Generators and bases for algebras over a monad

Stefan Zetzsche[1], Alexandra Silva[1], and Matteo Sammartino[1,2]

[1]University College London

[2]Royal Holloway University of London

Some of the fundamental notions of linear algebra are the concepts of a generator and a basis for a vector space. In the category theoretic formulation of universal algebra, vector spaces are the Eilenberg-Moore algebras over the free vector space monad on the category of sets. In this paper we investigate general notions of generators and bases for algebras over arbitrary monads on arbitrary categories. On the one hand, we establish purely algebraic results, for instance about the existence and uniqueness of generators and bases, the representation of algebra homomorphisms, and the product of generators. On the other hand, we use the general notion in the context of coalgebraic systems and show that a generator for the underlying algebra of a bialgebra gives rise to an equivalent free bialgebra. As a result, we are able to recover known constructions from automata theory such as the canonical residual finite state automaton and the minimal xor automaton. Finally, we instantiate the framework to a variety of example monads, both set and non set-based.

1 Introduction

One of the central concepts of linear algebra is the notion of a basis for a vector space: a subset of a vector space is called a basis for the former if every vector can be uniquely written as a finite linear combination of basis elements. Part of the importance of bases stems from the convenient consequences that follow from their existence. For example, linear transformations between vector spaces admit matrix representations relative to pairs of bases [1], which can be used for efficient numerical calculations. The idea of a basis however is not restricted to the theory of vector spaces: other algebraic theories have analogous notions of bases – and generators, by waiving the uniqueness constraint –, for instance modules, semi-lattices, Boolean algebras, convex sets, and many more. In fact, the theory of bases for vector spaces is different to others only in the sense that every vector space admits a basis, which is not the case for e.g. modules. In this paper we give compact definitions of generators and bases that subsume the well-known cases, and as a consequence allow us to lift results from one theory to the others. For example, one may wonder if there exists a matrix representation theory for convex sets that is analogous to the one of vector spaces.

In the category theoretic approach to universal algebra, algebraic structures are typically captured as algebras over a monad [2, 3]. Intuitively, monads may be seen as a generalisation of closure operators on partially ordered sets, and an algebra over a monad may be viewed as a set with an operation that allows the interpretation of formal linear combinations in a way that is coherent with the monad structure. For instance, a vector space over a field k, that is, an algebra for the free k-vector space monad, is given by a set X with a function h that coherently interprets a finitely supported X-indexed k-sequence λ as an actual linear combination $h(\lambda) = \sum_x \lambda_x \cdot x$ in X [4]. It is straightforward to see that under this perspective a basis for a vector space thus consists of a subset Y of X and a function d that assigns to a vector x in X a Y-indexed k-sequence $d(x)$ such that $h(d(x)) = x$ for all x in X and $d(h(\lambda)) = \lambda$ for all Y-indexed k-sequences λ. In other words, the restriction of h to Y-indexed k-sequences is an isomorphism with inverse d, and surjectivity corresponds to the fact that the subset Y generates the vector space, while injectivity captures that Y does so uniquely. As demonstrated in Definition 4.1, the concept easily generalises to arbitrary monads on arbitrary categories by making the subset relation explicit in form of a morphism.

Monads however not only occur in the context of universal algebra, but also play a role in algebraic topology [5] and theoretical computer science [6, 7, 8]. Among others, they are a convenient tool for capturing side-effects of coalgebraic systems [9]: popular examples include the powerset monad (non-determinism), the distribution monad (probability), and the neighbourhood monad (alternating). While coalgebraic systems with side-effects can be more compact than their deterministic counterparts, they often lack a unique minimal acceptor for a given language. For instance, every regular language admits a unique deterministic automaton with minimal state space, which can be computed via the Myhill-Nerode construction. On the other hand, for some regular languages there exist multiple non-isomorphic non-deterministic automata with minimal state space. The problem has been independently approached for different variants of side-effects, often with the common idea of restricting to a particular subclass [10, 11]. For example, for non-derministic automata, the subclass of so-called residual finite state automata has been identified as suitable. It moreover has turned out that in order to construct a unique minimal realisation in one of the subclasses it is often sufficient to derive an equivalent system with free state space from a particular given system [12]. As Arbib and Manes realised [13], instrumental to the former is what they call a *scoop*, or what we call a generator in Definition 3.1, a slight generalisation of bases. In other words, our definition of a basis for an algebra over a monad has its origin in a context that is not purely concerned with universal algebra. Throughout the paper we will value these roots by lifting results of Arbib and Manes from scoops to bases. We hope that our treatment helps to uncover ramifications between certain areas of universal algebra and the theory of coalgebras.

The paper is structured as follows. In Section 2 we recall the basic categorical notions of monads, algebras over a monad, coalgebras, distributive laws, and bialgebras. In Section 3 we introduce generators for algebras over monads and exemplify their relation with free bialgebras (Section 3.1), varieties (Section 3.2), and finitely presentable categories (Section 3.3). The definition of bases for algebras over a monad and preliminary results are covered in Section 4. Monoidal products of bases are the content of Section 4.1. In Section 4.2 we generalise the representation theory of linear maps between vector spaces to a representation theory of homomorphisms between algebras over a monad. Bases for the underlying algebra of a bialgebra and bases for the full bialgebra are considered in Section 4.3 and Section 4.4, respectively. In Section 4.5 we compare our notion of a basis with an existing coalgebraic perspective on the former. Related work and future work are discussed in Section 5 and Section 6, respectively.

2 Preliminaries

We only assume a basic knowledge of category theory, e.g. an understanding of categories, functors, natural transformations, and adjunctions. All other relevant definitions can be found in the paper. In this section, we recall monads, algebras over a monad, coalgebras, distributive laws, and bialgebras.

The concept of a monad can be traced back both to algebraic topology [5] and to an alternative to Lawvere theory as a category theoretic formulation of universal algebra [2, 3]. For an extended historical overview we refer to the survey of Hyland and Power [14]. In the context of computer science, monads have been introduced by Moggi as a general perspective on exceptions, side-effects, non-determinism, and continuations [6, 7, 8].

Definition 2.1 (Monad)**.** A *monad* on a category $\mathscr{C}$ is a tuple $\mathbb{T} = (T, \mu, \eta)$ consisting of an endofunctor $T : \mathscr{C} \to \mathscr{C}$ and natural transformations $\mu : T^2 \Rightarrow T$ and $\eta : \mathrm{id}_{\mathscr{C}} \Rightarrow T$ satisfying $\mu \circ T\mu = \mu \circ \mu_T$ and $\mu \circ \eta_T = \mathrm{id}_{\mathscr{C}} = \mu \circ T\eta$.

Below we cover some simple set-based examples: the powerset monad, the free vector space monad, the distribution monad, and the neighbourhood monad.

Examples 2.1.

- The *powerset monad* $\mathbb{P}$ on the category of sets may easily be the most well-known monad. Its underlying set endofunctor assigns to a set its subsets and to a function the function that maps a subset to its direct image under the former. The unit maps an element x in X to the singleton $\eta_X(x) = \{x\}$, and the multiplication flattens a set of subsets by taking their union, $\mu_X(U) = \bigcup_{A \in U} A$.
- The *free k-vector space monad* is an instance of the so-called multiset monad $\mathbb{M}_S$ over some semiring S on the category of sets, when S is given by the field k. The underlying set

endofunctor assigns to a set X the set of finitely-supported X-indexed sequences φ in S, typically written as formal sums $\sum_i s_i \cdot x_i$ for $s_i = \varphi(x_i)$; the unit η_X maps an element x in X to the singleton multiset $1 \cdot x$; and the multiplication μ_X satisfies $\mu_X(\sum_i s_i \cdot \varphi_i)(x) = \sum_i s_i \cdot \varphi_i(x)$ [4].

- The *distribution monad* $\mathbb{D}$ on the category of sets is given as follows. The underlying set endofunctor $\mathcal{D}$ assigns to a set X its distributions with finite support, $\mathcal{D}X = \{p : X \to [0,1] \mid \mathrm{supp}(p) \text{ finite and } \sum_{x \in X} p(x) = 1\}$ and to a function its direct image. Unit and multiplcation are given by $\eta_X(x)(y) = [x = y]$ and $\mu_X(\Phi)(x) = \sum_{p \in \mathcal{D}X} p(x) \cdot \Phi(p)$, respectively.

- The monad induced by the self-dual contravariant powerset adjunction is known under the name *neighbourhood monad* $\mathbb{H}$, since its coalgebras are related to neighbourhood frames in modal logic [15]. The underlying set endofunctor $\mathcal{H}$ satisfies $\mathcal{H}X = 2^{2^X}$ and $\mathcal{H}f(\Phi)(\varphi) = \Phi(\varphi \circ f)$. Unit and multiplication are given by $\eta_X(x)(\varphi) = \varphi(x)$ and $\mu_X(\Phi)(\varphi) = \Phi(\eta_{2^X}(\varphi))$, respectively.

A few more examples of non set-based monads are given below: the symmetric algebra monad, the nominal powerset monad, and the downset monad.

Examples 2.2.

- The *symmetric algebra monad* $\mathbb{S}$ on the category of k-vector spaces is given as follows. The underlying functor $\mathcal{S}$ assigns to a vector space V its symmetric algebra, that is $\amalg_{n \geq 0} S_n(V)$, the direct-sum of vector spaces (where $S_n(V)$ denotes the quotient of the n-fold product $V^{\otimes n}$ under the permutation action of the symmetric group of degree n); the unit injects the symmetric power $S_1(V) = V$; and the multiplication flattens tensors over tensors in the usual way.

