HAL open science

How to Kill Epsilons with a Dagger

Filippo Bonchi, Stefan Milius, Alexandra Silva, Fabio Zanasi

To cite this version:

Filippo Bonchi, Stefan Milius, Alexandra Silva, Fabio Zanasi. How to Kill Epsilons with a Dagger. 12th International Workshop on Coalgebraic Methods in Computer Science (CMCS), Apr 2014, Grenoble, France. pp.53-74, 10.1007/978-3-662-44124-4_4 . hal-01408752

HAL Id: hal-01408752

https://inria.hal.science/hal-01408752v1

Submitted on 5 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

How to Kill Epsilons with a Dagger

A Coalgebraic Take on Systems with Algebraic Label Structure

Filippo Bonchi[1], Stefan Milius[2], Alexandra Silva[3], and Fabio Zanasi[1]

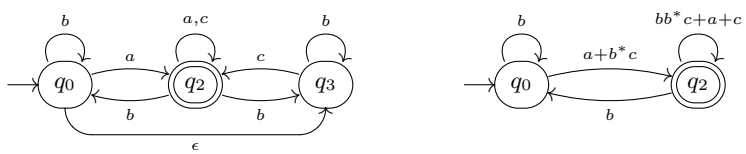
[1] ENS Lyon, U. de Lyon, CNRS, INRIA, UCBL, France
[2] Lehrstuhl für Theoretische Informatik, Friedrich-Alexander Universität Erlangen-Nürnberg
[3] Institute for Computing and Information Sciences, Radboud University Nijmegen*

Abstract. We propose an abstract framework for modeling state-based systems with internal behavior as e.g. given by silent or ϵ-transitions. Our approach employs monads with a parametrized fixpoint operator $\dagger$ to give a semantics to those systems and implement a sound procedure of abstraction of the internal transitions, whose labels are seen as the unit of a free monoid. More broadly, our approach extends the standard coalgebraic framework for state-based systems by taking into account the algebraic structure of the labels of their transitions. This allows to consider a wide range of other examples, including Mazurkiewicz traces for concurrent systems.

1 Introduction

The theory of coalgebras provides an elegant mathematical framework to express the semantics of computing devices: the operational semantics, which is usually given as a state machine, is modeled as a coalgebra for a functor; the denotational semantics as the unique map into the final coalgebra of that functor. While the denotational semantics is often *compositional* (as, for instance, ensured by the bialgebraic approach of [26]), it is sometimes not *fully-abstract*, i.e, it discriminates systems that are equal from the point of view of an external observer. This is due to the presence of internal transitions (also called ϵ-transitions) that are not observable but that are not abstracted away by the usual coalgebraic semantics using the unique homomorphism into the final coalgebra.

In this paper, we focus on the problem of giving trace semantics to systems with internal transitions. Our approach stems from an elementary observation (pointed out in previous work, e.g. [25]): the labels of transitions form a monoid and the internal transitions are those labeled by the unit of the monoid. Thus, there is an *algebraic structure* on the labels that needs to be taken into account when modeling the denotational semantics of those systems. To illustrate this point, consider the following two non-deterministic automata (NDA).



* Also affiliated to Centrum Wiskunde & Informatica (Amsterdam, The Netherlands) and HASLab / INESC TEC, Universidade do Minho (Braga, Portugal).

The one on the left (that we call $\mathbb{A}$) is an NDA with ϵ-transitions: its transitions are labeled either by the symbols of the alphabet $A = \{a, b, c\}$ or by the empty word $\epsilon \in A^*$. The one on the right (that we call $\mathbb{B}$) has transitions labeled by languages in $\mathcal{P}(A^*)$, here represented as regular expressions. The monoid structure on the labels is explicit on $\mathbb{B}$, while it is less evident in $\mathbb{A}$ since the set of labels $A \cup \{\epsilon\}$ does not form a monoid. However, this set can be trivially embedded into $\mathcal{P}(A^*)$ by looking at each symbols as the corresponding singleton language. For this reason each automaton with ϵ-transitions, like $\mathbb{A}$, can be regarded as an automaton with transitions labeled by languages, like $\mathbb{B}$. Furthermore, we can define the semantics of NDA with ϵ-transitions by defining the semantics of NDA with transitions labeled by languages: a word w is accepted by a state q if there is a path $q \xrightarrow{L_1} \cdots \xrightarrow{L_n} p$ where p is a final state, and there exist a decomposition $w = w_1 \cdots w_n$ such that $w_i \in L_i$. Observe that, with this definition, $\mathbb{A}$ and $\mathbb{B}$ accept the same language: all words over A that end with a or c. In fact, $\mathbb{B}$ was obtained from $\mathbb{A}$ in a well-known process to compute the regular expression denoting the language accepted by a given automaton [16].

We propose to define the semantics of systems with internal transitions following the same idea as in the above example. Given some transition type (i.e. an endofunctor) F, one first defines an embedding of F-systems with internal transitions into F^*-system, where F^* has been derived from F by making explicit the algebraic structure on the labels. Next one models the semantics of an F-system as the one of the corresponding F^*-system e. Naively, one could think of defining the semantics of e as the unique map $!_e$ into the final coalgebra for F^*. However, this approach turns out to be too fine grained, essentially because it ignores the underlying algebraic structure on the labels of e. The same problem can be observed in the example above: $\mathbb{B}$ and the representation of $\mathbb{A}$ as an automaton with languages as labels have different final semantics—they accept the same language only modulo the equations of monoids.

Thus we need to extend the standard coalgebraic framework by taking into account the algebraic structure on labels. To this end, we develop our theory for systems whose transition type F^* has a *canonical fixpoint*, i.e. its initial algebra and final coalgebra coincide. This is the case for many relevant examples, as observed in [14]. Our *canonical fixpoint semantics* will be given as the composite $¡ \circ !_e$, where $!_e$ is a coalgebra morphism given by finality and ¡ is an algebra morphism given by initiality. The target of ¡ will be an algebra for F^* encoding the equational theory associated with the labels of F^*-systems. Intuitively, ¡ being an *algebra* morphism, will take the quotient of the semantics given by $!_e$ modulo those equations. Therefore the extension provided by ¡ is the technical feature allowing us to take into account the algebraic structure on labels.

To study the properties of our canonical fixpoint semantics, it will be convenient to formulate it as an operator $e \mapsto e^\dagger$ assigning to systems (seen as sets of equations) a certain *solution*. Within the same perspective we will implement a different kind of solution $e \mapsto e^\ddagger$ turning any system e with internal transitions into one $e^\ddagger$ where those have been abstracted away. By comparing the operators $e \mapsto e^\dagger$ and $e \mapsto e^\ddagger$, we will then be able to show that such a procedure (also called *ϵ-elimination*) is sound with respect to the canonical fixpoint semantics.

To conclude, we will explore further the flexibility of our framework. In particular, we will model the case in which the algebraic structure of the labels is quotiented under

some equations, resulting in a coarser equivalence than the one given by the canonical fixpoint semantics. As a relevant example of this phenomenon, we give the first coalgebraic account of Mazurkiewicz traces.

Synopsis After recalling the necessary background in Section 2, we discuss our motivating examples—automata with ϵ-transitions and automata on words—in Section 3. Section 4 is devoted to present the canonical fixpoint semantics and the sound procedure of ϵ-elimination. This framework is then instantiated to the examples of Section 3. Finally, in Section 5 we show how a quotient of the algebra on labels induces a coarser canonical fixpoint semantics. We propose Mazurkiewicz traces as a motivating example for such a construction. A full version of this paper with all proofs and extra material can be found in `http://arxiv.org/abs/1402.4062`.

2 Preliminaries

In this section we introduce the basic notions we need for our abstract framework. We assume some familiarity with category theory. We will use boldface capitals $\mathbf{C}$ to denote categories, $X, Y, \ldots$ for objects and $f, g, \ldots$ for morphisms. We use Greek letters and double arrows, e.g. $\eta\colon F \Rightarrow G$, for natural transformations, monad morphisms and any kind of 2-cells. If $\mathbf{C}$ has coproducts we will denote them by $X + Y$ and use $\mathsf{inl}, \mathsf{inr}$ for the coproduct injections.

2.1 Monads

We recall the basics of the theory of monads, as needed here. For more information, see *e.g.* [20]. A monad is a functor $T\colon \mathbf{C} \to \mathbf{C}$ together with two natural transformations, a *unit* $\eta\colon \mathrm{id}_{\mathbf{C}} \Rightarrow T$ and a *multiplication* $\mu\colon T^2 \Rightarrow T$, which are required to satisfy the following equations, for every $X \in \mathbf{C}$: $\mu_X \circ \eta_{TX} = \mu_X \circ T\eta_X = \mathrm{id}$ and $T\mu_X \circ \mu_{TX} = \mu_X \circ \mu_X$.

A *morphism of monads* from (T, η^T, μ^T) to (S, η^S, μ^S) is a natural transformation $\gamma\colon T \Rightarrow S$ that preserves unit and multiplication: $\gamma_X \circ \eta^T_X = \eta^S_X$ and $\gamma_X \circ \mu^T_X = \mu^S_X \circ \gamma_{SX} \circ T\gamma_X$. A *quotient of monads* is a morphism of monads with epimorphic components.

Example 2.1. We briefly describe the examples of monads that we use in this paper.

