A Coalgebraic Perspective on Minimization, Determinization and Behavioural Metrics

Filippo Bonchi
ENS de Lyon
France

Mathias Hülsbusch and Barbara König
Universität Duisburg-Essen
Germany

Alexandra Silva
Centrum Wiskunde & Informatica
The Netherlands

***Abstract*—Coalgebra offers a unified theory of state based systems, including infinite streams, labelled transition systems and deterministic automata. In this paper, we use the coalgebraic view on systems to derive, in a uniform way, abstract procedures for checking behavioural equivalence in coalgebras, which perform (a combination of) minimization and determinization.**

First, we show that for coalgebras on categories equipped with *factorization structures*, there exists an abstract procedure for equivalence checking. For instance, when considering epi-mono factorizations in the category of sets and functions, this procedure corresponds to the usual minimization algorithm and two states are behaviourally equivalent if and only if they are mapped to the same state in the minimized coalgebra.

Second, motivated by several examples, we consider coalgebras on categories without suitable factorization structures: under certain conditions, it is possible to apply the above procedure after transforming coalgebras with *reflections*. This transformation can be thought of as some kind of determinization.

Finally, we show that the result of the procedure also induces a pseudo-metric on the states, in such a way that the distance between each pair of states is minimized.

I. Introduction

Finite automata are one of the most basic structures in computer science and much research has been devoted to studying their properties. One particular interesting problem is that of minimization: given a (non-)deterministic finite automaton is there an equivalent one which has a minimal number of states? Minimization is relevant in many areas of computer science such as model checking, concurrency theory, language theory and formal verification.

For deterministic automata (DA's), minimization algorithms are well-known ([1]) whereas for non-deterministic automata (NDA's) the situation is less clear. There have been algorithms proposed to minimize an NDA modulo bisimilarity [2] but when it comes to language equivalence it is known that the minimal automaton is not unique. In order to obtain a unique minimal automaton corresponding to an NDA, a determinization step is needed converting it into a DA. Minimization procedures are even less clear when one moves to more elaborate versions of automata, such as weighted automata [3], which are becoming popular nowadays due to their applicability in several areas of computer science, such as image recognition or speech processing.

It is the main aim of this paper to take a step towards a clearer understanding of minimization algorithms for a large class of automata, including the ones mentioned above. This encompasses two things: (i) cast the automata and the intended equivalence in a general framework; and (ii) use the general framework to devise algorithms to minimize (and determinize) the automata. To study all the types of automata mentioned above (and more) in a uniform setting we use coalgebras.

Coalgebras provide a general framework for the study of dynamical systems such as infinite streams, deterministic and non-deterministic automata. For a functor $F\colon \mathbf{C} \to \mathbf{C}$, an F-coalgebra is a pair (X, α), where X is an object of $\mathbf{C}$ representing the "state space" of the system and $\alpha\colon X \to FX$ is an arrow of $\mathbf{C}$ defining the "transitions" of the states. We call the functor F the type of the system. For instance, DA's can be readily seen to correspond to coalgebras of the functor $2 \times \mathsf{Id}^A\colon \mathbf{Set} \to \mathbf{Set}$ and NDA's to coalgebras of the functor $A \times \mathsf{Id} + 1\colon \mathbf{Rel} \to \mathbf{Rel}$, where $\mathbf{Set}$ is the category of sets and functions and $\mathbf{Rel}$ the category of sets and relations.

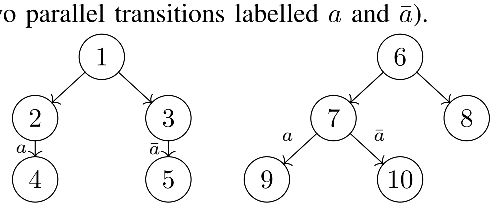
The strength of the coalgebraic modeling lies in the fact that many important notions, such as canonical representatives of behaviour (the so-called final coalgebra) and equivalence, are determined uniquely by the type of the system. Under mild conditions, functors F have a final coalgebra (unique up to isomorphism) into which every F-coalgebra can be mapped via a unique so-called F-homomorphism. The final coalgebra can be viewed as the universe of all possible F-behaviours: the unique homomorphism into the final coalgebra maps every state of a coalgebra to its behaviour. This provides a general notion of behavioural equivalence: two states are equivalent iff they are mapped to the same element of the final coalgebra. In the case of DA's, the final coalgebra is $\mathcal{P}(A^*)$ (the set of all languages over input alphabet A) and the unique morphism is a *function* mapping each state to the language that it accepts. In the case of NDA's, as shown in [4], the final coalgebra is A^* (the set of all finite words on A) and the unique morphism is a *relation* linking each state with all the words that it accepts. In both cases, the induced behavioural equivalence is language equivalence. The base category chosen to model the system plays an important role in the obtained equivalence. For instance, NDA's can alternatively be modelled as coalgebras of the functor $2 \times \mathcal{P}(\mathsf{Id})^A\colon \mathbf{Set} \to \mathbf{Set}$, where $\mathcal{P}$ is the powerset functor, but in this case the induced behavioural equivalence is bisimilarity (which is finer than language equivalence).

For a functor F on $\mathbf{Set}$, the *image* of a certain F-coalgebra through the unique morphism is its *minimal representative* (with respect to the induced behavioural equivalence) that, in the finite case, can be computed via ordinary partition refinement algorithms [5], [6]. For functors on categories not equipped with proper image factorization structures (such as $\mathbf{Rel}$, for instance) the situation is less clear-cut. This general observation instantiates to the well-known fact that for every DA there exists an equivalent minimal automaton, while for NDA's the uniqueness of minimal automata is not guaranteed.

It is our aim to, on the one hand, offer a procedure to perform ordinary partition refinement for categories with suitable factorization structures (such as $\mathbf{Set}$, wherein DA's are modelled). On the other hand, we want to offer an alternative procedure for categories without proper factorization structures: we describe a general setting for determinizations and show how both can be combined to yield a single algorithm that does determinization and minimization simultaneously.

Our work was motivated by several examples, considering coalgebras in various underlying categories. In this paper, we take one example in $\mathbf{Set}$ and three examples in $\mathcal{K}\ell(T)$, the Kleisli category for a monad T. More precisely, we consider DA's in $\mathbf{Set}$ and NDA's in $\mathbf{Rel}$, which is $\mathcal{K}\ell(\mathcal{P})$, where $\mathcal{P}$ is the powerset functor. Moreover, we consider linear weighted automata (LWA), over vector spaces for a field $\mathbb{F}$, which can also be seen as a Kleisli category. For DA's, we recover the usual minimization algorithm. Instantiation to NDA's gives us (a part of) Brzozowski's algorithm [7]: the obtained automaton is *backwards-deterministic*, that is, each state has only one incoming transition labeled by the same letter and there is only one final state. It also has the nice property that the languages recognized by the states form a partition of the set of all words. For LWA's, we obtain Boreale's minimization algorithm [8] as a special case.

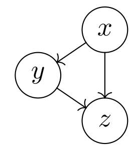
As a new example we consider transition systems labelled with conditions that have similarly been considered in [9], [10]. We model these automata in $\mathcal{K}\ell(T)$, for $T = X^A$ the so-called input monad. Consider the following transition system where transitions are decorated with conditions $a, \bar{a}$, where intuitively $\bar{a}$ stands for "not a". Labelled transitions are either present or absent, depending on whether a or $\bar{a}$ hold. Unlabelled transitions are always present (they can be thought of as two parallel transitions labelled a and $\bar{a}$).



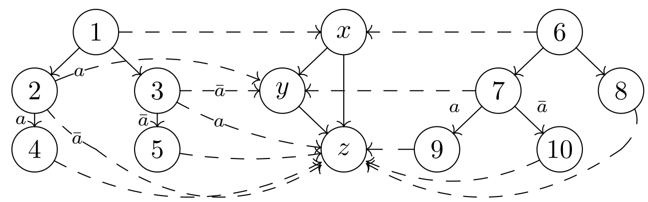
The environment can make one choice: it decides whether to take either a or $\bar{a}$. Once this choice has been made, it can not be changed. Regardless of the specific choice of the environment, the two states 1, 6 will be bisimilar in the resulting unlabelled transition systems, containing non-deterministic branching.

One possible way to solve the question whether two states are *always* bisimilar is to enumerate all conditions and to create suitably many instantiations of the transition system. This will be shown to be closely related to (backwards) determinizations, since both constructions are categorical reflections. Then, in a second step, the resulting transition system can be minimized with respect to bisimilarity. We will discuss both constructions separately, in a general setting, and also show how they can be combined into a single algorithm.

The adequacy of the combined determinization/minimization algorithm is further strengthened by considering behavioural metrics on the state spaces. More specifically, we show that the mapping into the minimized coalgebra induces a pseudo-metric that provides minimal distances for all pairs of states. In the example above, the minimization is of the form:



The arrow from the original transition system into the one above maps both 1 and 6 to x, 7 to y, but 2 to y whenever a holds and to z whenever $\bar{a}$ holds. (Remember that we are not working in $\mathbf{Set}$, but in a suitable Kleisli category that allows us to express such effects.) The full mapping is represented below.



In this case, distances are sets of conditions. The metric we consider is such that while states 2 and 7 have distance $\{a, \bar{a}\}$ in the original coalgebra (they are completely different), the distance of their images in the minimization decreases to $\{\bar{a}\}$ (their behaviour is only different if $\bar{a}$ holds).

In summary, the contributions of the present article are:

1) an algorithm that generalizes the usual refinement-based minimization or bisimulation checking algorithms to categories with a suitable factorization structure;
2) a general setting for determinization-like constructions, which we show how to combine with 1) to yield a single algorithm that does determinization and minimization simultaneously;
3) a proof that the mapping to the minimization induces a pseudo-metric which provides minimal distances for all pairs of states;
4) an extensive set of interesting examples, for which we work out the details of the constructions above.

In this paper we use four examples to illustrate various aspects of the theory. In order to make it simpler for the reader to keep the examples apart we will use the following abbreviations: (CTS) conditional transition systems (the last example discussed in the introduction); (DA) deterministic automata; (NDA) non-deterministic automata and (LWA) linear weighted automata [8]. Proofs can be found in Appendix D.

II. Background material on coalgebras

We assume some prior knowledge of category theory (categories, functors, monads, limits and adjunctions). Definitions can be found in [11]. However, to establish some notation, we recall some basic definitions. Let $\mathbf{Set}$ be the category of sets and functions. Sets (and other objects) are denoted by capital letters $X, Y, \ldots$ and functions (and other arrows) by lower case $f, g, \ldots, \alpha, \beta, \ldots$. We write $\emptyset$ for the empty set, 1 for the singleton set, typically written as $1 = \{\bullet\}$, and 2 for the two elements set $2 = \{0, 1\}$. The collection of all subsets of a set X is denoted by $\mathcal{P}(X)$ and the collection of all countable subsets of X by $\mathcal{P}_c(X)$. The collection of functions from a set X to a set Y is denoted by Y^X. We write $g \circ f$ for function composition, when defined. The product of two sets X, Y is written as $X \times Y$, while the coproduct, or disjoint union, of two sets X, Y is usually written as $X + Y$. These operations, defined on sets, can analogously be defined on functions, yielding (bi-)functors.

Definition 1 (Concrete Category [11]). *A concrete category (over* $\mathbf{Set}$*) is a pair* $(\mathbf{C}, U)$*, where* $\mathbf{C}$ *is a category and* $U: \mathbf{C} \to \mathbf{Set}$ *is a faithful functor.*

Definition 2 (Coalgebra). *Given an endofunctor* $F: \mathbf{C} \to \mathbf{C}$ *an* (F-)coalgebra *is a pair* (X, α)*, where* X *is an object of* $\mathbf{C}$ *and* $\alpha: X \to FX$ *an arrow in* $\mathbf{C}$*. A* coalgebra morphism $f: (X, \alpha) \to (Y, \beta)$ *between two coalgebras* $\alpha: X \to FX$ *and* $\beta: Y \to FY$ *is a* $\mathbf{C}$*-arrow* $f: X \to Y$ *such that* $Ff \circ \alpha = \beta \circ f$*.* F*-coalgebras and their morphisms form a category.*

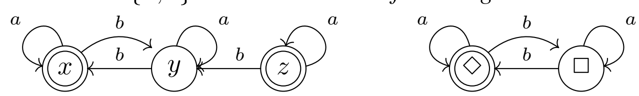
When it is clear from the context, we will often use α instead of (X, α) to refer to the coalgebra.

An F-coalgebra (Ω, ω) is *final* if for any F-coalgebra (X, α) there exists a unique morphism $beh_X: (X, \alpha) \to (\Omega, \omega)$. If a final F-coalgebra exists and $\mathbf{C}$ is concrete, we can define behavioural equivalence. For F-coalgebras (X, α) and (Y, β), $x \in UX$, $y \in UY$, we say that x and y are *behaviourally equivalent*, written $x \approx y$, if and only if $U(beh_X)(x) = U(beh_Y)(y)$.