- The *nominal powerset monad* $\mathbb{P}_{\mathrm{fs}}$ on the category of finitely-supported nominal sets and equivariant functions is the nominal equivalent of the powerset monad on the category of sets [16]. The underlying functor assigns to a nominal set X the nominal set $\mathcal{P}_{\mathrm{fs}}(X)$ of finitely-supported subsets; the equivariant unit η_X maps an element x to the singleton nominal set $\{x\}$; and the equivariant multiplication μ_X takes the union of subsets.

- The *downset monad* $\mathbb{P}_{\downarrow}$ on the category of posets is defined in the following way. The underlying endofunctor $\mathcal{P}_{\downarrow}$ assigns to a poset the inclusion-ordered poset of its downclosed [1] subsets, and to a monotone function the monotone function mapping a downclosed subset to the downclosure of its direct image. The natural transformations η and μ are given by the monotone functions $\eta_X(x) = \{x\}_{\downarrow}$ and $\mu_X(U) = \bigcup_{A \in U} A$.

If a monad results from a free-forgetful adjunction induced by some algebraic structure, the latter may be recovered in the following sense.

Definition 2.2 (Alg($\mathbb{T}$))**.** An *algebra* over a monad $\mathbb{T} = (T, \mu, \eta)$ on a category $\mathscr{C}$ is a tuple (X, h) consisting of an object X in $\mathscr{C}$ and a morphism $h : TX \to X$ such that $h \circ \mu_X = h \circ Th$ and $h \circ \eta_X = \mathrm{id}_X$. A homomorphism $f : (X, h_X) \to (Y, h_Y)$ between $\mathbb{T}$-algebras is a morphism $f : X \to Y$ such that $h_Y \circ Tf = f \circ h_X$. The category of $\mathbb{T}$-algebras and homomorphisms is denoted by Alg($\mathbb{T}$).

The canonical example for an algebra over a monad is the free $\mathbb{T}$-algebra (TX, μ_X) for any object X in $\mathscr{C}$. Below we identify the algebras over the monads given in Examples 2.1 and Examples 2.2.

Examples 2.3.

- The category Alg($\mathbb{P}$) is isomorphic to the category of complete join-semi lattices and join-preserving functions.

- The category Alg($\mathbb{M}_k$) is isomorphic to the category of k-vector spaces and linear maps.

- The category Alg($\mathbb{D}$) is isomorphic to the category of convex sets and affine functions.

- The category Alg($\mathbb{H}$) is isomorphic to the category of complete atomic Boolean algebras and Boolean algebra homomorphisms that preserve all meets and all joins [17].

[1] The *downward closure* $Y_{\downarrow}$ of a subset $Y \subseteq X$ of a poset is the set of those elements in X for which there exists at least one element in Y above. A subset is called *downset* or *downclosed*, if it coincides with its downward closure.

- The category $\mathrm{Alg}(\mathbb{S})$ is isomorphic to the category of commutative k-algebras and algebra homomorphisms [18].
- The category $\mathrm{Alg}(\mathbb{P}_{\mathrm{fs}})$ is isomorphic to the category of nominal complete join-semilattices for which the the join-operation is equivariant, and join-preserving equivariant functions [19].
- The category $\mathrm{Alg}(\mathbb{P}_{\downarrow})$ is isomorphic to $\mathrm{Alg}(\mathbb{P})$ [20].

We now turn our attention to the dual of algebras: coalgebras [9]. While algebras have been used to model finite data types, coalgebras deal with infinite data types and have turned out to be suited as an abstraction for a variety of state-based systems [21].

Definition 2.3 ($\mathrm{Coalg}(F)$)**.** A *coalgebra* for an endofunctor $F : \mathscr{C} \to \mathscr{C}$ is a tuple (X, k) consisting of an object X in $\mathscr{C}$ and a morphism $k : X \to FX$. A homomorphism $f : (X, k_X) \to (Y, k_Y)$ between F-coalgebras is a morphism $f : X \to Y$ satisfying $k_Y \circ f = Ff \circ k_X$. The category of coalgebras and homomorphisms is denoted by $\mathrm{Coalg}(F)$.

We are particularly interested in systems with side-effects, for instance non-deterministic automata. Often such systems can be realised as coalgebras for an endofunctor composed of a monad and an endofunctor similar to F satisfying $FX = X^A \times B$. In these cases the compatibility between the dynamics of the system and its side-effects can be captured by a so-called distributive law. Distributive laws have originally occurred as a way to compose monads [22], but now also exist in a wide range of other forms [23]. For our particular case it is sufficient to consider distributive laws between a monad $\mathbb{T}$ and an endofunctor F, such that $\mathbb{T}$ *distributives over* F. We follow the terminology of [24] and call a distributive law of such a form an *Eilenberg-Moore law.*

Definition 2.4 (Eilenberg-Moore law)**.** Let $\mathbb{T} = (T, \mu, \eta)$ be a monad on a category $\mathscr{C}$ and $F : \mathscr{C} \to \mathscr{C}$ an endofunctor. A natural transformation $\lambda : TF \Rightarrow FT$ is called *Eilenberg-Moore law*, if it satisfies $F\eta = \lambda \circ \eta_F$ and $F\mu \circ \lambda_T \circ T\lambda = \lambda \circ \mu_F$.

Given an Eilenberg-Moore law, it is straightforward to model the determinisation of a system with side-effects [25]. Indeed, for any morphism $f : Y \to X$ and $\mathbb{T}$-algebra (X, h), the free-algebra adjunction yields a $\mathbb{T}$-algebra homomorphism

$$f^\sharp := h \circ Tf : (TY, \mu_Y) \longrightarrow (X, h). \tag{1}$$

Thus, in particular, any FT-coalgebra $k : X \to FTX$ lifts to a $\mathbb{T}$-algebra homomorphism

$$k^\sharp := (F\mu_X \circ \lambda_{TX}) \circ Tk : (TX, \mu_X) \longrightarrow (FTX, F\mu_X \circ \lambda_{TX}). \tag{2}$$

For instance, if $\mathbb{P}$ is the powerset monad and F is the set endofunctor for deterministic automata satisfying $FX = X^A \times 2$, the disjunctive $\mathbb{P}$-algebra structure $h : \mathcal{P}2 \to 2$ with $h(\varphi) = \varphi(1)$ induces a canonical Eilenberg-Moore law [24], such that the lifting (2) is given by the classical determinisation procedure for non-deterministic automata [25].

One can show that the F-coalgebra structure $(TX, k^\sharp)$ obtained by the lifting (2) is in some sense compatible along λ with the free $\mathbb{T}$-algebra structure (TX, μ_X): it is a λ-bialgebra.

Definition 2.5 ($\mathrm{Bialg}(\lambda)$)**.** Let λ be an Eilenberg-Moore law between a monad $\mathbb{T}$ and an endofunctor F. A λ-*bialgebra* is a tuple (X, h, k) consisting of a $\mathbb{T}$-algebra (X, h) and an F-coalgebra (X, k) satisfying $Fh \circ \lambda_X \circ Tk = k \circ h$. A homomorphism $f : (X, h_X, k_X) \to (Y, h_Y, k_Y)$ between λ-bialgebras is a morphism $f : X \to Y$ that is simultaneously a $\mathbb{T}$-algebra homomorphism and an F-coalgebra homomorphism. The category of λ-bialgebras and homomorphisms is denoted by $\mathrm{Bialg}(\lambda)$.

One class of examples of bialgebras is given by determinised probabilistic automata.

Example 2.1 (Probabilistic Automata)**.** Let F be the set endofunctor satisfying $FX = X^A \times [0, 1]$. Coalgebras for $F\mathcal{D}$ are known as unpointed Rabin probabilistic automata over the alphabet A [26, 25]. The unit interval can be equipped with a $\mathbb{D}$-algebra structure h that is defined by $h(p) = \sum_{x \in [0,1]} x \cdot p(x)$ and induces a canonical Eilenberg-Moore law between $\mathbb{D}$ and F [24]. The respective bialgebras consist of unpointed Moore automata with input A and output $[0, 1]$, a convex set as state space, and affine transition and output functions. For instance, every Moore automaton derived from a probabilistic automaton by assigning to the state space of the latter all its distributions with finite support constitutes such a bialgebra.

The lifting or *expansion* in (2) can be summarised as a functor between the category of FT-coalgebras and the category of λ-bialgebras.

Lemma 2.1 ([24])**.** *Defining* $\exp(X, k) := (TX, \mu_X, F\mu_X \circ \lambda_{TX} \circ Tk)$ *and* $\exp(f) := Tf$ *yields a functor* $\exp : \mathrm{Coalg}(FT) \to \mathrm{Bialg}(\lambda)$.

Sometimes it is convenient to be able to refer to the functor which arises from the one above by precomposition with the canonical embedding of F-coalgebras into FT-coalgebras.

Corollary 2.1. *Defining* $\mathrm{free}(X, k) := (TX, \mu_X, \lambda_X \circ Tk)$ *and* $\mathrm{free}(f) := Tf$ *yields a functor* $\mathrm{free} : \mathrm{Coalg}(F) \to \mathrm{Bialg}(\lambda)$ *satisfying* $\mathrm{free}(X, k) = \exp(X, F\eta_X \circ k)$.

3 Generators for algebras

In this section we define what it means to be a generator for an algebra over a monad. Our notion coincides with what is called a *scoop* by Arbib and Manes [13]. One may argue that the morphism i in the definition below should be mono, but we choose to continue without this requirement.

Definition 3.1 (Generator [13])**.** A *generator* for a $\mathbb{T}$-algebra (X, h) on a category $\mathscr{C}$ is a tuple (Y, i, d) consisting of an object Y in $\mathscr{C}$ and a pair of morphisms

$$\begin{tikzcd} TY \arrow[r, bend left, "i^\sharp"] & X \arrow[l, bend left, "d"] \end{tikzcd} \qquad \text{satisfying} \qquad i^\sharp \circ d = \mathrm{id}_X,$$

where $i^\sharp := h \circ Ti : TY \to X$ is the unique extension of $i : Y \to X$ to a $\mathbb{T}$-algebra homomorphism.

