A Coalgebraic Perspective on Minimization and Determinization

Jiří Adámek[1], Filippo Bonchi[2], Mathias Hülsbusch[3], Barbara König[3], Stefan Milius[1], Alexandra Silva[4]

[1] Technische Universität Braunschweig [2]CNRS - ENS Lyon
[2] Universität Duisburg-Essen [4] Centrum Wiskunde & Informatica

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 in categories equipped with *factorization structures*, there exists an abstract procedure for equivalence checking. Then, we consider coalgebras in 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. We will apply our theory to the following examples: conditional transition systems, (non-deterministic) automata and linear weighted automata.

1 Introduction

Finite automata are one of the most basic structures in computer science. One particularly interesting problem is that of minimization: given a (non-)deterministic finite automaton is there an equivalent one which has a minimal number of states?

Given a regular language L, minimal deterministic automata (DA) can be thought of as the canonical acceptors of the given language L. A minimal automaton is universal, in the sense that given any automaton which recognizes the same language (and where all states are reachable) there is a unique mapping into the minimal one. Similar notions exist for other kinds of transition systems such as Mealy machines or labelled transition systems. However, in many interesting cases, such as for non-deterministic automata (NDA) or for weighted automata, what it means to be a minimal system is not yet clear. Typically, for NDA one first determinizes the automaton and then minimizes it, since for DA minimization algorithms are well-known ([17]).

It is the main aim of this paper to find a general notion of canonicity for a large class of transition systems, in a uniform manner. This encompasses two things: (i) casting the automata and the intended equivalence in a general framework; and (ii) using the general framework to devise algorithms to minimize (and determinize) the automata, yielding a canonical representative. To study all the types of automata mentioned above (and more) in a uniform setting, we use *coalgebras*.

For a functor $F\colon \mathbf{C} \to \mathbf{C}$, on a category $\mathbf{C}$, 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 a morphism of $\mathbf{C}$ defining the "transitions" of the states. For instance, given an input alphabet A, DAs are coalgebras for the functor $2 \times (-)^A\colon \mathbf{Set} \to \mathbf{Set}$ and NDAs are coalgebras for the functor $A \times (-) + 1\colon \mathbf{Rel} \to \mathbf{Rel}$, where $\mathbf{Set}$ is the category of sets and functions and $\mathbf{Rel}$ the category of sets and relations.

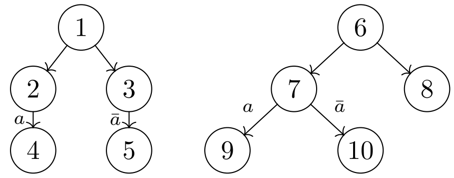
The strength of the coalgebraic approach lies in the fact that many important notions, such as behavioural equivalence, are uniquely determined by the type of the system. Under mild conditions, functors F have a final coalgebra (unique up to isomorphism) into which every F-coalgebra can be mapped via a unique homomorphism. The final coalgebra can be viewed as the universe of all possible 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 DAs, the final coalgebra is $\mathcal{P}(A^*)$ (the set of all languages over input alphabet A) and the unique homomorphism is a *function* mapping each state to the language that it accepts. In the case of NDAs, as shown in [14], the final coalgebra is A^* (the set of all finite words over A) and the unique homomorphism 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, NDAs can alternatively be modelled as coalgebras for the functor $2 \times \mathcal{P}(-)^A \colon \mathbf{Set} \to \mathbf{Set}$, where $\mathcal{P}$ is the powerset functor, but then the induced behavioural equivalence is bisimilarity (which is finer than language equivalence).

For a functor F on $\mathbf{Set}$, the *image* of an F-coalgebra under 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. For functors on categories not equipped with proper image factorization structures (such as $\mathbf{Rel}$, for instance) the situation is less clear-cut. This observation instantiates to the well-known fact that for every DA there exists an equivalent minimal automaton, while for NDAs 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 DAs are modelled), yielding the *minimization* of a coalgebra. 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 to obtain a single algorithm that does determinization and minimization simultaneously. It is worth to note that the latter approach holds for functors for which a final coalgebra does not exist.

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 DAs in $\mathbf{Set}$ and NDAs in $\mathbf{Rel}$, which is $\mathcal{K}\ell(\mathcal{P})$, where $\mathcal{P}$ is the powerset monad. 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 DAs, we recover the usual Hopcroft minimization algorithm [17]. Instantiation to NDAs gives us (a part of) Brzozowski's algorithm [7]: the obtained automata coincide with *átomata*, that are a new kind of "canonical" NDAs recently introduced in [8]. For LWAs, we obtain Boreale's minimization algorithm [6].

Conditional Transition Systems (CTS). To better illustrate our work, we employ transition systems labelled with *conditions* that have similarly been studied in [15, 10]. 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 transitions labelled a and $\bar{a}$).



(1)

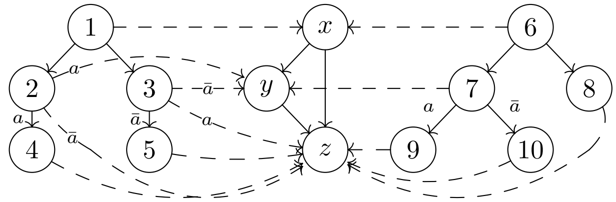
The environment can make one choice, which can not be changed later: it decides whether to take either a or $\bar{a}$. Regardless of the specific choice of the environment, the two states 1 and 6 in (1) above will be bisimilar. If a holds then the systems above would be instantiated to the transition system on the left below. Instead if a does not hold then the systems we obtain the system on the right. In both cases, the instances of the states 1 and 6 are bisimilar.

(2)

This shows that one possible way to solve the question whether two states are *always* bisimilar consists in enumerating all conditions and to create suitably many instantiations of the transition system. Then the resulting transition system can be minimized with respect to bisimilarity. This is analogous to the steps of determinization and minimization for NDAs. Indeed, the base category of coalgebras of CTSs, as $\mathbf{Rel}$ for NDAs, has no suitable factorization structures. In order to minimize (both NDAs and CTSs), coalgebras should be transformed via *reflections* that, in the case of NDAs means determinizing, while for CTS, means instiantiating the automata for all the conditions.

In this work, we will study both constructions in a general setting and also show how they can be combined into a single algorithm. For CTSs this mean that we will provide an algorithm that checks if two states are bisimilar under all the possible conditions, without performing all the possible instantiations.

Now, what would be a canonical representative of the systems above? In other words, is there a system into which CTS (1) can be mapped? In the example above, it is relatively easy to see that system would be the transition system consisting of states x, y, z in (3) below. One would map both 1 and 6 to x, 7 to y, $4, 5, 9$ and 10 to z. What about 2 and 3? We want to map 2 to y whenever a holds and to z whenever $\bar{a}$ holds, dually for 3. In order to do that we need to work in a category where we can represent such *conditional* maps. As we will show in the sequel, by modelling CTS as coalgebras in a Kleisli category this will be possible. The full mapping is represented below.



(3)

2 Background material on coalgebras

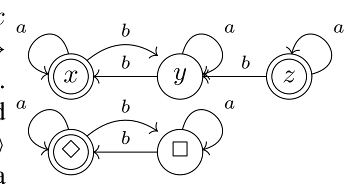
We assume some prior knowledge of category theory (categories, functors, monads, limits and adjunctions). Definitions can be found in [2]. However, to establish some notation, we recall some basic definitions. We denote by Ord the class of all ordinals. Let $\mathbf{Set}$ be the category of sets and functions. Sets (and other objects) are denoted by capital letters $X, Y, \ldots$ and functions (and other morphisms) 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 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, as $X + Y$. These operations, defined on sets, can analogously be defined on functions, yielding (bi-)functors. A category $\mathbf{C}$ is called *concrete* if a faithful functor $U : \mathbf{C} \to \mathbf{Set}$ is given.

Definition 2.1 (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 $\alpha : X \to FX$ a morphism in $\mathbf{C}$. A* (coalgebra) homomorphism *$f : (X, \alpha) \to (Y, \beta)$ between two coalgebras $\alpha : X \to FX$ and $\beta : Y \to FY$ is a $\mathbf{C}$-morphism $f : X \to Y$ such that $Ff \circ \alpha = \beta \circ f$.*

An F-coalgebra (Ω, ω) is *final* if for any F-coalgebra (X, α) there exists a unique homomorphism $beh_X : (X, \alpha) \to (\Omega, \omega)$. If $\mathbf{C}$ is concrete we can define behavioural equivalence. Given an F-coalgebra (X, α) and $x, y \in UX$, we say that x and y are *behaviourally equivalent*, written $x \approx y$, if and only if there exist an F-coalgebra (Z, γ) and a homomorphism $f : (X, \alpha) \to (Z, \gamma)$ such that $Uf(x) = Uf(y)$. If a final F-coalgebra exists, we have a simpler characterization of behavioural equivalence: $x \approx y$ iff $Ubeh_X(x) = Ubeh_X(y)$.

Example 2.2. (DA) A deterministic automaton over the 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 $a \in A$ the next state. DAs are coalgebras for the functor $FX = 2 \times X^A$ on $\mathbf{Set}$. The final coalgebra for 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 homomorphism 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.

Take $A = \{a, b\}$ and consider the DAs on the right. We call the topmost (X, α) where $X = \{x, y, z\}$ 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 bottom one 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 homomorphism, take 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 $2 \times \mathcal{P}(-)^A$ (on $\mathbf{Set}$): to each input in A, we assign a set of possible successors states. Unfortunately, the resulting behavioural equivalence is not language equivalence (as for DAs), but bisimilarity (i.e., it only identifies states having the same branching structure). In [25, 14], it is shown that in order to retrieve language equivalence for NDAs, one should consider coalgebras in a Kleisli category. In what follows, we introduce Kleisli categories, in which we model NDAs, LWAs and CTSs as coalgebras. While objects in a Kleisli category are sets, morphisms are generalized functions that incorporate side effects, such as non-determinism, specified by a monad (see Appendix A or [2, 14, 20]).

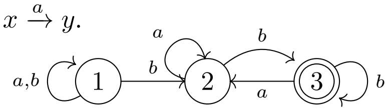
Definition 2.3 (Kleisli Category). *Let* $(T\colon \mathbf{Set} \to \mathbf{Set}, \eta, \mu)$ *(or simply* T*) be a monad on* $\mathbf{Set}$*. Its* Kleisli category $\mathcal{K}\ell(T)$ *has sets as objects and a morphism* $X \to Y$ *in* $\mathcal{K}\ell(T)$ *is a function* $X \to TY$*. The identity* id_X *is* η_X *and the composition* $g \circ f$ *of* $f\colon X \to Y$*,* $g\colon Y \to Z$ *(i.e., functions* $f\colon X \to TY$*,* $g\colon Y \to TZ$*) is* $\mu_Z \circ Tg \circ f$*.*

In the following we will employ overloading and use the same letter to both denote a morphism in $\mathcal{K}\ell(T)$ and the corresponding function in $\mathbf{Set}$. Furthermore, note that $\mathbf{Set}$ can be seen as a (non-full) subcategory of $\mathcal{K}\ell(T)$, where each function $f\colon X \to Y$ 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 a morphism $f\colon X \to Y$.

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 [14, 23]): a functor $\overline{F}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ is called a *lifting of* $F\colon \mathbf{Set} \to \mathbf{Set}$ 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 $\mathcal{K}\ell(T)$ 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 morphism 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 2.4. (NDA) Consider the powerset monad $TX = \mathcal{P}(X)$, fully described in Example A.2 (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 in $\mathbf{Rel}$ for the functor $\overline{F}$. 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 on the right is 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 [14], 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 homomorphism beh_X into A^* is the relation that links every state in X with all the words in A^* that it accepts.

Example 2.5. (CTS) We shortly discuss how to specify the example from the introduction in a Kleisli category. All the details can be found in Appendix C.

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 $f: X \to Y$, $Tf: TX \to TY$ is $f^A: X^A \to Y^A$ defined for all $g \in X^A$ and $a \in A$ as $f^A(g)(a) = f(g(a))$.

Note that a morphism $f: X \to Y$ in the Kleisli category over the input monad is a function $f: X \to Y^A$. For instance, the dashed arrows in the introduction describe a morphism 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: a morphism $f: X \to Y$ in $\mathcal{K}\ell(T)$, which is a function of the form $f: X \to Y^A$, is mapped to $\overline{F}f: \mathcal{P}_c(X) \to \mathcal{P}_c(Y)$ with $\overline{F}f(X')(a) = \{f(x)(a) \mid x \in X'\}$ for $X' \subseteq X$, $a \in A$. Hence, CTS (1) from the introduction is modelled by a morphism $\alpha: X \to \mathcal{P}_c(X)$ in $\mathcal{K}\ell(T)$ (i.e., a function $\alpha: X \to \mathcal{P}_c(X)^A$), where $X = \{1, \ldots, 10\}$ 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 the matrix on the right.

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 $\alpha: X \to \mathcal{P}_c(X)^A$ 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. 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 Appendix C, we prove that behavioural equivalence of $\overline{F}$-coalgebras coincides with the expected one.

3 Minimization via $(\mathcal{E}, \mathcal{M})$-Factorizations

We now introduce the notion of minimization of a coalgebra and its iterative construction that generalizes the minimization of transition systems via partition refinement. This notion is parametrized by two classes $\mathcal{E}$ and $\mathcal{M}$ of morphisms that form a factorization structure for the considered category $\mathbf{C}$.