1. Let $\mathbf{C} = \mathbf{Sets}$. The powerset monad $\mathcal{P}$ maps a set X to the set $\mathcal{P}X$ of subsets of X, and a function $f\colon X \to Y$ to $\mathcal{P}f\colon \mathcal{P}X \to \mathcal{P}Y$ given by direct image. The unit is given by the singleton set map $\eta_X(x) = \{x\}$ and multiplication by union $\mu_X(U) = \bigcup_{S \in U} S$.
2. Let $\mathbf{C}$ be a category with coproducts and E an object of $\mathbf{C}$. The exception monad $\mathcal{E}$ is defined on objects as $\mathcal{E}X = E + X$ and on arrows $f\colon X \to Y$ as $\mathcal{E}f = \mathrm{Id}_E + f$. Its unit and multiplication are given on $X \in \mathbf{C}$ respectively as $\mathsf{inr}_X : X \to E + X$ and $\nabla_E + \mathrm{Id}_X : E + E + X \to E + X$, where $\nabla_E = [\mathrm{id}_E, \mathrm{id}_E]$ is the codiagonal. When $\mathbf{C} = \mathbf{Sets}$, E can be thought as a set of *exceptions* and this monad is often used to encode computations that might fail throwing an exception chosen from the set E.

3. Let H be an endofunctor on a category $\mathbf{C}$ such that for every object X there exists a free H-algebra H^*X on X (equivalently, an initial $H + X$-algebra) with the structure $\tau_X : HH^*X \to H^*X$ and universal morphism $\eta_X : X \to H^*X$. Then as proved by Barr [7] (see also Kelly [18]) $H^*\colon \mathbf{C} \to \mathbf{C}$ is the functor part of a *free monad* on H with the unit given by the above η_X and the multiplication given by the freeness of H^*H^*X: μ_X is the unique H-algebra homomorphism from (H^*H^*X, τ_{H^*X}) to (H^*X, τ_X) such that $\mu_X \cdot \eta_{H^*X} = \eta_X$. Also notice that for a complete category every free monad arises in this way. Finally, for later use we fix the notation $\kappa = \tau \cdot H\eta\colon H \Rightarrow H^*$ for the universal natural transformation of the free monad.

Given a monad $M\colon \mathbf{C} \to \mathbf{C}$, its *Kleisli category* $\mathcal{K}\ell(M)$ has the same objects as $\mathbf{C}$, but morphisms $X \to Y$ in $\mathcal{K}\ell(M)$ are morphisms $X \to MY$ in $\mathbf{C}$. The identity map $X \to X$ in $\mathcal{K}\ell(M)$ is M's unit $\eta_X\colon X \to MX$; and composition $g \circ f$ in $\mathcal{K}\ell(M)$ uses M's multiplication: $g \circ f = \mu \circ Mg \circ f$. There is a forgetful functor $\mathcal{U}\colon \mathcal{K}\ell(T) \to \mathbf{C}$, sending X to TX and f to $\mu \circ Tf$. This functor has a left adjoint $\mathcal{J}$, given by $\mathcal{J}X = X$ and $\mathcal{J}f = \eta \circ f$. The Kleisli category $\mathcal{K}\ell(M)$ inherits coproducts from the underlying category $\mathbf{C}$. More precisely, for every objects X and Y their coproduct $X + Y$ in $\mathbf{C}$ is also a coproduct in $\mathcal{K}\ell(M)$ with the injections $\mathcal{J}\mathsf{inl}$ and $\mathcal{J}\mathsf{inr}$.

2.2 Distributive laws and liftings

The most interesting examples of the theory that we will present in Section 4 concern coalgebras for functors $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$ that are obtained as liftings of endofunctors H on $\mathbf{Sets}$. Formally, given a monad $M\colon \mathbf{C} \to \mathbf{C}$, a *lifting* of $H\colon \mathbf{C} \to \mathbf{C}$ to $\mathcal{K}\ell(M)$ is an endofunctor $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$ such that $\mathcal{J} \circ H = \widehat{H} \circ \mathcal{J}$. The lifting of a monad (T, η, μ) is a monad $(\widehat{T}, \widehat{\eta}, \widehat{\mu})$ such that $\widehat{T}$ is a lifting of T and $\widehat{\eta}$, $\widehat{\mu}$ are given on $X \in \mathcal{K}\ell(M)$ (i.e. $X \in \mathbf{Sets}$) respectively as $\mathcal{J}(\eta_X)$ and $\mathcal{J}(\mu_X)$.

A natural way of lifting functors and monads is by mean of distributive laws. A *distributive law* of a monad (T, η^T, μ^T) over a monad (M, η^M, μ^M) is a natural transformation $\lambda\colon TM \Rightarrow MT$, that commutes appropriately with the unit and multiplication of both monads; more precisely, the diagrams below commute:

$$\begin{tikzcd}
TX \arrow[r, equal] \arrow[d, "T\eta^M_X"'] & TX \arrow[d, "\eta^M_{TX}"] \\
TMX \arrow[r, "\lambda_X"'] & MTX \\
MX \arrow[u, "\eta^T_{MX}"] \arrow[r, equal] & MX \arrow[u, "M\eta^T_X"']
\end{tikzcd}
\qquad
\begin{tikzcd}
TM^2X \arrow[r, "\lambda_{MX}"] \arrow[d, "T\mu^M_X"'] & MTMX \arrow[r, "M\lambda_X"] & M^2TX \arrow[d, "\mu^M_{TX}"] \\
TMX \arrow[rr, "\lambda_X"'] & & MTX \\
T^2MX \arrow[u, "\mu^T_{MX}"] \arrow[r, "T\lambda_{MX}"'] & TMTX \arrow[r, "\lambda_{TX}"'] & T^2MX \arrow[u, "M\mu^T_X"']
\end{tikzcd}$$

A distributive law of a *functor* T over a *monad* (M, η^M, μ^M) is a natural transformation $\lambda\colon TM \Rightarrow MT$ such that only the two topmost squares above commute.

The following "folklore" result gives an alternative description of distributive laws in terms of liftings to Kleisli categories, see also [17], [22] or [6].

Proposition 2.2 ([22]). *Let (M, η^M, μ^M) be a monad on a category $\mathbf{C}$. Then the following holds:*

1. *For every endofunctor T on $\mathbf{C}$, there is a bijective correspondence between liftings of T to $\mathcal{K}\ell(M)$ and distributive laws of T over M.*
2. *For every monad (T, η^T, μ^T) on $\mathbf{C}$, there is a bijective correspondence between liftings of (T, η^T, μ^T) to $\mathcal{K}\ell(M)$ and distributive laws of T over M.*

In what follows we shall simply write $\widehat{H}$ for the lifting of an endofunctor H.

Proposition 2.3 ([14]). *Let $M\colon \mathbf{C} \to \mathbf{C}$ be a monad and $H\colon \mathbf{C} \to \mathbf{C}$ be a functor with a lifting $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$. If H has an initial algebra $\iota\colon HI \overset{\cong}{\Rightarrow} I$ (in $\mathbf{C}$), then $\mathcal{J}\iota\colon \widehat{H}I \to I$ is an initial algebra for $\widehat{H}$ (in $\mathcal{K}\ell(M)$).*

In our examples, we will often consider the free monad (Example 2.1.3) $\widehat{H}^*$ generated by a lifted functor $\widehat{H}$. The following result will be pivotal.

Proposition 2.4. *Let $H\colon \mathbf{C} \to \mathbf{C}$ be a functor and $M\colon \mathbf{C} \to \mathbf{C}$ be a monad such that there is a lifting $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$. Then the free monad $H^*\colon \mathbf{C} \to \mathbf{C}$ lifts to a monad $\widehat{H^*}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$. Moreover, $\widehat{H^*} = \widehat{H}^*$.*

Recall from [14] that for every polynomial endofunctor H on $\mathbf{Sets}$ there exists a canonical distributive law of H over any *commutative* monad M (equivalently, a canonical lifting of H to $\mathcal{K}\ell(M)$); this result was later extended to so-called analytic endofunctors of $\mathbf{Sets}$ (see [21]). This can be used in our applications since the power-set functor $\mathcal{P}$ is commutative, and so is the exception monad $\mathcal{E}$ iff $E = 1$.

2.3 $\mathbf{Cppo}$-enriched categories

For our general theory we are going to assume that we work in a category where the hom-sets carry a cpo structure. Recall that a *cpo* is a partially ordered set in which all ω-chains have a join. A cpo with bottom is a cpo with a least element $\perp$. A function between cpos is called *continuous* if it preserves joins of ω-chains. Cpos with bottom and continuous maps form a category that we denote by $\mathbf{Cppo}$.

A $\mathbf{Cppo}$-*enriched category* $\mathbf{C}$ is a category where (a) each hom-set $\mathbf{C}(X, Y)$ is a cpo with a bottom element $\perp_{X,Y}\colon X \to Y$ and (b) composition is continuous, that is:

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

The composition is called *left strict* if $\perp_{Y,Z} \circ f = \perp_{X,Z}$ for all arrows $f\colon X \to Y$.

In our applications, $\mathbf{C}$ will mostly be a Kleisli category for a monad on $\mathbf{Sets}$. Throughout this subsection we assume that $\mathbf{C}$ is a $\mathbf{Cppo}$-enriched category.

