(Fd \cdot f = w \) and \( h \cdot d = z \).

\(\square\)

**Notes on section 5**

In probabilistic systems, it is not clear how paths should look like. Consider the probabilistic system \( A \):

\[
\begin{array}{ccc}
& & e \\
& & \downarrow 0.3 \\
b & \xleftarrow{0.2} & a \\
& & \downarrow 0.4 \\
c & \xrightarrow{0.3} & f
\end{array}
\]

There are different possibilities how to model a path in such a system:

1. Simply \( \bullet \rightarrow \bullet \rightarrow \bullet \) as in LTS for a singleton input alphabet. Of course, this makes the probabilities in the system entirely meaningless for runs.
2. A path and a probability \( (p, \bullet \rightarrow \bullet \rightarrow \bullet) \) meaning that there is a path whose weights multiply to at least \( p \). So the above system has a run for such a path if \( p \leq 0.08 = 0.2 \cdot 0.4 \).
3. A path is a probabilistic system in which each state has at most one successor, and so the notion of run of path coincides with a functional simulation from that path to the system. E.g. the above system has a run for \( \bullet \xrightarrow{0.1} \bullet \xrightarrow{0.25} \bullet \).
Clearly, there is a functional bisimulation $f : A \to X$ from the above system to the system $X$ given by

$$\longrightarrow x \xrightarrow{0.5} y \xrightarrow{0.7} z.$$  

While $A$ has no run for the path $\bullet \xrightarrow{0.5} \bullet$, the system $X$ has. So $f : A \to X$ is not an open map, even though it is a functional bisimulation, i.e. coalgebra homomorphism for the subdistribution functor.