Note that above definition is slightly stronger than just asking $i^\sharp$ to be a split-epimorphism. The latter is a *property*, while we require the additional choice of a *datum*, or witness, d. In fact, right-inverses are not necessarily unique.

Below we characterise the generators for the algebras over the set based monads of Examples 2.1.

Examples 3.1.

- (Y, i, d) is a generator for a $\mathbb{P}$-algebra $(X, \vee)$ iff for all $x \in X$ there exists a subset $d(x) \subseteq Y$ such that $x = \bigvee_{y \in d(x)} i(y)$.
- (Y, i, d) is a generator for a $\mathbb{M}_S$-algebra $(X, +, \cdot)$ iff for all $x \in X$ there exists a finitely-supported Y-indexed S-sequence $d(x) = (s_y)_{y \in Y}$, such that $x = \sum_{y \in Y} s_y \cdot i(y)$.
- (Y, i, d) is a generator for a $\mathbb{D}$-algebra $(X, +, \cdot)$ iff for all $x \in X$ there exists a finitely-supported distribution $d(x) = (\lambda_y)_{y \in Y}$ such that $x = \sum_{y \in Y} \lambda_y \cdot i(y)$.
- (Y, i, d) is a generator for a $\mathbb{H}$-algebra $(X, \vee, \wedge, \neg)$ iff for all $x \in X$ there exists a subset $d(x) \subseteq \mathcal{P}(Y)$ such that $x = \bigvee_{A \in d(x)} (\bigwedge_{y \in A} i(y) \wedge \bigwedge_{y \notin A} \neg i(y))$ for all $x \in X$.

The generators for the algebras over the non-set based monads of Examples 2.2 are identified below.

Examples 3.2.

- (Y, i, d) is a generator for a $\mathbb{S}$-algebra $(V, +, \cdot)$ iff it consists of a k-vector space and linear maps, such that for all $x \in X$ there exists a vector $d(x) \in \mathcal{S}(Y)$ such that $x = \sum_{n \geq 0} \widehat{p_n}(d(x)(n))$. where $\widehat{p_n} : S_n(Y) \to V$ is the extension of the multilinear map $p_n : Y^n \to V$ with $(y_i)_{i=0}^n \mapsto \prod_{i=0}^n i(y_i)$. This is equivalent to the existence of a set Y and function $i : Y \to V$ such that for all $x \in V$ there exists a polynomial $d(x) \in k[Y]$ such that $x = i^\sharp(d(x))$ for $i^\sharp : k[Y] \to V$ the obvious interpretation.
- (Y, i, d) is a generator for a $\mathbb{P}_{\mathrm{fs}}$-algebra iff it consists of a finitely-supported nominal set and equivariant functions turning it into a generator for the underlying $\mathbb{P}$-algebra.
- (Y, i, d) is a generator for a $\mathbb{P}_{\downarrow}$-algebra iff it consists of a partially-ordered set and monotone functions turning it into a generator for the underlying $\mathbb{P}$-algebra.

Every algebra over a monad can be generated by itself.

Lemma 3.1 ([13])**.** *$(X, \mathrm{id}_X, \eta_X)$ is a generator for any $\mathbb{T}$-algebra (X, h).*

Proof. Follows immediately from the equality $h \circ \eta_X = \mathrm{id}_X$. $\square$

For free algebras there exists a slightly more compact characterisation of generators.

Lemma 3.2. *Let $i : Y \to X$ and $d : X \to TY$ be morphisms such that $\mu_X \circ T^2 i \circ Td = \mathrm{id}_{TX}$. Then $(Y, \eta_X \circ i, \mu_Y \circ Td)$ is a generator for the free $\mathbb{T}$-algebra (TX, μ_X).*

Proof. Follows immediately from the naturality of μ, the monad law $\mu_X \circ T\eta_X = \mathrm{id}_X$, and the assumption. $\square$

Corollary 3.1. *$(X, \eta_X, \mathrm{id}_{TX})$ is a generator for the free $\mathbb{T}$-algebra (TX, μ_X).*

Proof. Using the equality $\mu_X \circ T\eta_X = \mathrm{id}_{TX}$, the claim follows from Lemma 3.2 with $i = \mathrm{id}_X$ and $d = \eta_X$. $\square$

We extend Definition 3.1 by introducing the following category.

Definition 3.2 ($\mathrm{GAlg}(\mathbb{T})$)**.** The category $\mathrm{GAlg}(\mathbb{T})$ of algebras with a generator is defined as follows:

- objects are pairs $(\mathbb{X}_\alpha, \alpha)$, where $\mathbb{X}_\alpha = (X_\alpha, h_\alpha)$ is a $\mathbb{T}$-algebra with generator $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$; and
- a morphism $(f, p) : (\mathbb{X}_\alpha, \alpha) \to (\mathbb{X}_\beta, \beta)$ consists of a $\mathbb{T}$-algebra homomorphism $f : \mathbb{X}_\alpha \to \mathbb{X}_\beta$ and a Kleisli-morphism $p : Y_\alpha \to TY_\beta$, such that the diagram below commutes:

$$\begin{tikzcd} X_\alpha \arrow[r, "f"] & X_\beta \\ TY_\alpha \arrow[u, "i_\alpha^\sharp"] \arrow[r, "p^\sharp"] & TY_\beta \arrow[u, "i_\beta^\sharp"'] \\ X_\alpha \arrow[u, "d_\alpha"] \arrow[r, "f"] & X_\beta \arrow[u, "d_\beta"'] \end{tikzcd} . \tag{3}$$

  The composition of morphisms is defined componentwise as $(g, q) \circ (f, p) := (g \circ f, q \cdot p)$, where $q \cdot p$ denotes the usual Kleisli-composition $\mu_{Y_\gamma} \circ Tq \circ p$.

As an immediate consequence we obtain the following adjoint relation with the category of Eilenberg-Moore algebras.

Lemma 3.3. *There exists a free-forgetful adjunction*

$$\begin{tikzcd} \mathrm{GAlg}(\mathbb{T}) \arrow[r, bend right, "U"'] \arrow[r, phantom, "\perp"] & \mathrm{Alg}(\mathbb{T}) \arrow[l, bend right, "F"'] \end{tikzcd} .$$

Proof. We define the forgetful functor U as projection on the first component, i.e. $U(\mathbb{X}_\alpha, \alpha) := \mathbb{X}_\alpha$ and $U(f, p) := f$. For the definition of F we propose

$$F(\mathbb{X}) := (\mathbb{X}, (X, \mathrm{id}_X, \eta_X)) \quad \text{and} \quad F(f : \mathbb{X} \to \mathbb{Y}) := (f, \eta_Y \circ f).$$

By Lemma 3.1 the definition for F is well-defined on objects; for morphisms one easily establishes (3) from the naturality of η, the monad law $\mu_Y \circ T\eta_Y = \mathrm{id}_{TY}$, and the commutativity of f with algebra structures. The compositionality of F follows analogously; preservation of identity is trivial. For the natural isomorphism

$$\mathrm{Hom}_{\mathrm{GAlg}(\mathbb{T})}(F(\mathbb{X}), (\mathbb{X}_\alpha, \alpha)) \simeq \mathrm{Hom}_{\mathrm{Alg}(\mathbb{T})}(\mathbb{X}, U(\mathbb{X}_\alpha, \alpha))$$

we propose mapping (f, p) to f, and conversely, f to $(f, d_\alpha \circ f)$. The latter is well-defined since $(d_\alpha \circ f)^\sharp \circ \eta_X = d_\alpha \circ f$ and $i_\alpha^\sharp \circ (d_\alpha \circ f)^\sharp = i_\alpha^\sharp \circ d_\alpha \circ f^\sharp = f^\sharp = f \circ (\mathrm{id}_X)^\sharp$. Composition in one of the directions trivially yields the identity; for the other direction we note that if (f, p) satisfies (3), then $p = p^\sharp \circ \eta_X = d_\alpha \circ f$. $\square$

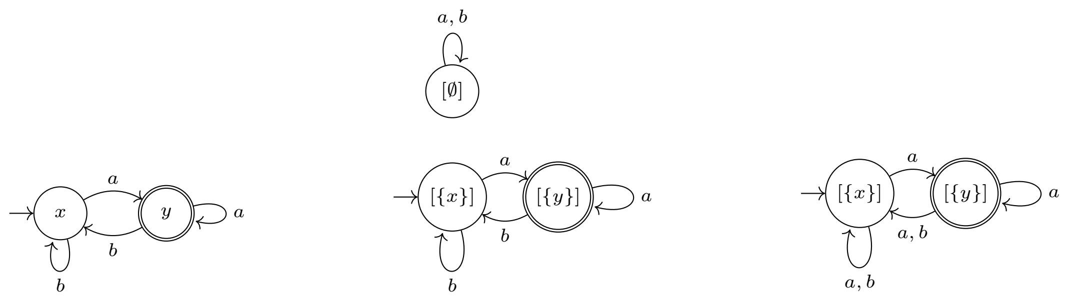

Figure 1: (a) The minimal DFA $\mathrm{M}(L)$ for $L = (a + b)^*a$; (b) The minimal CSL-structured DFA $\mathrm{M}_{\mathbb{P}}(L)$ for $L = (a + b)^*a$; (c) The canonical RFSA for $L = (a + b)^*a$.