Definition 3.1 (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* $\mathbf{C}$ *whenever*

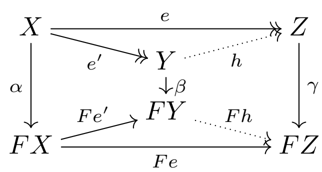
- $\mathcal{E}$ *and* $\mathcal{M}$ *are closed under composition with isos.*
- $\mathbf{C}$ *has* $(\mathcal{E}, \mathcal{M})$-factorizations *of morphisms, i.e., each morphism* f *of* $\mathbf{C}$ *has a factorization* $f = m \circ e$ *with* $e \in \mathcal{E}$ *and* $m \in \mathcal{M}$.
- $\mathbf{C}$ *has the unique* $(\mathcal{E}, \mathcal{M})$-diagonalization property*: for each commutative square as shown on the left-hand side 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* $m \circ d = g$*). If all morphisms in* $\mathcal{E}$ *are epis we call* $(\mathcal{E}, \mathcal{M})$ *a* right factorization structure.

$$\begin{array}{ccc} A & \xrightarrow{e} & B \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle g} \\ C & \xrightarrow{m} & D \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{e} & B \\ {\scriptstyle f}\downarrow & \overset{d}{\swarrow} & \downarrow{\scriptstyle g} \\ C & \xrightarrow{m} & D \end{array}$$

In any category with an $(\mathcal{E}, \mathcal{M})$-factorization structure, the classes $\mathcal{E}, \mathcal{M}$ are closed under composition and factorizations of morphisms are unique up to iso (see [2]). For $\mathbf{Set}$ we always consider below the factorization structure $(\mathcal{E}, \mathcal{M})$ with $\mathcal{E} =$ epimorphims (surjections) and $\mathcal{M} =$ monomorphisms (injections); for the category $\mathbf{Set^{op}}$ we take the corresponding structure $(\mathcal{M}, \mathcal{E})$, i.e., where the epic part consists of functions that in $\mathbf{Set}$ are monomorphisms, analogously with $\mathcal{E}$. Morphisms from $\mathcal{E}$ are drawn using double-headed arrows $A \twoheadrightarrow B$, whereas morphisms from $\mathcal{M}$ are depicted using arrows of the form $A \rightarrowtail B$. Whenever the endofunctor F preserves $\mathcal{M}$-morphisms, which we assume in the following, the factorization structure can be straightforwardly lifted to coalgebra homomorphisms (see Lemma G.1 in Appendix G or [19]).

Assumption 3.2 *We assume that* $\mathbf{C}$ *is a complete category with a right* $(\mathcal{E}, \mathcal{M})$*-factorization structure and* $\mathbf{C}$ *is* $\mathcal{E}$*-cowellpowered, i. e., every object* X *only has a set of* $\mathcal{E}$*-quotients (i.e.,* $\mathcal{E}$*-morphisms with domain* X *up to isomorphism of the codomains). We also assume that* $F\colon \mathbf{C} \to \mathbf{C}$ *is a functor preserving* $\mathcal{M}$*, i. e., if* $m \in \mathcal{M}$ *then* $Fm \in \mathcal{M}$*.*

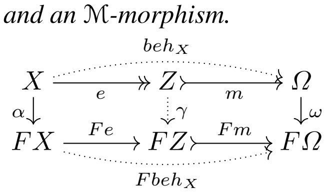
Definition 3.3 (Minimization). *The* minimization *of a coalgebra* $\alpha\colon X \to FX$ *is the greatest* $\mathcal{E}$*-quotient coalgebra. More precisely, the minimization is a coalgebra* (Z, γ) *with a homomorphism* $e\colon (X, \alpha) \twoheadrightarrow (Z, \gamma)$ *with* $e \in \mathcal{E}$ *such that for any other coalgebra homomorphism* $e'\colon (X, \alpha) \twoheadrightarrow (Y, \beta)$ *with* $e' \in \mathcal{E}$ *there exists a (necessarily) unique coalgebra homorphism* $h\colon (Y, \beta) \to (Z, \gamma)$ *such that* $e = h \circ e'$*.*



Remark 3.4. (1) Since $\mathbf{C}$ is $\mathcal{E}$-cowellpowered and $\mathcal{E}$ consists of epimorphisms, the $\mathcal{E}$-quotient coalgebras of a coalgebra (X, α) form a pre-ordered set: a quotient coalgebra $e'\colon (X, \alpha) \twoheadrightarrow (Y', \beta')$ is larger than $e\colon (X, \alpha) \twoheadrightarrow (Y, \beta)$ iff there exists a coalgebra homomorphism $h\colon (Y, \beta) \to (Y', \beta')$ with $e' = h \circ e$; notice that h is uniquely determined and $h \in \mathcal{E}$ by the properties of factorization systems. Thus, the minimization is simply the greatest element in the pre-order of $\mathcal{E}$-quotient coalgebras of (X, α).
(2) While in $\mathbf{Set}$ the minimization is also determined by the strict minimality of the number of states, this is not necessarily true for other categories, as we will show in Example 4.10.
(3) We often speak about (Z, γ) (without explicitly referring to the morphism e) or even just the object Z as the minimization of the given coalgebra.

Theorem 3.8 will show that under Assumption 3.2 the minimization always exists, even when there is no final coalgebra. When the final coalgebra exists, minimization is the quotient of the unique morphism.

Proposition 3.5 (Minimization and Final Coalgebra). *If the final coalgebra* $\omega\colon \Omega \to F\Omega$ *exists, then – for a given coalgebra* $\alpha\colon X \to FX$ *– the minimization* $\gamma\colon Z \to FZ$ *can be obtained by factoring the unique coalgebra homomorphism* $beh_X\colon (X, \alpha) \to (\Omega, \omega)$ *into an* $\mathcal{E}$*-morphism and an* $\mathcal{M}$*-morphism.*



Note that whenever the concretization functor $U\colon \mathbf{C} \to \mathbf{Set}$ maps $\mathcal{M}$-morphisms to injections, $x, y \in UX$ are behaviourally equivalent ($x \approx y$) iff $Ue(x) = Ue(y)$.

Example 3.6 (DA, Minimal Automata). Recall that DAs are coalgebras for the functor $FX = 2 \times X^A$ on $\mathbf{Set}$ (Example 2.2). In this case, minimization corresponds to the well known minimization of deterministic automata. For instance, the minimization of the top automaton (X, α) in Example 2.2 yields the automaton (Z, γ) (on the bottom).

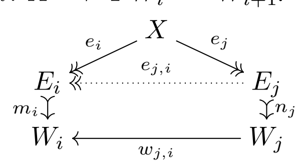
We now describe a construction that – given a coalgebra (X, α) – obtains the minimization γ without going via the final coalgebra. This closely resembles the partition refinement algorithm for minimizing deterministic automata or for computing bisimilarity. Whenever the construction below becomes stationary, 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 (in the dual setting) of the initial algebra by Adámek [1], for the coalgebraic version see Worrel [29] and Adámek and Koubek [3]. As in those papers, our construction works for ordinals beyond ω. Hereafter 1 denotes the final object of $\mathbf{C}$.

Construction 3.7 *Recall the final chain* $W\colon \mathsf{Ord} \to \mathbf{C}$ *given by*

$$W_0 = 1, \qquad W_{i+1} = FW_i, \qquad W_j = \lim_{i<j} W_j \quad (j \text{ a limit ordinal.})$$

This is the unique chain, up to natural isomorphism, whose connecting morphisms $w_{i,j}$ *fulfil (a)* $w_{i+1,j+1} = Fw_{i,j}$ *and (b) for limit ordinals they form a limit cone.*

As we do not assume that F *has a final coalgebra, the chain* W *need not converge. Every coalgebra* $\alpha\colon X \to FX$ *defines a unique* canonical cone $(\alpha_i\colon X \to W_i)_{i\in\mathsf{Ord}}$ *on* W *with the property that* $\alpha_{i+1} = F\alpha_i \circ \alpha\colon X \to FW_i = W_{i+1}$*. Let* $e_i\colon X \twoheadrightarrow E_i$*,* $m_i\colon E_i \rightarrowtail W_i$ *be an* $(\mathcal{E},\mathcal{M})$*-factorization of* α_i*. Then, we obtain an ordinal indexed chain* (E_i) *of quotients of* X *with the connecting morphisms* $e_{j,i}$ *obtained by diagonalization for* $i < j$*, as depicted on the right.*



Theorem 3.8. *For every* F*-coalgebra* (X, α)*, its minimization is* E_i*, for some* $i \in \mathsf{Ord}$.

More precisely, there exists an ordinal i such that E_i carries a coalgebra structure $\varepsilon\colon E_i \to FE_i$ such that $e_i\colon (X, \alpha) \to (E_i, \varepsilon)$ is the minimization; for details see the proof of Theorem 3.8 in Appendix G.

By the above theorem, minimizations always exist even when there is no final coalgebra. Worrell [29] shows that for a finitary functor $F\colon \mathbf{Set} \to \mathbf{Set}$, the final chain W_i converges at the final coalgebra in $\omega + \omega$ iterations. The chain E_i, instead, converges at the minimization in ω iterations.

Theorem 3.9. *Let* $F\colon \mathbf{Set} \to \mathbf{Set}$ *be a finitary functor. Then for every* F*-coalgebra* (X, α)*, its minimization is* E_ω*.*

In our examples, we will use the following construction which is closer to the standard minimzation algorithm and to any reasonable implementation of Construction 3.7.

Theorem 3.10. *The chain* $(E_i)_{i\in\mathsf{Ord}}$ *of Construction 3.7 can also be defined as follows:*

(a) Factor the unique morphism $d_0\colon X \to 1$ *into* $e_0\colon X \twoheadrightarrow E_0$ *and* $n_0\colon E_0 \rightarrowtail 1$*.*

(b) Given $e_i\colon X \twoheadrightarrow E_i$*, factor* $d_{i+1} = Fe_i \circ \alpha$ *into* $e_{i+1}\colon X \twoheadrightarrow E_{i+1}$ *and* $n_{i+1}\colon E_{i+1} \rightarrowtail FE_i$*.*

(c) For a limit ordinal j*, form a limit of the preceding chain* $(E_i)_{i<j}$*, obtaining* $\hat{E}_j$ *and* $\hat{e}_j\colon X \to \hat{E}_j$ *as mediating morphism. Then factor* $\hat{e}_j$ *into* $e_j\colon X \twoheadrightarrow E_j$ *and* $n_j\colon E_j \rightarrowtail \hat{E}_j$*.*

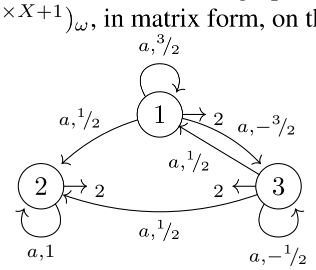
By instantiating the above construction to the case of DAs, we obtain the standard minimization algorithm by Hopcroft [17], as will be illustrated in Appendix D.

Example 3.11. (LWA) We study automata with weights taken from a field (linear weighted automata, see [6]). 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 morphism $X \to Y$ 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 morphism 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 $\bullet$ – denoting termination – stands for the only element of 1. We lift F to $\mathcal{K}\ell(T)$ as follows: a morphism $f\colon X \to Y$ 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 otherwise. Hence transitions carry labels from A (for the explicit branching) and weights (for the implicit branching).

An example LWA for $A = \{a\}$ and $\mathbb{F} = \mathbb{R}$ 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 $\mathcal{E}$-morphisms the matrices of full row rank (i.e., the monos) and as $\mathcal{M}$-morphisms 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 [6] (see Appendix F for an elaboration of this claim and for an example minimization involving the automaton above).

4 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 minimization of non-deterministic automata is not unique. 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 conditional transition systems. 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.

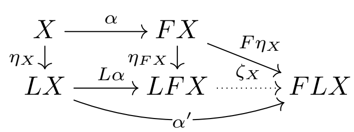
Definition 4.1 (Reflective Subcategory). *Let* $\mathbf{S}$ *be a subcategory of* $\mathbf{C}$*. Let* X *be an object of* $\mathbf{C}$*. An* $\mathbf{S}$-reflection *for* X *is a morphism* $\eta_X : X \to X'$*, where* X' *is an* $\mathbf{S}$*-object, such that for every other morphism* $f\colon X \to Y$ *with* Y *in* $\mathbf{S}$ *there exists a unique* $\mathbf{S}$*-morphism* $f'\colon X' \to Y$ *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.*

This definition is equivalent to saying that the functor embedding $\mathbf{S}$ into $\mathbf{C}$ has a left adjoint $L\colon \mathbf{C} \to \mathbf{S}$ called *reflector*. The morphisms η_X form the unit of this adjunction. In our examples in $\mathcal{K}\ell(T)$, the unit η of the reflection will *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.

Example 4.2. (NDA) The category $\mathbf{Set^{op}}$ is a reflective subcategory of $\mathbf{Rel}$. The reflector L is the contravariant 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 $\mathbf{Set^{op}}$ where $L(R)$ maps $Y' \subseteq Y$ to $R^{-1}(Y')$. The reflection $\eta_X : X \to \mathcal{P}(X)$ 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 $f\colon X \to Y^A$ corresponds to a function $f'\colon A \times X \to Y$ by currying, the category $\mathcal{K}\ell(T)$ 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 reflection is the Kleisli morphism $\eta_X : X \to A \times X$ with $\eta_X(x)(a) = \langle a, x\rangle$. The reflector L coincides with V on objects and takes the product of the state set X with the label set A. More concretely, for a morphism $f\colon X \to Y$ in $\mathcal{K}\ell(T)$ we obtain a morphism $Lf\colon A \times X \to A \times Y$ in $\mathbf{Set}$ with $Lf(\langle a, x\rangle) = \langle a, f(x)(a)\rangle$.