Example 3. *(DA) A deterministic automaton over the input alphabet* A *is a pair* (X, α)*, where* X *is a set of states and* $\alpha: X \to 2 \times X^A$ *is a function that to each state* x *associates a pair* $\alpha(x) = \langle o_x, t_x \rangle$*, where* o_x*, the output value, determines if a state* x *is final* ($o_x = 1$) *or not* ($o_x = 0$)*; and* t_x*, the transition function, returns for each input letter* $a \in A$ *the next state. DA's are coalgebras for the functor* $FX = 2 \times X^A$ *on* $\mathbf{Set}$*. The final coalgebra of this functor is* $(\mathcal{P}(A^*), \omega)$ *where* $\mathcal{P}(A^*)$ *is the set of languages over* A *and, for a language* L*,* $\omega(L) = \langle \varepsilon_L, L_a \rangle$*, where* ε_L *determines whether or not the empty word is in the language* ($\varepsilon_L = 1$ *or* $\varepsilon_L = 0$*, resp.) and, for each input letter* a*,* L_a *is the* derivative *of* L*:* $L_a = \{w \in A^* \mid aw \in L\}$*. From any DA* (X, α)*, there is a unique map* beh_X *into* $\mathcal{P}(A^*)$ *which assigns to each state its behaviour (that is, the language that the state recognizes). Two states are behaviourally equivalent iff they accept the same language.*

$$\begin{array}{ccc} X & \xrightarrow{\;beh_X\;} & \mathcal{P}(A^*) \\ \alpha \downarrow & & \downarrow \omega \\ 2 \times X^A & \xrightarrow[id \times beh_X^A]{} & 2 \times \mathcal{P}(A^*)^A \end{array}$$

Take $A = \{a, b\}$ *and consider the following DA's.*



We call the leftmost (X, α) *where* $X = \{x, y, x\}$ *and* $\alpha: X \to 2 \times X^A$ *maps* x *to the pair* $\langle 1, \{a \mapsto x,\ b \mapsto y\} \rangle$*,* y *to* $\langle 0, \{a \mapsto y,\ b \mapsto x\} \rangle$ *and* z *to* $\langle 1, \{a \mapsto z,\ b \mapsto y\} \rangle$*. The rightmost is* (Z, γ) *where* $Z = \{\Diamond, \Box\}$ *and* $\gamma: Z \to 2 \times Z^A$ *maps* $\Diamond$ *to* $\langle 1, \{a \mapsto \Diamond,\ b \mapsto \Box\} \rangle$ *and* $\Box$ *to* $\langle 0, \{a \mapsto \Box,\ b \mapsto \Diamond\} \rangle$*. As an example of a coalgebra morphism consider the function* $e: X \to Z$ *mapping* x, z *to* $\Diamond$ *and* y *to* $\Box$*.*

Non-deterministic automata (NDA) can be described as coalgebras for the functor $\mathcal{F}X = 2 \times \mathcal{P}(X)^A$ (on $\mathbf{Set}$): to each input in A, we assign a set in $\mathcal{P}(X)$ of possible successors states. Unfortunately, the resulting behavioural equivalence is not language equivalence (as for DA), but bisimilarity (i.e., it only identifies states having the same branching structure). In [4], [12] it is shown that in order to retrieve language equivalence for NDA's, one should consider coalgebras on a Kleisli category.

In what follows, we introduce Kleisli categories, in which we model non-deterministic automata and conditional transition systems as coalgebras. While objects in a Kleisli category are sets, arrows are generalized functions that incorporate side effects, specified by a monad. For the definition of a monad see Appendix A or [4], [13].

Definition 4 (Kleisli Category). *Let* $(T: \mathbf{Set} \to \mathbf{Set}, \eta, \mu)$ *(or simply* T*) be a monad on* $\mathbf{Set}$*. Its* Kleisli category $\mathcal{K}\ell(T)$ *has sets as objects and an arrow* $X \to Y$ *in* $\mathcal{K}\ell(T)$ *is a function* $X \to TY$*. The identity* id_X *is* η_X *and the composition* $g \circ f$ *of two arrows* $f: X \to Y$*,* $g: Y \to Z$ *(given as arrows* $f: X \to TY$*,* $g: Y \to TZ$ *in* $\mathbf{Set}$*) is* $\mu_Z \circ Tg \circ f$*.*

Note that in the following we will employ overloading and use the same letter to both denote an arrow in $\mathcal{K}\ell(T)$ and the corresponding arrow in $\mathbf{Set}$. We will usually specify which arrow is meant in order to avoid confusion. Furthermore, note that $\mathbf{Set}$ can be seen as a subcategory of $\mathcal{K}\ell(T)$, where each function $f: X \to Y$ is identified with $\eta_Y \circ f$.

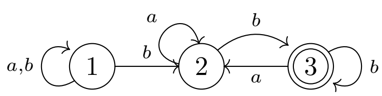
Every Kleisli category $\mathcal{K}\ell(T)$ is a concrete category where $UX = TX$ and $Uf = \mu_X \circ Tf$ for an object X and an arrow $f: X \to Y$.

To define coalgebras over Kleisli categories we need the notion of lifting of a functor, which we define here directly, but could otherwise be specified via a distributive law (for details see [4], [14]): a functor $\overline{F}: \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ is called a *lifting of* $F: \mathbf{Set} \to \mathbf{Set}$ whenever it coincides with F on $\mathbf{Set}$, seen as a subcategory of $\mathcal{K}\ell(T)$.

Since F and $\overline{F}$ coincide on objects, $\overline{F}$-coalgebras in a Kleisli category $\mathcal{K}\ell(T)$ are of the form $X \to TFX$, where intuitively the functor F describes the explicit branching, i.e. choices which are visible to the observer, and the monad T the implicit branching, i.e. side-effects, which are there but cannot be observed directly. In this way, the implicit branching is part of the underlying category and is also present in the arrow from any coalgebra into the final coalgebra. As in functional programming languages such as Haskell, the idea is to "hide" computational effects underneath a monad and to separate them from the (functional) behaviour as much as possible.

Example 5. *(NDA) Consider the powerset monad $TX = \mathcal{P}(X)$, fully described in Example 32 (Appendix A). The Kleisli category $\mathcal{K}\ell(\mathcal{P})$ coincides with the category $\mathbf{Rel}$ of sets and relations. As an example of a lifting, take $FX = A \times X + 1$ in $\mathbf{Set}$ (with $1 = \{\bullet\}$). The functor F lifts to $\overline{F}$ in $\mathbf{Rel}$ as follows: For any $f\colon X \to Y$ in $\mathbf{Rel}$ (that is $f\colon X \to \mathcal{P}(Y)$ in $\mathbf{Set}$), $\overline{F}f\colon A \times X + 1 \to A \times Y + 1$ is defined as $\overline{F}f(\bullet) = \{\bullet\}$ and $\overline{F}f(\langle a, x\rangle) = \{\langle a, y\rangle \mid y \in f(x)\}$.*

Non-deterministic automata over the input alphabet A can be regarded as coalgebras on $\mathbf{Rel}$ for the functor $\overline{F}$ (described above). A coalgebra $\alpha\colon X \to \overline{F}X$ is a function $\alpha\colon X \to \mathcal{P}(A \times X + 1)$, which assigns to each state $x \in X$ a set which contains $\bullet$ if x is final and $\langle a, y\rangle$ for all transitions $x \xrightarrow{a} y$. For instance, the automaton



corresponds to the coalgebra (X, α), where $X = \{1, 2, 3\}$ and $\alpha\colon X \to \mathcal{P}(\{a, b\} \times X + \{\bullet\})$ is defined as follows: $\alpha(1) = \{\langle a, 1\rangle, \langle b, 1\rangle, \langle b, 2\rangle\}$, $\alpha(2) = \{\langle a, 2\rangle, \langle b, 3\rangle\}$ and $\alpha(3) = \{\bullet, \langle a, 2\rangle, \langle b, 3\rangle\}$.

In [4], it is shown that the final $\overline{F}$-coalgebra (in $\mathbf{Rel}$) is the set A^* of words. For an NDA (X, α), the unique coalgebra morphism beh_X into A^* is a relation that links every state in X with all the words in A^* that it accepts.

Example 6. *(CTS) We shortly discuss how to specify the example from the introduction in a Kleisli category.*

We use the input monad $TX = X^A$, where A is a set of conditions or inputs (for the example of the introduction $A = \{a, \bar{a}\}$). Given a function $f\colon X \to Y$, $Tf\colon TX \to TY$ is $f^A\colon X^A \to Y^A$ defined for all $g \in X^A$ and $a \in A$ as $f^A(g)(a) = f(g(a))$.

Note that an arrow $f\colon X \to Y$ in the Kleisli category over the input monad is a function $f\colon X \to Y^A$. For instance, the dashed arrows in the example of the introduction describe an arrow in $\mathcal{K}\ell(T)$: state 2 is mapped to y if condition a holds and to z if $\bar{a}$ holds.

We will use the countable powerset functor $FX = \mathcal{P}_c(X)$ as endofunctor, which is lifted to $\mathcal{K}\ell(T)$ as follows: an arrow $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, which is a function of the form $f\colon X \to Y^A$, is mapped to $\overline{F}f\colon \mathcal{P}_c(X) \to \mathcal{P}_c(Y)$ with $\overline{F}f(X')(a) = \{f(x)(a) \mid x \in X'\}$ for $X' \subseteq X$, $a \in A$.

Hence, the example from the introduction is modelled by an arrow $\alpha\colon X \to \mathcal{P}_c(X)$ in $\mathcal{K}\ell(T)$ (i.e., a function $\alpha\colon X \to \mathcal{P}_c(X)^A$), where $X = \{1, \dots, 10\}$ and $A = \{a, \bar{a}\}$. For instance $\alpha(1)(a) = \alpha(1)(\bar{a}) = \{2, 3\}$, $\alpha(2)(a) = \{4\}$, $\alpha(2)(\bar{a}) = \emptyset$. The entire coalgebra α is represented by:

α	1	2	3	4	5	6	7	8	9	10
a	$\{2,3\}$	$\{4\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{7,8\}$	$\{9\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\bar{a}$	$\{2,3\}$	$\emptyset$	$\{5\}$	$\emptyset$	$\emptyset$	$\{7,8\}$	$\{10\}$	$\emptyset$	$\emptyset$	$\emptyset$

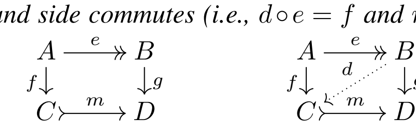
Note that the above $\alpha\colon X \to \mathcal{P}_c(X)^A$ can be seen as a coalgebra for the functor $FX = \mathcal{P}_c(X)^A$ in $\mathbf{Set}$, which yields ordinary A-labelled transition systems, where intuitively all branching decisions are explicit. However, the resulting behavioural equivalence (that is, ordinary bisimilarity) would be inadequate for our intuition, since it would distinguish the states 1 and 6. In Example 30, we will show that 1 and 6 are behaviourally equivalent in $\overline{F}$-coalgebras in $\mathcal{K}\ell(T)$.

III. "Minimization" via $(\mathcal{E}, \mathcal{M})$-Factorizations

We will now present the construction of the minimized coalgebra that is intended to mimic the minimization of transition systems via partition refinement in a general setting. This notion is parametrized by two classes $\mathcal{E}, \mathcal{M}$ of morphisms that form a factorization structure for the category $\mathbf{C}$ under consideration.

Definition 7 (Factorization Structures)**.** *Let $\mathbf{C}$ be a category and let $\mathcal{E}$, $\mathcal{M}$ be classes of morphisms in $\mathbf{C}$. The pair $(\mathcal{E}, \mathcal{M})$ is called a* factorization structure *for morphisms in $\mathbf{C}$ (and $\mathbf{C}$ is called $(\mathcal{E}, \mathcal{M})$-*structured*) whenever*

- *$\mathcal{E}$ and $\mathcal{M}$ are closed under composition with isos.*
- *$\mathbf{C}$ has* $(\mathcal{E}, \mathcal{M})$-factorizations *of morphisms, i.e., each morphism f of $\mathbf{C}$ has a factorization $f = m \circ e$ with $e \in \mathcal{E}$ and $m \in \mathcal{M}$.*
- *$\mathbf{C}$ has the unique* $(\mathcal{E}, \mathcal{M})$-diagonalization property*, i.e., for each commutative square as shown on the left-hand side below with $e \in \mathcal{E}$ and $m \in \mathcal{M}$ there exists a unique* diagonal*, i.e., a morphism d such that the diagram on the right-hand side commutes (i.e., $d \circ e = f$ and $m \circ d = g$).*



The classical example of $(\mathcal{E}, \mathcal{M})$-factorization in $\mathbf{Set}$ is the factorization of a function f into a surjective and an injective function (epi-mono factorization). In the following, morphisms from $\mathcal{E}$ are drawn using double-headed arrows of the form $A \twoheadrightarrow B$, whereas morphisms from $\mathcal{M}$ are depicted using arrows of the form $A \rightarrowtail B$.

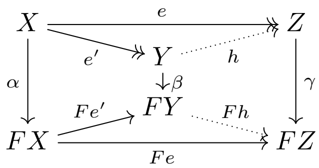
In any $(\mathcal{E}, \mathcal{M})$-structured category $(\mathcal{E}, \mathcal{M})$-factorizations of morphisms are unique up to iso. Furthermore the classes $\mathcal{E}, \mathcal{M}$ are both closed under composition. For more details see [11].

We will also use the classes $\mathcal{E}$ and $\mathcal{M}$ to classify coalgebra morphisms, depending on whether the underlying $\mathbf{C}$-arrow belongs to one of the classes. Whenever the endofunctor F preserves $\mathcal{M}$-arrows, which will be assumed in the following, the factorization structure can be straightforwardly lifted to coalgebra morphisms (see Lemma 37 in Appendix D or [15]).

We are now ready to define the minimization. We assume that a factorization structure $(\mathcal{E}, \mathcal{M})$ has been fixed.

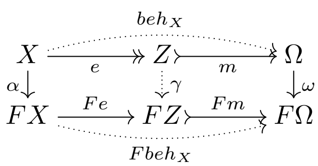
Definition 8 (Minimization)**.** *Let $\alpha\colon X \to FX$ be a coalgebra. We call $\gamma\colon Z \to FZ$ the minimization for α if*

- *There exists a coalgebra morphism $e\colon (X,\alpha) \to (Z,\gamma)$, which is contained in $\mathcal{E}$.*
- *Furthermore, for any other coalgebra morphism $e'\colon (X,\alpha) \to (Y,\beta)$, which is in $\mathcal{E}$, there exists a unique coalgebra morphism $h\colon (Y,\beta) \to (Z,\gamma)$ such that $e = h \circ e'$.*



Note that minimization is unique up to isomorphism if it exists. Furthermore, while in **Set** the minimization will also be minimal in the number of states, this is not necessarily true for other categories.

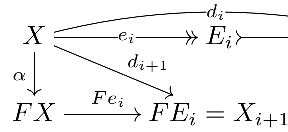
Proposition 9 (Minimization and Final Coalgebras)**.** *If the final coalgebra $\omega\colon \Omega \to F\Omega$ exists, then – for a given coalgebra $\alpha\colon X \to FX$ – the minimization $\gamma\colon Z \to FZ$ for α can be obtained by factoring the unique coalgebra morphism $beh_X\colon (X,\alpha) \to (\Omega,\omega)$ into an $\mathcal{E}$-morphism and an $\mathcal{M}$-morphism.*



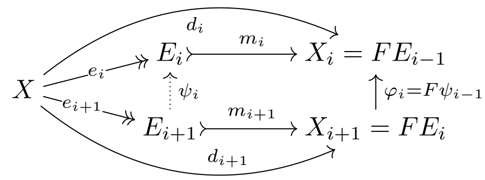
Example 10 (DA, Minimal Automata)**.** *Recall that DA's are coalgebras for the functor $FX = 2 \times X^A$ on* **Set** *(Example 3) and take surjective-injective factorization of* **Set**. *In this case, minimizations correspond to the well known minimal deterministic automata. For instance, the minimization of the (leftmost) automaton (X,α) in Example 3 is the automaton (Z,γ) (on its right).*

We now describe a construction that – given a coalgebra α – obtains the minimization γ without going via the final coalgebra. This closely resembles the partition refinement algorithm for minimizing deterministic automata or for computing bisimilarity. Whenever the limit in the construction below exists, we obtain the minimization. In many examples the constructed sequence might even become stationary after finitely many steps. The construction is reminiscent of the construction of the final coalgebra given by Adámek and Koubek in [16]. We believe that, as in [16], our construction can be generalized to ordinals beyond ω. In the following, we assume that the category $\mathbf{C}$ has a final object 1.