An endofunctor $H\colon \mathbf{C} \to \mathbf{C}$ is said to be *locally continuous* if for any ω-chain $f_n\colon X \to Y$, $n < \omega$ in $\mathbf{C}(X, Y)$ we have:

$$H\left(\bigsqcup_{n<\omega} f_n \right) = \bigsqcup_{n<\omega} H(f_n).$$

We are going to make use of the fact that a locally continuous endofunctor H on $\mathbf{C}$ has a *canonical fixpoint*, i.e. whenever its initial algebra exists it is also its final coalgebra:

Theorem 2.5 ([11]). *Let $H\colon \mathbf{C} \to \mathbf{C}$ be a locally continuous endofunctor on the $\mathbf{Cppo}$-enriched category $\mathbf{C}$ whose composition is left-strict. If an initial H-algebra $\iota\colon HI \overset{\cong}{\Rightarrow} I$ exists, then $\iota^{-1}\colon I \overset{\cong}{\Rightarrow} HI$ is a final H-coalgebra.*

In the sequel, we will be interested in free algebras for a functor H on $\mathbf{C}$ and the free monad H^* (cf. Example 2.1.3). For this observe that coproducts in $\mathbf{C}$ are always $\mathbf{Cppo}$-enriched, i.e. all copairing maps $[-,-] : \mathbf{C}(X,Y) \times \mathbf{C}(X',Y) \to \mathbf{C}(X+X',Y)$ are continuous; in fact, it is easy to show that this map is continuous in both of its arguments using that composition with the coproduct injections is continuous.

Proposition 2.6. *Let $\mathbf{C}$ be $\mathbf{Cppo}$-enriched with composition left-strict. Furthermore, let $H : \mathbf{C} \to \mathbf{C}$ be locally continuous and assume that all free H-algebras exist. Then the free monad H^* is locally continuous.*

2.4 Final Coalgebras in Kleisli categories

In our applications the $\mathbf{Cppo}$-enriched category will be the Kleisli category $\mathbf{C} = \mathcal{K}\ell(M)$ of a monad on $\mathbf{Sets}$ and the endofunctors of interest are liftings $\widehat{H}$ of endofunctors H on $\mathbf{Sets}$. It is known that in this setting a final coalgebra for the lifting $\widehat{H}$ can be obtained as a lifting of an initial H-algebra (see Hasuo et al. [14]). The following result is a variation of Theorem 3.3 in [14]:

Theorem 2.7. *Let $M\colon \mathbf{Sets} \to \mathbf{Sets}$ be a monad and $H\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor such that*

(a) $\mathcal{K}\ell(M)$ is $\mathbf{Cppo}$-enriched with composition left strict;
(b) H is accessible (i.e., H preserves λ-filtered colimits for some cardinal λ) and has a lifting $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$ which is locally continuous.

If $\iota\colon HI \overset{\cong}{\Rightarrow} I$ is the initial algebra for the functor H, then

1. *$\mathcal{J}\iota\colon \widehat{H}I \to I$ is the initial algebra for the functor $\widehat{H}$;*
2. *$\mathcal{J}\iota^{-1}\colon I \to \widehat{H}I$ is the final coalgebra for the functor $\widehat{H}$.*

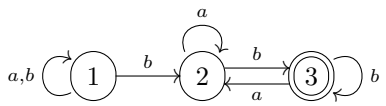
The first item follows from Proposition 2.3 and the second one follows from Theorem 2.5. There are two differences with Theorem 3.3 in [14]:

(1) The functor $H\colon \mathbf{Sets} \to \mathbf{Sets}$ is supposed to preserve ω-colimits rather that being accessible. We use the assumption of accessibility because it guarantees the existence of all free algebras for H and for $\widehat{H}$, which implies also that for all $Y \in \mathcal{K}\ell(M)$ an initial $\widehat{H}^*(\mathrm{Id} + Y)$-algebra exists. This property of $\widehat{H}^*$ will be needed for applying our framework of Section 4.

(2) We assume that the lifting $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$ is locally continuous rather than locally monotone. We will need continuity to ensure the double dagger law in Remark 2.9. This assumption is not really restrictive since, as explained in Section 3.3.1 of [14], in all the meaningful examples where $\widehat{H}$ is locally monotone, it is also locally continuous.

Example 2.8 (NDA). Consider the powerset monad $\mathcal{P}$ (Example 2.1.1) and the functor $HX = A \times X + 1$ on $\mathbf{Sets}$ (with $1 = \{\checkmark\}$). The functor H lifts to $\widehat{H}$ on $\mathcal{K}\ell(\mathcal{P})$ as follows: for any $f\colon X \to Y$ in $\mathcal{K}\ell(\mathcal{P})$ (that is $f\colon X \to \mathcal{P}(Y)$ in $\mathbf{Sets}$), $\widehat{H}f\colon A \times X + 1 \to A \times Y + 1$ is given by $\widehat{H}f(\checkmark) = \{\checkmark\}$ and $\widehat{H}f(\langle a, x\rangle) = \{\langle a, y\rangle \mid y \in f(x)\}$.

Non-deterministic automata (NDA) over the input alphabet A can be regarded as coalgebras for the functor $\widehat{H}\colon \mathcal{K}\ell(\mathcal{P}) \to \mathcal{K}\ell(\mathcal{P})$. Consider, on the left, a 3-state NDA, where the only final state is marked by a double circle.



$$X = \{1, 2, 3\} \quad A = \{a, b\}$$
$$e(1) = \{\langle a, 1\rangle, \langle b, 1\rangle, \langle b, 2\rangle\}$$
$$e(2) = \{\langle a, 2\rangle, \langle b, 3\rangle\} \quad e(3) = \{\checkmark, \langle a, 2\rangle, \langle b, 3\rangle\}$$

It can be represented as a coalgebra $e\colon X \to \widehat{H}X$, that is a function $e\colon X \to \mathcal{P}(A \times X + 1)$, given above on the right, which assigns to each state $x \in X$ a set which: contains $\checkmark$ if x is final; and $\langle a, y\rangle$ for all transitions $x \xrightarrow{a} y$.

It is easy to see that $M = \mathcal{P}$ and H above satisfy the conditions of Theorem 2.7 and therefore both the final $\widehat{H}$-coalgebra and the initial $\widehat{H}$-algebra are the lifting of the initial algebra for the functor $HX = A \times X + 1$, given by A^* with structure $\iota\colon A \times A^* + 1 \to A^*$ which maps $\langle a, w\rangle$ to aw and $\checkmark$ to ϵ.

For an NDA (X, e), the final coalgebra homomorphism $!_e\colon X \to A^*$ is the function $X \to \mathcal{P}A^*$ that maps every state in X to the language that it accepts. In $\mathcal{K}\ell(\mathcal{P})$:

$$\begin{tikzcd}
X \arrow[rr, dashed, "!_e"] \arrow[d, "e"'] & & A^* \arrow[d, "\wr\,\iota^{-1}"] \\
A \times X + 1 \arrow[rr, dashed, "A \times !_e + 1"'] & & A \times A^* + 1
\end{tikzcd}$$

$$\begin{aligned} \epsilon \in !_e(x) &\iff \checkmark \in e(x) \\ aw \in !_e(x) &\iff \text{for some } y \in X,\ (a, y) \in e(x) \text{ and } w \in !_e(y) \end{aligned}$$

2.5 Monads with Fixpoint Operators

In order to develop our theory of systems with internal behavior, we will adopt an equational perspective on coalgebras. In the sequel we recall some preliminaries on this viewpoint.

Let $T : \mathbf{C} \to \mathbf{C}$ be a monad on any category $\mathbf{C}$. Any morphism $e : X \to T(X+Y)$ (i.e. a coalgebra for the functor $T(\mathrm{Id} + Y)$) may be understood as a system of mutually recursive equations. In our applications we are interested in the case where $\mathbf{C} = \mathcal{K}\ell(M)$ and $T = \widehat{H^*}$ is a (lifted) free monad. As in the example of NDA (Example 2.8) take

$M = \mathcal{P}$ and $HX = 1 + A \times X$. Now, set $TX = A^* + A^* \times X$ and consider the following system of mutually recursive equations

$$x_0 \approx \{c, (ab, x_1)\}, \qquad x_1 \approx \{d, (a, x_0), (\epsilon, y)\},$$

where $x_0, x_1 \in X$ are *recursion variables*, $y \in Y$ is a *parameter* and $a, b, c, d \in A$. A *solution* assigns to each of the two variables x_0, x_1 an element of $\mathcal{P}(TY)$ such that the formal equations $\approx$ become actual identities in $\mathcal{K}\ell(\mathcal{P})$:

$$x_0 \mapsto \{(aba)^*c, (aba)^*abd, ((aba)^*ab, y)\}, \quad x_1 \mapsto \{(aab)^*d, (aab)^*ac, ((aab)^*, y)\}.$$

Observe that the above system of equations corresponds to an *equation morphism* $e\colon X \to T(X+Y)$ and the solution to a morphism $e^\dagger\colon X \to TY$, both in $\mathcal{K}\ell(M)$. The property that $e^\dagger$ is a solution for e is expressed by the following equation in $\mathcal{K}\ell(M)$:

$$e^\dagger = (X \xrightarrow{e} T(X+Y) \xrightarrow{T[e^\dagger, \eta^T_Y]} TTY \xrightarrow{\mu^T_Y} TY). \tag{1}$$

So $e \mapsto e^\dagger$ is a *parametrized fixpoint operator*, i.e. a family of fixpoint operators indexed by parameter sets Y.

Remark 2.9. In our applications we shall need a certain equational property of the operator $e \mapsto e^\dagger$: for all $Y \in \mathbf{C}$ and equation morphism $e : X \to T(X + X + Y)$, the following equation, called *double dagger law*, holds:

$$e^{\dagger\dagger} = (X \xrightarrow{e} T(X+X+Y) \xrightarrow{T(\nabla_X + Y)} T(X+Y))^\dagger.$$

This and other laws of parametrized fixpoint operators have been studied by Bloom and Ésik in the context of *iteration theories* [8]. A closely related notion is that of *Elgot monads* [2, 3].