Definition 4.3 (Reflection of Coalgebras). *Let* $\mathbf{S}$ *be a reflective subcategory of a category* $\mathbf{C}$ *and let* $L\colon \mathbf{C} \to \mathbf{S}$ *be the reflector. Assume that* $\mathbf{S}$ *is preserved by the endofunctor* F*. Then, given a coalgebra* $\alpha\colon X \to FX$ *in* $\mathbf{C}$ *we reflect it into* $\mathbf{S}$*, obtaining a coalgebra* $\alpha'\colon LX \to FLX$ *by the following construction:*



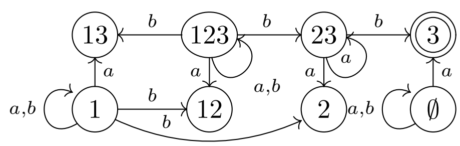
Note that the existence of a unique morphism ζ_X *is guaranteed by Definition 4.3, since* F *preserves* $\mathbf{S}$ *and hence* FLX *is an object of* $\mathbf{S}$*.*

That the above construction indeed gives a reflection of coalgebras for F is a special instance of a known result (see for instance Hermida and Jacobs [16], Corollary 2.15). In Appendix G we give a proof for the convenience of the reader.

Proposition 4.4. *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* $\mathbf{C}$*.*

A limit in a reflective subcategory $\mathbf{S}$ is also a limit in $\mathbf{C}$. Hence, if the final coalgebra exists in the subcategory $\mathbf{S}$, it is also the final coalgebra in $\mathbf{C}$. In particular, whenever $\mathbf{S}$ is complete, the chain (W_i) (Construction 3.7) in $\mathbf{S}$ will coincide with the chain in $\mathbf{C}$.

Example 4.5. (NDA) We will first study the effect of a reflection on a non-deterministic automaton, for which we use the reflective subcategory $\mathbf{Set^{op}}$ of $\mathbf{Rel}$ (see Example 4.2). The effect of the reflection on coalgebras is a powerset automaton which is however "backwards-deterministic": more specifically, given a coalgebra $\alpha\colon X \to A \times X + 1$ in $\mathbf{Rel}$, the reflected coalgebra $\alpha'\colon \mathcal{P}(X) \to A \times \mathcal{P}(X) + 1$ is a relation which lives in $\mathbf{Set^{op}}$ and, when seen as a function, maps $\langle a, X' \rangle$ with $X' \subseteq X$ to $\{x \in X \mid \exists x' \in X'\colon \langle a, x' \rangle \in \alpha(x)\}$ (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).

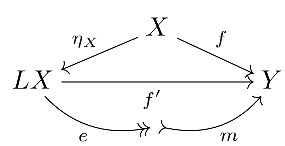


For instance, the reflection of the NDA (X, α) in Example 2.4 is the above backwards-deterministic automaton. Note that the above automaton has a single final state (consisting of the set of final states of the original automaton) and every state has a unique predecessor for each alphabet letter. For this reason, it can be seen as a function $\alpha'\colon A \times Y + 1 \to Y$ (i.e., an algebra for the functor $FY = A \times Y + 1$). 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 $\mathbf{Set^{op}}$, we have exactly the same final coalgebra as for $\mathbf{Rel}$, which, as shown in [14], is the *initial algebra* in $\mathbf{Set}$.

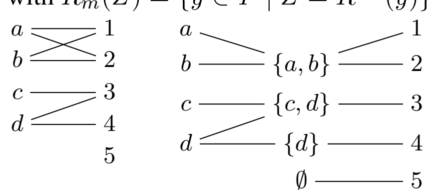
(CTS) Now we come back to the Kleisli category $\mathcal{K}\ell(T)$ over the input monad T (see Example 2.5) and coalgebras with endofunctor $\mathcal{P}_c$. As discussed in Example 4.2, $\mathbf{Set}$ is a reflective subcategory of $\mathcal{K}\ell(T)$. On coalgebras reflection has the following effect: given a coalgebra $\alpha\colon X \to \mathcal{P}_c(X)$ in $\mathcal{K}\ell(T)$ we obtain a reflected coalgebra $\alpha'\colon A \times X \to \mathcal{P}_c(A \times X)$ 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 CTS (1) (formally introduced in Example 2.5, see also the introduction) is CTS (2) from the introduction.

We now consider other forms of factorizations that do not conform to Definition 3.1.

Definition 4.6 (Pseudo-Factorization). *Let* $\mathbf{C}$ *be a category and let* $\mathbf{S}$ *be a reflective subcategory with a factorization structure* $(\mathcal{E}, \mathcal{M})$*. Let* $f\colon X \to Y$ *be a morphism of* $\mathbf{C}$ *where* Y *is an object of* $\mathbf{S}$*. Take the unique morphism* $f'\colon LX \to Y$ *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 4.7. (NDA) Consider $\mathbf{Set^{op}}$ as the reflective subcategory of $\mathbf{Rel}$ (Example 4.2). Given a relation $R\colon X \to Y$, 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 $R_c\colon X \to \mathcal{Z}$ with $R_c(x) = \{Z \in \mathcal{Z} \mid x \in Z\}$ and $R_m\colon \mathcal{Z} \to Y$ with $R_m(Z) = \{y \in Y \mid Z = R^{-1}(y)\}$. Note that R_m lies in $\mathcal{E}$, and $R_m \circ R_c = R$. As an example consider the relation R between sets $X = \{a, b, c, d\}$ and $Y = \{1, 2, 3, 4, 5\}$ visualized 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 lies in $\mathcal{E}$, mapping elements of Y to their preimage under R in $\mathcal{P}(X)$.

(CTS) For $\mathbf{Set}$, 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 a morphism $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, seen as a function $f\colon X \to Y^A$, we define $Y' = \{y \in Y \mid \exists x \in X, a \in A\colon f(x)(a) = y\}$. Then $f_c\colon X \to Y'^A$ with $f_c(x)(a) = f(x)(a)$ and $f_m\colon Y' \to Y^A$ 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 3.1 whenever g is a morphism of $\mathbf{S}$ (see Lemma G.5 in Appendix G). However pseudo-factors are not necessarily closed under composition with the isos of $\mathbf{C}$.

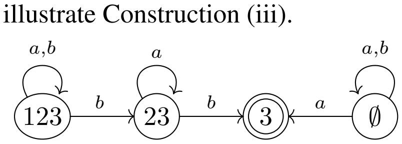
Assumption 4.8 *We assume that* $\mathbf{S}$ *is a reflective subcategory of* $\mathbf{C}$. *We also assume that an endofunctor* F *of* $\mathbf{C}$ *is given preserving* $\mathbf{S}$. *And* $\mathbf{S}$ *and* F *fulfil Assumption 3.2.*

Theorem 4.9. *Given a coalgebra* $\alpha\colon X \to FX$ *in* $\mathbf{C}$*, the following four constructions obtain the same result (we also call this result the* minimization*):*
- *(i) Apply Construction 3.7 using the* $(\mathcal{E}, \mathcal{M})$*-pseudo-factorizations of Definition 4.6.*
- *(ii) Reflect* α *into the subcategory* $\mathbf{S}$ *according to Definition 4.3 and then apply Construction 3.7 using* $(\mathcal{E}, \mathcal{M})$*-factorizations.*
- *(iii) Apply the construction of Theorem 3.10 using* $(\mathcal{E}, \mathcal{M})$*-pseudo-factorizations.*
- *(iv) Reflect* α *into the subcategory* $\mathbf{S}$ *and then apply the construction of Theorem 3.10 using* $(\mathcal{E}, \mathcal{M})$*-factorizations.*

Note that we do not have to require here that $\mathbf{C}$ is complete. As will become clear in the proof of Theorem 4.9 (see Appendix G) Construction 3.7 and the construction in Theorem 3.10 can be straightforwardly adapted to pseudo-factorizations instead of factorizations: The quotients E_i and the chain $e_{j,i}$ of connecting morphisms obtained in variants (i)–(iv) are identical and live in the subcategory $\mathbf{S}$. Since $\mathbf{S}$ is reflective in $\mathbf{C}$ we obtain the same results when taking the limit in $\mathbf{C}$ or in $\mathbf{S}$, respectively.

Variant (iii) allows to tightly integrate minimization with a determinization-like construction, i.e., to do both simultaneously instead of sequentially. For practical purposes it is usually the most efficient solution, since it avoids building the final chain of Construction 3.7 and the reflected coalgebra of Definition 4.3 which both usually involve significant combinatorial explosion.

Example 4.10. (NDA) Theorem 4.9 suggests two ways to build the minimization of an NDA (and thus checking the equivalence of its states). We first apply Construction (iv) to the NDA (X, α) in Example 2.4 and then we illustrate Construction (iii).



Recall that the reflection of (X, α) into $\mathbf{Set^{op}}$ is $(\mathcal{P}(X), \alpha')$ in Example 4.5. By applying Construction 3.7 (with the factorization structure of $\mathbf{Set^{op}}$), 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. The resulting automaton is illustrated above.

Construction (iii) can be understood as an efficient implementation of Construction (iv): 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 are shown in Appendix E.

The minimized NDA can be thought of as a *canonical* representative of its equivalence class. The quest for canonical NDAs (also referred to as "universal") started in the seventies and, recently, an interesting kind of canonical NDAs (called *átomata*) has been proposed in [8]. In Appendix B, we show that our minimized NDAs coincide with átomata of [8]. This provides a universal property that uniquely characterizes átomata (up to isomorphism), namely the átomaton of a regular language is the minimization of any NDA accepting the language.

It is worth noting 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.

Example 4.11. (CTS) Recall the coalgebraic description of CTS given in Example 2.5: the base category is $\mathcal{K}\ell(T)$, where T is the input monad and $\overline{F} = \mathcal{P}_c$ is the countable powerset functor. CTS (1) of the introduction is the coalgebra $\alpha\colon X \to \mathcal{P}_c(X)$ represented by the table in Example 2.5.

We describe the algorithm in Theorem 4.9(iii) with the pseudo-factorization of Example 4.7 (Construction (iv) only consists in the standard minimization of the reflected coalgebra α', that is CTS (2) of the introduction). We start by taking the unique morphism $d_0\colon X \to 1$ into the final object of $\mathcal{K}\ell(T)$, that is $1 = \{\bullet\}$. At the iteration i, we obtain e_i via the pseudo-factorization of $d_i = n_i \circ e_i$, and then we build $d_{i+1} = \overline{F} e_i \circ \alpha$. The iterations of the algorithm are shown in the following tables below.

$d_0\colon X \to 1 = \{\bullet\} = E_0$

d_0, e_0	1	2	3	4	5	6	7	8	9	10
a	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
$\bar{a}$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$

$d_1\colon X \to \mathcal{P}_c(E_0) = \{\emptyset, \{\bullet\}\} = E_1$

d_1, e_1	1	2	3	4	5	6	7	8	9	10
a	$\{\bullet\}$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\bullet\}$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\bar{a}$	$\{\bullet\}$	$\emptyset$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\{\bullet\}$	$\{\bullet\}$	$\emptyset$	$\emptyset$	$\emptyset$

$d_2\colon X \to \mathcal{P}_c(E_1),\ E_2 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\bullet\}\}\}$

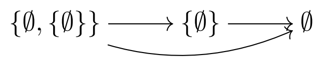
d_2, e_2	1	2	3	4	5	6	7	8	9	10
a	$\{\emptyset, \{\bullet\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\emptyset, \{\bullet\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\bar{a}$	$\{\emptyset, \{\bullet\}\}$	$\emptyset$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\{\emptyset, \{\bullet\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$

$d_3\colon X \to \mathcal{P}_c(E_2),\ E_3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$

d_3, e_3	1	2	3	4	5	6	7	8	9	10
a	$\{\emptyset, \{\emptyset\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\{\emptyset, \{\emptyset\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$
$\bar{a}$	$\{\emptyset, \{\emptyset\}\}$	$\emptyset$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\{\emptyset, \{\emptyset\}\}$	$\{\emptyset\}$	$\emptyset$	$\emptyset$	$\emptyset$

Each table represents both d_i and $e_i\colon X \twoheadrightarrow E_i$ (the morphisms n_i such that $d_i = n_i \circ e_i$ are just the obvious injections). At the iterations 0 and 1, $E_0 = 1$ and $E_1 = \mathcal{P}_c(E_0)$. At the iteration 2 instead, $E_2 \neq \mathcal{P}_c(E_1)$, since nothing maps to $\{\{\bullet\}\} \in \mathcal{P}_c(E_1)$.

The algorithm reaches a fixed-point at iteration 3, since there is an iso $\iota\colon E_2 \to E_3$. The minimization $(E_3, \mathcal{P}_c(\iota) \circ n_3)$ is depicted below.



It is easy to see that the above transition system is isomorphic to the one from the introduction having states x, y, z. Moreover, the coalgebra morphism $e_3\colon (X, \alpha) \twoheadrightarrow (E_3, \mathcal{P}_c(\iota) \circ n_3)$, illustrated in the table above, corresponds to the dashed arrow of the introduction, where 2 is mapped to $\{\emptyset\}$ $(= y)$ if a holds, and to $\emptyset$ $(= z)$ if $\bar{a}$ holds.