Construction 11. *Let $\alpha\colon X \to FX$ be a coalgebra.*
Step 0*: Take the unique arrow $d_0\colon X \to 1$ $(= X_0)$.*
Step $\mathbf{i+1}$*: Assume that $d_i\colon X \to X_i$ is given. Take the unique $(\mathcal{E},\mathcal{M})$-factorization of d_i as $d_i = e_i \circ m_i$, where $e_i\colon X \twoheadrightarrow E_i$, $m_i\colon E_i \rightarrowtail X_i$. Then set $d_{i+1} = Fe_i \circ \alpha$ (see diagram below). Note that $X_{i+1} = FE_i$.*

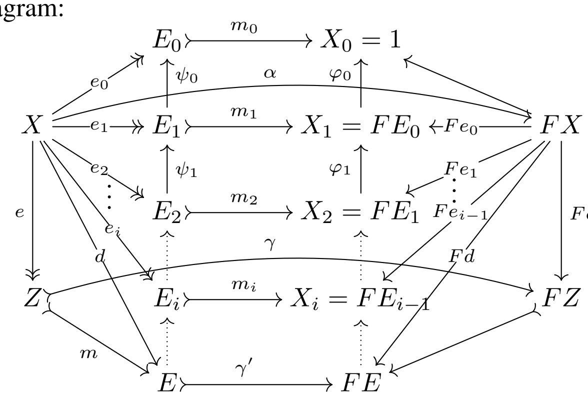


Take the sequences of arrows $\varphi_i\colon X_{i+1} \to X_i$ and $\psi_i\colon E_{i+1} \to E_i$ defined as follows: (a) $\varphi_0\colon X_1 \to 1$ $(= X_0)$ is the unique arrow, (b) $\varphi_i = F\psi_{i-1}$ for $i > 0$ and (c) ψ_i is the unique diagonal as depicted below.



Now take the limits of the sequences ψ_i and φ_i and under the assumption that F preserves the limit, we obtain E, FE and $\gamma'\colon E \to FE$ as mediating arrow. Furthermore there is a coalgebra morphism $d\colon (X,\alpha) \to (E,\gamma)$. Factoring $d = m \circ e$ with $m\colon X \to Z \in \mathcal{M}$, $e\colon Z \to E \in \mathcal{E}$ gives us the resulting coalgebra $\gamma\colon Z \to FZ$ as diagonal.[1]

The entire construction can be summarized in the following diagram:



Theorem 12. *If the limits in Construction 11 exist and the endofunctor F preserves the limit, then the coalgebra $\gamma\colon Z \to FZ$ is the minimization for α.*

In some cases it is straightforward to show that F preserves the limit of the ψ_i: in our examples the sequence will usually become stationary, i.e., consist only of isos from some point onwards. In this case the arrow d is already an element of $\mathcal{E}$ and the last factorization step can be omitted. In this case $\gamma = \varphi_n^{-1} \circ m_n\colon E_n \to FE_n$, where n is the index from which onwards the arrows ψ_i, φ_i are isos.

For any category if we take as $\mathcal{E}$ the class of all arrows and $\mathcal{M}$ the class of all isos we recover the standard construction of the final coalgebra, but, in this case, the construction never terminates in finitely many steps. For a coalgebra in **Set**, with $\mathcal{M}$ the injections and $\mathcal{E}$ the surjections the procedure described above is nothing else than the classical partition refinement algorithm, which we illustrate next for deterministic automata.

[1] Note that the arrow γ' must be an $\mathcal{M}$-arrow, since $\mathcal{M}$-arrows are preserved by limits. Then γ must also be an $\mathcal{M}$-arrow (due to Proposition 14.9 in [11]). However, as far as we know, d need not necessarily be an arrow in $\mathcal{E}$.

Example 13. *(DA) We apply the construction to the DA (X, α) of Example 3: at the beginning the function $d_0 \colon X \to 1$ maps x, y, z to the singleton $\bullet$. By factorizing d_0, we obtain $e_0 \colon X \twoheadrightarrow 1$ and $m_0 \colon 1 \rightarrowtail 1$ (both uniquely defined). The surjection e_0 corresponds to the partition $\{x, y, z\}$ (i.e., the partition where all the states are equivalent).*

Then, $d_1 \colon X \to 2$ $(= 2 \times 1^A)$ maps x and z to 1 (since both states are final) and y to 0. By factoring d_1, we obtain $e_1 \colon X \twoheadrightarrow \{\Diamond, \Box\}$ (mapping x, z to $\Diamond$ and y to $\Box$) and $m_i \colon \{\Diamond, \Box\} \rightarrowtail 2$. The surjection e_i corresponds to the partition $\{x, z\}, \{y\}$ (i.e., the partition equating x and z).

Finally, $d_2 \colon X \to 2 \times \{\Diamond, \Box\}^A$, maps x and z to the pair $\langle 1, \{a \mapsto \Diamond,\ b \mapsto \Box\}\rangle$ and y to $\langle 0, \{a \mapsto \Box,\ b \mapsto \Diamond\}\rangle$. By factoring d_2, we obtain $e_2 \colon X \twoheadrightarrow \{\Diamond, \Box\}$ (mapping x, z to $\Diamond$ and y to $\Box$) and $m_2 \colon \{\Diamond, \Box\} \rightarrowtail 2 \times \{\Diamond, \Box\}^A$. Note that m_2 is exactly the function γ of Example 3 and thus $(\{\Diamond,\ \Box\}, m_2)$ is the DA (Z, γ) which, as shown in Example 10, is the minimization of (X, α).

Example 14. *(LWA) We study automata with weights taken from a field (linear weighted automata, see [8]). Consider the Kleisli category $\mathcal{K}\ell(T)$ for the monad $T \colon \mathbf{Set} \to \mathbf{Set}$ where $TX = (\mathbb{F}^X)_\omega$, where $(\mathbb{F}^X)_\omega$ denotes the set of all mappings from X into $\mathbb{F}$ with finite support. For a function $f \colon X \to Y$ in $\mathbf{Set}$ define $Tf \colon TX \to TY$ as follows: let $a \in (\mathbb{F}^X)_\omega$, then*

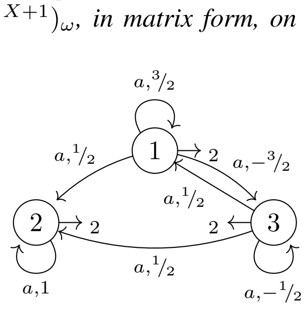
$$Tf(a)(y) = \sum \{a(x) \mid x \in X, f(x) = y\}$$

If we restrict to finite sets, we obtain the category of finite-dimensional vector spaces: a Kleisli arrow $X \to Y$ for finite sets X, Y is a matrix with entries from $\mathbb{F}$, where the columns are indexed by X and the rows are indexed by Y. If we view a Kleisli arrow as a function $TX \to TY$ we obtain exactly the linear maps from an $|X|$-dimensional vector space into a $|Y|$-dimensional vector space (both over $\mathbb{F}$).

For a set A of labels we take the endofunctor $FX = A \times X + 1$ on $\mathbf{Set}$ where A is a fixed set of labels and $\bullet$ – denoting termination – stands for the element of the one-element set 1. This functor is lifted to $\mathcal{K}\ell(T)$ as follows: an arrow $f \colon X \to Y$ in $\mathcal{K}\ell(T)$, which is a function of the form $f \colon X \to (\mathbb{F}^Y)_\omega$, is mapped to $\overline{F}f \colon A \times X + 1 \to A \times Y + 1$ with $\overline{F}f(\langle a, x\rangle)(\langle a, y\rangle) = f(x)(y)$, $\overline{F}f(\bullet)(\bullet) = 1$ and 0 in all other cases. Hence transition carry labels from A (for the explicit branching) and weights (for the implicit branching).

An example LWA for $A = \{a\}$ and $\mathbb{F} = \mathbb{R}$ (taken from [8]) is shown below (graphical representation on the right and coalgebra $\alpha \colon X \to (\mathbb{R}^{A \times X + 1})_\omega$, in matrix form, on the left):

$$\begin{array}{c} \\ \langle a,1\rangle \\ \langle a,2\rangle \\ \langle a,3\rangle \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 3/2 & 0 & 1/2 \\ 1/2 & 1 & 1/2 \\ -3/2 & 0 & -1/2 \\ 2 & 2 & 2 \end{pmatrix} \end{array}$$



As factorization structure we use as $\mathcal{E}$-arrows the matrices of full row rank (i.e., the monos) and as $\mathcal{M}$-arrows the matrices of full column rank (i.e., the epis). Let E be the morphism (matrix) into the minimization: two vectors $\mathbf{x}, \mathbf{y}$ satisfy $E\mathbf{x} = E\mathbf{y}$ iff they are equivalent in the sense of [8] (see Appendix C for an elaboration of this claim and for an example minimization involving the automaton above).

Boreale also observes that bisimilarity in his setting coincides with trace equivalence, which is consistent with the intuition behind implicit and explicit branching: here the endofunctor F does not provide any explicit branching, but simply observations in A.

IV. "Determinization" via Reflections

For several categories there are no suitable factorization structures. This can for instance be observed in $\mathbf{Rel}$, wherein we model non-deterministic automata as coalgebras. It is known that there is no unique minimal non-deterministic automata. The usual procedure is to first construct the corresponding deterministic automaton (via the powerset construction), which is then minimized in a second step. In this section, we will give a general framework for determinization-like constructions in the form of reflections, which can also be applied to other settings, such as the example discussed in the introduction. For non-deterministic automata we will obtain an automaton which is "backward-deterministic", i.e., for every state and each letter there is exactly one predecessor. Then we will show how reflections can be combined with the minimization construction. For the sake of the reader we will define the frequently used concept of a reflective subcategory.

Definition 15 (Reflective Subcategory)**.** *Let $\mathbf{S}$ be a subcategory of $\mathbf{C}$. Let X be an object of $\mathbf{C}$. An* $\mathbf{S}$-reflection *for X is an arrow $\eta_X \colon X \to X'$, where X' is an $\mathbf{S}$-object, such that for every other arrow $f \colon X \to Y$ with Y in $\mathbf{S}$ there exists a unique $\mathbf{S}$-morphism f' such that $f = f' \circ \eta_X$. $\mathbf{S}$ is called a* reflective subcategory *of $\mathbf{C}$ whenever each $\mathbf{C}$-object has an $\mathbf{S}$-reflection.*

Note that this definition is equivalent to saying that the functor embedding $\mathbf{S}$ into its host category $\mathbf{C}$ has a left adjoint $L \colon \mathbf{C} \to \mathbf{S}$. The arrows η_X form the unit of this adjunction.

It is important to remark here that the unit η will in our examples *not* coincide with the natural transformation η of the monad T. It is well-known that for a monad $T \colon \mathbf{Set} \to \mathbf{Set}$ the category $\mathbf{Set}$ is coreflective in $\mathcal{K}\ell(T)$, whereas here we need a reflective subcategory.

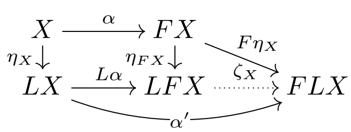
We will first study two examples of reflections: a reflective subcategory of $\mathbf{Rel}$ (needed for NDA's) and a reflective subcategory of $\mathcal{K}\ell(T)$, for T the input monad (needed for CTS's).

Example 16. *(NDA) The category of inverse functions $\mathbf{Set}^{\mathbf{op}}$ is a reflective subcategory of $\mathbf{Rel}$. The reflection L is the inverse powerset functor, i.e., for a relation $R \colon X \to Y$ we have $L(R) \colon \mathcal{P}(X) \to \mathcal{P}(Y)$ in $\mathbf{Set}^{\mathbf{op}}$ where $L(R)$, seen as an inverse function, maps $Y' \subseteq Y$ to $R^{-1}(Y')$. The adjunction* *has as unit $\eta_X : X \to \mathcal{P}(X)$, which relates an element $x \in X$ with $X' \subseteq X$ if and only if $x \in X'$.*

(CTS) For $\mathcal{K}\ell(T)$ where T is the input monad, we have the following situation: since every function $f : X \to Y^A$ corresponds to a function $f' : A \times X \to Y$ by currying, the Kleisli category over the input monad is isomorphic to the co-Kleisli category over the comonad $VX = A \times X$ on $\mathbf{Set}$. Hence $\mathbf{Set}$ is both reflective and coreflective in $\mathcal{K}\ell(T)$. The unit of the reflection is the Kleisli arrow $\eta_X : X \to A \times X$ with $\eta_X(x)(a) = \langle a, x\rangle$. The reflection L coincides with V on objects and takes the product of the state set X with the label set A. More concretely, for an arrow $f : X \to Y$ in $\mathcal{K}\ell(T)$ we obtain an arrow $Lf : A \times X \to A \times Y$ in $\mathbf{Set}$ with $Lf(\langle a, x\rangle) = \langle a, f(x)(a)\rangle$.

Now we are ready to define the reflection of a coalgebra into the subcategory.

Definition 17 (Reflection of Coalgebras)**.** *Let $\mathbf{S}$ be a reflective subcategory of a category $\mathbf{C}$ and let $L : \mathbf{C} \to \mathbf{S}$ be the left adjoint of the embedding functor. Assume that $\mathbf{S}$ is preserved by the endofunctor F. Then, given a coalgebra $\alpha : X \to FX$ in $\mathbf{C}$ we reflect it into $\mathbf{S}$, obtaining a coalgebra $\alpha' : LX \to FLX$ by the following construction:*



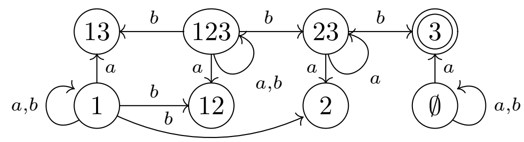
Note that η is the unit of the adjunction and that ζ_X is the mediating morphism that exists since F preserves $\mathbf{S}$ and hence FLX is an object of $\mathbf{S}$.