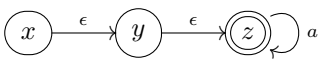
Example 2.10 (Least fixpoint solutions). Let $T : \mathbf{C} \to \mathbf{C}$ be a locally continuous monad on the $\mathbf{Cppo}$-enriched category $\mathbf{C}$. Then T is equipped wi th a parametrized fixpoint operator obtained by taking least fixpoints: given a morphism $e : X \to T(X+Y)$ consider the function Φ_e on $\mathbf{C}(X, TY)$ given by $\Phi_e(s) = \mu^T_Y \circ T[s, \eta^T_Y] \circ e$. Then Φ_e is continuous and we take $e^\dagger$ to be the least fixpoint of Φ_e. Since $e^\dagger = \Phi_e(e^\dagger)$, equation (1) holds, and it follows from the argument in Theorem 8.2.15 and Exercise 8.2.17 in [8] that the operator $e \mapsto e^\dagger$ satisfies the axioms of iteration theories (or Elgot monads, respectively). In particular the double dagger law holds for the least fixpoint operator $e \mapsto e^\dagger$.

3 Motivating examples

The work of [14] bridged a gap in the theory of coalgebras: for certain functors, taking the final coalgebra directly in $\mathbf{Sets}$ does not give the right notion of equivalence. For instance, for NDA, one would obtain bisimilarity instead of language equivalence. The change to Kleisli categories allowed the recovery of the usual language semantics for NDA and, more generally, led to the development of *coalgebraic trace semantics*.

In the Introduction we argued that there are relevant examples for which this approach still yields the unwanted notion of equivalence, the problem being that it does not consider the extra algebraic structure on the label set. In the sequel, we motivate the reader for the generic theory we will develop by detailing two case studies in which this phenomenon can be observed: NDA with ϵ-transitions and NDA with word transitions. Later on, in Example 5.7, we will also consider Mazurkiewicz traces [19].

NDA with ϵ-transitions. In the world of automata, ϵ-transitions are considered in order to enable easy composition of automata and compact representations of languages. These transitions are to be interpreted as the empty word when computing the language accepted by a state. Consider, on the left, the following simple example of an NDA with ϵ-transitions, where states x and y just make ϵ transitions. The intended semantics in this example is that all states accept words in a^*.

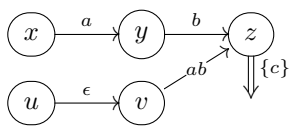


$$\begin{array}{rcl} e(x) & = & \{(\epsilon, y)\} \\ e(y) & = & \{(\epsilon, z)\} \\ e(z) & = & \{(a, z), \checkmark\} \end{array} \qquad \begin{array}{rcl} !_e(x) & = & \epsilon\epsilon a^* \\ !_e(y) & = & \epsilon a^* \\ !_e(z) & = & a^* \end{array}$$

Note that, more explicitly, these are just NDA where the alphabet has a distinguished symbol ϵ. So, they are coalgebras for the functor $\widehat{H + \mathrm{Id}}\colon \mathcal{K}\ell(\mathcal{P}) \to \mathcal{K}\ell(\mathcal{P})$ (where H is the functor of Example 2.8), i.e. functions $e\colon X \to \mathcal{P}((A \times X + 1) + X) \cong \mathcal{P}((A + 1) \times X + 1)$, as made explicit for the above automaton in the middle.

The final coalgebra for $\widehat{H + \mathrm{Id}}$ is simply $(A + 1)^*$ and the final map $!_e\colon X \to (A + 1)^*$ assigns to each state the language in $(A + 1)^*$ that it accepts. However, the equivalence induced by $!_e$ is too fine grained: for the automata above, $!_e$ maps x, y and z to three different languages (on the right), where the number of ϵ plays an explicit role, but the intended semantics should disregard ϵ's.

NDA with word transitions. This is a variation on the motivating example of the introduction: instead of languages, transitions are labeled by words[4]. Formally, consider again the functor H from Example 2.8. Then NDA with word transitions are coalgebras for the functor $\widehat{H^*}\colon \mathcal{K}\ell(\mathcal{P}) \to \mathcal{K}\ell(\mathcal{P})$, that is, functions $e\colon X \to \mathcal{P}(A^* \times X + A^*) \cong \mathcal{P}(A^* \times (X + 1))$. We observe that they are like NDA but (1) transitions are labeled by words in A^*, rather than just symbols of the alphabet A, and (2) states have associated output languages, rather than just $\checkmark$. We will draw them as ordinary automata plus an arrow $\overset{L}{\Rightarrow}$ to denote the output language of a state (no $\Rightarrow$ stands for the empty language). For an example, consider the following word automaton and associated transition function e.



$$e(x) = \{(a, y)\} \quad e(y) = \{(b, z)\} \quad e(z) = \{c\}$$
$$e(u) = \{(\epsilon, v)\} \quad e(v) = \{(ab, z)\}$$

[4] More generally, one could consider labels from an arbitrary monoid.

The semantics of NDA with word transitions is given by languages over A, obtained by concatenating the words in the transitions and ending with a word from the output language. For instance, x above accepts word abc but not ab.

However, if we consider the final coalgebra semantics we again have a mismatch. The initial H^*-algebra has carrier $(A^*)^* \times A^*$ that can be represented as the set of non-empty lists of words over A^*, where $(A^*)^*$ indicates possibly empty lists of words. Its structure $\iota\colon A^* \times ((A^*)^* \times A^*) + A^* \to (A^*)^* \times A^*$ maps w into $(\langle\rangle, w)$ and $(w', (l, w))$ into $(w' :: l, w)$. Here, we use $\langle\rangle$ to denote the empty list and $::$ is the append operation. By Theorem 2.7, the final $\widehat{H^*}$-coalgebra has the same carrier and structure $\jmath\iota^{-1}$. The final map, as a function $!_e\colon X \to \mathcal{P}((A^*)^* \times A^*)$, is then defined by commutativity of the following square (in $\mathcal{K}\ell(\mathcal{P})$):

$$\begin{tikzcd}
X \arrow[r, dashed, "!_e"] \arrow[d, "e"'] & (A^*)^* \times A^* \arrow[d, "\jmath\iota^{-1}"] \\
A^* \times X + A^* \arrow[r, dashed, "\mathrm{id}_{A^*} \times !_e + \mathrm{id}_{A^*}"'] & A^* \times ((A^*)^* \times A^*) + A^*
\end{tikzcd} \qquad (2)$$

$$(\langle\rangle, w) \in !_e(x) \;\Leftrightarrow\; w \in e(x)$$
$$(w :: l, w') \in !_e(x) \;\Leftrightarrow\; \exists_y\, (w, y) \in e(x) \text{ and } (l, w') \in !_e(y).$$

Once more, the semantics given by $!_e$ is too fine grained: in the above example, $!_e(x) = \{([a, b], c)\}$ and $!_e(u) = \{([\epsilon, ab], c)\}$ whereas the intended semantics would equate both x and u, since they both accept the language $\{abc\}$.

Note that any NDA can be regarded as word automaton. Recall the natural transformation $\kappa\colon \widehat{H} \Rightarrow \widehat{H^*}$ defined in Example 2.1.3: for the functor $\widehat{H}$ of NDA,

$$\kappa_X\colon A \times X + 1 \to A^* \times X + A^*$$

maps any pair $(a, x) \in A \times X$ into $\{(a, x)\} \in \mathcal{P}(A^* \times X + A^*)$ and $\checkmark \in 1$ into $\{\epsilon\} \in \mathcal{P}(A^* \times X + A^*)$. Composing an NDA $e\colon X \to \widehat{H}X$ with $\kappa_X\colon \widehat{H}X \to \widehat{H^*}X$, one obtains the word automaton $\kappa_X \circ e$.

In the same way, every NDA with ϵ-transitions can also be seen as a word automaton by postcomposing with the natural transformation $[\kappa, \eta]\colon \widehat{H + \mathrm{Id}} \Rightarrow \widehat{H^*}$. Here, $\eta\colon \mathrm{Id} \Rightarrow \widehat{H^*}$ is the unit of the free monad $\widehat{H^*}$ defined on a given set X below (the multiplication $\mu\colon \widehat{H^*}\widehat{H^*} \Rightarrow \widehat{H^*}$ is shown on the right).

$$\eta_X\colon X \to A^* \times X + A^* \qquad x \mapsto \{(\epsilon, x)\}$$

$$\mu_X\colon A^* \times ((A^* \times X + A^*) + A^* \to A^* \times X + A^* \qquad (w, (w', x)) \mapsto \{(w \cdot w', x)\} \quad (w, w') \mapsto \{w \cdot w'\} \qquad w \mapsto \{w\}$$

In the next section, we propose to define the semantics of $\widehat{H^*}$-coalgebras via a canonical fixpoint operator rather than with the final map which as we saw above might yield unwanted semantics. Then, using the observation above, the semantics of $\widehat{H}$-coalgebras and $\widehat{H + \mathrm{Id}}$-coalgebras will be defined by embedding them into $\widehat{H^*}$-coalgebras via the natural transformations κ and $[\kappa, \eta]$ described above.

4 Canonical Fixpoint Solutions

In this section we lay the foundations of our approach. A construction is introduced assigning canonical solutions to coalgebras seen as equation morphisms (*cf.* Section 2.5) in a $\mathbf{Cppo}$-enriched setting. We will be working under the following assumptions.

Assumption 4.1. Let $\mathbf{C}$ be a $\mathbf{Cppo}$-enriched category with coproducts and composition left-strict. Let T be a locally continuous monad on $\mathbf{C}$ such that, for all object Y, an initial algebra for $T(\mathrm{Id} + Y)$ exists.