5 Conclusion, Related and Future Work

In this work, we have introduced a notion of minimization, which encompasses several concepts of "canonical" systems in the literature, and abstract procedures to compute it. Our approach only relies on *(pseudo-)factorization structures* and it is completely independent of the base category and of the endofunctor F. Together with appropriate reflections, this allows to compute minimizations of interesting types of systems that, for the purpose of minimization, cannot be regarded as coalgebras over $\mathbf{Set}$, such as non-deterministic automata, linear weighted automata and conditional transition sytems.

For non-deterministic automata, which we model as coalgebras in $\mathbf{Rel}$ following [14], the result of the proposed algorithm coincides with the one of the third step of Brzozowski's algorithm [7]. The resulting automata are not minimal in the number of states (it is well-known that there exists no unique minimal non-deterministic automata), but they correspond to átomata, recently introduced in [8] (as shown in Appendix B).

The construction of linear weighted automata as coalgebras in a Kleisli category is new, while the resulting algorithm coincides with the one in [6]. The example of conditional transition systems is completely original, but it has been motivated by the work in [15, 10], which introduces notions of bisimilarity depending on conditions (which are fixed once and for all). The setting of [10] is closer to ours, but no algorithm is given there. Our algorithm can be made more efficient by considering CTSs where conditions are boolean expressions. We already have a prototype implementation performing the fixed-point iteration based on binary decision diagrams. Moreover, our coalgebraic model of CTSs provides a notion of *quantitative bisimulations* (Definition C.2 in Appendix C) that can be seen as a behavioural (pseudo-)metric. We plan to study how our approach can be integrated to define and compute behavioural metrics.

As related work, we should also mention that the notion of minimization generalizes simple [26] and minimal [13] coalgebras in the case where the base category is $\mathbf{Set}$ with epi-mono factorizations. Moreover, several previous studies (e.g. [19, 9, 28]) have pointed out the relationship between the construction of the final coalgebra (via the final chain [29, 3]) and the minimization algorithm. For instance, in case of regular categories the chain of quotients $e_i\colon X \twoheadrightarrow E_i$ (Construction 3.7) corresponds to the chain $K_i \rightarrowtail X \times X$ of their kernel pairs, which is precisely the relation refinement sequence of Staton [**?**, Section 5.1]. However, none of these works employed reflections for determinization-like constructions, that is exactly what allows us to minimize coalgebras in categories not equipped with a proper factorization structure, such as non-deterministic automata and conditional transition systems.

In future work we will study general conditions ensuring finite convergence: it is immediate to see that for any functor on $\mathbf{Set}$ with epi-mono factorizations, the sequence E_i of a finite coalgebra converges in a finite number of iterations. However, discovering general conditions encompassing all the examples of this paper seems to be non-trivial.

Preliminary research suggests that by integrating our approach with well-pointed coalgebras [4], we might obtain an explicit account of initial states. Indeed, given the reachable part of a pointed coalgebra for a set functor (which is defined through the canonical graph of Gumm [12]), the result of its minimization is a well-pointed coalgebra, i. e., a pointed coalgebra with no proper subcoalgebra and no proper quotient.

Acknowledgements. We would like to thank Ana Sokolova, Paolo Baldan and Walter Tholen for answering our questions and giving generous feedback.

References

1. J. Adámek. Free algebras and automata realizations in the language of categories. *Comment. Math. Univ. Carolin.*, 15:589–602, 1974.
2. J. Adámek, H. Herrlich, and G.E. Strecker. *Abstract and Concrete Categories – The Joy of Cats*. Wiley, 1990.
3. J. Adámek and V. Koubek. On the greatest fixed point of a set functor. *TCS*, 150:57–75, 1995.
4. J. Adámek, S. Milius, L. S. Moss, and L. Sousa. Well-pointed coalgebras. In *Proc. of FOSSACS '12*. Springer, 2012. LNCS, to appear.
5. M. Boreale. Weighted bisimulation in linear algebraic form. In *Proc. of CONCUR '09*, pages 163–177. Springer, 2009. LNCS 5710.
6. J. A. Brzozowski. Canonical regular expressions and minimal state graphs for definite events. In *Mathematical Theory of Automata*, volume 12(6), pages 529–561. Polytechnic Press, NY, 1962.
7. J.A. Brzozowski and H. Tamm. Theory of átomata. In *Proc. of DLT '11*, pages 105–116. Springer, 2011. LNCS 6795.
8. G.L. Ferrari, U. Montanari, and E.Tuosto. Coalgebraic minimization of HD-automata for the pi-calculus using polymorphic types. *TCS*, 331(2–3):325–365, 2005.
9. M. Fitting. Bisimulations and boolean vectors. In *Advances in Modal Logic*, volume 4, pages 1–29. World Scientific Publishing, 2002.
10. H.P. Gumm. From T-coalgebras to filter structures and transition systems. In *Proc. of CALCO '05*, pages 194–212. Springer, 2005. LNCS 3629.
11. H.P. Gumm. On minimal coalgebras. *Applied Categorical Structures*, 16:313–332, 2008.
12. H.P. Gumm. Copower functors. *TCS*, 410(12-13):1129–1142, 2009.
13. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace semantics via coinduction. *LMCS*, 3(4:11):1–36, 2007.
14. M. Hennessy and H. Lin. Symbolic bisimulations. *TCS*, 138(2):353–389, 1995.
15. C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145:107–152, 1998.
16. J.E. Hopcroft, R. Motwani, and J.D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Wesley, 2006.
17. B. Klin. Structural operational semantics for weighted transition systems. In *Semantics and Algebraic Specification*, pages 121–139. Springer, 2009. LNCS 5700.

18. A. Kurz. *Logics for Coalgebras and Applications to Computer Science*. PhD thesis, Ludwigs-Maximilians-Universität München, 2000.
19. S. Mac Lane. *Categories for the Working Mathematician*. Springer, 1971.
20. J. MacDonald and W. Tholen. Decomposition of morphisms into infinitely many factors. In *Proc. of Category Theory – Applications to Algebra, Logic and Topology*, number 962 in Lecture Notes in Mathematics, pages 175–189. Springer, 1981.
21. R. Milner and D. Sangiorgi. Barbed bisimulation. In *Proc. of ICALP '92*, pages 685–695. Springer, 1992. LNCS 623.
22. P.S. Mulry. Lifting theorems for Kleisli categories. In *Proc. of MFPS '93*, pages 304–319. Springer, 1993. LNCS 802.
23. J. Power and D. Turi. A coalgebraic foundation for linear time semantics. In *Proc. of CTCS '99*, volume 29 of *ENTCS*, pages 259–274, 1999.
24. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *TCS*, 249:3–80, 2000.
25. S. Staton. Relating coalgebraic notions of bisimulation. In *Proc. of CALCO '09*, pages 191–205. Springer, 2009. LNCS 5728.
26. S. Staton. Relating coalgebraic notions of bisimulation. *LMCS*, 7(1), 2011.
27. J. Worrell. On the final sequence of a finitary set functor. *TCS*, 338(1-3):184–199, 2005.

This paper is equipped with an extended appendix, where we provide some additional (but standard) definitions and back up some of the claims made in the main text of the paper, especially concerning the correspondence to related formalisms (átomata and linear weighted automata – LWA). Furthermore we present more minimization examples, for DAs, NDAs and LWAs, and give the proofs for all the results in the paper. The appendix is simply an addendum, which is not strictly necessary to understand the main part of the paper.

In more detail, we present in Appendix A additional definitions, especially the (standard) notion of monad. Then in Appendix B we discuss the relationship between our notion of canonicity with the átomata of [8]. In Appendix C we work out the example on conditional transition systems in more detail and show that the coalgebraic notion of behavioural equivalence coincides with the intuitive notion of the introduction. Then in Appendices D, E and F we will spell out in detail the minimization construction for three examples (deterministic automata, non-deterministic automata and linear weighted automata). In Appendix F we additionally show that the equivalence arising in our setting coincides with the one of [6]. Finally, we give the proofs in Appendix G.

A Additional Definitions (Monad)

We will now formally define the notion of monad (see also [2, 14, 20]).

Definition A.1 (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 morphisms* $\eta_X\colon X \to TX$ *for each set* X *satisfying suitable naturality conditions;*
- *a* multiplication *natural transformation* $\mu\colon T^2 \Rightarrow T$*, that is morphisms* $\mu_X\colon TTX \to X$ *for each set* X *again satisfying suitable naturality conditions.*

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

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

Example A.2. In the running examples of this paper we use the following monads:

(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 morphisms 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 morphisms 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 morphism 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 morphisms are $\eta_X \colon X \to \mathcal{P}(X)$ with $\eta_X(x) = \{x\}$. Furthermore the multiplication morphisms 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 $\mathcal{Z}$.

It is interesting to remark that the monad T of LWA can be defined also on a semiring $\mathbb{S}$ (in place of the field $\mathbb{F}$). The coalgebraic models that we have developed for $\mathbb{F}$, could be extended to $\mathbb{S}$ (but in some cases, the algorithm will not be guaranteed to terminate anymore). By considering more general structures, such as commutative monoids or semilattices, we could still define endofunctors on $\mathbf{Set}$ [11, 18], but the lack of multiplicative structures forbids to define (the multiplication of) a monad. Coalgebras for such functors on $\mathbf{Set}$ are (some kind of) weighted transition systems, but the associated notion of equivalence keeps into account the branching structure [11, 18], while for LWA we are interested in (weighted) language equivalence.

B Canonical NDA, Minimization and Átomata

Given a regular language L over an alphabet A, minimal deterministic automata can be thought of as *canonical* acceptors of the given language L. Does an analogous notion of canonicity exist for non-deterministic automata? Several works (starting in the seventies) have tried to answer this question and recently a new kind of canonical NDA has been introduced in [8]. In this appendix, we show that the coalgebraic notion of minimization instantiated to NDAs (discussed in Example 4.10) provides the same notion of canonicity as [8].

First, we report the notion of atoms and átomata from Section 3 of [8].

For a non-empty regular language L, let $L_1, L_2 \ldots, L_n$ be the quotients of L (recall that for any word $w \in A^*$, the quotient of L w.r.t. w is $w^{-1}L = \{u \in A^* \mid wu \in L\}$ and, moreover, each regular language has finitely many different quotients). An *atom* of L is a language of the form $\widehat{L_1} \cap \widehat{L_2} \cap \cdots \cap \widehat{L_n}$ (where $\widehat{L_i}$ is either L_i or its complement $\overline{L_i}$) such that (1) it is non-empty and (2) at least one of the $\widehat{L_i}$ is not complemented. It is easy to prove that there exists exactly one atom containing ε.

Definition B.1 (Átomata). *Let L be a regular language and $L_1, L_2, \dots, L_n$ be its atoms. The* átomaton *of L ($\mathcal{A}_L$) is the non-deterministic automaton having the atoms as states. The transition relation is defined (for all $a \in A$ and atoms L_i, L_j) as $L_i \xrightarrow{a} L_j$ iff $aL_j \subseteq L_i$ and the final state is the only atom containing ε.*

In order to describe the correspondence between átomata and our notion of minimization for NDA, it is convenient to first observe the following property of reflections of NDAs (described in Example 4.5).

Lemma B.2. *Let $(\mathcal{P}(X), \alpha')$ be the reflection of an NDA (X, α). For all $Y \in \mathcal{P}(X)$, the language recognized by Y is*

$$beh_{\mathcal{P}(X)}(Y) = \bigcap_{x' \in Y} beh_X(x') \cap \bigcap_{x'' \notin Y} \overline{beh_X(x'')}$$

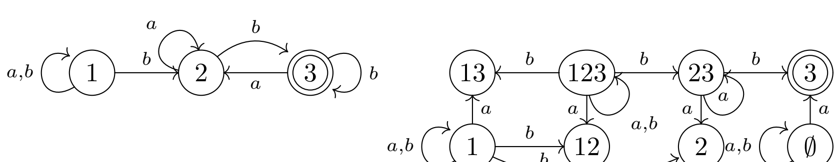
where $beh_X(x')$ is the language recognized by x' and $\overline{beh_X(x')}$ is its complement.

Proof. Recall that the unit $\eta_X \colon X \to \mathcal{P}(X)$ (of the reflection) relates x to all the sets $Y \in \mathcal{P}(X)$ containing x. Moreover observe that η_X is a coalgebra morphism $\eta_X \colon (X, \alpha) \to (\mathcal{P}(X), \alpha')$. Since there is a unique morphism into a final coalgebra, then $beh_{\mathcal{P}(X)} \circ \eta_X = beh_X$. This means that for all words $w \in A^*$ and states $x' \in X$:
(1) if $w \in beh_X(x')$, then exists $Y \in \mathcal{P}(X)$ such that $x' \in Y$ and $w \in beh_{\mathcal{P}(X)}(Y)$,
(2) if $w \notin beh_X(x')$, then for all $Y \in \mathcal{P}(X)$ such that $x' \in Y$, $w \notin beh_{\mathcal{P}(X)}(Y)$.
Then observe that since $beh_{\mathcal{P}(X)}$ is a morphism in $\mathbf{Set^{op}}$, each word $w \in A^*$ is recognized by exactly one state in $\mathcal{P}(X)$. More formally, for all $w \in A^*$,
(3) there exists $Y \in \mathcal{P}(X)$ such that $w \in beh_{\mathcal{P}(X)}(Y)$;
(4) for all $Y, Z \in \mathcal{P}(X)$, if $w \in beh_{\mathcal{P}(X)}(Y)$ and $w \in beh_{\mathcal{P}(X)}(Z)$ then $Y = Z$.
Suppose that $w \in \bigcap_{x' \in Y} beh_X(x') \cap \bigcap_{x'' \notin Y} \overline{beh_X(x'')}$. By (3) there exists a $Z \in \mathcal{P}(X)$ such that $w \in beh_{\mathcal{P}(X)}(Z)$. Since $w \in \bigcap_{x' \in Y} beh_X(x')$, by (1), we have that $Y \subseteq Z$. Since $w \in \bigcap_{x'' \notin Y} \overline{beh_X(x'')}$, by (2), we have that $Z \subseteq Y$. That is $w \in beh_{\mathcal{P}(X)}(Y)$.