3.1 Generators for the underlying algebra of a bialgebra

In this section we consider generators for the algebraic structure underlying a bialgebra. We are particularly interested in using the construction for an unified view on the theory of residual finite state automata and variations [27, 28, 29, 30]; more details are given in Example 3.1. The following result is a slight generalisation of a statement by Arbib and Manes [13].

Proposition 3.1 ([31])**.** *Let (X, h, k) be a λ-bialgebra and let (Y, i, d) be a generator for the $\mathbb{T}$-algebra (X, h). Then $i^\sharp : TY \to X$ is a λ-bialgebra homomorphism $i^\sharp : \exp(Y, Fd \circ k \circ i) \to (X, h, k)$.*

While non-deterministic automata can be much more succinct than their deterministic counterpart, they often lack a unique minimal acceptor for a given language, which makes them challenging to deduce with active learning algorithms. The problem is typically approached by restricting the class of non-deterministic accceptors to a subclass. For instance, residual finite state automata (RFSA) [30] are non-deterministic finite state automata (NFA) that share with deterministic finite state automata (DFA) two important properties: for any regular language they admit a unique size-minimal acceptor, called the *canonical RFSA*, and the language of each state is a residual[2] of the language of its initial state. The following example shows that Proposition 3.1 can be used to recover the canonical RFSA.

Example 3.1 (Canonical RFSA [31])**.** As before, let $\mathbb{P}$ be the powerset monad and F the set endofunctor for deterministic automata over the alphabet A satisfying $FX = X^A \times 2$. One verifies that the disjunctive $\mathbb{P}$-algebra structure $h : \mathcal{P}2 \to 2$ with $h(\varphi) = \varphi(1)$ induces a canonical Eilenberg-Moore law λ between $\mathbb{P}$ and F, such that λ-bialgebras are deterministic unpointed automata in the category of complete lattices and join-preserving functions [24].

We are particularly interested in the λ-bialgebra that is typically called the minimal $\mathbb{P}$-automaton $\mathrm{M}_{\mathbb{P}}(L)$ for a regular language L [28, 12] over an alphabet A. The bialgebra $\mathrm{M}_{\mathbb{P}}(L)$ may be recognised as an algebraic closure of the well-known minimal deterministic automaton $\mathrm{M}(L)$ for L in the category of sets and functions; formally, it is given as as the image of the unique final bialgebra homomorphism of $\mathrm{free}(\mathrm{M}(L))$. The automaton $\mathrm{M}(L)$ can be derived via the classical Myhill-Nerode construction, that is, its (necessarily finite) state space is given by the residuals of L. The (necessarily finite) state-space of $\mathrm{M}_{\mathbb{P}}(L)$ is thus given by the complete lattice that arises by taking the union-closure of residuals of L, equipped with the usual transition and output functions for languages inherited by the final coalgebra for F. For example, for the regular language $L = (a+b)^*a$ over the alphabet $\{a, b\}$ the automaton $\mathrm{M}(L)$ and the (pointed) bialgebra $\mathrm{M}_{\mathbb{P}}(L)$ are depicted in Figure 1. The latter is given the partial order generated by $[\emptyset] \leq [\{x\}] \leq [\{y\}]$.

Since the underlying lattice of $\mathrm{M}_{\mathbb{P}}(L)$ is finite, one can show that the tuple $(\mathcal{J}(\mathrm{M}_{\mathbb{P}}(L)), i, d)$, with i the subset-embedding of join-irreducibles[3] and d the function assigning to a language the

[2]A language is called a residual of $L \subseteq A^*$, if it is of the form $\{u \in A^* \mid vu \in L\}$ for some $v \in A^*$.

[3]A non-zero element x in a lattice L is called join-irreducible if for all $y, z \in L$ such that $x = y \vee z$ one finds $x = y$ or $x = z$

join-irreducible languages below it, is a generator for $\mathrm{M}_{\mathbb{P}}(L)$. Since $d \circ i^\sharp(\{[\{y\}]\}) = d([\{y\}]) = \{[\{x\}], [\{y\}]\}$, the decomposition is not unique. Writing k for the F-coalgebra structure of $\mathrm{M}_{\mathbb{P}}(L)$, it is easy to verify that the NFA-structure $Fd \circ k \circ i$ on $\mathcal{J}(\mathrm{M}_{\mathbb{P}}(L))$ mentioned in Proposition 3.1 corresponds precisely to the canonical RFSA for L [30]. For example, for $L = (a+b)^*a$, the join-irreducibles of the underlying lattice of $\mathrm{M}_{\mathbb{P}}(L)$ are easily verified to be $[\{x\}]$ and $[\{y\}]$, resulting in the canonical RFSA in Figure 1.

3.2 Signatures, equations, and finitary monads

Most of the algebras over set monads we previously considered generators for constitute finitary varieties in the sense of universal algebra, i.e. classes of algebraic structures for a finitary signature satisfying a finite set of equations. In this section, we will briefly explore the consequences for generators that arise from this observation.

Formally, a finitary signature consists of a finite set Σ, whose elements are thought of as operations, and a function $\mathrm{ar} : \Sigma \to \mathbb{N}$ that assigns to an operation its finite arity. Any signature induces a set endofunctor H_Σ defined on a set as the coproduct $H_\Sigma X = \coprod_{\sigma \in \Sigma} X^{\mathrm{ar}(\sigma)}$, and consequently, a set monad $\mathbb{S}_\Sigma$ that assigns to a set V of variables the initial algebra $S_\Sigma V = \mu X.V + H_\Sigma X$, i.e. the set of Σ-terms generated by V [32]. One can show that the categories of H_Σ-algebras and $\mathbb{S}_\Sigma$-algebras are isomorphic. We say that a $\mathbb{S}_\Sigma$-algebra $\mathbb{X}$ satisfies a finite set of equations $E \subseteq S_\Sigma V \times S_\Sigma V$, if for all $(s,t) \in E$ and valuations $v : V \to X$ it holds $v^\sharp(s) = v^\sharp(t)$, where $v^\sharp : (S_\Sigma V, \mu_V) \to \mathbb{X}$ is the unique extension of v to a $\mathbb{S}_\Sigma$-algebra homomorphism [33]. The set of $\mathbb{S}_\Sigma$-algebras that satisfy E is denoted by $\mathrm{Alg}(\Sigma, E)$. As one verifies, the forgetful functor $U : \mathrm{Alg}(\Sigma, E) \to \mathsf{Set}$ admits a left-adjoint $F : \mathsf{Set} \to \mathrm{Alg}(\Sigma, E)$, thus resulting in a set monad $\mathbb{T}_{\Sigma,E}$ with underlying endofunctor $U \circ F$ that preserves directed colimits. It further can be shown to be monadic, that is, the comparison functor $K : \mathrm{Alg}(\Sigma, E) \to \mathrm{Alg}(\mathbb{T}_{\Sigma,E})$ is an isomorphism [34]. In other words, the category of algebras over $\mathbb{T}_{\Sigma,E}$ and the finitary variety of algebras over Σ and E coincide. In fact, set monads preserving directed colimits and finitary varieties are in *bijection*. In consequence, monads of such a form are sometimes called finitary [33].

The following result characterises generators for algebras over $\mathbb{T}_{\Sigma,E}$. It can be seen as a unifying proof for the observations in Examples 3.1.

Lemma 3.4. *A morphism $i : Y \to X$ is part of a generator for a $\mathbb{T}_{\Sigma,E}$-algebra $\mathbb{X}$ iff every element of X can be expressed as a Σ-term in $i[Y]$ modulo E.*

Proof. First note that the equivalence K preserves the underlying carrier set of an algebra. Under its inverse, the $\mathbb{T}_{\Sigma,E}$-algebra homomorphism $i^\sharp$ corresponds to the function $K^{-1}(i^\sharp)$ between $\mathbb{S}_\Sigma$-algebras satisfying E inductively defined by $K^{-1}(i^\sharp)(y) = i(y)$ and $K^{-1}(i^\sharp)(\sigma(t_1, ..., t_{\mathrm{ar}(\sigma)})) = \sigma^{\mathbb{X}}(K^{-1}(i^\sharp)(t_1), ..., K^{-1}(i^\sharp)(t_{\mathrm{ar}(\sigma)}))$. In consequence, the identity $i^\sharp \circ d = \mathrm{id}_{\mathbb{X}}$ thus translates to the observation that any element $x \in K^{-1}(\mathbb{X})$ can be expressed as the Σ-term $K^{-1}(i^\sharp)(t)$ in $i[Y]$, where $t = K^{-1}(d(x))$. ◻

3.3 Finitely generated objects

In this section we shortly relate our definition of a generator to the theory of locally finitely presentable categories, in particular, to the notions of finitely generated and finitely presentable objects – categorical abstractions of finitely generated algebraic structures.

For intuition, recall that one calls an element $x \in X$ of a partially ordered set finite, if for each directed set $D \subseteq X$ with $x \leq \bigvee D$, there exists $d \in D$, such that $x \leq d$. An algebraic lattice is a partially ordered set that has all joins, and every element is a directed join of finite elements. One can show that the naive categorification of finite elements is equivalent to the following compact definition: a object Y in $\mathscr{C}$ is *finitely presentable (generated)*, if $\mathrm{Hom}_{\mathscr{C}}(Y, -) : \mathscr{C} \to \mathsf{Set}$ preserves filtered colimits (of monomorphisms). Consequently, one can categorify algebraic lattices as *locally finitely presentable* (*lfp*) categories, which are cocomplete and admit a set of finitely presentable objects, such that every object is a filtered colimit in the former [33].

The result below states that for finitary monads on lfp categories, an algebra is a finitely generated object iff it is a strong quotient of an algebra generated by a finitely presentable object.