As seen in Example 2.10, in this setting an equation morphism $e : X \to T(X + Y)$ may be given the least solution. Here, we take a different approach, exploiting the initial algebra-final coalgebra coincidence of Theorem 2.5.

For every parameter object $Y \in \mathbf{C}$, the endofunctor $T(\mathrm{Id} + Y)$ is a locally continuous monad because it is the composition of T with the (locally continuous) exception monad $\mathrm{Id}+Y$. Thus, by Theorem 2.5 applied to $T(\mathrm{Id}+Y)$, the initial $T(\mathrm{Id}+Y)$-algebra $\iota_Y : T(I_Y + Y) \xrightarrow{\cong} I_Y$ yields a final $T(\mathrm{Id} + Y)$-coalgebra $\iota_Y^{-1} : I_Y \xrightarrow{\cong} T(I_Y + Y)$. This allows us to associate with any equation morphism $e : X \to T(X+Y)$ a canonical morphism of type $X \to TY$ as in the following diagram.

$$\begin{tikzcd}
X \arrow[r, dashed, "!_e"] \arrow[d, "e"'] & I_Y \arrow[r, dashed, "¡"] \arrow[d, bend right, "\iota_Y^{-1}"'] & TY \\
T(X+Y) \arrow[r, dashed, "T(!_e+\mathrm{id}_Y)"'] & T(I_Y+Y) \arrow[r, dashed, "T(¡+\mathrm{id}_Y)"'] \arrow[u, bend right, "\iota_Y"'] & TTY \arrow[u, "\mu_Y^T"'] \\
 & & T(TY+Y) \arrow[u, "T[\mathrm{id}_{TY},\eta_Y^T]"']
\end{tikzcd} \tag{3}$$

In (3), the map $!_e \colon X \to I_Y$ is the unique morphism of $T(\mathrm{Id} + Y)$-coalgebras given by finality of $\iota_Y^{-1} \colon I_Y \to T(I_Y + Y)$, whereas $¡ : I_Y \to TY$ is the unique morphism of $T(\mathrm{Id} + Y)$-algebras given by initiality of $\iota_Y \colon T(I_Y + Y) \to I_Y$.

We call the composite $¡ \circ !_e \colon X \to TY$ the *canonical fixpoint solution* of e. In the following we check that the canonical fixpoint solution is indeed a solution of e, in fact, it coincides with the least solution.

Proposition 4.2. *Given a morphism $e \colon X \to T(X + Y)$, then the least solution of e as in Example 2.10 is the canonical fixpoint solution: $e^\dagger = ¡ \circ !_e \colon X \to TY$ as in* (3).

As recalled in Example 2.10, the least fixpoint operator $e \mapsto e^\dagger$ satisfies the double dagger law. Thus Proposition 4.2 yields the following result[5]

Corollary 4.3. *Let $\mathbf{C}$ and $T \colon \mathbf{C} \to \mathbf{C}$ be as in Assumption 4.1. Then the canonical fixpoint operator $e \mapsto e^\dagger$ associated with T satisfies the double dagger law.*

[5] The equality of least and canonical fixpoint solutions can be used to state a stronger result, namely that canonical fixpoint solutions satisfy the axioms of iteration theories (*cf.* Example 2.10). However, the double dagger law is the only property that we need here, explaining the statement of Corollary 4.3.

We now introduce a factorisation result on the operator $e \mapsto e^\dagger$, which is useful for comparing solutions provided by different monads connected via a monad morphism.

Proposition 4.4 (Factorisation Lemma). *Suppose that T and T' are monads on $\mathbf{C}$ satisfying Assumption 4.1 and $\gamma\colon T \Rightarrow T'$ is a monad morphism. For any morphism $e : X \to T(X+Y)$:*

$$\gamma_Y \circ e^\dagger = (\gamma_{X+Y} \circ e)^\dagger : X \to T'Y,$$

where $e^\dagger$ is provided by the canonical fixpoint solution for T and $(\gamma_{X+Y} \circ e)^\dagger$ by the one for T'.

4.1 A Theory of Systems with Internal Behavior

We now use canonical fixpoint solutions to provide an abstract theory of systems with internal behavior, that we will later instantiate to the motivating examples of Section 3. Throughout this section, we will develop our framework for the following ingredients.

Assumption 4.5. Let $\mathbf{C}$ be a $\mathbf{Cppo}$-enriched category with coproducts and composition left-strict and let $F\colon \mathbf{C} \to \mathbf{C}$ be a locally continuous functor for which all free F-algebras exist. Consider the following two monads derived from F:

- the free monad $F^*\colon \mathbf{C} \to \mathbf{C}$ (*cf.* Example 2.1.3), for which we suppose that an initial $F^*(\mathrm{Id}+Y)$-algebra exists for all $Y \in \mathbf{C}$;
- for a fixed $X \in \mathbf{C}$, the exception monad $FX + \mathrm{Id}\colon \mathbf{C} \to \mathbf{C}$ (*cf.* Example 2.1.2), for which we suppose that an initial $FX + \mathrm{Id} + Y$-algebra exists for all $Y \in \mathbf{C}$.

In the next proposition we verify that the construction introduced in the previous section applies to the two monads of Assumption 4.5.

Proposition 4.6. *Let $\mathbf{C}$, F, F^* and $FX + \mathrm{Id}$ be as in Assumption 4.5. Then $\mathbf{C}$ and the monads $F^*\colon \mathbf{C} \to \mathbf{C}$ and $FX + \mathrm{Id}\colon \mathbf{C} \to \mathbf{C}$ satisfy Assumption 4.1. Thus both F^* and $FX + \mathrm{Id}$ are monads with canonical fixpoint solution (which satisfy the double dagger law by Corollary 4.3).*

To avoid ambiguity, we denote with $e \mapsto e^\dagger$ the canonical fixpoint operator associated with F^* and with $e \mapsto e^\ddagger$ the one associated with $FX + \mathrm{Id}$.

We will employ the additional structure of those two monads for the analysis of F-*systems with internal transitions*. An F-system is simply an F-coalgebra $e\colon X \to FX$, where we take the operational point of view of seeing X as a space of states and F as the transition type of e. An F-system with internal transitions is an $(F + \mathrm{Id})$-coalgebra $e\colon X \to FX + X$, where the component X of the codomain is targeted by those transitions representing the internal (non-interacting) behavior of system e.

A key observation for our analysis is that F-systems—with or without internal transitions—enjoy a standard representation as F^*-systems, that is, coalgebras of the form $e\colon X \to F^*X$.

Definition 4.7 (F-systems as F^*-systems). *Let $\kappa : F \to F^*$ be as in Example 2.1.3. We introduce the following encoding $e \mapsto \bar{e}$ of F-systems and F-systems with internal transitions as F^*-systems.*

– *Given an F-system $e\colon X \to FX$, define $\bar{e}\colon X \to F^*X$ as*

$$\bar{e} : X \xrightarrow{e} FX \xrightarrow{\kappa_X} F^*X.$$

– *Given an F-system with internal transitions $e : X \to FX + X$, define $\bar{e} : X \to F^*X$ as $\bar{e} : X \xrightarrow{e} FX + X \xrightarrow{[\kappa_X, \eta_X^{F^*}]} F^*X$.*

Thus F-systems (with or without internal transitions) may be seen as equation morphisms $X \to F^*(X+0)$ for the monad F^* (with the initial object $Y = 0$ as parameter), with solutions by canonical fixpoint (*cf.* Section 2.5). This will allow us to achieve the following.

§**1** We supply a uniform trace semantics for F-systems, possibly with internal transitions, and F^*-systems, based on the canonical fixpoint solution operator of F^*.
§**2** We use the canonical fixpoint operator of $FX + \mathrm{Id}$ to transform any F-system $e : X \to FX + X$ with internal transitions into an F-system $e \backslash \epsilon : X \to FX$ without internal transitions.
§**3** We prove that the transformation of §**2** is sound with respect to the semantics of §**1**.

§**1: Uniform trace semantics.** The canonical fixpoint semantics of F-systems, with or without internal transitions, and F^*-systems is defined as follows.

Definition 4.8 (Canonical Fixpoint Semantics).

– *For an F^*-system $e\colon X \to F^*X$, its semantics $[\![e]\!]\colon X \to F^*0$ is defined as $e^\dagger$ (note that e can be seen as an equation morphism for F^* on parameter $Y = 0$).*
– *For an F-system $e : X \to FX$, its semantics $[\![e]\!]\colon X \to F0$ is defined as $\bar{e}^\dagger = (\kappa_X \circ e)^\dagger$.*
– *For an F-system with internal transitions $e : X \to FX + X$, its semantics $[\![e]\!]\colon X \to F0$ is defined as $\bar{e}^\dagger = ([\kappa_X, \eta_X^{F^*}] \circ e)^\dagger$.*

The underlying intuition of Definition 4.8 is that canonical fixpoint solutions may be given an operational understanding. Given an F^*-system $e\colon X \to F^*X$, its solution $e^\dagger\colon X \to F^*0$ is formally defined as the composite ¡ $\circ\, !_e$ (*cf.* (3)): we can see the coalgebra morphism $!_e$ as a map that gives the *behavior* of system e without taking into account the structure of labels and the algebra morphism ¡ as evaluating this structure, e.g. flattening words of words, using the initial algebra $\mu_0\colon F^*F^*0 \to F^*0$ for the monad F^*. In particular, the action of ¡ is what makes our semantics suitable for modeling "algebraic" operations on internal transitions such as ϵ-elimination, as we will see in concrete instances of our framework.