If $w \in beh_{\mathcal{P}(X)}(Y)$ then, by (2), $w \in \bigcap_{x' \in Y} beh_X(x')$.

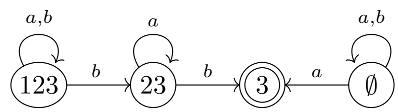
If $w \in beh_X(x'')$ for some $x'' \notin Y$, then by (1), there exists a Z such that $x'' \in Z$ and $w \in beh_{\mathcal{P}(X)}(Z)$. Since $Z \neq Y$, by (4), we have that $w \notin beh_{\mathcal{P}(X)}(Y)$. This means that, if $w \in beh_{\mathcal{P}(X)}(Y)$ then $w \in \bigcap_{x'' \notin Y} \overline{beh_X(x'')}$. □

For an example, consider the NDA (X, α) and its reflection $(\mathcal{P}(X), \alpha')$, depicted below on the left and right, respectively.



The languages recognized by the states of $(\mathcal{P}(X), \alpha')$ can be computed from the languages recognized by the states of (X, α). For instance, $beh_{\mathcal{P}(X)}(3) = \{\varepsilon\} = beh_X(3) \cap \overline{beh_X(1)} \cap \overline{beh_X(2)}$ and $beh_{\mathcal{P}(X)}(23) = a^*b = beh_X(2) \cap beh_X(3) \cap \overline{beh_X(1)}$.

As we have shown in Example 4.10, the minimization of an NDA (X, α) can be obtained from the reflection $(\mathcal{P}(X), \alpha')$, just by removing those states that accept the empty language (i.e., those states that cannot reach the final state). For instance, the minimization of the NDA above is depicted below.



It is easy to see that in a minimization all the states accept different languages (actually disjoint languages) and therefore we can safely identify states of a minimization with the language that they recognize. Intuitively, the states of a minimization are all the atoms plus the state $\emptyset$ that recognizes the language $beh_{\mathcal{P}(X)}(\emptyset) = \bigcap_{x'' \in X} \overline{beh_X(x'')}$ (that is the complement of all the languages recognized by all the states of the original NDA). Note that this language is *not* an atom by condition (2) of the definition of atom.

Hereafter, we formalize the above intuition. Recall that we model NDA with alphabet A as relation $\alpha\colon X \to A \times X + 1$ where (1) X is the set of states, (2) $1 = \{\bullet\}$ and $\bullet \in \alpha(x)$ iff $x \in X$ is a final state, and (3) $(a, x_j) \in \alpha(x_i)$ iff $x_i \xrightarrow{a} x_j$.

Proposition B.3. *Let (X, α) be an NDA with states $x_1, x_2, \ldots, x_n$ accepting the languages $L_1, L_2, \ldots, L_n$, respectively. Let $Z = \{z_1, \ldots z_m\}$ be the set of* non-empty *languages of the form $\widehat{L_1} \cap \widehat{L_2} \cap \cdots \cap \widehat{L_n}$ where $\widehat{L_i}$ is either L_i or $\overline{L_i}$. Let $\gamma\colon Z \to A \times Z + 1$ be the relation defined as (1) $\bullet \in \gamma(z_i)$ iff $\varepsilon \in z_i$ and (2) $(a, z_j) \in \delta(z_i)$ iff $az_j \subseteq z_i$.*

Then (Z, γ) is the minimization of (X, α).

Proof. According to Example 4.10, the minimization of an NDA (X, α) can be built by the reflection $(\mathcal{P}(X), \alpha')$ by simply removing those states $Y \in \mathcal{P}(X)$ that accept the empty language. Let us call (Z', δ') the resulting automaton, i.e., $Z' \subseteq \mathcal{P}(X)$ and δ' is equal to α' restricted to Z'.

By Lemma B.2, each $Y \in Z'$ recognizes a non-empty language of the form $\widehat{L_1} \cap \widehat{L_2} \cap \cdots \cap \widehat{L_n}$ and since all these languages are distinct, we can identify each state $Y \in Z'$ with the language that it recognizes. More precisely, for each $Y \in Z' \subseteq \mathcal{P}(X)$ there is a corresponding (non-empty) language in Z defined as

$$beh_{Z'}(Y) = \bigcap_{x_i \in Y} L_i \cap \bigcap_{x_j \notin Y} \overline{L_j}$$

Analogously, each $z \in Z$ is a language such that $z = \widehat{L_1} \cap \widehat{L_2} \cap \cdots \cap \widehat{L_n}$ that corresponds to a set of states in $Z' \subseteq \mathcal{P}(X)$ defined as

$$Y_z = \{x_i \mid \widehat{L_i} = L_i \text{ in } z\}$$

It is easy to see that this correspondence defines an isomorphism between Z and Z', that is

$$beh_{Z'}(Y_z) = z \qquad Y_{beh_{Z'}(Y)} = Y$$

Let us now check that also the relations δ and δ' are isomorphic. Recall that δ' is just the restriction (on Z') of α' defined as in Example 4.5, that is

(1) $\bullet \in \delta'(\{x \in X \mid \bullet \in \alpha(x)\})$,
(2) $\langle a, Y'\rangle \in \delta'(\{x \in X \mid \exists x' \in Y' \text{ s.t. } \langle a, x'\rangle \in \alpha(x)\})$.

Let $z \in Z$ be a language of the shape $\widehat{L_1} \cap \widehat{L_2} \cap \cdots \cap \widehat{L_n}$.

Then, $\bullet \in \delta'(Y_z)$ iff $Y_z = \{x \in X \mid \bullet \in \alpha(x)\}$, that is iff Y_z is the set of all and only the final states. By definition of Y_z this means that all those x_i with $\widehat{L_i} = L_i$ are final, while all those x_j with $\widehat{L_j} = \overline{L_j}$ are *not* final. Thus $\varepsilon \in L_i$ (for $\widehat{L_i} = L_i$) and $\varepsilon \notin L_j$ (for $\widehat{L_j} = \overline{L_j}$) which means $\varepsilon \in z$ (i.e., $\bullet \in \delta(z)$).

Finally, we should prove that $\langle a, Y'\rangle \in \delta'(Y_z)$ iff $\langle a, beh_{Z'}(Y')\rangle \in \delta(z)$.

Observe that, by hypothesis, $\langle a, beh_{Z'}(Y')\rangle \in \delta(z)$ iff $a(beh_{Z'}(Y')) \subseteq z$. Therefore, we show that if $\langle a, Y'\rangle \in \delta'(Y_z)$ then $a(beh_{Z'}(Y')) \subseteq z$. The other direction can be proved analogously. Suppose that $\langle a, Y'\rangle \in \delta'(Y_z)$, then Y_z is the set of all states x_i such that $x_i \xrightarrow{a} x' \in Y'$. Now, suppose that $w \in beh_{Z'}(Y')$ then, by Lemma B.2, for all $x' \in Y'$, $w \in beh_X(x')$ and for all $x'' \notin Y'$, $w \notin beh_X(x'')$. Therefore, $aw \in beh_X(x_i)$ for all $x_i \in Y_z$. Analogously we can prove that $aw \notin beh_X(x_j)$ for all $x_j \notin Y_z$: observe that for all $x_j \notin Y_z$, $x_j \xrightarrow{a} x'_j$ iff $x'_j \notin Y'$ (otherwise $x_j \in Y_z$) and thus $w \notin beh_X(x'_j)$. By Lemma B.2, this means that $aw \in beh_{Z'}(Y_z) = z$. ☐

Now what is the exact relationship between $\mathcal{A}_L$ (that is the átomaton of L) and our minimization?

Let $\mathcal{D}_L$ be the minimal DA corresponding to L: its states are the quotients $L_1, \ldots, L_n$ of L and each state L_i recognizes itself. Let $\mathcal{N}_L$ be the (NDA) minimization of $\mathcal{D}_L$. By virtue of Proposition B.3, the states of $\mathcal{N}_L$ are exactly the atoms of L (with the extra state $\bigcap_{i \in 1 \ldots n} \overline{L_i}$ corresponding to $\emptyset$ in the construction of Example 4.10). Since transition relation and final states are defined in the same way, $\mathcal{N}_L$ is $\mathcal{A}_L$ (plus the extra state).

C Conditional Transition Systems, in detail

In the introduction, we have given an intuitive description of conditional transition systems (CTSs). In this appendix, we first give a formal definition of CTSs and their behavioural equivalence and then we show all the details of their coalgebraic modelling.

Given a set of conditions A, a *conditional transition system* consists of a set of states X and a transition function $\alpha \colon X \to \mathcal{P}_c(X)^A$. Intuitively $x' \in \alpha(x)(a)$ (written $x \xrightarrow{a} x'$) means that x can make a transition into x' if the condition $a \in A$ holds. Two states x and y are behaviourally equivalent if they are bisimilar under all the possible conditions $a \in A$. This can be formalized by defining the *instantiation* of a CTS (X, α) as the "unlabeled" transition function

$$\alpha' \colon A \times X \to \mathcal{P}_c(A \times X)$$

such that $\alpha'(a, x) = \{(a, x') \mid x \xrightarrow{a} x'\}$. We write $(a, x) \to (a, x')$ to mean that $(a, x') \in \alpha'(a, x)$.

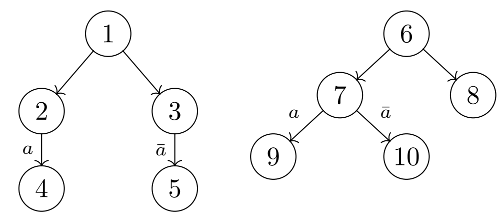
Definition C.1. *A relation $R \subseteq X \times X$ is a* bisimulation under the condition $a \in A$ *if whenever $(x, y) \in R$:*

- *if* $(a,x) \to (a,x')$ *then there exists* $y' \in X$ *such that* $(a,y) \to (a,y')$ *and* $(x',y') \in R$*;*
- *if* $(a,y) \to (a,y')$ *then there exists* $x' \in X$ *such that* $(a,x) \to (a,x')$ *and* $(x',y') \in R$*.*

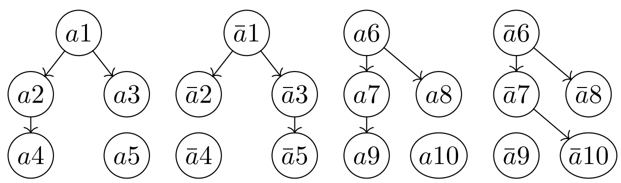
We say that x *and* y *are* bisimilar under the condition a *(written* $x \sim_a y$*) if there exists a bisimulation* R *under* a *such that* $(x,y) \in R$*. We say that* x *and* y *are* instantiated equivalent *(written* $x \approx_i y$*) iff for all conditions* $a \in A$*,* $x \sim_a y$*.*

It is interesting to note that this kind of definition is analogous to the one of *barbed congruence* of [22]: first, one defines bisimilarity and then take the intersection with respect to all possible contexts (that in the case of CTS are all the possible conditions).

As an example consider the following CTS (X, α) for $A = \{a, \bar{a}\}$ (unlabeled transition can be thought of two transitions labeled with a and $\bar{a}$).



The instantiation of the above is the following $\alpha' \colon A \times X \to \mathcal{P}_c(A \times X)$.



Note that $1 \sim_a 6$ and $1 \sim_{\bar{a}} 6$ and thus $1 \approx_i 6$. Moreover $2 \sim_a z$, but $2 \not\sim_{\bar{a}} 7$.

We can give also an alternative "more efficient" characterization of $\approx_i$ that avoids to instantiate the CTS w.r.t. all the conditions. We make use of *conditional relations*, that are functions $R \colon X \times X \to \mathcal{P}(A)$. Intuitively R assigns to each pair of states (x,y) the set of conditions under which x and y are equivalent. By looking at ordinary relations as functions $R \colon X \times X \to 2$, it is easy to see that they are a special case of conditional relations where $|A| = 1$. Moreover, we can generalize the notion of equivalence as follows. A *conditional equivalence* is a conditional relation $R \colon X \times X \to \mathcal{P}(A)$ such that for all $x,y,z \in X$ and $a \in A$: (1) $R(x,x) = A$ (reflexivity), (2) $R(x,y) = R(y,x)$ (symmetry) and (3) if $a \in R(x,y)$ and $a \in R(y,z)$, then $a \in R(x,z)$ (transitivity).