Proposition 18. *Let $\mathbf{S}$ be a reflective subcategory of $\mathbf{C}$, which is preserved by the endofunctor F. The category of F-coalgebras in $\mathbf{S}$ is a reflective subcategory of the category of F-coalgebras in $\mathbf{C}$.*

The result above is a special case of Corollary 2.15 by Hermida and Jacobs [17].

Note that a limit in a reflective subcategory $\mathbf{S}$ is also a limit in the entire category $\mathbf{C}$. Hence, if the final coalgebra exists in the subcategory $\mathbf{S}$, it is also the final coalgebra in $\mathbf{C}$.

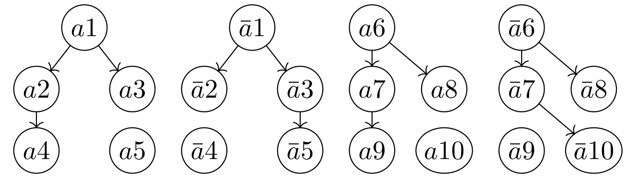
Example 19. *(NDA) We will first study the effect of a reflection on a non-deterministic automaton, for which we use the reflective subcategory $\mathbf{Set}^{\mathbf{op}}$ of $\mathbf{Rel}$ (see Example 16). The effect of the reflection on coalgebras is a powerset automaton which is however "backwards-deterministic": more specifically, given a coalgebra $\alpha : X \to A \times X + 1$ in $\mathcal{K}\ell(\mathcal{P}) = \mathbf{Rel}$, the reflected coalgebra $\alpha' : \mathcal{P}(X) \to A \times \mathcal{P}(X) + 1$ is a relation which lives in $\mathbf{Set}^{\mathbf{op}}$ and, when seen as a function, maps $\langle a, X'\rangle$ with $X' \subseteq X$ to $\{x \in X \mid \exists x' \in X' : \langle a, x'\rangle \in \alpha(x)\}$ (the set of a-predecessors of X') and $\bullet$ to $\{x \in X \mid \bullet \in \alpha(x)\}$ (the set of final states, the unique final state of the new automaton). For instance, the reflection of the NDA (X, α) in Example 5 is the following backwards-deterministic automaton*



Note that the above automaton has a single final state (consisting of the set of final states of the original automaton) and every state has a unique predecessor for each alphabet letter. For this reason, it can be seen as a function $\alpha' : A \times X + 1 \to X$ (i.e., an algebra for the functor $FX = A \times X + 1$).

Note that $\mathbf{Set}$ is not *a reflective subcategory of $\mathbf{Rel}$ – it is instead coreflective – and hence both categories have different final coalgebras. However for the reflective subcategory $\mathbf{Set}^{\mathbf{op}}$, we have exactly the same final coalgebra as for $\mathbf{Rel}$, which, as shown in [4], is the* initial algebra *in $\mathbf{Set}$.*

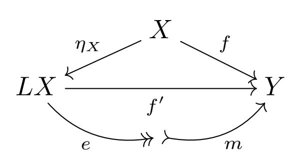
(CTS) Now we come back to the Kleisli category $\mathcal{K}\ell(T)$ over the input monad T (see Example 6) and coalgebras with endofunctor $\mathcal{P}_c$. As discussed in Example 16, $\mathbf{Set}$ is a reflective subcategory of $\mathcal{K}\ell(T)$. On coalgebras reflection has the following effect: given a coalgebra $\alpha : X \to \mathcal{P}_c(X)$ in $\mathcal{K}\ell(T)$ we obtain a reflected coalgebra $\alpha' : A \times X \to \mathcal{P}_c(A \times X)$ in $\mathbf{Set}$ with $\alpha'(\langle a, x\rangle) = \{\langle a, x'\rangle \mid x' \in \alpha(x)(a))\}$. That is, we generate the disjoint union of $|A|$ different transition systems, each of which describes the behaviour for some $a \in A$. For instance, the reflection of the CTS from the introduction (formally introduced in Example 6) is the following.



We now consider other forms of factorizations that do not conform to the conditions of Definition 7. They make use of a reflective subcategory with a "good" factorization structure.

Definition 20 (Pseudo-Factorization)**.** *Let $\mathbf{C}$ be a category and let $\mathbf{S}$ be a reflective subcategory which is $(\mathcal{E}, \mathcal{M})$-structured.*

*Let $f : X \to Y$ be an arrow of $\mathbf{C}$ where Y is an object of $\mathbf{S}$. Now take the unique arrow $f' : LX \to Y$ with $f' \circ \eta_X = f$ (which exists due to the reflection) and factor $f' = m \circ e$ with $m \in \mathcal{M}$, $e \in \mathcal{E}$. Then the decomposition $f = m \circ c$ with $c = e \circ \eta_X$ is called the $(\mathcal{E}, \mathcal{M})$-*pseudo-factorization *of f.*

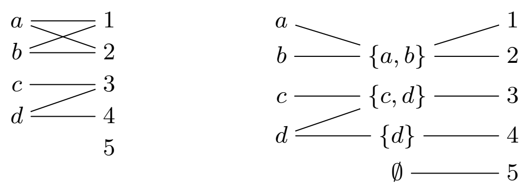


Example 21. *We study pseudo-factorizations, using the reflections of Example 19.*

(NDA) For $\mathbf{Set}^{\mathbf{op}}$, the reflective subcategory of $\mathbf{Rel}$, we consider as $\mathcal{E}$ the class of all inverse injections and as $\mathcal{M}$ the class of all inverse surjections. Given a relation $R : X \to Y$, let $\mathcal{Z} = \{R^{-1}(y) \mid y \in Y\} \subseteq \mathcal{P}(X)$ be the set of pre-images of elements of Y under R. Now define relations $R_c : X \to \mathcal{Z}$

with $R_c(x) = \{Z \in \mathcal{Z} \mid x \in Z\}$ and $R_m: \mathcal{Z} \to Y$ with $R_m(Z) = \{y \in Y \mid Z = R^{-1}(y)\}$. Note that R_m is an inverse surjection, and $R_m \circ R_c = R$.

As a concrete example consider the relation R between sets $X = \{a, b, c, d\}$ and $Y = \{1, 2, 3, 4, 5\}$ visualized below on the left (where $R(a) = R(b) = \{1, 2\}$, $R(c) = \{3\}$, $R(d) = \{3, 4\}$). Its pseudo-factorization into R_c and R_m is shown on the right. Here R_m is indeed an inverse surjection, mapping elements of Y to their preimage in $\mathcal{P}(X)$.



*(CTS) For **Set**, the reflective subcategory of $\mathcal{K}\ell(T)$, where T is the input monad, we use the classical factorization structure with surjective and injective functions. Given an arrow $f: X \to Y$ in $\mathcal{K}\ell(T)$, seen as a function $f: X \to Y^A$, we define $Y' = \{y \in Y \mid \exists x \in X, a \in A: f(x)(a) = y\}$. Then $f_c: X \to Y'^A$ with $f_c(x)(a) = f(x)(a)$ and $f_m: Y' \to Y^A$ with $f_m(y)(a) = y$ for all $a \in A$, i.e., f_m is simply an injection without side-effects. Note that $f_m \circ f_c = f$ in $\mathcal{K}\ell(T)$.*

Note that pseudo-factorizations enjoy the diagonalization property as in Definition 7 whenever g is an arrow of $\mathbf{S}$ (see Lemma 38 in Appendix D). However pseudo-factors are not necessarily closed under composition with the isos of $\mathbf{C}$.

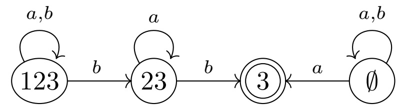
We will in the following fix a reflective subcategory $\mathbf{S}$ (possibly $\mathbf{C}$ itself) and work under the assumption that $\mathbf{S}$ has a final object and is preserved by the chosen endofunctor F. As before we assume that F preserves $\mathcal{M}$-arrows.

Proposition 22. *Let $\mathbf{C}$ be a category with a reflective subcategory $\mathbf{S}$. Furthermore assume that $\mathbf{S}$ is $(\mathcal{E}, \mathcal{M})$-structured. Then, given a coalgebra $\alpha: X \to FX$ in $\mathbf{C}$, the following two constructions obtain the same resulting minimization $\gamma: Z \to FZ$ (provided that the limit exists):*

- (i) *Apply Construction 11 using the $(\mathcal{E}, \mathcal{M})$-pseudo-factorizations of Definition 20.*
- (ii) *First reflect α into the subcategory $\mathbf{S}$ according to Definition 17 and then apply Construction 11 using $(\mathcal{E}, \mathcal{M})$-factorizations.*

Example 23. *(NDA) Proposition 22 suggests two procedures for building the minimization of an NDA (and thus checking the equivalence of its states). We first apply Construction (ii) to the NDA (X, α) in Example 5 and then we briefly illustrate Construction (i).*

Recall that the reflection of (X, α) into $\mathbf{Set}^{\mathbf{op}}$ is $(\mathcal{P}(X), \alpha')$ in Example 19. By applying Construction 11 (with the factorization structure of $\mathbf{Set}^{\mathbf{op}}$), we remove from $(\mathcal{P}(X), \alpha')$ the states that are not related to any word in the final coalgebra or, in other words, those states from which there is no path to the final state. Intuitively, we perform a backwards breadth-first search and the factorizations make sure that unreachable states are discarded and duplicated states are merged. The resulting automata is illustrated below.



Construction (i) can be understood as an efficient implementation of Construction (ii): we do not build the entire $(\mathcal{P}(X), \alpha')$, but we construct directly the above automaton by iteratively adding states and transitions. We start with state 3, then we add 23 and $\emptyset$ and finally we add 123. All the details of this construction are shown in Appendix B.

It is worth noting that in the minimization the languages accepted the states form a partition of A^*, i.e., they are all disjoint and their union is exactly A^*. For the above automaton, this is easily observed by looking at the regular expressions denoting them:

$$L(123) = (a+b)^* b a^* b \qquad L(23) = a^* b$$
$$L(3) = \varepsilon \qquad L(\emptyset) = (a+b)^* a$$

Recall that the final coalgebra consists of all words A^*. The unique mapping from the minimization into the final coalgebra is an inverse surjection (i.e. it is in $\mathcal{E}$ of $\mathbf{Set}^{\mathbf{op}}$) and assigns every word to the unique state that accepts it.

It is interesting to remark that the automaton obtained above is precisely the automaton in the third step of the well-known Brzozowski algorithm for minimization of non-deterministic automata [7], which, in a nutshell, works as follows: 1) given an NDA reverse it, by reversing all arrows and exchanging final and initial states; 2) determinize it, using the subset construction, and remove unreachable states; 3) reverse it again; 4) determinize it, using the subset construction, and remove unreachable states. In our example, we are doing steps 1)–3) but without the explicit reversal. Our automata do not have initial states, but steps 1)–3) are independent on the specific choice of initial states, because of the two reversals.

V. Metrics and Isometric Maps

The construction of the previous section is parametrized by the chosen classes of $\mathcal{E}$ and $\mathcal{M}$ arrows. It is a natural question which classes are *reasonable*. One possible answer is to define metrics and to choose as $\mathcal{M}$-arrows isometric maps, i.e., those maps that preserve distances. This means that, if a final coalgebra should exist, one obtains the same distances regardless of whether they are computed on the final coalgebra or on the minimization. In order to be able to define metrics, we will from now on restrict ourselves to concrete categories over $\mathbf{Set}$ (see Definition 1).

Definition 24 (Metric and Pseudo-Metric)**.** *Let A be an arbitrary set and let $(D, \leq, \oplus)$ be a partially ordered set with a commutative monoidal operation $\oplus$ with unit 0. A* pseudo-metric *on A is a function $d: A \times A \to D$ satisfying the following laws for $a, b, c \in A$: (i) $d(a, b) \geq 0$; (ii) $d(a, a) = 0$; (iii) $d(a, b) = d(b, a)$; (iv) $d(a, c) \leq d(a, b) \oplus d(b, c)$, where 0*

is the unit of the monoidal operation $\oplus$. It is called a metric *if in addition $d(a,b)=0$ implies $a=b$.*

Example 25. *(NDA,LWA) In our examples involving non-deterministic automata or linear weighted automata we simply take the discrete metrics, obtaining distance 0 if two elements are equal and 1 otherwise (where $0 \le 1$). The monoidal operation takes the maximum of two values.*

(CTS) In the case of the conditional transition systems, defined via the input monad, we consider a more interesting metric: let $\mathbf{x},\mathbf{y} \in TX$, i.e., $\mathbf{x},\mathbf{y}\colon A \to X$. Then $d(\mathbf{x},\mathbf{y}) = \{a \mid \mathbf{x}(a) \neq \mathbf{y}(a)\}$, ordered by inclusion. The unit is $\emptyset$ and the monoidal operation corresponds to the union.

Proposition 26 (Induced Pseudo-Metric)**.** *Let $(\mathbf{C},U)$ be a concrete category. Assume that there is a metric $d_Y\colon UY \times UY \to D$ for every object Y of $\mathbf{C}$. Every arrow $f\colon X \to Y$ induces a pseudo-metric $d_f\colon UX \times UX \to D$ defined by $d_f(a,b) = d_Y(Uf(a),Uf(b))$.*

Definition 27 (Non-Expansive, Isometric Maps)**.** *Let $(\mathbf{C},U)$ be a concrete category and assume a family of metrics as in Proposition 26. An arrow $f\colon X \to Y$ of $\mathbf{C}$ is called* non-expansive *whenever $d_f(a,b) \le d_X(a,b)$ for all $a,b \in UX$. It is called* isometric *whenever $d_f(a,b) = d_X(a,b)$.*

If all arrows are non-expansive, it can easily be shown that all sections are isometric.[2] This also means that η_X, as the unit of an adjunction, is isometric, since it has a left-inverse ε_X (the co-unit). This implies that the reflection into the subcategory does not modify the distance between elements.