Remark 4.9. The canonical fixpoint semantics of Definition 4.8 encompasses the framework for traces in [14], where the semantics of an F-system $e : X \to FX$—without internal transitions—is defined as the unique morphism $!_e$ from X into the final F-coalgebra F^*0. Indeed, using finality of F^*0, it can be shown that $!_e = [\![e]\!]$. Theorem 4.10 below guarantees compatibility with Assumption 4.5.

The following result is instrumental in our examples and in comparing our theory with the one developed in [14] for trace semantics in Kleisli categories.

Theorem 4.10. *Let $M\colon \mathbf{Sets} \to \mathbf{Sets}$ be a monad and $H\colon \mathbf{Sets} \to \mathbf{Sets}$ be a functor satisfying the assumptions of Theorem 2.7, that is:*

(a) $\mathcal{K}\ell(M)$ is $\mathbf{Cppo}$-enriched and composition is left strict;
(b) H is accessible and has a locally continuous lifting $\widehat{H}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$.

Then $\mathcal{K}\ell(M)$, $\widehat{H}$, $\widehat{H}^*$ and $\widehat{H}\mathcal{J}X + \mathrm{Id}$ (for a given set X) satisfy Assumption 4.5.

Example 4.11 (Semantics of NDA with word transitions). In Section 3, we have modeled NDA with word transitions as $\widehat{H^*}$-coalgebras on $\mathcal{K}\ell(M)$, where H and M are defined as for NDA (see Example 2.8). By Proposition 2.4, $\widehat{H^*} = \widehat{H}^*$ and thus, by virtue of Theorem 4.10, $\widehat{H^*}$ satisfies Assumption 4.5. Therefore we can define the semantics of NDA with word transitions $e\colon X \to \mathcal{P}(A^* \times X + A^*)$ via canonical fixpoint solutions as $[\![e]\!] = e^\dagger = ¡ \circ !_e$:

$$
\begin{tikzcd}
X \arrow[r, dashed, "!_e"] \arrow[d, "e"'] & (A^*)^* \times A^* \arrow[r, dashed, "¡"] \arrow[d, shift right, "\cong"'] & A^* \\
A^* \times X + A^* \arrow[r, dashed, "\mathrm{id}\times !_e + \mathrm{id}"'] & A^* \times ((A^*)^* \times A^*) + A^* \arrow[r, dashed, "\mathrm{id}\times ¡ + \mathrm{id}"'] \arrow[u, shift right, "\cong"'] & A^* \times A^* + A^* \arrow[u, "\mu_0"']
\end{tikzcd}
$$

$$
\begin{aligned}
¡(\langle\rangle, w) &= \{w\} \\
¡(w :: l, w') &= \{wu \mid u \in ¡(l, w')\}
\end{aligned}
$$

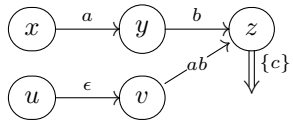
Observe that the above diagram is just (3) instantiated with $T = \widehat{H^*}$ and $Y = 0$. Moreover, this diagram is in $\mathcal{K}\ell(\mathcal{P})$ and hence the explicit definition of $e^\dagger$ as a function $X \to \mathcal{P}(A^*)$ is given by $e^\dagger(x) = \bigcup \mathcal{P}(¡)(!_e(x))$.

Both $!_e$ and ¡ can be defined uniquely by the commutativity of the above diagram. We have already defined $!_e$ in diagram (2) and the definition of ¡ is given in the right-hand square of the above diagram. The isomorphism in the middle and μ_0 were defined in Section 3.

Using the above formula $e^\dagger(x) = \bigcup \mathcal{P}(¡)(!_e(x))$ we now have the semantics of e:

$$
\begin{aligned}
w \in e^\dagger(x) \Leftrightarrow\ & w \in e(x) \ \textbf{or} \\
& \exists_{y \in X, w_1, w_2 \in A^*}\ (w_1, y) \in e(x), w_2 \in e^\dagger(y) \text{ and } w = w_1 w_2.
\end{aligned} \tag{5}
$$

This definition is precisely the language semantics: a word w is accepted by a state x if there exists a decomposition $w = w_1 \cdots w_n$ such that $x \xrightarrow{w_1} y_1 \xrightarrow{w_2} \cdots \xrightarrow{w_{n-1}} y_{n-1} \overset{w_n}{\Longrightarrow}$. Take again the automaton of the motivating example. We can calculate the semantics and observe that we now get exactly what was expected: $e^\dagger(u) = e^\dagger(v)$.

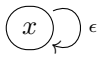


$$
\begin{aligned}
!_e(x) &= \{([a,b],c)\} & e^\dagger(x) &= \{abc\} \\
!_e(y) &= \{([b],c)\} & e^\dagger(y) &= \{bc\} \\
!_e(z) &= \{(\langle\rangle,c)\} & e^\dagger(z) &= \{c\} \\
!_e(u) &= \{([\epsilon,ab],c)\} & e^\dagger(u) &= \{abc\} \\
!_e(v) &= \{([ab],c)\} & e^\dagger(v) &= \{abc\}
\end{aligned}
$$

The key role played by the monad structure on A^* can be appreciated by comparing the graphs of $!_e$ and $e^\dagger = ¡ \circ !_e$ as in the example above. The algebra morphism ¡: $(A^*)^* \times$

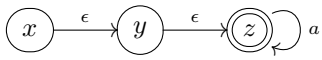
$A^* \to A^*$ maps values from the initial algebra $(A^*)^* \times A^*$ for the *endofunctor* $\widehat{H^*}$ into the initial algebra A^* for the *monad* $\widehat{H^*}$: its action is precisely to take into account the additional equations encoded by the algebraic theory of the monad $\widehat{H^*}$. For instance, we can see the mapping of $!_e(u) = \{([\epsilon, ab], c)\}$ into the word abc as the result of concatenating the words ϵ, ab, c and then quotienting out of the equation $\epsilon abc = abc$ in the monoid A^*.

Remark 4.12 (Multiple Solutions). The canonical solution $e^\dagger$ is not the unique solution. Indeed, the uniqueness of $!_e$ in the left-hand square and of ¡ in the right-hand square of the diagram above does not imply the uniqueness of $e^\dagger$. To see this, take for instance the automaton



Both $s(x) = \emptyset$ and $s'(x) = A^*$ are solutions. The canonical one is the least one, i.e., $e^\dagger(x) = s(x) = \emptyset$.

Example 4.13 (Semantics of NDA with ϵ-transitions). NDA with ϵ-transitions are modeled as $\widehat{H + \mathrm{Id}}$-coalgebras on $\mathcal{K}\ell(M)$, where H and M are defined as for NDA (see Example 2.8). We can define the semantics of NDA with ϵ-transitions via canonical fixpoint solutions as $[\![e]\!] = \bar{e}^\dagger$, where $\bar{e}$ is the automaton with word transitions corresponding to e (see Definition 4.7). The first example in Section 3 would be represented as follows,



$$\begin{aligned} \bar{e}(x) = [\kappa_X, \eta_X] \circ e(x) &= \{(\epsilon, y)\} \\ \bar{e}(y) = [\kappa_X, \eta_X] \circ e(y) &= \{(\epsilon, z)\} \\ \bar{e}(z) = [\kappa_X, \eta_X] \circ e(z) &= \{(a, z), \epsilon\} \end{aligned}$$

where η and κ are defined as at the end of Section 3. By using (5), it can be easily checked that the semantics $[\![e]\!] = \bar{e}^\dagger \colon X \to \mathcal{P}A^*$ maps x, y and z into a^*.

§**2: Elimination of internal transitions.** We view an F-system $e \colon X \to FX + X$ with internal transitions as an equation morphism for the monad $FX + \mathrm{Id}$, with parameter $Y = 0$. Thus we can use the canonical fixpoint solution of $FX + \mathrm{Id}$ to obtain an F-system $e^\ddagger \colon X \to FX + 0 = FX$, which we denote by $e\backslash\epsilon$. The construction is depicted below.

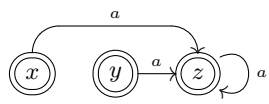
$$\begin{tikzcd} X \arrow[r, dashed, "!_e"] \arrow[d, "e"'] \arrow[rr, bend left, "e\backslash\epsilon \overset{\mathrm{def}}{=} e^\ddagger"] & \mathbb{N} \times FX \arrow[r, dashed, "¡"] \arrow[d, bend left, "\cong"] & FX \\ FX + X \arrow[r, "\mathrm{id}_{FX} + !_e"'] & FX + \mathbb{N} \times FX \arrow[r, "\mathrm{id}_{FX} + ¡"'] \arrow[u, bend left, "\cong"] & FX + FX \arrow[u, "\mu_0 = \nabla_{FX}"'] \end{tikzcd} \tag{6}$$

Example 4.14 (ϵ-elimination). Using the automaton of Example 4.13, we can perform ϵ-elimination, as defined in (6), using the canonical solution for the monad $\widehat{H}\mathcal{J}X + \mathrm{Id}$:

$$\begin{tikzcd}
X \arrow[r, dashed, "!_e"] \arrow[d, "e"'] & \mathbb{N} \times (A \times X + 1) \arrow[r, dashed, "¡"] \arrow[d, bend left, "\cong"] & (A \times X + 1) \\
(A \times X + 1) + X \arrow[r, "\mathrm{id} + !_e"'] & (A \times X + 1) + \mathbb{N} \times (A \times X + 1) \arrow[r, "\mathrm{id} + ¡"'] \arrow[u, bend left, "\cong"] & (A \times X + 1) + (A \times X + 1) \arrow[u, "\mu_0 = \nabla"']
\end{tikzcd}$$

We obtain the following NDA $e\backslash\epsilon \stackrel{\text{def}}{=} ¡ \circ !_e \colon X \to A \times X + 1$.

$$\begin{array}{rclrcl}
!_e(x) & = & \{(2, a, z), (2, \checkmark)\} & e\backslash\epsilon(x) & = & \{(a, z), \checkmark\} \\
!_e(y) & = & \{(1, a, z), (1, \checkmark)\} & e\backslash\epsilon(y) & = & \{(a, z), \checkmark\} \\
!_e(z) & = & \{(0, a, z), (0, \checkmark)\} & e\backslash\epsilon(x) & = & \{(a, z), \checkmark\}
\end{array}$$



The semantics $[\![e\backslash\epsilon]\!]$ is defined as $\overline{e\backslash\epsilon}^{\dagger}$, where $\overline{e\backslash\epsilon} = \kappa_X \circ e\backslash\epsilon$ is the representation of the NDA $e\backslash\epsilon$ as an automaton with word transitions (Definition 4.7). It is immediate to see, in this case, that $[\![e\backslash\epsilon]\!] = [\![e]\!]$. This fact is an instance of Theorem 4.17 below.