Definition C.2. *A conditional equivalence* $R \colon X \times X \to \mathcal{P}(A)$ *is a* conditional bisimulation *if for all* $x,y \in X$ *and* $a \in R(x,y)$*:*

- *if* $x \xrightarrow{a} x'$ *then there exists* $y' \in X$ *such that* $y \xrightarrow{a} y'$ *and* $a \in R(x',y')$*;*
- *if* $y \xrightarrow{a} y'$ *then there exists* $x' \in X$ *such that* $x \xrightarrow{a} x'$ *and* $a \in R(x',y')$*.*

We say that x *and* y *are* conditionally bisimilar *(written* $x \approx_c y$*) if there exists a conditional bisimulation* R *such that* $R(x,y) = A$*.*

For instance, the following table (where "$\dots$" is an abbreviation for for $4, 5, 8, 9, 10$) shows a conditional bisimulation for the CTS α above:

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

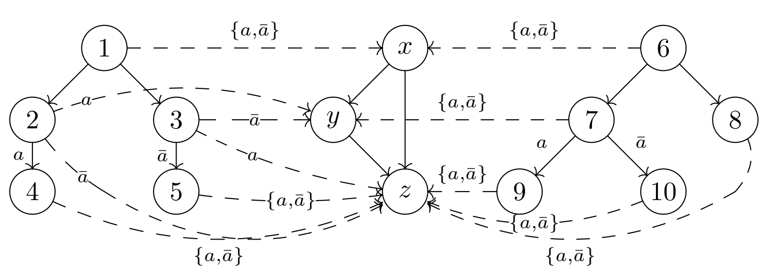
Since $R(1, 6) = \{a, \bar{a}\} = A$, then $1 \approx_c 6$.

It is interesting to observe that, differently from Definition C.1, Definition C.2 is really coinductive and indeed, as we will show later, conditional bisimulations are in close correspondence with coalgebra morphisms. Definition C.1 instead consists in first instantianting the CTSs and then considering the ordinary bisimilarity. We will show that $\approx_c$ coincides with $\approx_i$ and thus, in order to check them, we could either consider conditional bisimulations or first instantiate and then compute ordinary bisimilarity. Intuitively, these correspond to the Constructions (iii) and (iv) of Theorem 4.9. Hereafter, we will prove that $\approx_c = \approx_i$ by showing that the behvaioural equivalence ($\approx$) induced by our coalgebraic construction coincides with both.

The input monad. For a given set A of conditions (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 $Tf = f^A\colon X^A \to Y^A$ with $f^A(\varphi)(a) = f(\varphi(a))$ for all $\varphi\colon A \to X$ 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 $\mu_X\colon (X^A)^A \to X^A$ is defined as $\mu_X(\varphi)(a) = \varphi(a)(a)$ for all functions $\varphi\colon A \to X^A$.

The Kleisli category for the input monad. The Kleisli category $\mathcal{K}\ell(T)$ over the input monad T has sets as objects and morphisms $f\colon X \to Y$ are functions $f\colon X \to Y^A$. The identity is the unit η described above and the composition of morphisms is defined as follows: for all morphisms $f\colon X \to Y$ and $g\colon Y \to Z$ (i.e., functions $f\colon X \to Y^A$ and $g\colon Y \to Z^A$), $g \circ f\colon X \to Z$ is the function of the shape $X \to Z^A$ such that $g \circ f(x)(a) = g(f(x)(a))(a)$, for all $x \in X$ and $a \in A$. The following dashed arrow describes a morphism in $\mathcal{K}\ell(T)$: the state 2 is mapped to y if condition a holds and to z if $\bar{a}$ holds.



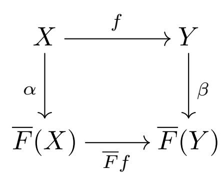
$\overline{F}$*-coalgebras.* Consider the endofunctor $\overline{F}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ defined as follows: for all sets X, $\overline{F}X = \mathcal{P}_c(X)$ and for all arrows $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, $\overline{F}f\colon \mathcal{P}_c(X) \to \mathcal{P}_c(Y)$ is

$$\overline{F}f(X')(a) = \{f(x)(a) \mid x \in X'\}$$

for all $X' \subseteq X$, $a \in A$. It is easy to see that each $\overline{F}$-coalgebra $\alpha\colon X \to \overline{F}X$ corresponds to a CTS and viceversa each CTS corresponds to an $\overline{F}$-coalgebra.

Proposition C.3. *$\overline{F}$-coalgebras are in one-to one correspondence with CTSs.*

A homomorphism between $\overline{F}$-coalgebras (X, α) and (Y, β) is a morphisms $f\colon X \to Y$ in $\mathcal{K}\ell(T)$ such that the following diagram commutes in $\mathcal{K}\ell(T)$.



By spelling out the definition of $\overline{F}f$ and $\circ$, we have that a function $f\colon X \to Y^A$ is a $\overline{F}$-homomorphism iff

$$\beta(f(x)(a))(a) = \{f(x')(a) \mid x' \in \alpha(x)(a)\}.$$

For instance, the dashed arrow above is an $\overline{F}$-homomorphism.

Given a morphism $f\colon X \to Y$ in $\mathcal{K}\ell(T)$, we can define the conditional equivalence $R_f\colon X \times X \to \mathcal{P}(A)$ as $R(x, y) = \{a \mid f(x)(a) = f(y)(a)\}$.

Proposition C.4. *If f is a $\overline{F}$-homomorphism, then R_f is a conditional bisimulation.*

Proof. We check that R_f satisfies the conditions of Definition C.2.

For all $x, y \in X$ and $a \in R_f(x, y)$ (i.e., such that $f(x)(a) = f(y)(a)$), we have that $\{f(x')(a) \mid x' \in \alpha(x)(a)\} = \beta(f(x)(a))(a) = \beta(f(y)(a))(a) = \{f(y')(a) \mid y' \in \alpha(y)(a)\}$, because f is a $\overline{F}$-homomorphism.

Since $\{f(x')(a) \mid x' \in \alpha(x)(a)\} \subseteq \{f(y')(a) \mid y' \in \alpha(y)(a)\}$, then if $x \xrightarrow{a} x'$ then exists $y' \in X$ such that $y \xrightarrow{a} y'$ and $f(x')(a) = f(y')(a)$, that means $a \in R_f(x', y')$.

Since $\{f(y')(a) \mid y' \in \alpha(y)(a)\} \subseteq \{f(x')(a) \mid x' \in \alpha(x)(a)\}$, then if $y \xrightarrow{a} y'$ then exists $x' \in X$ such that $x \xrightarrow{a} x'$ and $f(x')(a) = f(y')(a)$, i.e., $a \in R_f(x', y')$. $\square$

Viceversa, given a conditional bisimulation R, there exists a $\overline{F}$-homomorphism ε_R. Given $x \in X$ and $a \in A$, let $[x]_R^a$ be the set $\{y \mid a \in R(x, y)\}$. Let X/R be the set $\{[x]_R^a \mid x \in X, a \in A\}$ and $\varepsilon_R\colon X \to (X/R)^A$ be the function defined as $\varepsilon_R(x)(a) = [x]_R^a$ for all $x \in X$ and $a \in A$. Finally, consider $\alpha_R\colon (X/R) \to (X/R)^A$ defined as $\alpha_R([x]_R^a)(b) = \{[x']_R^a \mid x \xrightarrow{a} x'\}$ if $a = b$ and $\emptyset$ otherwise.

Proposition C.5. *If R is a conditional bisimulation, then $\varepsilon_R\colon (X, \alpha) \to (X/R, \alpha_R)$ is a $\overline{F}$-homomorphism.*

Proof. We first prove that α_R is well defined, i.e., that for all $y \in [x]^a_R$, $\alpha_R([x]^a_R)(b) = \alpha_R([y]^a_R)(b)$. If $b \neq a$, then trivially both are equal to $\emptyset$. If $b = a$, then $\alpha_R([y]^a_R)(a) = \{[y']^a_R \mid y \xrightarrow{a} y'\}$. Since $a \in R(x,y)$ and R is a conditional bisimulation, then $\{[x']^a_R \mid x \xrightarrow{a} x'\} = \{[y']^a_R \mid y \xrightarrow{a} y'\}$. The fact that ε_R is a $\overline{F}$-homomorphism follows immediately from the fact that $\alpha_R([x]^a_R)(a) = \{[x']^a_R \mid x \xrightarrow{a} x'\}$. □

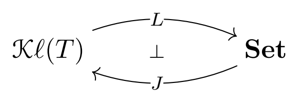
With the two above propositions, it is easy to see that the coalgebraic definition of behavioural equivalence ($\approx$) coincides with conditional bisimilarity ($\approx_c$).

Theorem C.6. $\approx \, = \, \approx_c$

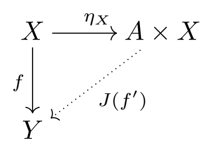
Proof. If $x \approx y$, then there exists an $\overline{F}$-coalgebra (Z,γ) and an $\overline{F}$-homomorphism $f\colon (X,\alpha) \to (Z,\gamma)$ such that $f(x)\colon A \to Z$ and $f(y)\colon A \to Z$ are the same function. Therefore, by definition, $R_f(x,y) = A$ and, by Proposition C.4, R_f is a conditional bisimulation. This means $x \approx_c y$.

If $x \approx_c y$, then there exists a conditional bisimulation R such that $R(x,y) = A$. By Proposition C.5, $\varepsilon_R\colon (X,\alpha) \to (X/R, \alpha_R)$ is a $\overline{F}$-homomorphism. Since $R(x,y) = A$, $\varepsilon_R(x)\colon A \to X/R$ and $\varepsilon_R(y)\colon A \to X/R$ are the same function, i.e., $x \approx y$. □

Reflection. As explained after Definition 2.3, $\mathbf{Set}$ can be regarded as a subcategory of $\mathcal{K}\ell(T)$. For this aim, consider the embedding functor $J\colon \mathbf{Set} \to \mathcal{K}\ell(T)$ mapping all sets X in $J(X) = X$ and all functions $f\colon X \to Y$ in $J(f)\colon X \to Y = \mu_Y \circ f$ (that is the function $J(f)\colon X \to Y^A$ such that $J(f)(x)(a) = f(x)$). Intuitively a morphism $f\colon X \to Y$ in Kleisli is also a morphism in $\mathbf{Set}$ iff it ignores the inputs in A.



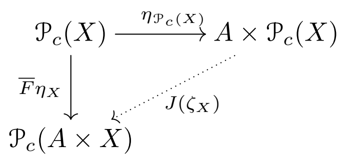
The embedding J has a left adjoint $L\colon \mathcal{K}\ell(T) \to \mathbf{Set}$ mapping all sets X in $L(X) = A \times X$ and all arrows $f\colon X \to Y$ in $\mathcal{K}\ell(T)$ in the arrow $Lf\colon A \times X \to A \times Y$ in $\mathbf{Set}$ with $Lf(\langle a, x\rangle) = \langle a, f(x)(a)\rangle$. The unit of the adjunction is the Kleisli arrow $\eta_X\colon X \to A \times X$ with $\eta_X(x)(a) = \langle a, x\rangle$. Note that this has nothing to do with the unit of the input monad described above. For all Kleisli arrows $f\colon X \to Y$, the unique arrow $f'{:}A \times X \to Y$ (in $\mathbf{Set}$) such that the following diagram commutes (in $\mathcal{K}\ell(T)$) is $f'(\langle a, x\rangle) = f(x)(a)$.



Reflecting Coalgebras. Recall the endofunctor $\overline{F}\colon \mathcal{K}\ell(T) \to \mathcal{K}\ell(T)$ introduced above. It is easy to see that if a Kleisli morphism f is a morphism in $\mathbf{Set}$ (i.e., it ignores the inputs in A) then also $\overline{F}f$ is a morphism in $\mathbf{Set}$. This means that $\overline{F}$ preserves the reflective subcategory $\mathbf{Set}$ and thus we can apply the construction shown in Definition 4.3 in order to reflect $\overline{F}$-coalgebras. Given a Kleisli arrow $\alpha\colon X \to \mathcal{P}_c(X)$, we can get the function

$$\alpha' = \; A \times X \xrightarrow{L\alpha} A \times \mathcal{P}_c(X) \xrightarrow{\zeta_X} \mathcal{P}_c(A \times X)$$

where ζ_X is the unique function such that the following diagram commutes in $\mathcal{K}\ell(T)$,



that is, for all $X' \subseteq X$ and $a \in A$

$$\zeta_X(\langle a, X' \rangle) = \overline{F}\eta_X(X')(a) = \{\eta_X(x)(a) \mid x \in X'\} = \{\langle a, x \rangle \mid x \in X'\}.$$

Since $L\alpha(\langle a, x \rangle) = \langle a, \alpha(x)(a) \rangle$ for all $a \in A$ and $x \in X$, then

$$\alpha'(\langle a, x \rangle) = \{\langle a, x' \rangle \mid x' \in \alpha(x)(a)\}$$

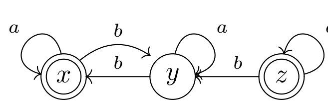
that is α' is exactly the *instantiation* defined at the beginning of this section.