Now assume that we are in the following setting: let $(\mathbf{C},U)$ be a concrete category with a reflective subcategory $\mathbf{S}$, which is $(\mathcal{E},\mathcal{M})$-structured. We also assume a family of metrics as defined above, such that all $\mathbf{C}$-arrows are non-expansive and all $\mathcal{M}$-arrows are isometric. It can be shown that these requirements hold for our running examples concerning NDA's, LWA's and CTS's. Then the construction of the minimization in $\mathbf{S}$, done in one of the two ways described in Proposition 22, induces a pseudo-metric on the original coalgebra, which is minimal with respect to distances.

Theorem 28. *Let $\alpha\colon X \to FX$ be a coalgebra in $\mathbf{C}$, let $\alpha'\colon LX \to FLX$ be the reflection of α into $\mathbf{S}$ and let $\gamma\colon Z \to FZ$ be the minimization for α'. In addition, let $c = e \circ \eta_X\colon (X,\alpha) \to (Z,\gamma)$ be the coalgebra morphism into the minimization where $e \in \mathcal{E}$ with $e\colon (LX,\alpha') \to (Z,\gamma)$. Then, for every other coalgebra morphism $f\colon (X,\alpha) \to (Y,\beta)$, we have $d_c(a,b) \le d_f(a,b)$ for all $a,b \in UX$.*

We prove that two elements have distance 0 in the minimization if and only if they are behaviourally equivalent.

Corollary 29 (Behavioural Equivalence and Induced Pseudo–Metrics)**.** *Let $(\mathbf{C},U)$ be a concrete category, let $F\colon \mathbf{C} \to \mathbf{C}$ be an endofunctor and we assume that the final F-coalgebra exists. Let (X,α) be a coalgebra with mapping $c\colon X \to Z$ into its minimization (as in Theorem 28). Then for $a,b \in UX$ it holds that $a \approx b$ if and only if $d_c(a,b) = 0$.*

Finally, we summarize all notions introduced earlier, including metrics, by considering once more the CTS example.

Example 30. *(CTS) Recall the coalgebraic description of CTS given in Example 6: the base category is $\mathcal{K}\ell(T)$, where T is the input monad and $F = \mathcal{P}_c$ is the countable powerset functor. The CTS of the introduction is the coalgebra $\alpha\colon X \to \mathcal{P}_c(X)$ represented by the table in Example 6 (with column indices in $X = \{1,\dots 10\}$ and row indices in $A = \{a,\bar{a}\}$).*

We first apply the algorithm in Proposition 22(i) with the pseudo-factorization of Example 21 (Construction (ii) only consists in minimizing the reflected coalgebra of Example 19). The terminal object of $\mathcal{K}\ell(T)$ is the one-element set $1 = \{\bullet\}$ and hence the unique morphism $d_0\colon X \to 1$ is the following:

d_0	1	2	3	4	5	6	7	8	9	10
a	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
$\bar{a}$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$

Via pseudo-factorization we obtain $d_0 = m_0 \circ c_0$ with $c_0 = d_0$ and m_0 the identity. The algorithm now computes $\overline{F}c_0 \circ \alpha = d_1\colon X \to \mathcal{P}_c(\{\bullet\}) = \{\emptyset,\{\bullet\}\}$:

d_1	1	2	3	4	5	6	7	8	9	10
a	$\{\bullet\}$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\bullet\}$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\bar{a}$	$\{\bullet\}$	$\emptyset$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\{\bullet\}$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\emptyset$

Again $d_1 = c_1$ and hence the algorithm computes $\overline{F}c_1 \circ \alpha = d_2\colon X \to \mathcal{P}_c(\{\emptyset,\{\bullet\}\}) = \{\emptyset,\{\emptyset\},\{\{\bullet\}\},\{\emptyset,\{\bullet\}\}\}$:

d_2	1	2	3	4	5	6	7	...
a	$\{\emptyset,\{\bullet\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\emptyset,\{\bullet\}\}$	$\{\emptyset\}$	$\emptyset$
$\bar{a}$	$\{\emptyset,\{\bullet\}\}$	$\emptyset$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\{\emptyset,\{\bullet\}\}$	$\{\emptyset\}$	$\emptyset$

where "..." is an abbreviation for for $8,9,10$.

The pseudo-factorization gives $c_2\colon X \to \{\emptyset,\{\emptyset\},\{\emptyset,\{\bullet\}\}\}$ by restricting the image of d_2, since nothing maps to $\{\{\bullet\}\}$. Hence c_2 has exactly the same table as the one given for d_2.

By iterating the construction once again, $\overline{F}c_2 \circ \alpha = d_3\colon X \to \mathcal{P}_c(\{\emptyset,\{\emptyset\},\{\emptyset,\{\bullet\}\}\})$:

d_3	1	2	3	4	5	6	7	...
a	$\{\emptyset,\{\emptyset\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\emptyset,\{\emptyset\}\}$	$\{\emptyset\}$	$\emptyset$
$\bar{a}$	$\{\emptyset,\{\emptyset\}\}$	$\emptyset$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\{\emptyset,\{\emptyset\}\}$	$\{\emptyset\}$	$\emptyset$

Note that $c_3\colon X \to \{\emptyset,\{\emptyset\},\{\emptyset,\{\emptyset\}\}\}$ (which is d_3 restricted to its image) and c_2 are isomorphic and thus the algorithm terminates. The resulting minimization is (Z,γ) depicted below (note that γ is an element of $\mathcal{M}$ and hence simply an injective function from Z to $\mathcal{P}_c(Z)$).

$$\{\emptyset,\{\emptyset\}\} \longrightarrow \{\emptyset\} \longrightarrow \emptyset$$

It is easy to see that the above transition system is isomorphic to the one of introduction having states x,y,z. Moreover, the morphism c_3 (whose table is equal to the one of d_3 above) corresponds to the dashed arrow of the introduction.

We now show that $c_3\colon(X,\alpha) \to (Z,\gamma)$ induces a metric on X. A specific state $x \in X$ can be represented by a function $\mathbf{x}\colon A \to X$ in TX with $\mathbf{x}(a) = x$ for all a. Now

[2]Sections are the arrows $f\colon X \to Y$ for which there exists a $g\colon Y \to X$ with $g \circ f = id_X$.

consider the states 2, 7 *and take the respective functions* $\mathbf{x}_2, \mathbf{x}_7$*;* $Uc_3(\mathbf{x}_2)\colon A \to Z$ *maps* a *to* $\{\emptyset\}$ *and* $\bar{a}$ *to* $\emptyset$*, while* $Uc_3(\mathbf{x}_7)$ *maps both* $a, \bar{a}$ *to* $\{\emptyset\}$*. Hence the distance between* 2 *and* 7 *is* $\{\bar{a}\}$*, meaning that they have the same behaviour only when* a *holds. Consider now states* 1 *and* 6*: the functions* $Uc_3(\mathbf{x}_1)\colon A \to Z$ *and* $Uc_3(\mathbf{x}_6)\colon A \to Z$ *map both* a *and* $\bar{a}$ *to* $\{\emptyset, \{\emptyset\}\}$*. Thus* 1 *and* 6 *are at distance* $\emptyset$*, i.e., they are behaviourally equivalent. The entire table of distances is as follows (where "*$\ldots$*" is an abbreviation for for* $4, 5, 8, 9, 10$*):*

	1	2	3	$\ldots$	6	7
1	$\emptyset$	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$	$\emptyset$	$\{a,\bar{a}\}$
2	$\{a,\bar{a}\}$	$\emptyset$	$\{a,\bar{a}\}$	$\{a\}$	$\{a,\bar{a}\}$	$\{\bar{a}\}$
3	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$	$\emptyset$	$\{\bar{a}\}$	$\{a,\bar{a}\}$	$\{a\}$
$\ldots$	$\{a,\bar{a}\}$	$\{a\}$	$\{\bar{a}\}$	$\emptyset$	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$
6	$\emptyset$	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$	$\emptyset$	$\{a,\bar{a}\}$
7	$\{a,\bar{a}\}$	$\{\bar{a}\}$	$\{a\}$	$\{a,\bar{a}\}$	$\{a,\bar{a}\}$	$\emptyset$

VI. Conclusion and Related Work

Several previous papers (e.g., [5], [6]) have pointed out the relationship between the construction of the final coalgebra (via the so-called "final sequence" [18], [16]) and the minimization algorithm. Inspired by these, our paper is the first one that proposes an abstract algorithm (Construction 11) for minimizing coalgebras. The construction only relies on *(pseudo-)factorization structures* and it is completely independent of the base category of the endofunctor F. Together with appropriate reflections, this allows to perform "minimization" of interesting class of systems that cannot be regarded as coalgebras over $\mathbf{Set}$, such as non-deterministic automata, conditional transition systems and linear weighted automata.

For non-deterministic automata, which we model as coalgebras following [4], the result of the proposed algorithm coincides with the one of the third step of the Brzozowski's algorithm [7]. The resulting automata are not really minimal in the number of states (it is well know that there exists no unique minimal non-deterministic automata), but they correspond to backwards-deterministic automata and have the nice property that the languages recognized by the states form a partition of the set of all words.

The example on conditional transition systems is completely original, but it has been motivated by the work in [9], [10], which introduce notions of bisimilarity depending on conditions (which are fixed once and for all). The setting of [10] is closer to ours, but no minimization algorithm is given there. For practical purposes we think it will be beneficial to work with matrices where entries correspond to Boolean formulas, specifying whether there exists a transition between two states. We already have a prototype implementation performing the fixed-point iteration based on binary decision diagrams.

In order to further witness the expressiveness of our approach, we showed that Construction 11 subsumes the minimization algorithm in [8] for linear weighted automata.

Our study on factorization structures and minimizations also allows to define interesting behavioural pseudo-metrics on the state spaces of dynamical systems. A closely related approach relying on coalgebras on the category of metric spaces and non-expansive maps is studied in [19], [20] for the case of probabilistic transition systems. In these works the functor F also acts on the metrics (via the so-called Kantorovich metrics) and, consequently, the algorithm presented in [20] computes a new pseudo-metric at every iteration. In the approach of this paper instead, the minimization is first computed and from that the resulting behavioural pseudometrics.

We plan to investigate under what circumstances one can obtain an algorithm that computes the distances on states directly, without going via the minimization. We know that this is feasible for conditional transition systems, but it is unclear how far this can be generalized. Moreover, we plan to further study applications to weighted automata over semirings and to probabilistic transition systems, using either the (sub-)distribution monad for the discrete case or the Giry monad as considered by Doberkat in [21].

Finally, we should mention that factorization structures for coalgebras have also been studied in [15] (for axiomatizing a Co-Birkhoff theorem) but, to our knowledge, they have not been related to minimization algorithms. Moreover, the notion of minimizations generalizes simple [22] and minimal [23] coalgebras in the case where the base category is $\mathbf{Set}$ with epi-mono factorizations.

Acknowledgements: We would like to thank Ana Sokolova, Paolo Baldan, Walter Tholen, Jiří Adámek and Stefan Milius for answering our questions and giving generous and extremely helpful feedback.

References

[1] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Wesley, 2006.

[2] R. Paige and R. E. Tarjan, "Three partition refinement algorithms," *SIAM Journal on Computing*, vol. 16, no. 6, pp. 973–989, 1987.

[3] M. P. Schützenberger, "On the definition of a family of automata," *Information and Control*, vol. 4, no. 2-3, pp. 245–270, 1961.

[4] I. Hasuo, B. Jacobs, and A. Sokolova, "Generic trace semantics via coinduction," *LMCS*, vol. 3, no. 4:11, pp. 1–36, 2007.

[5] S. Staton, "Relating coalgebraic notions of bisimulation," in *Proc. of CALCO '09*. Springer, 2009, pp. 191–205, LNCS 5728.

[6] F. Bonchi and U. Montanari, "Coalgebraic models for reactive systems," in *Proc. of CONCUR '07*. Springer, 2007, pp. 364–379, LNCS 4703.

[7] J. A. Brzozowski, "Canonical regular expressions and minimal state graphs for definite events," in *Mathematical Theory of Automata*, vol. 12(6). Polytechnic Press, NY, 1962, pp. 529–561.

[8] M. Boreale, "Weighted bisimulation in linear algebraic form," in *Proc. of CONCUR '09*. Springer, 2009, pp. 163–177, LNCS 5710.

[9] M. Hennessy and H. Lin, "Symbolic bisimulations," *TCS*, vol. 138, no. 2, pp. 353–389, 1995.

[10] M. Fitting, "Bisimulations and boolean vectors," in *Advances in Modal Logic*. World Scientific Publishing, 2002, vol. 4, pp. 1–29.

[11] J. Adámek, H. Herrlich, and G. Strecker, *Abstract and Concrete Categories - The Joy of Cats*. Wiley, 1990.

[12] J. Power and D. Turi, "A coalgebraic foundation for linear time semantics," in *Proc. of CTCS '99*, ser. ENTCS, vol. 29, 1999, pp. 259–274.

[13] S. Mac Lane, *Categories for the Working Mathematician*. Springer-Verlag, 1971.

[14] P. S. Mulry, "Lifting theorems for Kleisli categories," in *Proc. of MFPS*. Springer, 1993, pp. 304–319, LNCS 802.

[15] A. Kurz, "Logics for coalgebras and applications to computer science," Ph.D. dissertation, Ludwigs-Maximilians-Universität München, 2000.

[16] J. Adámek and V. Koubek, "On the greatest fixed point of a set functor," *TCS*, vol. 150, pp. 57–75, 1995.

[17] C. Hermida and B. Jacobs, "Structural induction and coinduction in a fibrational setting," *Information and Computation*, vol. 145, pp. 107–152, 1998.

[18] J. Worrell, "On the final sequence of a finitary set functor," *TCS*, vol. 338, no. 1-3, pp. 184–199, 2005.

[19] F. van Breugel and J. Worrell, "A behavioural pseudometric for probabilistic transition systems," *TCS*, vol. 331, pp. 115–142, 2005.

[20] ——, "Approximating and computing behavioural distances in probabilistic transition systems," *TCS*, vol. 360, pp. 373–385, 2005.

[21] E.-E. Doberkat, "Kleisli morphisms and randomized congruences for the giry monad," *J. Pure and Appl. Algebra*, vol. 211, pp. 638–664, 2007.

[22] J. Rutten, "Universal coalgebra: a theory of systems," *TCS*, vol. 249, pp. 3–80, 2000.

[23] H. P. Gumm, "On minimal coalgebras," *Applied Categorical Structures*, vol. 16, pp. 313–332, 2008.