Remark 4.15. Note that ϵ-elimination was recently defined using a trace operator on a Kleisli category [13, 24, 5]. These works are based on the trace semantics of Hasuo et al. [14] and tailored for ϵ-elimination. They do not take into account any algebraic structure of the labels and are hence not applicable to the other examples we consider in this paper.

§**3: Soundness of ϵ-elimination.** We now formally prove that the canonical fixpoint semantics of e and $e\backslash\epsilon$ coincide. To this end, first we show how the construction $e \mapsto e\backslash\epsilon$ can be expressed in terms of the canonical fixpoint solution of F^*. This turns out to be an application of the factorisation lemma (Proposition 4.4), for which we introduce the natural transformation $\pi \colon FX + \mathrm{Id} \Rightarrow F^*(X + \mathrm{Id})$ defined at $Y \in \mathbf{C}$ by

$$\pi_Y \colon\ FX + Y \xrightarrow{[\kappa_X\,,\,\eta_Y^{F^*}]} F^*X + F^*Y \xrightarrow{[F^*\mathsf{inl}, F^*\mathsf{inr}]} F^*(X + Y)\ .$$

Since F^* is a monad with canonical fixpoint solutions, it can be verified that so is $F^*(X + \mathrm{Id})$. Moreover, π is a monad morphism between $FX + \mathrm{Id}$ and $F^*(X + \mathrm{Id})$. These observations allow us to prove the following.

Proposition 4.16 (Factorisation property of $e \mapsto e\backslash\epsilon$). *For any F-system $e : X \to FX + X$ with internal transitions, consider the equation morphism $\pi_X \circ e : X \to F^*(X + X)$. Then:*

$$\pi_0 \circ e\backslash\epsilon = (\pi_X \circ e)^{\dagger} : X \to F^*X.$$

Proof. This follows simply by an application of Proposition 4.4 to $e\backslash\epsilon = e^{\ddagger}$ and $\gamma = \pi$ with $Y = 0$. □

We are now in position to show point §**3**: soundness of ϵ-elimination.

Theorem 4.17 (Eliminating internal transitions is sound). *For any F-system $e : X \to FX + X$ with internal transitions,*

$$[\![e \backslash \epsilon]\!] = [\![e]\!].$$

Proof. The statement is shown by the following derivation.

$$\begin{aligned}
[\![e \backslash \epsilon]\!] &= [\![e^{\ddagger}]\!] && \text{Definition of } e \backslash \epsilon \\
&= (\kappa_X \circ e^{\ddagger})^{\dagger} && \text{Definition of } [\![-]\!] \text{ (Def. 4.8)} \\
&= (\pi_0 \circ e^{\ddagger})^{\dagger} && \text{Definition of } \pi_0 \\
&= (\pi_X \circ e)^{\dagger\dagger} && \text{Proposition 4.16} \\
&= (F^*(\nabla_X) \circ (\pi_X \circ e))^{\dagger} && \text{double dagger law} \\
&= \bar{e}^{\dagger} && \text{Definition of } \bar{e} \text{ (Def. 4.7) and } \pi_X \\
&= [\![e]\!] && \text{Definition of } [\![-]\!].
\end{aligned}$$

□

5 Quotient Semantics

When considering behavior of systems it is common to encounter spectrums of successively coarser equivalences. For instance, in basic process algebra trace equivalence can be obtained by quotienting bisimilarity with an axiom stating the distributivity of action prefixing by non-determinism [23]. There are many more examples of this phenomenon, including Mazurkiewicz traces, which we will describe below.

In this section we develop a variant of the canonical fixpoint semantics, where we can encompass in a uniform manner behaviors which are quotients of the canonical behaviors of the previous section (that is, the object F^*0).

Assumption 5.1. Let $\mathbf{C}$, F, F^* and $FX + \mathrm{Id}$ be as in Assumption 4.5 and $\gamma\colon F^* \Rightarrow Q$ a monad quotient for some monad Q. Moreover, suppose that for all $Y \in \mathbf{C}$ an initial $Q(\mathrm{Id} + Y)$-algebra exists.

Observe that, as Assumption 5.1 subsumes Assumption 4.5, we are within the framework of previous section, with the canonical fixpoint solution of F^* providing semantics for F^*- and F-systems. For our extension, one is interested in $Q0$ as a semantic domain coarser than F^*0 and we aim at defining an interpretation for F-systems in $Q0$. To this aim, we first check that Q has canonical fixpoint solutions.

Proposition 5.2. *Let $\mathbf{C}$, F, Q and $\gamma : F^* \Rightarrow Q$ be as in Assumption 5.1. Then Assumption 4.1 holds for $\mathbf{C}$ and Q, meaning that Q is a monad with canonical fixpoint solutions (which satisfy the double dagger law by Corollary 4.3).*

We use the notation $e \mapsto e^{\sim}$ for the canonical fixpoint operator of Q. This allows us to define the semantics of Q-systems, analogously to what we did for F^*-systems in Definition 4.8. Moreover, the connecting monad morphism $\gamma\colon F^* \Rightarrow Q$ yields an extension of this semantics to include also systems of transition type F^* and F.

Definition 5.3 (Quotient Semantics). *The quotient semantics of F-systems, with or without internal transitions, F^*-systems and Q-systems is defined as follows.*

- *For a Q-system $e\colon X \to QX$, its semantics $[\![e]\!]_\sim : X \to Q0$ is defined as $e^\sim$ (note that e can be regarded as an equation morphism for Q with $Y = 0$).*
- *For an F^*-system $e\colon X \to F^*X$, its semantics $[\![e]\!]_\sim : X \to Q0$ is defined as $(\gamma_X \circ e)^\sim$.*
- *For an F-system e—with or without internal transitions—its semantics $[\![e]\!]_\sim : X \to Q0$ is defined as $(\gamma_X \circ \bar{e})^\sim$, where $\bar{e}$ is as in Definition 4.7.*

The Factorisation Lemma (Proposition 4.4) allows us to establish a link between the canonical fixpoint semantics $[\![-]\!]$ and the quotient semantics $[\![-]\!]_\sim$.

Proposition 5.4 (Factorisation for the quotient semantics). *Let e be either an F^*-system or an F-system (with or without internal transitions). Then:*

$$[\![e]\!]_\sim = \gamma_0 \circ [\![e]\!]. \tag{7}$$

As a corollary we obtain that eliminating internal transitions is sound also for quotient semantics.

Corollary 5.5. *For any F-system $e : X \to FX + X$ with internal transitions,*

$$[\![e]\!]_\sim = [\![e \backslash \epsilon]\!]_\sim.$$

The quotient semantics can be formulated in a Kleisli category $\mathcal{K}\ell(M)$ by further assuming (c) below. This is needed to lift a quotient of monads from $\mathbf{Sets}$ to $\mathcal{K}\ell(M)$.

Theorem 5.6. *Let $M\colon \mathbf{Sets} \to \mathbf{Sets}$ be a monad and $H\colon \mathbf{Sets} \to \mathbf{Sets}$ be an accessible functor satisfying the assumptions of Theorem 2.7. By Proposition 2.4 the free monad H^* on H lifts to a monad $\widehat{H^*}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$ via a distributive law $\lambda\colon H^*M \Rightarrow MH^*$ with $\widehat{H^*} = \widehat{H}^*$. Let $R\colon \mathbf{Sets} \to \mathbf{Sets}$ be a monad and $\xi\colon H^* \Rightarrow R$ a monad quotient such that*

(c) for each set X, there is a map $\lambda'_X\colon RMX \to MRX$ making the following commute.

$$\begin{tikzcd} H^*MX \arrow[r, "\lambda_X"] \arrow[d, "\xi_{MX}"'] & MH^*X \arrow[d, "M\xi_X"] \\ RMX \arrow[r, "\lambda'_X"'] & MRX \end{tikzcd}$$

Then the following hold:

1. *there is a monad $\widehat{R}\colon \mathcal{K}\ell(M) \to \mathcal{K}\ell(M)$ lifting R and a monad morphism $\widehat{\xi}\colon \widehat{H}^* \Rightarrow \widehat{R}$ defined as $\widehat{\xi_X} = \mathcal{J}(\xi_X)$;*
2. *$\mathcal{K}\ell(M)$, $\widehat{H}$, $\widehat{H}^*$, $\widehat{H}\mathcal{J}X + \mathrm{Id}$ (for a given set X), $\widehat{R}$ and $\widehat{\xi}\colon \widehat{H}^* \Rightarrow \widehat{R}$ satisfy Assumption 5.1.*

Notice that condition (c) and the first part of statement 1 are related to [9, Theorem 1]; however, that paper treats distributive laws of monads over endofunctors.