Theorem C.7. $\approx = \approx_i$

Proof. Since $\mathbf{Set}$ is a reflective subcategory of $\mathcal{K}\ell(T)$, then a final colagebra of $\mathcal{P}_c$ in $\mathbf{Set}$ is a final coalgebra for $\overline{F}$ in $\mathcal{K}\ell(T)$. We can build the unique morphism from a generic $\overline{F}$-coalgebra (X, α) to a final coalgebra as follows: first we take the coalgebra morphism $\eta_X : (X, \alpha) \to (A \times X, \alpha')$ (where α' is the reflection of α) and then we take the unique morphism $beh_{A\times X}$ from α' to a final coalgebra. Therefore, two states $x, y \in X$ are in $\approx$ iff that is for all $a \in A$, $beh_{A\times X}(\eta_X(x)(a))(a) = beh_{A\times X}(\eta_X(y)(a))(a)$. By definition of η_X, this just means that $beh_{A\times X}(a, x)(a) = beh_{A\times X}(a, y)(a)$ for all $a \in A$. We can think of $beh_{A\times X}$ as a morphism in $\mathbf{Set}$ (that ignores the second input parameter a) and thus the above condition becomes $beh_{A\times X}(a, x) = beh_{A\times X}(a, y)$ for all a. By standard results in coalgebras, we know that the unique morphism into a final coalgebra for the endofunctor $\mathcal{P}_c$ equates all and only the "bisimilar" states (where by "bisimilar" we mean in the standard sense). Thus, $x \approx y$ iff (a, x) is "bisimilar" to (a, y) for all $a \in A$. $\square$

D Additional Examples: Deterministic Automata

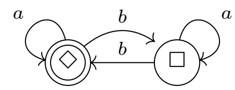
Example D.1. (DA) We apply the construction of Theorem 3.10 to the DA (X, α) of Example 2.2, depicted below.



At the beginning, we take $d_0 : X \to 1$ as the unique function into the final object $1 = \{\bullet\}$ (it maps x, y, z to the singleton $\bullet$). By factoring d_0, we obtain $e_0 : X \twoheadrightarrow 1$ and $n_0 : 1 \rightarrowtail 1$ (both uniquely defined). The surjection e_0 corresponds to the partition $\{x, y, z\}$ (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 $n_i \colon \{\Diamond, \Box\} \rightarrowtail 2$. 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 $n_2 \colon \{\Diamond, \Box\} \rightarrowtail 2 \times \{\Diamond, \Box\}^A$. The coalgebra $(\{\Diamond,\ \Box\}, n_2)$ is depicted below.



Note that n_2 is exactly the function γ of Example 2.2 and thus $(\{\Diamond,\ \Box\}, n_2)$ is the DA (Z, γ) which, as shown in Example 3.6, is the minimization of (X, α).

E Additional Examples: Non-deterministic Automata

Example E.1. (NDA) We are considering the non-deterministic automaton of Example 2.4. We are working in the category $\mathcal{K}\ell(\mathcal{P}) = \mathbf{Rel}$ and $\overline{F}X = A \times X + 1$. 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 $A \times X + 1$. Composition of morphisms 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.

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

The final object of $\mathbf{Rel}$ is the empty set $\emptyset$ and hence $d_0 = e_0$ is a 0×3-matrix. In the next step we obtain $d_1 \colon X \to 1$ $(= A \times \emptyset + 1)$.

$$d_1 = \overline{F}e_0 \circ \alpha \ = \begin{array}{c} \\ \bullet \end{array} \begin{array}{c} 1\,2\,3 \\ \begin{pmatrix} 0&0&1 \end{pmatrix} \end{array}$$

We compute the pseudo-factorization (see Example 4.7) and obtain $e_1 = d_1$. In the next step we get $d_2 \colon X \to A \times 1 + 1$.

$$d_2 = \overline{F}e_1 \circ \alpha \ = \begin{array}{c} \\ a\bullet \\ b\bullet \\ \bullet \end{array} \begin{array}{c} a1\ b1\ a2\ b2\ a3\ b3\ \bullet \\ \begin{pmatrix} 0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1 \end{pmatrix} \end{array} \circ \alpha \ = \begin{array}{c} \\ a\bullet \\ b\bullet \\ \bullet \end{array} \begin{array}{c} 1\,2\,3 \\ \begin{pmatrix} 0&0&0 \\ 0&1&1 \\ 0&0&1 \end{pmatrix} \end{array}$$

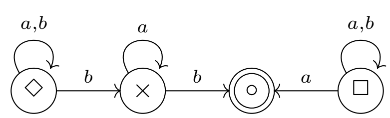
Again, via the pseudo-factorization we obtain $e_2 = d_2$. By iterating again, we obtain $d_3 \colon X \to A \times (A \times 1 + 1) + 1$.

$$d_3 = \overline{F}e_2 \circ \alpha \; = \begin{array}{r}\\ aa\bullet \\ ba\bullet \\ ab\bullet \\ bb\bullet \\ a\bullet \\ b\bullet \\ \bullet\end{array}\!\!\begin{array}{c} 1\,2\,3 \\ \begin{pmatrix} 0&0&0\\0&0&0\\0&1&1\\1&1&1\\0&0&0\\0&1&1\\0&0&1\end{pmatrix}\end{array} = \begin{array}{r}\\ aa\bullet \\ ba\bullet \\ ab\bullet \\ bb\bullet \\ a\bullet \\ b\bullet \\ \bullet\end{array}\!\!\begin{array}{c} \Box\times\Diamond\circ \\ \begin{pmatrix} 1&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&1&0\\1&0&0&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\end{array} \circ \begin{array}{c} 1\,2\,3 \\ \begin{pmatrix} 0&0&0\\0&1&1\\1&1&1\\0&0&1\end{pmatrix}\end{array}\!\!\begin{array}{l}\\ \Box \\ \times \\ \Diamond \\ \circ\end{array} = n_3 \circ e_3$$

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 n_4, e_4 with $e_4 = e_3$, i.e., we have reached the fixed-point.

$$d_4 = \overline{F}e_3 \circ \alpha \; = \begin{array}{r}\\ a\Box \\ b\Box \\ a\times \\ b\times \\ a\Diamond \\ b\Diamond \\ a\circ \\ b\circ \\ \bullet\end{array}\!\!\begin{array}{c} 1\,2\,3 \\ \begin{pmatrix} 0&0&0\\0&0&0\\0&1&1\\1&1&1\\1&1&1\\1&1&1\\0&0&0\\0&1&1\\0&0&1\end{pmatrix}\end{array} = \begin{array}{r}\\ a\Box \\ b\Box \\ a\times \\ b\times \\ a\Diamond \\ b\Diamond \\ a\circ \\ b\circ \\ \bullet\end{array}\!\!\begin{array}{c} \Box\times\Diamond\circ \\ \begin{pmatrix} 1&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&1&0\\0&0&1&0\\1&0&0&0\\0&1&0&0\\0&0&0&1\end{pmatrix}\end{array} \circ \begin{array}{c} 1\,2\,3 \\ \begin{pmatrix} 0&0&0\\0&1&1\\1&1&1\\0&0&1\end{pmatrix}\end{array}\!\!\begin{array}{l}\\ \Box \\ \times \\ \Diamond \\ \circ\end{array} = n_4 \circ e_4$$

The morphism $n_4 = \gamma$ gives us the following minimization (compare with the automaton in Example 4.10).

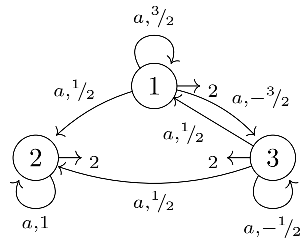


Intuitively, we are performing a breadth-first backwards search, starting from the set of final states.

F Additional Examples: Linear Weighted Automata

Example F.1. (LWA) We come back to Example 3.11 and consider the following linear weighted automaton from [6] 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 $d_0 = e_0$ is a 0×3-matrix. In the next step, we obtain:

$$d_1 = \overline{F}e_0 \circ \alpha = \begin{array}{c} \\ \bullet \end{array} \begin{array}{c} 1\ 2\ 3 \\ \begin{pmatrix} 2 & 2 & 2 \end{pmatrix} \end{array}$$

The morphism d_1 is a matrix of full row rank (i.e., an element of $\mathcal{E}$) and hence $e_1 = d_1$. In the next step we obtain:

$$d_2 = \overline{F}e_1 \circ \alpha = \begin{array}{c} \\ \circ \\ \bullet \end{array} \begin{array}{c} 1\ 2\ 3 \\ \begin{pmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \end{array}$$

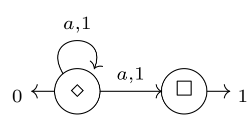
Note that d_2 is an element of $\mathcal{E}$, since its row vectors are linearly independent and hence $e_2 = d_2$. In the next step we obtain:

$$d_3 = \overline{F}e_2 \circ \alpha = \begin{array}{c} \\ \diamond \\ \square \\ \bullet \end{array} \begin{array}{c} 1\ 2\ 3 \\ \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \end{array}$$

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

$$d_3 = \begin{array}{c} \\ \diamond \\ \square \\ \bullet \end{array} \begin{array}{c} \diamond\ \square \\ \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \circ \begin{array}{c} 1\ 2\ 3 \\ \begin{pmatrix} 1 & 2 & 1 \\ 2 & 2 & 2 \end{pmatrix} \end{array} \begin{array}{c} \\ \diamond \\ \square \end{array} = n_3 \circ e_3$$

Since $e_3 = e_2$, we have reached a fixed-point and set $\gamma = n_3$. The corresponding transition system looks as follows:



This linear weighted automaton is equivalent to the one obtained in [6].

Comparison to Boreale's Linear Weighted Automata

We will compare the setting of Example 3.11 with the linear weighted automata of Boreale [6], where we use the reals as field.

Definition F.2 (Weighted Automaton in Linear Form [6]). *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 3.11.

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

$$\alpha(x)(\langle a, y\rangle) = T_a(\eta_X(x))(y)$$
$$\alpha(x)(\bullet) = \varphi(\eta_X(x))(\bullet)$$

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 F.3 (Weighted L-Bisimulation [6]). *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 ($\mathbf{u} \sin_L \mathbf{v}$) 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 F.3 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 F.3 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'.

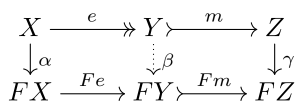
It is straightforward to show that two vectors are L-bisimilar iff their images in the minimization coincide. Finally, note that Boreale's algorithm computes the orthogonal complement $U^\perp$ rather than U itself, similar to our algorithm.

G Proofs

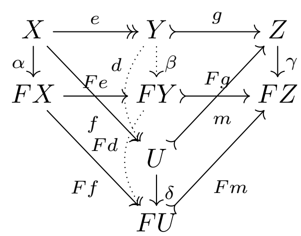
Lemma G.1. *Assume that the functor F preserves $\mathcal{M}$-morphisms. Then $(\mathcal{E}, \mathcal{M})$ is a factorization structure for the category of F-coalgebras, whenever this holds for the underlying category* **C**.

Proof. We check that the conditions of Definition 3.1 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 $\mathcal{M}$-morphisms $Fm \in \mathcal{M}$ and hence the coalgebra β can be obtained as the unique diagonal morphism.



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



The morphism d is obtained as the unique diagonal morphism 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. □

Lemma G.2. *Let $h\colon (X,\alpha) \to (Y,\beta)$ be a coalgebra homomorphism. Then for the canonical cones $\alpha_i\colon X \to W_i$ and $\beta_i\colon Y \to W_i$ we have*

$$\alpha_i = (\ X \xrightarrow{h} Y \xrightarrow{\beta_i} W_i\ ) \qquad \textit{for every } i \in \mathsf{Ord}.$$

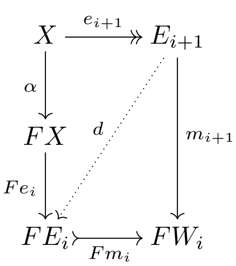
Proof. Easy by transfinite induction on i. □

Remark G.3. Observe that, by transfinite induction, each W_i is an F-algebra via $w_{i+1,i}\colon FW_i = W_{i+1} \to W_i$. Moreover it is not difficult to prove that for every F-coalgebra (X, α) the morphisms $\alpha_i\colon A \to W_i$ are the unique coalgebra-to-algebra homomorphisms, i. e., for every ordinal i, α_i is unique such that $\alpha_i = w_{i+1,i} \circ F\alpha_i \circ \alpha$.

Theorem 3.8.
For every F-coalgebra (X, α), its minimization is E_i, for some $i \in \mathsf{Ord}$.

Remark G.4. More explicitly, there exists an ordinal number i such that E_i carries the structure of an F-coalgebra such that $e_i\colon X \twoheadrightarrow E_i$ is the greatest quotient coalgebra, i. e., for every coalgebra homomorphism $e'\colon X \twoheadrightarrow Y$ in $\mathcal{E}$ there exists a unique coalgebra homomorphism $h\colon Y \to E_i$ such that $e_i = h \circ e'$.