[24] J. MacDonald and W. Tholen, "Decomposition of morphisms into infinitely many factors," in *Proc. of Category Theory – Applications to Algebra, Logic and Topology*, ser. Lecture Notes in Mathematics, no. 962. Springer, 1981, pp. 175–189.

Appendix

A. Additional Definitions

We will now formally define the notion of monad (see also [11], [4]).

Definition 31 (Monad)**.** *A* monad *on* $\mathbf{Set}$ *is an endofunctor* $T\colon \mathbf{Set} \to \mathbf{Set}$ *together with two natural transformations:*

- *a* unit *natural transformation* $\eta\colon \mathsf{Id} \Rightarrow T$*, that is arrows* $\eta_X\colon X \to TX$ *for each set* X *satisfying suitable naturality conditions;*
- *a* multiplication *natural transformation* $\mu\colon T^2 \Rightarrow T$*, that is arrows* $\mu_X\colon TTX \to X$ *for each set* X *again satisfying suitable naturality conditions.*

The unit and multiplication have to satisfy the follow compatibility conditions:

$$\begin{array}{ccccc} TX & \xrightarrow{\eta_{TX}} & T^2X & \xleftarrow{T\eta_X} & TX \\ & \searrow^{id} & \downarrow{\scriptstyle \mu_X} & \swarrow_{id} & \\ & & TX & & \end{array} \qquad \begin{array}{ccc} T^3X & \xrightarrow{T\mu_X} & T^2X \\ {\scriptstyle \mu_{TX}}\downarrow & & \downarrow{\scriptstyle \mu_X} \\ T^2X & \xrightarrow[\mu_X]{} & TX \end{array}$$

Example 32. *In the running examples of this paper we use the following monads:*

(CTS) Input monad: *for a given set* A *of inputs define* $T\colon \mathbf{Set} \to \mathbf{Set}$ *with* $TX = X^A$ *for a set* X*. For a function* $f\colon X \to Y$ *in* $\mathbf{Set}$ *define* $Tf\colon X^A \to Y^A$ *with* $Tf(g)(a) = f(g(a))$ *for* $g\colon A \to X$ *and* $a \in A$*.*

The unit arrows are $\eta_X\colon X \to X^A$ *with* $\eta_X(x)(a) = x$ *for all* $a \in A$*. Furthermore the multiplication arrows have the form* $\mu_X\colon (X^A)^A \to X^A$ *with* $\mu_X(g)(a) = g(a)(a)$ *for a function* $g\colon A \to X^A$*.*

(LWA) Monad assigning weights from a field: *let* $\mathbb{F}$ *be a field and define* $T\colon \mathbf{Set} \to \mathbf{Set}$ *with* $TX = (\mathbb{F}^X)_\omega$*, which is the set of all mappings from* X *to* $\mathbb{F}$ *of finite support, i.e., only finitely many function values may be different from* 0*. For a function* $f\colon X \to Y$ *in* $\mathbf{Set}$ *define* $Tf\colon (\mathbb{F}^X)_\omega \to (\mathbb{F}^Y)_\omega$ *as follows: let* $a \in (\mathbb{F}^X)_\omega$*, where* a *has finite support, then*

$$Tf(a)(y) = \sum\{a(x) \mid x \in X, f(x) = y\}$$

The unit arrows are $\eta_X\colon X \to (\mathbb{F}^X)_\omega$ *with*

$$\eta_X(x)(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

Furthermore, the multiplication arrows have the form $\mu_X\colon \left(\mathbb{F}^{(\mathbb{F}^X)_\omega}\right)_\omega \to (\mathbb{F}^X)_\omega$ *with*

$$\mu_X(g)(x) = \sum_{f \in (\mathbb{F}^X)_\omega} g(f) \cdot f(x)$$

for a function $g \in \left(\mathbb{F}^{(\mathbb{F}^X)_\omega}\right)_\omega$*. This definition implies that arrow composition in the corresponding Kleisli category corresponds to matrix multiplication.*

(NDA) Powerset monad: *let* $T\colon \mathbf{Set} \to \mathbf{Set}$ *be the powerset monad with* $TX = \mathcal{P}(X)$ *for a set* X*. Furthermore* T *acts on a function* $f\colon X \to Y$ *as follows:* $Tf\colon \mathcal{P}(X) \to \mathcal{P}(Y)$ *with* $Tf(X') = \{y \in Y \mid \exists x \in X'\colon f(x) = y\}$*.*

The unit arrows are $\eta_X\colon X \to \mathcal{P}(X)$ *with* $\eta_X(x) = \{x\}$*. Furthermore the multiplication arrows have the form* $\mu_X\colon \mathcal{P}(\mathcal{P}(X)) \to \mathcal{P}(X)$ *with*

$$\mu_X(\mathcal{Z}) = \bigcup_{Z \in \mathcal{Z}} Z, \qquad \mathcal{Z} \subseteq \mathcal{P}(X)$$

i.e., we take the union of all the sets contained in $\mathcal{Z}$*.*

B. Additional Examples: Non-deterministic Automata

Example 33. *(NDA) We are considering the non-deterministic automaton of Example 5. We are working in the category* $\mathcal{K}\ell(\mathcal{P}) = \mathbf{Rel}$ *and* $FX = A \times X + 1$*. We will in the following denote a relation* $\alpha\colon X \to A \times X + 1$ *by a Boolean matrix with column indices from* X *and row indices from* $A \times X + 1$*. Arrow composition is done via matrix multiplication (using logical* or *for addition and logical* and *for multiplication). In our specific example* α *looks as follows. Note that* $\langle a, 1\rangle$ *is abbreviated by* $a1$*, etc.*

$$\alpha = \begin{array}{c} \\ a1 \\ b1 \\ a2 \\ b2 \\ a3 \\ b3 \\ \bullet \end{array} \!\!\begin{array}{c} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

The final object is the empty set $\emptyset$ *and hence* $d_0 = c_0$ *is a* 0×3*-matrix. In the next step we obtain* $d_1\colon X \to 1$ $(= A \times \emptyset + 1)$*.*

$$d_1 = \overline{F}c_0 \circ \alpha = \bullet \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$$

We compute the pseudo-factorization (see Example 21) and obtain $c_1 = d_1$*. In the next step we get* $d_2\colon X \to A \times 1 + 1$*.*

$$d_2 = \overline{F}c_1 \circ \alpha = \begin{array}{c} \\ a\bullet \\ b\bullet \\ \bullet \end{array} \!\!\begin{array}{c} \begin{matrix} a1 & b1 & a2 & b2 & a3 & b3 & \bullet \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \circ \alpha$$

$$= \begin{array}{c} \\ a\bullet \\ b\bullet \\ \bullet \end{array} \!\!\begin{array}{c} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

Again, via the pseudo-factorization we obtain $c_2 = d_2$*. By iterating again, we obtain* $d_3\colon X \to A \times (A \times 1 + 1) + 1$*.*

$$d_3 = \overline{F}c_2 \circ \alpha = \begin{array}{c} \\ aa\bullet \\ ba\bullet \\ ab\bullet \\ bb\bullet \\ a\bullet \\ b\bullet \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

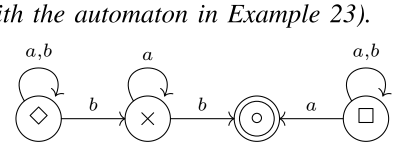
$$= \begin{array}{c} \\ aa\bullet \\ ba\bullet \\ ab\bullet \\ bb\bullet \\ a\bullet \\ b\bullet \\ \bullet \end{array} \begin{array}{c} \begin{array}{cccc} \Box & \times & \Diamond & \circ \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \circ \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ \Box \\ \times \\ \Diamond \\ \circ \end{array} = m_3 \circ c_3$$

By iterating the construction once again we obtain $d_4\colon X \to A \times \{\Box, \times, \diamond, \circ\} + 1$. *Via pseudo factorization we obtain* m_4, c_4 *with* $c_4 = c_3$, *i.e., we have reached the fixed-point.*

$$d_4 = \overline{F}c_3 \circ \alpha = \begin{array}{c} \\ a\Box \\ b\Box \\ a\times \\ b\times \\ a\Diamond \\ b\Diamond \\ a\circ \\ b\circ \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$= \begin{array}{c} \\ a\Box \\ b\Box \\ a\times \\ b\times \\ a\Diamond \\ b\Diamond \\ a\circ \\ b\circ \\ \bullet \end{array} \begin{array}{c} \begin{array}{cccc} \Box & \times & \Diamond & \circ \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \circ \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} \\ \Box \\ \times \\ \Diamond \\ \circ \end{array} = m_4 \circ c_4$$

The arrow $m_4 = \gamma$ *gives us the following minimization (compare with the automaton in Example 23).*

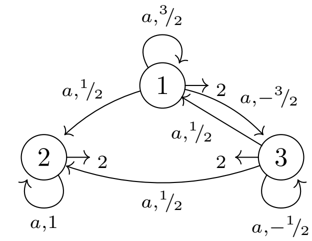


Intuitively, we are here basically doing a breadth-first backwards search, starting from the set of final states.

C. Additional Examples: Linear Weighted Automata

Example 34. *(LWA) We come back to Example 14 and consider the following linear weighted automaton from [8] with* $X = \{1, 2, 3\}$, $A = \{a\}$ *and* $\mathbb{F} = \mathbb{R}$ *(graphical representation on the right and coalgebra* $\alpha\colon X \to (\mathbb{R}^{A \times X + 1})_\omega$, *in matrix form, on the left):*

$$\begin{array}{c} \\ \langle a, 1\rangle \\ \langle a, 2\rangle \\ \langle a, 3\rangle \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 3/2 & 0 & 1/2 \\ 1/2 & 1 & 1/2 \\ -3/2 & 0 & -1/2 \\ 2 & 2 & 2 \end{pmatrix} \end{array}$$



There is only a single label a, *hence we omit labels in the following.*

The final object is the empty set and hence $d_0 = c_0$ *is a* 0×3*-matrix. Thus we obtain:*

$$d_1 = \overline{F}c_0 \circ \alpha = \begin{array}{c} \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 2 & 2 & 2 \end{pmatrix} \end{array}$$

The arrow d_1 *is a matrix of full row rank (i.e., an element of* $\mathcal{E}$*) and hence* $c_1 = d_1$. *In the next step we obtain:*

$$d_2 = \overline{F}c_1 \circ \alpha = \begin{array}{c} \\ \circ \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \end{array}$$

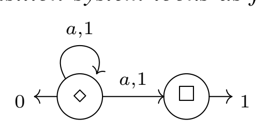
Note that d_2 *is an element of* $\mathcal{E}$, *since its row vectors are linearly independent and hence* $c_2 = d_2$. *In the next step we obtain:*

$$d_3 = \overline{F}c_2 \circ \alpha = \begin{array}{c} \\ \diamond \\ \Box \\ \bullet \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \end{array}$$

Note that d_3 *is not of full row rank, since it contains two identical row vectors. We factor out an arrow of* $\mathcal{M}$ *as follows:*

$$d_3 = \begin{array}{c} \\ \diamond \\ \Box \\ \bullet \end{array} \begin{array}{c} \begin{array}{cc} \diamond & \Box \end{array} \\ \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \circ \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \end{array} \begin{array}{c} \\ \diamond \\ \Box \end{array} = m_3 \circ c_3$$

Hence we have reached a fixed-point and set $\gamma = m_3$. *The corresponding transition system looks as follows:*



This linear weighted automaton is equivalent to the one obtained in [8].

Comparison to Boreale's Linear Weighted Automata

We will compare the setting of Example 14 with the linear weighted automata of Boreale [8], where we use the reals as field.

Definition 35 (Weighted Automaton in Linear Form [8])**.** *A* linear weighted automaton *(LWA for short) is a triple* $L = (V, \{T_a\}_{a \in A}, \varphi)$, *where* V *is a (finite-dimensional) vector space over* $\mathbb{R}$, *and* $T_a\colon V \to V$, *for* $a \in A$, *and* $\varphi\colon V \to \mathbb{R}$ *are linear maps.*

We assume in the following that the vector space has as elements mappings of the form $X \to \mathbb{R}$ for a finite set X, i.e., vectors are elements of TX in the notation of Example 14.

First, we show how to convert LWA's into coalgebras and vice versa. Given an LWA L we define the following coalgebra $\alpha\colon X \to TFX$, where $TFX = (\mathbb{R}^{A\times X+1})_\omega$:

$$\begin{aligned}\alpha(x)(\langle a,y\rangle) &= T_a(\eta_X(x))(y)\\ \alpha(x)(\bullet) &= T_a(\eta_X(x))(\bullet)\end{aligned}$$

Note that $\eta_X(x)$, where η_X is the unit of the monad, stands for the function that maps x to 1 and all other elements to 1. It corresponds to a unit vector.

Given a coalgebra $\alpha\colon X \to TFX$ we define an LWA with vector space $(\mathbb{R}^X)_\omega$, $T_a(\mathbf{u})(y) = (\alpha\cdot\mathbf{u})(\langle a,y\rangle)$ and $\varphi(\mathbf{u}) = (\alpha\cdot\mathbf{u})(\bullet)$. Here we abuse the notation and interpret α as a matrix where columns are indexed by X and rows by $A\times X+1$. Then $\alpha\cdot\mathbf{u}$ denotes the multiplication of matrix α with vector $\mathbf{u}$.

Definition 36 (Weighted L-Bisimulation [8])**.** *A relation R on V is called* weighted L-bisimulation *whenever*

1) *R is* linear*, i.e., there exists a subspace U of V such that for $\mathbf{u},\mathbf{v}\in V$ it holds that $\mathbf{u}\,R\,\mathbf{v} \iff \mathbf{u}-\mathbf{v}\in U$.*
2) *Whenever $\mathbf{u}\,R\,\mathbf{v}$ for $\mathbf{u},\mathbf{v}\in V$, then*
   a) $\varphi(\mathbf{u}) = \varphi(\mathbf{v})$
   b) $T_a(\mathbf{u})\,R\,T_a(\mathbf{v})$ *for all $a\in A$.*

Two vectors $\mathbf{u},\mathbf{v}$ are L-bisimilar ($\mathbf{u}\sim_L\mathbf{v}$) if there exists an L-bisimulation R with $\mathbf{u}\,R\,\mathbf{v}$.

Instead of using the definition above, an alternative definition is to require an LWA L' on a vector space V' and a linear map $f\colon V\to V'$, such that:

(i) $\varphi'(f(\mathbf{u})) = \varphi(\mathbf{u})$;
(ii) $f\circ T_a = T'_a\circ f$.