Example 5.7 (Mazurkiewicz traces). This example, using a known equivalence in concurrency theory, illustrates the use of the quotient semantics developed in Section 5.

The trace semantics proposed by Mazurkiewicz [19] accounts for concurrent actions. Intuitively, let A be the action alphabet and $a, b \in A$. We will call a and b concurrent, and write $a \equiv b$, if the order in which these actions occur is not relevant. This means that we equate words that only differ in the order of these two actions, e.g. $uabv$ and $ubav$ denote the same Mazurkiewicz trace.

To obtain the intended semantics of Mazurkiewicz traces we use the quotient semantics defined above[6]. In particular, for Mazurkiewisz traces one considers a symmetric and irreflexive "independence" relation I on the label set A. Let $\equiv$ be the least congruence relation on the free monoid A^* such that

$$(a, b) \in I \Rightarrow ab \equiv ba.$$

We now have two monads on $\mathbf{Sets}$, namely $H^*X = A^* \times X + A^*$ and $RX = A^*/_\equiv \times X + A^*/_\equiv$. There is the canonical quotient of monads $\xi\colon H^* \Rightarrow R$ given by identifying words of the same $\equiv$-equivalence class. It can be checked that those data satisfy the assumptions of Theorem 5.6 and thus we are allowed to apply the quotient semantics $[\![-]\!]_\sim$. This will be given on an NDA $e\colon X \to \widehat{H}X$ by first embedding it into $\widehat{H}^*$ as $\bar{e} = \kappa_X \circ e\colon X \to \widehat{H}^*X$ and then into $\widehat{R}$ as $\widehat{\xi}_X \circ \bar{e}\colon X \to \widehat{R}X$. To this morphism we apply the canonical fixpoint operator of $\widehat{R}$ to obtain $(\widehat{\xi}_X \circ \bar{e})^\sim$, that is, the semantics $[\![e]\!]_\sim\colon X \to R0 = A^*/\equiv$. It is easy to see that this definition captures the intended semantics: for all states $x \in X$

$$[\![e]\!]_\sim(x) = \{[w]_\equiv \mid w \in [\![e]\!](x)\}.$$

Indeed, by Proposition 5.4, $[\![e]\!]_\sim = \widehat{\xi}_0 \circ [\![e]\!]$ and $\widehat{\xi}_0\colon \widehat{H}^*0 \to \widehat{R}0$ is just $\mathcal{J}\xi_0$ where $\xi_0\colon A^* \to A^*/_\equiv$ maps every word w into its equivalence class $[w]_\equiv$.

6 Discussion

The framework introduced in this paper provides a uniform way to express the semantics of systems with internal behaviour via canonical fixpoint solutions. Moreover, these solutions are exploited to eliminate internal transitions in a sound way, i.e., preserving the semantics. We have shown our approach at work on NDA with ϵ-transitions but, by virtue of Theorem 4.10, it also covers all the examples in [14] (like probabilistic systems) and more (like the weighted automata on positive reals of [24]).

It is worth noticing that, in principle, our framework is applicable also to examples that do not arise from Kleisli categories. Indeed the theory of Section 4 is formulated for a general category $\mathbf{C}$: Assumption 4.5 only requires $\mathbf{C}$ to be $\mathbf{Cppo}$-enriched and the monad T to be locally continuous. The role of these assumptions is two-fold: (a) ensuring the initial algebra-final coalgebra coincidence and (b) guaranteeing that the canonical fixpoint operator $e \mapsto e^\dagger$ satisfies the *double dagger law*. If (a) implies (b),

[6] Mazurkiewicz traces were defined over labelled transition systems which are similar to NDA but where every state is final. For simplicity, we consider LTS here immediately as NDA.

we could have formulated our theory just assuming the coincidence of initial algebra and final coalgebra and without any $\mathbf{Cppo}$-enrichment. Condition (a) holds for some interesting examples not based on Kleisli categories, e.g. for examples in the category of join semi-lattices. Therefore it is of relevance to investigate the following question: given a monad T with initial algebra-final coalgebra coincidence, under which conditions does the canonical fixpoint solution provided by T satisfy the double dagger law?

As a concluding remark, let us recall that our original question concerned the problem of modeling the semantics of systems where labels carry an algebraic structure. In this paper we have mostly been focusing on automata theory, but there are many other examples in which the information carried by the labels has relevance for the semantics of the systems under consideration: in logic programming labels are substitutions of terms; in (concurrent) constraint programming they are elements of a lattice; in process calculi they are actions representing syntactical contexts and in tile systems [12] they are morphisms in a category. We believe that our approach provides various insights towards a coalgebraic semantics for these computational models.

Acknowledgments. We are grateful to the anonymous referees for valuable comments. The work of Alexandra Silva is partially funded by the ERDF through the Programme COMPETE and by the Portuguese Foundation for Science and Technology, project ref. `FCOMP-01-0124-FEDER-020537` and `SFRH/BPD/71956/2010`. The first and the fourth author acknowledge support by project `ANR 12IS0 2001 PACE`.

References

[1] Adámek, J.: Free algebras and automata realizations in the language of categories. Comment. Math. Univ. Carolin. 15, 589–602 (1974)

[2] Adámek, J., Milius, S., Velebil, J.: Equational properties of iterative monads. Inform. and Comput. 208, 1306–1348 (2010), doi:10.1016/j.ic.2009.10.006

[3] Adámek, J., Milius, S., Velebil, J.: Elgot theories: a new perspective of the equational properties of iteration. Math. Structures Comput. Sci. 21(2), 417–480 (2011)

[4] Adámek, J., Porst, H.E.: On tree coalgebras and coalgebra presentations. Theoret. Comput. Sci. 311, 257–283 (2004)

[5] Asada, K., Hidaka, S., Kato, H., Hu, Z., Nakano, K.: A parameterized graph transformation calculus for finite graphs with monadic branches. In: Peña, R., Schrijvers, T. (eds.) PPDP. pp. 73–84. ACM (2013)

[6] Balan, A., Kurz, A.: On coalgebras over algebras. Theoret. Comput. Sci. 412(38), 4989–5005 (2011)

[7] Barr, M.: Coequalizers and free triples. Math. Z. 116, 307–322 (1970)

[8] Bloom, S.L., Ésik, Z.: Iteration Theories: the equational logic of iterative processes. EATCS Monographs on Theoretical Computer Science, Springer (1993)

[9] Bonsangue, M.M., Hansen, H.H., Kurz, A., Rot, J.: Presenting distributive laws. In: Heckel and Milius [15], pp. 95–109

[10] Brookes, S.D., Main, M.G., Melton, A., Mislove, M.W., Schmidt, D.A. (eds.): Proceedings MFPS 1993. Lecture Notes in Computer Science, vol. 802. Springer (1994)

[11] Freyd, P.J.: Remarks on algebraically compact categories. London Mathematical Society Lecture Notes Series, vol. 177. Cambridge University Press (1992)

[12] Gadducci, F., Montanari, U.: The tile model. In: Plotkin, G.D., Stirling, C., Tofte, M. (eds.) Proof, Language, and Interaction. pp. 133–166. The MIT Press (2000)

[13] Hasuo, I., Jacobs, B., Sokolova, A.: Generic forward and backward simulations. In: (Partly in Japanese) Proceedings of *JSSST Annual Meeting* (2006)
[14] Hasuo, I., Jacobs, B., Sokolova, A.: Generic trace semantics via coinduction. Log. Methods Comput. Sci. 3(4:11), 1–36 (2007)
[15] Heckel, R., Milius, S. (eds.): Proceedings CALCO 2013. Lecture Notes in Computer Science, vol. 8089. Springer (2013)
[16] Hopcroft, J., Motwani, R., Ullman, J.: Introduction to Automata Theory, Languages, and Computation (3rd Edition). Wesley (2006)
[17] Johnstone, P.: Adjoint lifting theorems for categories of algebras. Bull. London Math. Soc. 7, 294–297 (1975)
[18] Kelly, G.M.: A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on. Bull. Austral. Math. Soc. 22, 1–83 (1980)
[19] Mazurkiewicz, A.: Concurrent Program Schemes and Their Interpretation. Aarhus University, Comp. Science Depart., DAIMI PB-78 (July 1977)
[20] Mac Lane, S.: Categories for the Working Mathematician. Springer, Berlin (1971)
[21] Milius, S., Palm, T., Schwencke, D.: Complete iterativity for algebras with effects. In: Kurz et al. (eds.) Proc. CALCO'09. LNCS, vol. 5728, pp. 34–48. Springer (2009)
[22] Mulry, P.S.: Lifting theorems for kleisli categories. In: Brookes et al. [10], pp. 304–319
[23] Rabinovich, A.M.: A complete axiomatisation for trace congruence of finite state behaviors. In: Brookes et al. [10], pp. 530–543
[24] Silva, A., Westerbaan, B.: A coalgebraic view of ε-transitions. In: Heckel and Milius [15], pp. 267–281
[25] Sobociński, P.: Relational presheaves as labelled transition systems. In: Proc. Coalgebraic Methods in Computer Science (CMCS'12). LNCS, vol. 7399, pp. 40–50. Springer (2012)
[26] Turi, D., Plotkin, G.: Towards a Mathematical Operational Semantics. In: Proc. Logic in Computer Science (LICS'97). IEEE Computer Society (1997)