Lemma 3.5 ([35])**.** *An algebra $\mathbb{X}$ over a finitary monad $\mathbb{T}$ on an lfp category $\mathscr{C}$ is a finitely generated object of* $\mathrm{Alg}(\mathbb{T})$ *iff there exists a finitely presentable object Y of $\mathscr{C}$ and a morphism $i : Y \to X$, such that $i^\sharp : TY \to X$ is a strong epimorphism.*

Note that a generator in the sense of Definition 3.1 requires the existence and choice of a right-inverse to $i^\sharp$, turning the latter in particular into a *split* epimorphism, while above we are only given a *strong* epimorphism. In general, splitness is a rather heavy quality, as asking for all (ordinary) epimorphisms of some category to be split is equivalent to asserting its internal axiom of choice.

4 Bases for algebras

In Section 3 we adopted the notion of a scoop by Arbib and Manes [13] by introducing generators for algebras over a monad. In this section we extend the former to the definition of a basis for an algebra over a monad by adding a uniqueness constraint.

Definition 4.1 (Basis)**.** A *basis* for a $\mathbb{T}$-algebra (X, h) on a category $\mathscr{C}$ is a tuple (Y, i, d) consisting of an object Y in $\mathscr{C}$ and a pair of morphisms

$$\begin{tikzcd} TY \arrow[r, bend left, "i^\sharp"] & X \arrow[l, bend left, "d"] \end{tikzcd} \qquad \text{satisfying} \qquad i^\sharp \circ d = \mathrm{id}_X \qquad \text{and} \qquad d \circ i^\sharp = \mathrm{id}_{TY},$$

where $i^\sharp := h \circ Ti : TY \to X$ is the unique extension of $i : Y \to X$ to a $\mathbb{T}$-algebra homomorphism.

We define the category $\mathrm{BAlg}(\mathbb{T})$ of $\mathbb{T}$-algebras with a basis as the full subcategory of $\mathrm{GAlg}(\mathbb{T})$.

Any two morphisms d_α and d_β turning (Y, i, d_α) and (Y, i, d_β) into bases for (X, h), respectively, are in fact identical: $d_\alpha = d_\alpha \circ i^\sharp \circ d_\beta = d_\beta$. In other words, the definition of basis may keep d implicit. To emphasise the similarity with generators we keep it explicit.

It is also immediate that if $|TY| = |TZ|$ implies $|Y| = |Z|$ for a set monad $\mathbb{T}$, then any two bases for a fixed $\mathbb{T}$-algebra have the same cardinality. This is in particular the case for the free vector space monad of Examples 2.1, recovering the so-called dimension theorem for vector spaces.

While every algebra over a monad admits a generator, cf. Lemma 3.1, the latter is not necessarily true for a basis. It is however not hard to see that every *free* algebra admits a basis. The next result is the analogue of Lemma 3.2.

Lemma 4.1. *Let $i : Y \to X$ and $d : X \to TY$ be morphisms such that $\mu_X \circ T^2 i \circ Td = \mathrm{id}_{TX}$ and $\mu_Y \circ Td \circ Ti = \mathrm{id}_{TY}$. Then $(Y, \eta_X \circ i, \mu_Y \circ Td)$ is a basis for the free $\mathbb{T}$-algebra (TX, μ_X).*

Proof. One part of the claim follows from Lemma 3.2. The other part follows immediately from the monad law $\mu_X \circ T\eta_X = \mathrm{id}_{TX}$ and the assumption. □

Corollary 4.1. *$(X, \eta_X, \mathrm{id}_{TX})$ is a basis for the free $\mathbb{T}$-algebra (TX, μ_X).*

Proof. Using the equality $\mu_X \circ T\eta_X = \mathrm{id}_{TX}$, the claim follows from Lemma 4.1 with $i = \mathrm{id}_X$ and $d = \eta_X$. □

The following result establishes that the morphism d is in fact an algebra homomorphism, and, intuitively, that elements of a basis are uniquely generated by their image under the monad unit, that is, typically by themselves. The former implies a mild generalisation of Corollary 4.1: an algebra admits a basis *if and only if* it is *isomorphic* to a free algebra.

Lemma 4.2. *Let (Y, i, d) be a basis for a $\mathbb{T}$-algebra (X, h). Then $\mu_Y \circ Td = d \circ h$ and $d \circ i = \eta_Y$.*

Proof. Follows from the commutativity of the following two diagrams:

$$
\begin{tikzcd}
TX \arrow[rr, "Td"] \arrow[d, "\mathrm{id}_{TX}"'] & & T^2Y \arrow[dl, "T^2 i"] \arrow[dd, "\mu_Y"] \\
TX \arrow[dd, "h"'] & T^2X \arrow[l, "Th"] \arrow[d, "\mu_X"] & \\
& TX \arrow[dl, "h"] & TY \arrow[l, "Ti"] \arrow[d, "\mathrm{id}_{TY}"] \\
X \arrow[rr, "d"] & & TY
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[r, "i"] \arrow[d, "\eta_Y"'] & X \arrow[r, "\mathrm{id}_X"] \arrow[d, "\eta_X"'] & X \arrow[ddll, bend left, "d"] \\
TY \arrow[r, "Ti"] \arrow[d, "\mathrm{id}_{TY}"'] & TX \arrow[ur, "h"] & \\
TY & &
\end{tikzcd}
.
$$

Alternatively, the first equality can be deduced from Beck's monadicity theorem: since the forgetful functor $U : \mathrm{Alg}(\mathbb{T}) \to \mathscr{C}$ is monadic, it reflects isomorphisms. □

Corollary 4.2. *A $\mathbb{T}$-algebra admits a basis if and only if it is isomorphic to a free algebra.*

Proof. Assume that a $\mathbb{T}$-algebra (X, h) admits a basis (Y, i, d). As seen in (1) the lifting $i^\sharp = h \circ Ti : (TY, \mu_Y) \to (X, h)$ is a $\mathbb{T}$-algebra homomorphism. By Lemma 4.2 the morphism d constitutes an algebra homomorphism that is inverse to $i^\sharp$. Conversely, assume $\varphi : (X, h) \to (TY, \mu_Y)$ is an isomorphism. Then $(Y, \varphi^{-1} \circ \eta_Y, \varphi)$ is a basis for (X, h). □

We give an example of a basis for the theory of monoids.

Example 4.1 (Monoids)**.** Let $\mathbb{T} = (T, \mu, \eta)$ be the monad whose underlying set endofunctor T assigns to a set X the set of all finite words over the alphabet X; whose unit η_X assigns to a character in X the corresponding word of length one; and whose multiplication μ_X syntactically flattens words over words over the alphabet X in the usual way. The monad $\mathbb{T}$ is also known as the list monad. One verifies that the constraints for its algebras correspond to the unitality and associativity laws of monoids. A function $i : Y \to X$ is thus part of a basis (Y, i, d) for a $\mathbb{T}$-algebra with underlying set X if and only if for all $x \in X$ there exists a *unique* word $d(x) = [y_1, ..., y_n]$ over the alphabet Y satisfying $x = i(y_1) \cdot ... \cdot i(y_n)$.

The result below shows that an algebra with a basis embeds into the free algebra it spans. The monomorphism is fundamental to an alternative approach to bases [36]; for more details see Section 4.5.

Corollary 4.3. *Let (Y, i, d) be a basis for a $\mathbb{T}$-algebra (X, h). Then $Ti \circ d : X \to TX$ is a $\mathbb{T}$-algebra monomorphism $Ti \circ d : (X, h) \to (TX, \mu_X)$ with left-inverse $h : (TX, \mu_X) \to (X, h)$.*

Proof. The morphism h is a $\mathbb{T}$-algebra homomorphism since the equality $h \circ \mu_X = h \circ Th$ holds for all algebras over a monad. By the definition of a generator h is a left-inverse to $Ti \circ d$. Thus, by Beck's monadicity theorem, $Ti \circ d$ is a $\mathbb{T}$-algebra homomorphism. Alternatively, observe that Lemma 4.2 and the naturality of μ imply that $Ti \circ d$ is the composition of $\mathbb{T}$-algebra homomorphisms $d : (X, h) \to (TY, \mu_Y)$ and $Ti : (TY, \mu_Y) \to (TX, \mu_X)$. The morphism $Ti \circ d$ is mono since every morphism with left-inverse is mono. □

As one would expect, every algebra homomorphism is uniquely determined by its image on a basis. Essentially, the result is a simple rephrasing of the well-known free-algebra adjunction.

Corollary 4.4. *Let (Y, i, d) be a basis for a $\mathbb{T}$-algebra (X, h_X) and (Z, h_Z) any $\mathbb{T}$-algebra. For every morphism $f : Y \to Z$ there exists a $\mathbb{T}$-algebra homomorphism $f^\sharp : (X, h_X) \to (Z, h_Z)$ and for every $\mathbb{T}$-algebra homomorphism $g : (X, h_X) \to (Z, h_Z)$ there exists a morphism $g^\flat : Y \to Z$, such that $(f^\sharp)^\flat = f$ and $(g^\flat)^\sharp = g$.*

Proof. We propose the definitions $f^\sharp := h_Z \circ Tf \circ d$ and $g^\flat := g \circ i$. The morphism $f^\sharp$ is a $\mathbb{T}$-algebra homomorphism since by Lemma 4.2 and (1), respectively, it is the composition of algebra homomorphisms $d : (X, h_X) \to (TY, \mu_Y)$ and $h_Z \circ Tf : (TY, \mu_Y) \to (Z, h_Z)$. By Lemma 4.2 it satisfies

$$(f^\sharp)^\flat = h_Z \circ Tf \circ d \circ i = h_Z \circ Tf \circ \eta_Y = h_Z \circ \eta_Z \circ f = f.$$

Conversely, the commutativity of g with algebra structures and the definition of a generator implies

$$(g^\flat)^\sharp = h_Z \circ Tg \circ Ti \circ d = g \circ h_X \circ Ti \circ d = g.$$

□

It is well-known that the Kleisli-category of any monad is equivalent to the full subcategory of free Eilenberg-Moore algebras. In Corollary 4.2 it was shown that an algebra admits a basis if and only if it is isomorphic to a free algebra. In consequence, the following result can be expected.