Proof. Since $\mathbf{C}$ is $\mathcal{E}$-cowellpowered the chain (E_i) of quotients stabilizes, i. e., for some ordinal i the quotients e_i and e_{i+1} are the same (more precisely, $e_{i+1,i}\colon E_{i+1} \twoheadrightarrow E_i$ is an isomorphism). We obtain a commutative square



(4)

and since $Fm_i \in \mathcal{M}$ and $e_{i+1} \in \mathcal{E}$ we obtain the unique diagonal d. Thus, we have an F-coalgebra

$$\varepsilon = (\ E_i \xrightarrow{e_{i+1,i}^{-1}} E_{i+1} \xrightarrow{d} FE_i\ ) \tag{5}$$

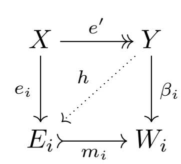
such that $e_i\colon X \twoheadrightarrow E_i$ is a homomorphism

$$\varepsilon \circ e_i = d \circ e_{i+1} = Fe_i \circ \alpha. \tag{6}$$

Observe that the canonical cone $\varepsilon_j\colon E_i \to W_j$ ($j \in \mathsf{Ord}$) of the coalgebra E_i is formed by

$$\varepsilon_j = m_j \circ e_{i,j} \quad \text{for all } j \le i \quad \text{and by} \quad \varepsilon_j = m_j \circ e_{j,i}^{-1} \quad \text{for all } j \ge i. \tag{7}$$

Now let $\beta\colon Y \to FY$ be a coalgebra with a homomorphism $e'\colon X \twoheadrightarrow Y$ in $\mathcal{E}$. Then the canonical cone $\beta_i\colon Y \to W_i$ satisfies $\beta_i \circ e' = \alpha_i$ by Lemma G.2. By diagonalization we obtain $h\colon Y \to E_i$:



(8)

We prove that h is a coalgebra homomorphism:

$$\varepsilon \circ h = Fh \circ \beta.$$

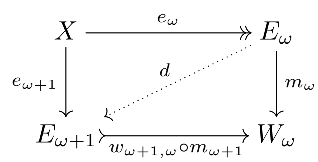
Indeed, this follows from e' being an epimorphism:

$$\begin{aligned}(\varepsilon \circ h) \circ e' &= \varepsilon \circ e_i && \text{by (8)}\\ &= Fe_i \circ \alpha && \text{by (6)}\\ &= Fh \circ Fe' \circ \alpha && \text{by (8)}\\ &= (Fh \circ \beta) \circ e' && e' \text{ homomorphism.}\end{aligned}$$

□

Theorem 3.9. *Let $F\colon \mathbf{Set} \to \mathbf{Set}$ be a finitary functor. Then for every F-coalgebra (X, α), its minimization is E_ω.*

Proof. We only need to prove that $e_{\omega+1,\omega}$:$E_{\omega+1} \to E_\omega$ is an isomorphism. Indeed, Worrell [29] proved that the connecting morphism $w_{\omega+1,\omega}\colon W_{\omega+1} \to W_\omega$ is a monomorphism. Therefore $w_{\omega+1,\omega} \circ m_{\omega+1}$ is a monomorphism and we obtain by diagonalization:



It is easy to show that d is an isomorphism with inverse $e_{\omega+1,\omega}$. □

Theorem 3.10. *The chain $(E_i)_{i\in\mathsf{Ord}}$ of Construction 3.7 can also be defined as follows:*

(a) Factor the unique morphism $d_0\colon X \to 1$ into $e_0\colon X \twoheadrightarrow E_0$ and $n_0\colon E_0 \rightarrowtail 1$.

(b) Given $e_i\colon X \twoheadrightarrow E_i$, factor $d_{i+1} = Fe_i \circ \alpha$ into $e_{i+1}\colon X \twoheadrightarrow E_{i+1}$ and $n_{i+1}\colon E_{i+1} \rightarrowtail FE_i$.

(c) For a limit ordinal j, form a limit of the preceding chain $(E_i)_{i<j}$, obtaining $\hat{E}_j$ and $\hat{e}_j\colon X \to \hat{E}_j$ as mediating morphism. Then factor $\hat{e}_j$ into $e_j\colon X \twoheadrightarrow E_j$ and $n_j\colon E_j \rightarrowtail \hat{E}_j$.

Proof. We will denote the arrows and objects obtained in the alternative construction by e'_i, n_i, E'_i. More specifically $e'_0\colon X \twoheadrightarrow E'_0, n_0\colon E'_0 \rightarrowtail 1$ are obtained by factoring the unique morphism $X \to 1$ and $e'_{i+1}\colon X \twoheadrightarrow E'_{i+1}, n_i\colon E'_{i+1} \rightarrowtail FE'_i$ are obtained by factoring $Fe'_i \circ \alpha$.

(a) Clearly since e'_0, n_0 and e_0, m_0 arise by factoring the same arrow, we can choose $e'_0 = e_0$.

(b) We now proceed by induction: we have $m_{i+1} \circ e_{i+1} = \alpha_{i+1} = F\alpha_i \circ \alpha = F(m_i \circ e_i) \circ \alpha = Fm_i \circ Fe_i \circ \alpha = Fm_i \circ Fe'_i \circ \alpha = Fm_i \circ n_{i+1} \circ e'_{i+1}$. Since

F preserves $\mathcal{M}$-morphisms and $\mathcal{M}$-morphisms compose we have that $Fm_i \circ n_{i+1}$ is contained in $\mathcal{M}$. Factorizations are unique and so we can choose $e'_{i+1} = e_{i+1}$.

(c) Take $\hat{E}_j$ as the limit of the preceding chain and obtain $\hat{e}_j : X \to \hat{E}_j$, $\hat{m}_j : \hat{E}_j \to W_j$ as mediating morphisms (for the limit of the E_i and the limit of the W_i $(i < j)$ respectively). Now both arrows $\alpha_j = m_j \circ e_j$ and $\hat{m}_j \circ \hat{e}_j$ are mediating morphisms for the cone $(\alpha_i)_{i<j}$ of X over the chain $(W_i)_{i<j}$. Hence, due to uniqueness $m_j \circ e_j = \hat{m}_j \circ \hat{e}_j$.

Next, consider the category $\mathbf{Mor}(\mathbf{C})$, whose objects are morphisms of $\mathbf{C}$ and arrows are commuting squares. In $\mathbf{Mor}(\mathbf{C})$, $\hat{m}_j$ is the limit object of the ω-chain

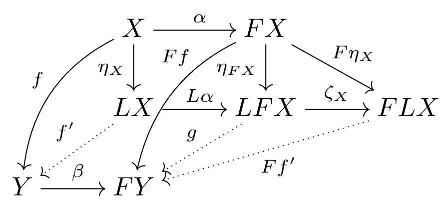
$$m_0 \leftarrow m_1 \leftarrow m_2 \leftarrow \cdots \leftarrow m_i \leftarrow \ldots$$

where each morphism $m_{i+1} \to m_i$ consists of a pair of morphisms of $\mathbf{C}$: $e_{i+1,i}$ and $w_{i+1,i}$. (The fact that $\hat{m}_j$ is the limit can be shown by standard diagram chasing.) It is known from [21] that the full subcategory of $\mathcal{M}$-arrows (seen as objects) is a reflective subcategory of $\mathbf{Mor}(\mathbf{C})$. Hence $\hat{m}_j$ as the limit object is in $\mathcal{M}$. Note however that $\hat{e}_j$ is not necessarily contained in $\mathcal{E}$. But by factoring $\hat{e}_j$ into an $\mathcal{E}$-morphisms e'_j and an $\mathcal{M}$-morphism, we obtain $e'_j = e_j$. □

Proposition 4.4. *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 morphism is constructed as described in Definition 4.3. Note especially that FLX is an object of $\mathbf{S}$, since F preserves $\mathbf{S}$, and hence the morphism ζ_X exists.

As in Definition 4.3 let $\alpha' = \zeta_X \circ L\alpha$. Now assume that $f : (X, \alpha) \to (Y, \beta)$ with $\beta : Y \to FY$ is a coalgebra morphism where β is an morphism of $\mathbf{S}$.



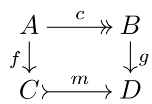
Let f' be the unique morphism in $\mathbf{S}$ for which $f' \circ \eta_X = f$ and let g be the unique morphism in $\mathbf{S}$ such that $g \circ \eta_{FX} = Ff$. We have to show that $Ff' \circ \alpha' = \beta \circ f$, i.e., f is indeed a coalgebra morphism.

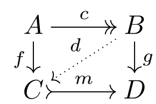
We first show that the square consisting of $L\alpha, g, f', \beta$ commutes: it holds that $(g \circ L\alpha) \circ \eta_X = g \circ \eta_{FX} \circ \alpha = Ff \circ \alpha = \beta \circ f = (\beta \circ f') \circ \eta_X$. Since η_X is the unit of a reflection and by uniqueness of the mediating morphism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

Next we show that the triangle consisting of the morphisms 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 $Ff' \circ \alpha' = Ff' \circ \zeta_X \circ L\alpha = g \circ L\alpha = \beta \circ f'$. □

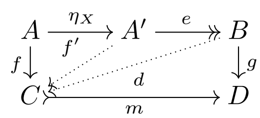
Lemma G.5 (Diagonalization for Pseudo-Factorizations). *Let* $\mathbf{S}$ *be a reflective subcategory of* $\mathbf{C}$ *and let* $(\mathcal{E},\mathcal{M})$ *be a factorization structure for* $\mathbf{S}$. *Assume a commuting diagram in* $\mathbf{C}$ *as shown on the left below where* $c = e \circ \eta_X$ *with* $e \in \mathcal{E}$ *and* $m \in \mathcal{M}$. *Furthermore let* g *be a morphism of* $\mathbf{S}$.





Then there exists a unique diagonal morphism 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 a morphism in $\mathbf{S}$, which implies the existence of a unique morphism $f'\colon A' \to C$ 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 morphisms). This commuting diagram lives in $\mathbf{S}$ and hence there exists a unique morphism $d\colon B \to C$ 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 $\mathbf{S}$. □

Theorem 4.9. *Given a coalgebra* $\alpha\colon X \to FX$ *in* $\mathbf{C}$*, the following four constructions obtain the same result (we also call this result the* minimization*):*

(i) Apply Construction 3.7 using the $(\mathcal{E},\mathcal{M})$*-pseudo-factorizations of Definition 4.6.*

(ii) Reflect α *into the subcategory* $\mathbf{S}$ *according to Definition 4.3 and then apply Construction 3.7 using* $(\mathcal{E},\mathcal{M})$*-factorizations.*

(iii) Apply the construction of Theorem 3.10 using $(\mathcal{E},\mathcal{M})$*-pseudo-factorizations.*

(iv) Reflect α *into the subcategory* $\mathbf{S}$ *and then apply the construction of Theorem 3.10 using* $(\mathcal{E},\mathcal{M})$*-factorizations.*

Proof. We start by showing that variant (i) and variant (ii) coincide if we use Construction 3.7.

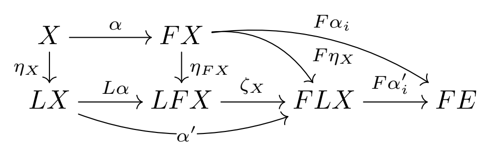
First note that the diagonal morphisms required in Construction 3.7 (variant (i)) exist due to Lemma G.5. The final sequence lives in $\mathbf{S}$, hence all morphisms w_{ji} are in $\mathbf{S}$. Furthermore m_j is an $\mathcal{M}$-morphism and hence in $\mathbf{S}$. This means that $w_{ji} \circ m_j$ is an $\mathbf{S}$-morphism and the conditions of Lemma G.5 are satisfied.

Assume that we apply Construction 3.7 (variant (i)) using the pseudo-factorizations, obtaining morphisms $\alpha_i\colon X \to W_i$, $c_i\colon X \to E_i$, $m_i\colon E_i \rightarrowtail W_i$.

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 morphisms arising in Construction 3.7 (variant (ii)) α'_i, e'_i, m'_i. We will show that we can choose the arrows in such a way that $\alpha_i = \alpha'_i \circ \eta_X, c_i = e'_i \circ \eta_X, m_i = m'_i$. This is true for $i = 0$ since α_0 is the unique morphism from X to 1 and $\alpha'_0 \circ \eta_X \colon X \to 1$. Now in order to obtain the pseudo-factorization of α_0 we first construct α'_0 and factor $\alpha'_0 = m'_0 \circ e'_0$. Hence $c_0 = e'_0 \circ \eta_X$ and $m_0 = m'_0$.

We assume by the induction hypothesis that $\alpha_i = \alpha'_i \circ \eta_X, c_i = e'_i \circ \eta_X, m_i = m'_i$.

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 $\alpha_i = \alpha'_i \circ \eta_X$.



Hence $\alpha_{i+1} = F\alpha_i \circ \alpha = F\alpha'_i \circ \alpha' \circ \eta_X = \alpha'_{i+1} \circ \eta_X$. Now, as argued above, the pseudo-factorization of α_{i+1} is obtained by factoring α'_{i+1} in the subcategory and hence $\alpha_{i+1} = \alpha'_{i+1} \circ \eta_X$, $c_{i+1} = e'_{i+1} \circ \eta_X$ and $m_{i+1} = m'_{i+1}$.

In both cases the diagonal morphisms $e_{j,i}$ agree and for limit ordinals we take limits of the same diagrams. Since $\mathbf{S}$ is a reflective subcategory of $\mathbf{C}$ those limits coincide. Hence we get the same chain of E_i's and an isomorphic minimization.

From Theorem 3.10 it follows that Construction 3.7 and the construction from Theorem 3.10 coincide if we first reflect α into $\mathbf{S}$ and then minimize using $(\mathcal{E}, \mathcal{M})$-factorizations (variants (ii) and (iv)).

Now it is left to show that we have a correspondence of the following two constructions: reflect first into $\mathbf{S}$ and then use the construction of Theorem 3.10 (variant (iv)) or use the construction of Theorem 3.10 with pseudo-factorizations (variant (iii)). The proof is more or less analogous to the proof above, the only critical case is construction step (c) (taking the limit for an ordinal j).

As above, in both cases we take limits of the same diagrams. In the case of pseudo-factorizations we obtain $\hat{c}_j \colon X \to \hat{E}_j$ as mediating morphism and in the other case we obtain $\hat{e}_j \colon LX \to \hat{E}_j$ as mediating morphism. It can be shown that $\hat{e}_j \circ \eta_X$ is also a mediating morphism from X and hence $\hat{c}_j = \hat{e}_j \circ \eta_X$. Hence by factoring $\hat{e}_j = n_j \circ e_j$ and pseudo-factoring $\hat{c}_j = n_j \circ c_j$ with $c_j = e_j \circ \eta_j$ we can continue the correspondence. □