Then two vectors $\mathbf{u}$, $\mathbf{v}$ are in relation if they have the same image under f.

Given such a linear map f, one can construct the subspace U in Definition 36 as the kernel of f and show that it has the required properties. On the other hand, if we are given an L-bisimulation R, one can construct f as a surjective linear mapping that has U as its kernel. Then one defines the linear weighted automaton L' via $T'_a(f(\mathbf{u})) = f(T_a(\mathbf{u}))$ and $\varphi'(f(\mathbf{u})) = \varphi(\mathbf{u})$. Due to the conditions of Definition 36 one can show that T'_a and φ' are well-defined.

Now Conditions (i) and (ii) above correspond to the condition for coalgebra morphisms, requiring that $\overline{F}f\circ\alpha = \alpha'\circ f$, where α is the coalgebra for L and α' the coalgebra for L'.

Since the minimization induces minimal distances, it is straightforward to show that two vectors are L-bisimilar iff their images in the relatively final coalgebra coincide, i.e., if they have distance 0.

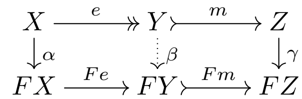
Finally, note that Boreale's algorithm computes the orthogonal complement $U^\perp$ (in our setting the row space of E in Example 14) rather than U itself, similar to our algorithm.

D. Proofs

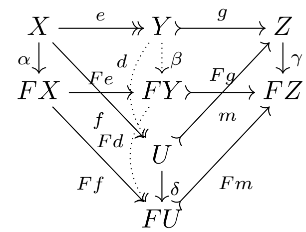
Lemma 37. *Assume that the functor F preserves $\mathcal{M}$-arrows. Then the category of F-coalgebras is $(\mathcal{E},\mathcal{M})$-structured, whenever this holds for the underlying category* $\mathbf{C}$.

Proof: We check that the conditions of Definition 7 are satisfied. Note that the isos in the underlying category agree with the isos in the category of F-coalgebras. Hence closure under composition with isos follows trivially.

The factors of a coalgebra morphism $f\colon(X,\alpha)\to(Z,\gamma)$ are obtained by factoring $f\colon X\to Z$ into $f = m\circ e$ with $e\colon X\to Y$, $m\colon Y\to Z$. Since F preserves $\mathcal{M}$-arrows $Fm\in\mathcal{M}$ and hence the coalgebra β can be obtained as the unique diagonal arrow.



Finally, take a commuting square in the category of coalgebras as depicted below and show that there is a diagonal arrow.



The arrow d is obtained as the unique diagonal arrow for the upper square consisting of e,g,f,m. Note that Fd makes the lower square commute. It is left to show that everything commutes, specifically that $\delta\circ d = Fd\circ\beta$. Consider the commuting square $(\gamma\circ g)\circ e = Fm\circ(Ff\circ\alpha)$ in the underlying category. It can be checked that both $\delta\circ d$ and $Fd\circ\beta$ are diagonals for this square and hence they coincide. Specifically:

$$\begin{aligned}&(\delta\circ d)\circ e = \delta\circ(d\circ e) = \delta\circ f = Ff\circ\alpha\\ &(Fd\circ\beta)\circ e = Fd\circ(\beta\circ e) = Fd\circ Fe\circ\alpha\\ &\qquad = F(d\circ e)\circ\alpha = Ff\circ\alpha\\ &Fm\circ(\delta\circ d) = (Fm\circ\delta)\circ d = \gamma\circ m\circ d = \gamma\circ g\\ &Fm\circ(Fd\circ\beta) = F(m\circ d)\circ\beta = Fg\circ\beta = \gamma\circ g\end{aligned}$$

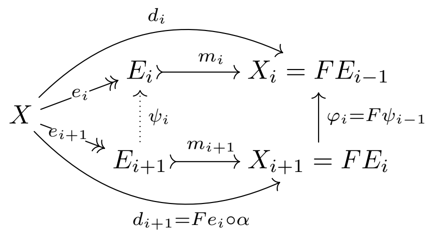
This shows that the factorization structure of the underlying category can be lifted to the category of F-coalgebras. ∎

Proposition 9 (Minimization and Final Coalgebras)**.** *If the final coalgebra $\omega\colon\Omega\to F\Omega$ exists, then – for a given coalgebra $\alpha\colon X\to FX$ – the minimization $\gamma\colon Z\to FZ$ for α can be obtained by factoring the unique coalgebra morphism $beh_X\colon(X,\alpha)\to(\Omega,\omega)$ into an $\mathcal{E}$-morphism and an $\mathcal{M}$-morphism.*

Proof: The proposition follows directly from the fact that the category of coalgebras is $(\mathcal{E},\mathcal{M})$-structured (cf. Lemma 37), especially from the diagonalization property. ∎

Theorem 12. *If the limits in Construction 11 exist and the endofunctor F preserves the limit, then the coalgebra $\gamma\colon Z\to FZ$ is the minimization for α.*

Proof: We first show that the construction is well-defined and then that $\gamma\colon Z \to FZ$ is the minimization.

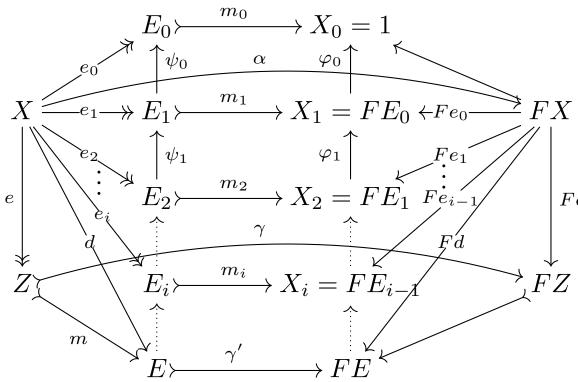


First, observe that the outer diagram above commutes: $\varphi_i \circ d_{i+1} = d_i$. For $i = 0$ follows immediately since both are arrows into the final object. For $i > 0$ we obtain by induction: $\varphi_i \circ d_{i+1} = F\psi_{i-1} \circ Fe_i \circ \alpha = F(\psi_{i-1} \circ e_i) \circ \alpha = Fe_{i-1} \circ \alpha = d_i$.

Next, consider the category $\mathbf{Mor}(\mathbf{C})$, whose objects are morphisms of $\mathbf{C}$ and arrows are commuting squares. In $\mathbf{Mor}(\mathbf{C})$, $\gamma'\colon E \to FE$ is the limit object of the ω-chain

$$m_0 \leftarrow m_1 \leftarrow m_2 \leftarrow \dots \leftarrow m_i \leftarrow \dots$$

where each morphism $m_{i+1} \to m_i$ consists of the pair of morphisms of $\mathbf{C}$: $\psi_i\colon E_{i+1} \to E_i$ and $\varphi_i\colon X_{i+1} \to X_i$. It is known from [24] that the full subcategory of $\mathcal{M}$-arrows (seen as objects) is a reflective subcategory of $\mathbf{Mor}(\mathbf{C})$. Hence γ' as the limit object is in $\mathcal{M}$. Note however that d is not necessarily contained in $\mathcal{E}$.



In the following let $p_i\colon E \to E_i$ and $r_i\colon FE \to X_i$ be the limit projections. Note that $r_i = Fp_{i-1}$ since F preserves this limit.

Observe that α is the tip of a cone over the ω-chain in $\mathbf{Mor}(\mathbf{C})$: $Fe_{i-1} \circ \alpha = d_i = m_i \circ e_i$ and $\psi_i \circ e_{i+1} = e_i$ by construction. Hence d, Fd exist as the mediating morphisms. Note that Fd must be the mediating morphism since it makes the triangles commute ($r_i \circ Fd = Fp_{i-1} \circ Fd = F(p_{i-1} \circ d) = Fe_{i-1}$.) Therefore $d\colon (X, \alpha) \to (E, \gamma')$ is a coalgebra morphism.

In the second part of the proof we will now assume the existence of a coalgebra morphism $g\colon (X, \alpha) \to (Y, \beta)$ with $g \in \mathcal{E}$. We first show that there is a unique coalgebra morphism $j\colon (Y, \beta) \to (E, \gamma')$.

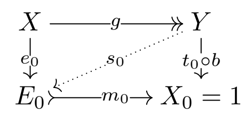
We start by showing that β is the tip of a unique cone (in $\mathbf{Mor}(\mathbf{C})$) consisting of morphisms $s_i\colon Y \to E_i$, $t_i\colon FY \to X_i$ over the ω-chain such that

$$t_i = Fs_{i-1} \qquad \text{for } i > 0 \tag{1}$$

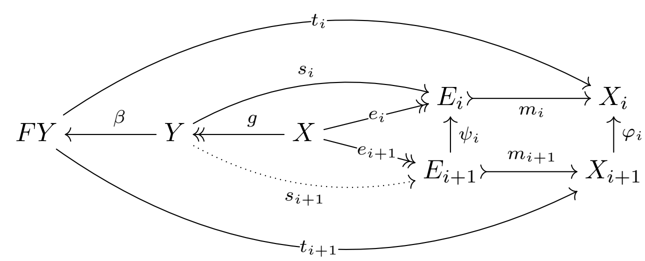
$$s_i \circ g = e_i \qquad \text{for } i \geq 0 \tag{2}$$

Being a cone amounts to $t_i \circ \beta = m_i \circ s_i$ and $\psi_i \circ s_{i+1} = s_i$. (Note that the latter implies $\varphi_i \circ t_{i+1} = F\psi_{i-1} \circ Fs_i = F(\psi_{i-1} \circ s_i) = Fs_{i-1} = t_i$.)

This cone is obtained as follows: we have $t_0\colon FY \to X_0 = 1$ as the unique arrow into the final element and $s_0\colon Y \to E_0$ as the unique diagonal arrow making the following diagram commutes (i.e., such that $s_0 \circ g = e_0$).



For the induction step assume that we have already constructed s_i, t_i. Set $t_{i+1} = Fs_i$ and we know that it is the unique choice due to (1). Now consider the diagram below and observe that $m_{i+1} \circ e_{i+1} = d_{i+1} = Fe_i \circ \alpha = F(s_i \circ g) \circ \alpha = Fs_i \circ (Fg \circ \alpha) = t_{i+1} \circ \beta \circ g$. Hence we obtain s_{i+1} as the unique diagonal arrow which satisfies Condition (2).

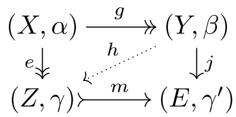


We have to check that s_{i+1}, t_{i+1} continue the cone: first $t_{i+1} \circ \beta = m_{i+1} \circ s_{i+1}$ due to the commutativity of the (lower) square above. It is left to show that $\psi_i \circ s_{i+1} = s_i$. We prove this by showing that $\psi_i \circ s_{i+1}$ is a diagonal arrow for the upper square in the diagram above: $(\psi_i \circ s_{i+1}) \circ g = \psi_i \circ e_{i+1} = e_i$ and $m_i \circ (\psi_i \circ s_{i+1}) = \psi_i \circ m_{i+1} \circ s_{i+1} = F\psi_{i-1} \circ Fs_i \circ \beta = F(\psi_{i+1} \circ s_i) \circ \beta = Fs_{i-1} \circ \beta = t_i \circ \beta$ and, by uniqueness $\psi_i \circ s_{i+1} = s_i$ (for $i > 0$). For $i = 0$ both arrows go to the final object and are hence equal.

Now we obtain $j\colon Y \to E$ and $Fj\colon FY \to FE$ as mediating morphisms between the cones consisting of s_i, t_i and p_i, r_i (and hence $p_i \circ j = s_i$). Note that Fj must be the mediating morphism since it makes the triangles commute. Furthermore $j \circ g = d$ since both are mediating morphisms from the cone (in $\mathbf{C}$) with tip X to the limit object E: d is a mediating morphism by construction and $j \circ g$ by $p_i \circ (j \circ g) = (p_i \circ j) \circ g = s_i \circ g = e_i$.

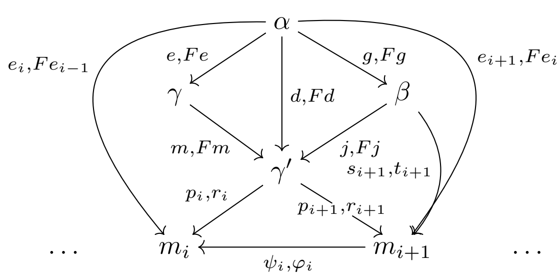
We now show uniqueness of j: let $j'\colon (Y, \beta) \to (E, \gamma')$ be another coalgebra morphism with $d = j' \circ g$. Then the arrows $p_i \circ j'$ and $r_i \circ Fj'$ form a cone satisfying the Conditions (1) and (2) above. Specifically: (1) $r_i \circ Fj' = Fp_{i-1} \circ Fj' = F(p_{i-1} \circ j')$ and (2) $(p_i \circ j') \circ g = p_i \circ d = e_i$ (where $p_i \circ j'$ plays the role of s_i and $r_i \circ Fj$ plays the role of t_i). Since the cone consisting of s_i, t_i is the unique cone satisfying these requirements we have $p_i \circ j' = s_i$, $r_i \circ Fj' = t_i$ and hence $j = j'$ (due to the uniqueness of mediating morphisms).

Finally obtain $h\colon (Y,\beta) \to (Z,\gamma)$ by constructing the diagonal arrow in the following diagram in the category of coalgebras.



In order to show that h is the unique coalgebra morphism with $e = h \circ g$, take another coalgebra morphism $h'\colon (Y,\beta) \to (Z,\gamma)$ with $h' \circ g = e$. Note that by using the diagonalization property it suffices to show that $m \circ h' = j$. We show that the arrows $p_i \circ (m \circ h')$ and $r_i \circ F(m \circ h')$ form a cone satisfying the Conditions (1) and (2) above. Condition (1) holds trivially and (2) $(p_i \circ (m \circ h')) \circ g = p_i \circ m \circ (h' \circ g) = p_i \circ m \circ e = p_i \circ d = e_i$. Since the cone consisting of s_i, t_i is the unique cone satisfying these requirements we have $p_i \circ (m \circ h') = s_i$ and hence $j = m \circ h'$ (due to the uniqueness of mediating morphisms).

In order to summarize the proof: the cones over the ω-chain in $\mathbf{Mor}(\mathbf{C})$ are as follows.