Lemma 4.3. *There exists a two-sided free-forgetful adjoint equivalence*

$$\begin{tikzcd} \mathrm{BAlg}(\mathbb{T}) \arrow[r, bend right, "U"'] \arrow[r, phantom, "\top \atop \bot"] & \mathrm{Kl}(\mathbb{T}) \arrow[l, bend right, "F"'] \end{tikzcd}.$$

Proof. We define the forgetful functor U by $U(\mathbb{X}_\alpha, \alpha) := Y_\alpha$ and $U(f, p) := p$. Its well-definedness is immediate. For F we propose the definitions

$$F(Y) := ((TY, \mu_Y), (Y, \eta_Y, \mathrm{id}_{TY})) \quad \text{and} \quad F(p) := (p^\sharp, p).$$

By Corollary 4.1 the definition is well-defined on objects; for morphisms one verifies that (3) is an immediate consequence of the identity $\eta_Y^\sharp = \mathrm{id}_{TY}$. Preservation of identities follows by the same argument, and compositionality is a consequence of $(q \cdot p)^\sharp = q^\sharp \circ p^\sharp$. For the natural isomorphism

$$\mathrm{Hom}_{\mathrm{BAlg}(\mathbb{T})}(F(Y), (\mathbb{X}_\alpha, \alpha)) \simeq \mathrm{Hom}_{\mathrm{Kl}(\mathbb{T})}(Y, U(\mathbb{X}_\alpha, \alpha))$$

we propose mapping (f, p) to p, and conversely, p to $(i_\alpha^\sharp \circ p^\sharp, p)$. Clearly $i_\alpha^\sharp \circ p^\sharp$ is a $\mathbb{T}$-algebra homomorphism. Equation (3) is a consequence of the definition of a basis, $d_\alpha \circ i_\alpha^\sharp \circ p^\sharp = p^\sharp$ and $(\eta_Y)^\sharp = \mathrm{id}_{TY}$. Composition of the two maps in one of the directions yields trivially the identity; for the other direction we observe that if (f, p) satisfies (3), then it immediately follows $f = i_\alpha^\sharp \circ p^\sharp$. For the natural isomorphism

$$\mathrm{Hom}_{\mathrm{BAlg}(\mathbb{T})}((\mathbb{X}_\alpha, \alpha), F(Y)) \simeq \mathrm{Hom}_{\mathrm{Kl}(\mathbb{T})}(U(\mathbb{X}_\alpha, \alpha), Y)$$

we propose sending (f, p) to p, and conversely, p to $(p^\sharp \circ d_\alpha, p)$. In essence, $p^\sharp \circ d_\alpha$ is a $\mathbb{T}$-algebra homomorphism by Lemma 4.2, and the tuple $(p^\sharp \circ d_\alpha, p)$ satisfies (3) because $p^\sharp \circ d_\alpha \circ i_\alpha^\sharp = p^\sharp$. Composition of the two maps in one of the directions again trivially yields the identity; for the other direction we observe that if (f, p) satisfies (3), then immediately $f = p^\sharp \circ d_\alpha$. □

4.1 Product of bases

In this section we show that, under certain assumptions, the monoidal product of a base category naturally extends to a product of algebras with bases within the former. As a natural example we obtain the tensor-product of vector spaces with fixed bases.

We assume basic familiarity with monoidal categories. A monoidal monad $\mathbb{T}$ on a monoidal category $(\mathscr{C}, \otimes, I)$ is a monad which is equipped with natural transformations $T_{X,Y} : TX \otimes TY \to T(X \otimes Y)$ and $T_0 : I \to TI$, satisfying certain coherence conditions. One can show that, given such additional data, the monoidal structure of $\mathscr{C}$ induces a monoidal category $(\mathrm{Alg}(\mathbb{T}), \boxtimes, (TI, \mu_I))$, if the following two assumptions are given [37, Corollary 2.5.6]:

(A1) for any two algebras $\mathbb{X}_\alpha = (X_\alpha, h_\alpha)$ and $\mathbb{X}_\beta = (X_\beta, h_\beta)$ the coequaliser $q_{\mathbb{X}_\alpha, \mathbb{X}_\beta}$ of the algebra homomorphisms $T(h_\alpha \otimes h_\beta)$ and $\mu_{X_\alpha \otimes X_\beta} \circ T(T_{X_\alpha, X_\beta})$ exists (we denote its codomain by $\mathbb{X}_\alpha \boxtimes \mathbb{X}_\beta = (X_\alpha \boxtimes X_\beta, h_{\alpha \boxtimes \beta})$);

(A2) left and right-tensoring with the induced functor $\boxtimes$ preserves reflexive coequaliser.

The two monoidal products $\otimes$ and $\boxtimes$ are related via the natural embedding $q_{\mathbb{X}_\alpha, \mathbb{X}_\beta} \circ \eta_{X_\alpha \otimes X_\beta}$, in the following referred to as $\iota_{\mathbb{X}_\alpha, \mathbb{X}_\beta}$. One can prove that the product $TY_\alpha \boxtimes TY_\beta$ is given by $T(Y_\alpha \otimes Y_\beta)$ and the coequaliser q_{TY_α, TY_β} by $\mu_{Y_\alpha \otimes Y_\beta} \circ T(T_{Y_\alpha, Y_\beta})$, where we abbreviate the free algebra (TY, μ_Y) as TY [37].

With the previous remarks in mind, we are able to claim the following product construction.

Lemma 4.4. *Let $\mathbb{T}$ be a monoidal monad on $(\mathscr{C}, \otimes, I)$ such that (A1) and (A2) are satisfied. Let $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$ and $\beta = (Y_\beta, i_\beta, d_\beta)$ be generators (bases) for $\mathbb{T}$-algebras $\mathbb{X}_\alpha$ and $\mathbb{X}_\beta$. Then $\alpha \boxtimes \beta = (Y_\alpha \otimes Y_\beta, \iota_{\mathbb{X}_\alpha,\mathbb{X}_\beta} \circ (i_\alpha \otimes i_\beta), (d_\alpha \boxtimes d_\beta))$ is a generator (basis) for the $\mathbb{T}$-algebra $\mathbb{X}_\alpha \boxtimes \mathbb{X}_\beta$.*

Proof. First, we note the equalities $Tf \boxtimes Tg = T(f \otimes g)$ and $q_{\mathbb{X}_\alpha,\mathbb{X}_\beta} = h_\alpha \boxtimes h_\beta$ [37]. Second, we calculate

$$\begin{aligned} &h_\alpha \boxtimes h_\beta = h_\alpha \boxtimes h_\beta \circ \mu_{X_\alpha \otimes X_\beta} \circ T(\eta_{X_\alpha \otimes X_\beta}) = h_{\alpha \boxtimes \beta} \circ T(h_\alpha \boxtimes h_\beta) \circ T(\eta_{X_\alpha \otimes X_\beta}) \\ = {}& h_{\alpha \boxtimes \beta} \circ T(\iota_{\mathbb{X}_\alpha,\mathbb{X}_\beta}). \end{aligned}$$

Finally, if α and β are generators, we compute

$$\begin{aligned} &h_{\alpha \boxtimes \beta} \circ T(\iota_{\mathbb{X}_\alpha,\mathbb{X}_\beta}) \circ T(i_\alpha \otimes i_\beta) \circ (d_\alpha \boxtimes d_\beta) = (h_\alpha \boxtimes h_\beta) \circ (T(i_\alpha) \boxtimes T(i_\beta)) \circ (d_\alpha \boxtimes d_\beta) \\ = {}& (h_\alpha \circ T(i_\alpha) \circ d_\alpha) \boxtimes (h_\beta \circ T(i_\beta) \circ d_\beta) = \mathrm{id}_{X_\alpha} \boxtimes \mathrm{id}_{X_\beta} = \mathrm{id}_{X_\alpha \boxtimes X_\beta}. \end{aligned}$$

The additional equality for the case in which α and β are bases follows analogously. □

Corollary 4.5. *Let $\mathbb{T}$ be a monoidal monad on $(\mathscr{C}, \otimes, I)$ such that (A1) and (A2) are satisfied. The definitions $(\mathbb{X}_\alpha, \alpha) \boxtimes (\mathbb{X}_\beta, \beta) := (\mathbb{X}_\alpha \boxtimes \mathbb{X}_\beta, \alpha \boxtimes \beta)$ and $(f, p) \boxtimes (g, q) := (f \boxtimes g, T_{Y_{\alpha'}, Y_{\beta'}} \circ (p \otimes q))$ yield monoidal structures with unit $((TI, \mu_I), (I, \eta_I, \mathrm{id}_{TI}))$ on $\mathrm{GAlg}(\mathbb{T})$ and $\mathrm{BAlg}(\mathbb{T})$.*

Proof. By Lemma 4.4 the construction is well-defined on objects. Its well-definedness on morphisms, i.e. the commutativity of (3), is a consequence of the previously mentioned equalities given in [37], which imply $(T_{Y_\alpha,Y_\beta} \circ (p \otimes q))^\sharp = (\mu_{Y_{\alpha'}} \boxtimes \mu_{Y_{\beta'}}) \circ (Tp \boxtimes Tq)$. The natural isomorphisms underlying the monoidal structure for $\mathrm{Alg}(\mathbb{T})$ can be extended to morphisms in $\mathrm{GAlg}(\mathbb{T})$ by associating canonical Kleisli-morphisms between generators as in (4). □

We conclude by instantiating above construction to the familiar setting of vector spaces.

Example 4.2 (Tensor product of vector spaces)**.** Recall the free k-vector space monad $\mathbb{M}_k$ introduced in Examples 2.1. The category of sets is monoidal (in fact, cartesian) with respect to the cartesian product $\times$ and the singleton set $\{\star\}$. The monad $\mathbb{M}_k$ is monoidal when equipped with the morphisms defined by $(\mathbb{M}_k)_{X,Y}(\varphi, \psi)(x, y) := \varphi(x) \cdot \psi(y)$ and $T_0(\star)(\star) = 1_k$ [38]. The category of $\mathbb{M}_k$-algebras is isomorphic to the category of k-vector spaces, and satisfies (A1) and (A2). The monoidal structure induced by $\mathbb{M}_k$ is the usual tensor product $\otimes$ of vector spaces with the unit field $\mathbb{M}_k(\{\star\}) \simeq k$. Lemma 4.4 captures the well-known fact that the dimension of the tensor product of two vector spaces equals the product of the dimensions of each vector space. The structure maps of the product generator map an element $\underline{(y_\alpha, y_\beta)}$ to the vector $i(y_\alpha) \otimes i(y_\beta)$, and a vector x to the function $(d_\alpha \otimes d_\beta)(x)$, where $d_\alpha \otimes d_\beta = \overline{d_\alpha \times d_\beta} : \mathbb{X}_\alpha \otimes \mathbb{X}_\beta \to (\mathbb{M}_k(Y_\alpha), \mu_{Y_\alpha}) \otimes (\mathbb{M}_k(Y_\beta), \mu_{Y_\beta}) \simeq (\mathbb{M}_k(Y_\alpha \times Y_\beta), \mu_{Y_\alpha \otimes Y_\beta})$ is the unique linear extension of the bilinear map defined by $(d_\alpha \times d_\beta)(x_\alpha, x_\beta)(y_\alpha, y_\beta) = d_\alpha(x_\alpha)(y_\alpha) \cdot d_\beta(x_\beta)(y_\beta)$.

4.2 Representation theory

In this section we use our general definition of a basis to derive a representation theory for homomorphisms between algebras over monads that is analogous to the representation theory for linear transformations between vector spaces.

In more detail, recall that a linear transformation $L : V \to W$ between k-vector spaces with finite bases $\alpha = \{v_1, ..., v_n\}$ and $\beta = \{w_1, ..., w_m\}$, respectively, admits a matrix representation $L_{\alpha\beta} \in \mathrm{Mat}_k(m, n)$ with

$$L(v_j) = \sum_i (L_{\alpha\beta})_{i,j} w_i,$$

such that for any vector v in V the coordinate vectors $L(v)_\beta \in k^m$ and $v_\alpha \in k^n$ satisfy the equality

$$L(v)_\beta = L_{\alpha\beta} v_\alpha.$$

A great amount of linear algebra is concerned with finding bases such that the corresponding matrix representation is in an efficient shape, for instance diagonalised. The following definitions generalise the situation by substituting Kleisli morphisms for matrices.

$$A = L_{\alpha'\alpha'} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \qquad L_{\alpha\alpha} = \begin{pmatrix} 3 & 2 \\ -5 & -3 \end{pmatrix}, \qquad P = \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}, \qquad P^{-1} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}$$

Figure 2: The basis representation of the counter-clockwise rotation by 90 degree $L : \mathbb{R}^2 \to \mathbb{R}^2$, $L(v) = Av$ with respect to $\alpha = \{(1,2),(1,1)\}$ and $\alpha' = \{(1,0),(0,1)\}$ satisfies $L_{\alpha'\alpha'} = P^{-1} L_{\alpha\alpha} P$.

Definition 4.2. Let $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$ and $\beta = (Y_\beta, i_\beta, d_\beta)$ be bases for $\mathbb{T}$-algebras $\mathbb{X}_\alpha = (X_\alpha, h_\alpha)$ and $\mathbb{X}_\beta = (X_\beta, h_\beta)$, respectively. The *basis representation* of a $\mathbb{T}$-algebra homomorphism $f : \mathbb{X}_\alpha \to \mathbb{X}_\beta$ with respect to α and β is the composition

$$f_{\alpha\beta} := Y_\alpha \xrightarrow{i_\alpha} X_\alpha \xrightarrow{f} X_\beta \xrightarrow{d_\beta} TY_\beta. \tag{4}$$

Conversely, the morphism *associated* with a Kleisli morphism $p : Y_\alpha \to TY_\beta$ with respect to α and β is the composition

$$p^{\alpha\beta} := X_\alpha \xrightarrow{d_\alpha} TY_\alpha \xrightarrow{Tp} T^2 Y_\beta \xrightarrow{\mu_{Y_\beta}} TY_\beta \xrightarrow{Ti_\beta} TX_\beta \xrightarrow{h_\beta} X_\beta. \tag{5}$$

Since for bases the structure morphisms $i^\sharp$ and d are algebra homomorphisms, above definitions in essence arise from the composition of the following isomorphisms:

$$\mathrm{Hom}_{\mathrm{Alg}(\mathbb{T})}(\mathbb{X}_\alpha, \mathbb{X}_\beta) \overset{(d_\beta)_* \circ (i_\alpha^\sharp)^*}{\simeq} \mathrm{Hom}_{\mathrm{Alg}(\mathbb{T})}((TY_\alpha, \mu_{Y_\alpha}), (TY_\beta, \mu_{Y_\beta})) \overset{(\eta_Y)^*}{\simeq} \mathrm{Hom}_{\mathrm{Kl}(\mathbb{T})}(Y_\alpha, Y_\beta). \tag{6}$$

The morphism associated with a Kleisli morphism should be understood as the linear transformation between vector spaces induced by some matrix of the right type. The following result confirms this intuition.

Lemma 4.5. *(5) is a $\mathbb{T}$-algebra homomorphism $p^{\alpha\beta} : \mathbb{X}_\alpha \to \mathbb{X}_\beta$.*

Proof. Follows immediately from (6).

□

The next result establishes a generalisation of the observation that for fixed bases, constructing a matrix representation of a linear transformation on the one hand, and associating a linear transformation to a matrix of the right type on the other hand, are mutually inverse operations.

Lemma 4.6. *The operations (4) and (5) are mutually inverse.*

Proof. Follows immediately from (6). □

At the beginning of this section we recalled the soundness identity $L(v)_\beta = L_{\alpha\beta} v_\alpha$ for the matrix representation $L_{\alpha\beta}$ of a linear transformation L. The next result is a natural generalisation of this statement.

Lemma 4.7. *$f_{\alpha\beta}$ is the unique Kleisli-morphism such that $f_{\alpha\beta} \cdot d_\alpha = d_\beta \circ f$. Conversely, $p^{\alpha\beta}$ is the unique $\mathbb{T}$-algebra homomorphism such that $p \cdot d_\alpha = d_\beta \circ p^{\alpha\beta}$.*

Proof. The definitions imply

$$f_{\alpha\beta} \cdot d_\alpha = \mu_{Y_\beta} \circ T(d_\beta \circ f \circ i_\alpha) \circ d_\alpha.$$

Using Lemma 4.2 we deduce the commutativity of the diagram below:

$$\begin{tikzcd}
X_\alpha \arrow[r, "d_\alpha"] \arrow[d, "\mathrm{id}_{X_\alpha}"'] & TY_\alpha \arrow[r, "Ti_\alpha"] & TX_\alpha \arrow[r, "Tf"] \arrow[dll, "h_\alpha"] & TX_\beta \arrow[r, "Td_\beta"] \arrow[dll, "h_\beta"] & T^2 Y_\beta \arrow[d, "\mu_{Y_\beta}"] \\
X_\alpha \arrow[r, "f"'] & X_\beta \arrow[rrr, "d_\beta"'] & & & TY_\beta
\end{tikzcd}.$$

Since an equality of the type $p \cdot d_\alpha = d_\beta \circ f$ implies

$$p = \mu_{Y_\beta} \circ \eta_{TY_\beta} \circ p = \mu_{Y_\beta} \circ Tp \circ \eta_{Y_\alpha} = \mu_{Y_\beta} \circ Tp \circ d_\alpha \circ i_\alpha = d_\beta \circ f \circ i_\alpha = f_{\alpha\beta},$$

the morphism $f_{\alpha\beta}$ is moreover uniquely determined. For the second part of the claim we observe on the one hand that by above and Lemma 4.6 it holds $p \cdot d_\alpha = (p^{\alpha\beta})_{\alpha\beta} \cdot d_\alpha = d_\beta \circ p^{\alpha\beta}$, and on the other hand, that an equality of the type $p \cdot d_\alpha = d_\beta \circ f$ implies

$$p^{\alpha\beta} = i_\beta^\sharp \circ (p \cdot d_\alpha) = i_\beta^\sharp \circ d_\beta \circ f = f.$$

□

The next result establishes the compositionality of basis representations: the matrix representation of the composition of two linear transformations is given by the multiplication of the matrix representations of the individual linear transformations.

Lemma 4.8. $g_{\beta\gamma} \cdot f_{\alpha\beta} = (g \circ f)_{\alpha\gamma}$.