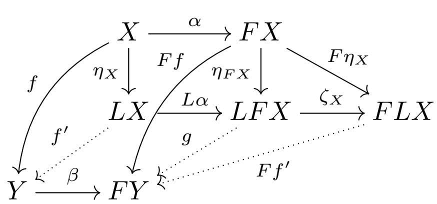


■

Proposition 18. *Let* $\mathbf{S}$ *be a reflective subcategory of* $\mathbf{C}$*, which is preserved by the endofunctor* F*. The category of* F*-coalgebras in* $\mathbf{S}$ *is a reflective subcategory of the category of* F*-coalgebras in* $\mathbf{C}$*.*

Proof: The reflection arrow is constructed as described in Definition 17. Note especially that FLX is an object of $\mathbf{S}$, since F preserves $\mathbf{S}$, and hence the arrow ζ_X exists.

As in Definition 17 let $\alpha' = \zeta_X \circ L\alpha$. Now assume that $f\colon (X,\alpha) \to (Y,\beta)$ with $\beta\colon Y \to FY$ is a coalgebra morphism where β is an arrow of $\mathbf{S}$.



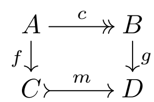
Let f' be the unique arrow in $\mathbf{S}$ for which $f' \circ \eta_X = f$ and let g be the unique arrow in $\mathbf{S}$ such that $g \circ \eta_{FX} = Ff$. We have to show that $Ff' \circ \alpha' = \beta \circ f$, i.e., f is indeed a coalgebra morphism.

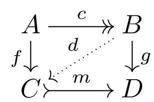
We first show that the square consisting of $L\alpha, g, f', \beta$ commutes: it holds that $(g \circ L\alpha) \circ \eta_X = g \circ \eta_{FX} \circ \alpha = Ff \circ \alpha = \beta \circ f = (\beta \circ f') \circ \eta_X$. Since η_X is the unit of a reflection and by uniqueness of the mediating arrow we obtain $g \circ L\alpha = \beta \circ f'$.

Next we show that the triangle consisting of the arrows g, ζ_X, Ff commutes: $g \circ \eta_{FX} = Ff = Ff' \circ F\eta_X = (Ff' \circ \zeta_X) \circ \eta_{FX}$. With the same argument as above (but for the unit η_{FX}) it follows that $g = Ff' \circ \zeta_X$.

Hence $Ff' \circ \alpha' = Ff' \circ \zeta_X \circ L\alpha = g \circ L\alpha = \beta \circ f'$. ■

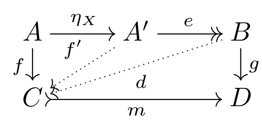
Lemma 38 (Diagonalization for Pseudo-Factorizations)**.** *Let* $\mathbf{S}$ *be a reflective subcategory of* $\mathbf{C}$*, which is* $(\mathcal{E},\mathcal{M})$*-structured. Assume a commuting diagram in* $\mathbf{C}$ *as shown on the left below where* $c = e \circ \eta_X$ *with* $e \in \mathcal{E}$ *and* $m \in \mathcal{M}$*. Furthermore let* g *be an arrow of* $\mathbf{S}$*.*





Then there exists a unique diagonal arrow d *which is contained in* $\mathbf{S}$ *and which makes the two triangles commute.*

Proof: In more detail the diagrams above look as follows:



Now C is an object of $\mathbf{S}$, since m is an arrow in $\mathbf{S}$, which implies the existence of a unique arrow $f'\colon A' \to C$ in $\mathbf{S}$ with $f' \circ \eta_X = f$.

It holds that $(g \circ e) \circ \eta_X = m \circ f = (m \circ f') \circ \eta_X$. Since both $g \circ e$ and $m \circ Ff'$ are contained in $\mathbf{S}$, it holds that $g \circ e = m \circ f'$ (uniqueness of mediating arrows). This commuting diagram lives in $\mathbf{S}$ and hence there exists a unique arrow $d\colon B \to C$ with $d \circ e = f'$ and $m \circ d = g$.

Assume there is another diagonal d' with $d' \circ e \circ \eta_X = f$ and $m \circ d' = g$. Since $d' \circ e \circ \eta_X = f' \circ \eta_X$ and since C is an object of $\mathbf{S}$ we have $d' \circ e = f'$. Uniqueness follows from the uniqueness requirement of factorization structures in $\mathbf{S}$. ■

Proposition 22. *Let* $\mathbf{C}$ *be a category with a reflective subcategory* $\mathbf{S}$*. Furthermore assume that* $\mathbf{S}$ *is* $(\mathcal{E},\mathcal{M})$*-structured. Then, given a coalgebra* $\alpha\colon X \to FX$ *in* $\mathbf{C}$*, the following two constructions obtain the same resulting minimization* $\gamma\colon Z \to FZ$ *(provided that the limit exists):*

(i) *Apply Construction 11 using the* $(\mathcal{E},\mathcal{M})$*-pseudo-factorizations of Definition 20.*
(ii) *First reflect* α *into the subcategory* $\mathbf{S}$ *according to Definition 17 and then apply Construction 11 using* $(\mathcal{E},\mathcal{M})$*-factorizations.*

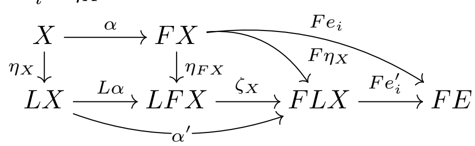
Proof: First note that the diagonal arrows required in Construction 11 (variant (i)) exist due to Lemma 38. Note that $\varphi_i \circ m_{i+1}$ is an arrow of $\mathbf{S}$, since $\varphi_i = F\psi_{i-1}$, ψ_{i-1} is a diagonal $\mathbf{S}$-arrow and F preserves $\mathbf{S}$-arrows. Furthermore $\varphi_o\colon X \to 1$ is the unique arrow into 1, which is also the final object of $\mathbf{S}$.

Assume that we apply Construction 11 (variant (i)) using the pseudo-factorizations, obtaining arrows $d_i\colon X \to X_i$, $e_i\colon X \to E_0$, $m_i\colon E_i \rightarrowtail X_i$, $\psi_i\colon E_i \to E_{i-1}$ and $\varphi_i\colon X_i \to X_{i-1}$.

Now let $\alpha'\colon LX \to FLX$ with $\alpha' = \zeta_X \circ L\alpha$ be the reflection of α into the subcategory. We call the arrows arising in Construction 11 (variant (ii)) $d'_i, e'_i, m'_i, \psi'_i, \varphi'_i$. We will show that $d_i = d'_i \circ \eta_X, e_i = e'_i \circ \eta_X, m_i = m'_i, \psi_i = \psi'_i, \varphi_i = \varphi'_i$. This is true for $i = 0$ since d_0 is the unique arrow from X to 1 and $d'_0 \circ \eta_X\colon X \to 1$. Now in order to obtain the pseudo-factorization of d_0 we first construct d'_0 and factorize $d'_0 = m'_0 \circ e'_0$. Hence $e_0 = e'_0 \circ \eta_X$ and $m_0 = m'_0$. (The arrows φ_0, ψ_0 are treated in the induction step.)

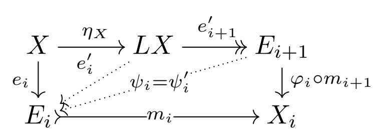
We assume by the induction hypothesis that $d_i = d'_i \circ \eta_X, e_i = e'_i \circ \eta_X, m_i = m'_i, \psi_{i-1} = \psi'_{i-1}, \varphi_{i-1} = \varphi'_{i-1}$.

For the induction step note that the diagram below commutes: the left-hand part arises from the reflection of α and the rightmost triangle commutes since it results from applying F to $e_i = e'_i \circ \eta_X$.



Hence $d_{i+1} = Fe_i \circ \alpha = Fe'_i \circ \alpha' \circ \eta_X = d'_{i+1} \circ \eta_X$. Now, as argued above, the pseudo-factorization of d_{i+1} is obtained by factorizing d'_{i+1} in the subcategory and hence $e_{i+1} = e'_{i+1} \circ \eta_X$ and $m_{i+1} = m'_{i+1}$.

In addition ψ_i and ψ'_i are both mediating arrows for the square consisting of the arrows $e'_i, m_i, e'_{i+1}, \varphi_i \circ m_{i+1}$ (see diagram below and compare with the proof of Lemma 38).



Furthermore φ_i and φ'_i are equal: for $i = 0$ this follows since they are the unique arrows from FE_0 into X_0. For $i > 0$ it holds that $\varphi_i = F\psi_{i-1} = F\psi'_{i-1} = \varphi'_i$.

Hence in both cases we take limits of the same diagrams. And since $\mathbf{S}$ is a reflective subcategory of $\mathbf{C}$ limits are preserved and we obtain the same coalgebra $\gamma'\colon E \to FE$ in both cases. Assume that $p_i\colon E \to E_i$ are the limit projections.

Let $d\colon X \to E$ and $d'\colon LX \to E$ be the mediating arrows from both coalgebras. It holds that $p_i \circ (d' \circ \eta_X) = e'_i \circ \eta_X = e_i$. Hence, by uniqueness of mediating morphisms, we have $d = d' \circ \eta_X$. Hence, the pseudo-factorization of d is obtained by factoring d' and we obtain the same resulting coalgebra $\gamma\colon Z \to FZ$ in both constructions. ∎

Proposition 26 (Induced Pseudo-Metric)**.** *Let $(\mathbf{C}, U)$ be a concrete category. Assume that there is a metric $d_Y\colon UY \times UY \to D$ for every object Y of $\mathbf{C}$. Every arrow $f\colon X \to Y$ induces a pseudo-metric $d_f\colon UX \times UX \to D$ defined by $d_f(a,b) = d_Y(Uf(a), Uf(b))$.*

Proof: In order to show that $d_f\colon UX \times UX \to D$ is a pseudo-metric we have to check that all conditions are satisfied: let $a, b, c \in UX$.

(i) $d_f(a,b) = d_Y(Uf(a), Uf(b)) \geq 0$

(ii) $d_f(a,a) = d_Y(Uf(a), Uf(a)) = 0$

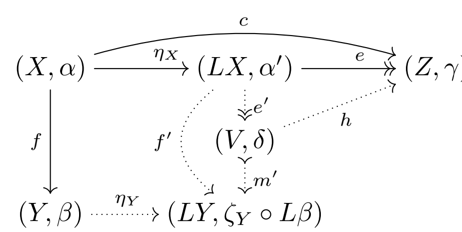
(iii) $d_f(a,b) = d_Y(Uf(a), Uf(b))$
$= d_Y(Uf(b), Uf(a)) = d_f(b,a)$

(iv) $d_f(a,c) = d_Y(Uf(a), Uf(c))$
$\leq d_Y(Uf(a), Uf(b)) \oplus d_Y(Uf(b), Uf(c))$
$= d_f(a,b) \oplus d_f(b,c)$

∎

Theorem 28. *Let $\alpha\colon X \to FX$ be a coalgebra in $\mathbf{C}$, let $\alpha'\colon LX \to FLX$ be the reflection of α into $\mathbf{S}$ and let $\gamma\colon Z \to FZ$ be the minimization for α'. In addition, let $c = e \circ \eta_X\colon (X,\alpha) \to (Z,\gamma)$ be the coalgebra morphism into the minimization where $e \in \mathcal{E}$ with $e\colon (LX,\alpha') \to (Z,\gamma)$. Then, for every other coalgebra morphism $f\colon (X,\alpha) \to (Y,\beta)$, we have $d_c(a,b) \leq d_f(a,b)$ for all $a, b \in UX$.*

Proof: We view this situation as a diagram in the category of coalgebras: f' is the morphism that is obtained by reflecting the coalgebra morphism f into the category of coalgebras in $\mathbf{S}$. Furthermore we take the $(\mathcal{E},\mathcal{M})$-factorization of f' with $f' = m' \circ e'$. Then, h is the unique morphism into the minimization γ, which makes the diagram commute.



Since η_Y and especially m' are isometric and h is non-expansive, we have for $a, b \in UX$:

$$\begin{aligned}
d_c(a,b) &= d_Z(U(h \circ e' \circ \eta_X)(a), U(h \circ e' \circ \eta_X)(b)) \\
&= d_h(U(e' \circ \eta_X)(a), U(e' \circ \eta_X)(b)) \\
&\leq d_V(U(e' \circ \eta_X)(a)), U(e' \circ \eta_X)(b)) \\
&= d_{m'}(U(e' \circ \eta_X)(a)), U(e' \circ \eta_X)(b)) \\
&= d_{LY}(U(m' \circ e' \circ \eta_X)(a)), U(m' \circ e' \circ \eta_X)(b)) \\
&= d_{LY}(U(f' \circ \eta_X)(a)), U(f' \circ \eta_X)(b)) \\
&= d_{LY}(U(\eta_Y \circ f)(a)), U(\eta_Y \circ f)(b)) \\
&= d_{\eta_Y}(Uf(a)), Uf(b)) \\
&= d_Y(Uf(a), Uf(b)) \\
&= d_f(a,b)
\end{aligned}$$

∎

Corollary 29 (Behavioural Equivalence and Induced Pseudo-Metrics)**.** *Let $(\mathbf{C}, U)$ be a concrete category, let $F\colon \mathbf{C} \to \mathbf{C}$*

be an endofunctor and we assume that the final F-coalgebra exists. Let (X, α) be a coalgebra with mapping $c\colon X \to Z$ into its minimization (as in Theorem 28). Then for $a, b \in UX$ it holds that $a \approx b$ if and only if $d_c(a, b) = 0$.

Proof: First note that there are coalgebra morphisms $beh_X\colon X \to \Omega$, $beh_Z\colon Z \to \Omega$ into the final coalgebra. In addition $beh_Z \circ c = beh_X$. According to Theorem 28 it holds that $d_c(a, b) \le d_{beh_X}(a, b)$. Furthermore, due to non-expansiveness, $d_{beh_X}(a, b) = d_{beh_Z \circ c}(a, b) = d_{beh_Z}(Uc(a), Uc(b)) \le d_Z(Uc(a), Uc(b)) = d_c(a, b)$ and thus $d_{beh_X}(a, b) = d_c(a, b)$.

Hence $x \approx y \iff Ubeh_X(a) = Ubeh_X(b) \iff d_\Omega(Ubeh_X(a), Ubeh_X(b)) = 0 \iff d_{beh_X}(a, b) = 0 \iff d_c(a, b) = 0$. ■