Proof. The definitions imply

$$\begin{aligned} g_{\beta\gamma} \cdot f_{\alpha\beta} &= \mu_{Y_\gamma} \circ T(d_\gamma \circ g \circ i_\beta) \circ d_\beta \circ f \circ i_\alpha \\ (g \circ f)_{\alpha\gamma} &= d_\gamma \circ (g \circ f) \circ i_\alpha. \end{aligned}$$

We delete common terms and use Lemma 4.2 to deduce the commutativity of the diagram below:

$$\begin{tikzcd}
X_\beta \arrow[r, "d_\beta"] \arrow[d, "\mathrm{id}_{X_\beta}"'] & TY_\beta \arrow[r, "Ti_\beta"] & TX_\beta \arrow[r, "Tg"] \arrow[dll, "h_\beta"] & TX_\gamma \arrow[r, "Td_\gamma"] \arrow[dll, "h_\gamma"] & T^2Y_\gamma \arrow[d, "\mu_{Y_\gamma}"] \\
X_\beta \arrow[r, "g"'] & X_\gamma \arrow[rrr, "d_\gamma"'] & & & TY_\gamma
\end{tikzcd}.$$

□

Similarly to the previous result, the next observation captures the compositionality of the operation that assigns to a Kleisli morphism its associated homomorphism.

Lemma 4.9. $q^{\beta\gamma} \circ p^{\alpha\beta} = (q \cdot p)^{\alpha\gamma}$.

Proof. The definitions imply

$$\begin{aligned} q^{\beta\gamma} \circ p^{\alpha\beta} &= (h_\gamma \circ Ti_\gamma \circ \mu_{Y_\gamma} \circ Tq \circ d_\beta) \circ (h_\beta \circ Ti_\beta \circ \mu_{Y_\beta} \circ Tp \circ d_\alpha) \\ (q \cdot p)^{\alpha\gamma} &= h_\gamma \circ Ti_\gamma \circ \mu_{Y_\gamma} \circ T\mu_{Y_\gamma} \circ T^2 q \circ Tp \circ d_\alpha. \end{aligned}$$

By deleting common terms and using the equality $d_\beta \circ h_\beta \circ Ti_\beta = \mathrm{id}_{TY_\beta}$ it is thus sufficient to show

$$\mu_{Y_\gamma} \circ Tq \circ \mu_{Y_\beta} = \mu_{Y_\gamma} \circ T\mu_{Y_\gamma} \circ T^2 q.$$

Above equation follows from the commutativity of the diagram below:

$$\begin{tikzcd}
T^2Y_\beta \arrow[r, "\mu_{Y_\beta}"] \arrow[d, "T^2q"'] & TY_\beta \arrow[r, "Tq"] & T^2Y_\gamma \arrow[d, "\mu_{Y_\gamma}"] \\
T^3Y_\gamma \arrow[r, "T\mu_{Y_\gamma}"'] \arrow[urr, "\mu_{TY_\gamma}"] & T^2Y_\gamma \arrow[r, "\mu_{Y_\gamma}"'] & TY_\gamma
\end{tikzcd}.$$

□

The previous statements may be summarised in terms of functors between the following two categories, which arise from the usual Eilenberg-Moore and Kleisli categories.

Definition 4.3 ($\mathrm{Alg}_\mathrm{B}(\mathbb{T})$ and $\mathrm{Kl}_\mathrm{B}(\mathbb{T})$)**.** Let $\mathrm{Alg}_\mathrm{B}(\mathbb{T})$ be the category defined as follows:

- objects are given by pairs $(\mathbb{X}_\alpha, \alpha)$, where $\mathbb{X}_\alpha$ is a $\mathbb{T}$-algebra with basis $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$; and
- a morphism $f : (\mathbb{X}_\alpha, \alpha) \to (\mathbb{X}_\beta, \beta)$ consists of a $\mathbb{T}$-algebra homomorphism $f : \mathbb{X}_\alpha \to \mathbb{X}_\beta$.

Similarly, let $\mathrm{Kl}_{\mathrm{B}}(\mathbb{T})$ be the category defined as follows:

- objects are given by pairs (Y_α, α), where $\alpha = (Y_\alpha, i_\alpha, d_\alpha)$ is a basis for some $\mathbb{T}$-algebra $\mathbb{X}_\alpha$; and
- a morphism $p : (Y_\alpha, \alpha) \to (Y_\beta, \beta)$ consists of a morphism $p : Y_\alpha \to TY_\beta$; the composition is given by the usual Kleisli-composition.

Corollary 4.6. *There exist the following isomorphisms of categories:*

$$\mathrm{BAlg}(\mathbb{T}) \simeq \mathrm{Alg}_{\mathrm{B}}(\mathbb{T}) \simeq \mathrm{Kl}_{\mathrm{B}}(\mathbb{T}).$$

Proof. For the first isomorphism we define a functor $F : \mathrm{BAlg}(\mathbb{T}) \to \mathrm{Alg}_{\mathrm{B}}(\mathbb{T})$ by $F(\mathbb{X}_\alpha, \alpha) = (\mathbb{X}_\alpha, \alpha)$ and $F(f, p) = f$; and a functor $G : \mathrm{Alg}_{\mathrm{B}}(\mathbb{T}) \to \mathrm{BAlg}(\mathbb{T})$ by $G(\mathbb{X}_\alpha, \alpha) = (\mathbb{X}_\alpha, \alpha)$ and $G(f) := (f, f_{\alpha\beta})$. Well-definedness and mutual invertibility are consequences of Lemma 4.8, and Lemma 4.7, respectively. For the second isomorphism we define a functor $F : \mathrm{Alg}_{\mathrm{B}}(\mathbb{T}) \to \mathrm{Kl}_{\mathrm{B}}(\mathbb{T})$ by $F(\mathbb{X}_\alpha, \alpha) = (Y_\alpha, \alpha)$ and $Ff = f_{\alpha\beta}$; and a functor $G : \mathrm{Kl}_{\mathrm{B}}(\mathbb{T}) \to \mathrm{Alg}_{\mathrm{B}}(\mathbb{T})$ by $G(Y_\alpha, \alpha) = (\mathbb{X}_\alpha, \alpha)$ and $Gp = p^{\alpha\beta}$. Well-definedness and mutual invertibility are consequences of Lemma 4.8, Lemma 4.9, and Lemma 4.6, respectively. ☐

Assume we are given bases α, α' and β, β' for $\mathbb{T}$-algebras (X_α, h_α) and (X_β, h_β), respectively. The following result clarifies how the two basis representations $f_{\alpha\beta}$ and $f_{\alpha'\beta'}$ are related.

Proposition 4.1. *There exist Kleisli isomorphisms p and q such that $f_{\alpha'\beta'} = q \cdot f_{\alpha\beta} \cdot p$.*

Proof. The Kleisli morphisms p and q and their respective candidates for inverses p^{-1} and q^{-1} are defined below

$$\begin{aligned} p &:= d_\alpha \circ i_{\alpha'} : Y_{\alpha'} \longrightarrow TY_\alpha & \qquad q &:= d_{\beta'} \circ i_\beta : Y_\beta \longrightarrow TY_{\beta'} \\ p^{-1} &:= d_{\alpha'} \circ i_\alpha : Y_\alpha \longrightarrow TY_{\alpha'} & \qquad q^{-1} &:= d_\beta \circ i_{\beta'} : Y_{\beta'} \longrightarrow TY_\beta. \end{aligned}$$

From Lemma 4.2 it follows that the diagram below commutes:

$$\begin{tikzcd}
Y_\alpha \arrow[rr, "i_\alpha"] \arrow[dd, "\eta_{Y_\alpha}"'] & & X_\alpha \arrow[rr, "d_{\alpha'}"] \arrow[d, "\mathrm{id}_{X_\alpha}"] & & TY_{\alpha'} \arrow[dd, "Ti_{\alpha'}"] \\
& & X_\alpha \arrow[dll, "d_\alpha"] & & \\
TY_\alpha & & T^2Y_\alpha \arrow[ll, "\mu_{Y_\alpha}"] & & TX_\alpha \arrow[ll, "Td_\alpha"] \arrow[ull, "h_\alpha"']
\end{tikzcd} .$$

This shows that p^{-1} is a Kleisli right-inverse of p. A symmetric version of above diagram shows that p^{-1} is also a Kleisli left-inverse of p. Analogously it follows that q^{-1} is a Kleisli inverse of q.

The definitions further imply the equalities

$$\begin{aligned} q \cdot f_{\alpha\beta} \cdot p &= \mu_{Y_{\beta'}} \circ T(d_{\beta'} \circ i_\beta) \circ \mu_{Y_\beta} \circ T(d_\beta \circ f \circ i_\alpha) \circ d_\alpha \circ i_{\alpha'} \\ f_{\alpha'\beta'} &= d_{\beta'} \circ f \circ i_{\alpha'}. \end{aligned}$$

We delete common terms and use Lemma 4.2 to establish the commutativity of the diagram below:

$$\begin{tikzcd}
X_\alpha \arrow[r, "d_\alpha"] \arrow[dd, "f"'] \arrow[dr, "\mathrm{id}_{X_\alpha}"] & TY_\alpha \arrow[r, "Ti_\alpha"] & TX_\alpha \arrow[r, "Tf"] \arrow[dl, "h_\alpha"'] & TX_\beta \arrow[d, "Td_\beta"] \arrow[ddll, "h_\beta"] \\
& X_\alpha \arrow[d, "f"] & & T^2Y_\beta \arrow[dd, "\mu_{Y_\beta}"] \\
X_\beta \arrow[d, "d_{\beta'}"'] & X_\beta \arrow[l, "\mathrm{id}_{X_\beta}"'] \arrow[drr, "d_\beta"] & & \\
TY_{\beta'} & T^2Y_{\beta'} \arrow[l, "\mu_{Y_{\beta'}}"] & TX_\beta \arrow[l, "Td_{\beta'}"] \arrow[ull, "h_\beta"'] & TY_\beta \arrow[l, "Ti_\beta"]
\end{tikzcd} .$$

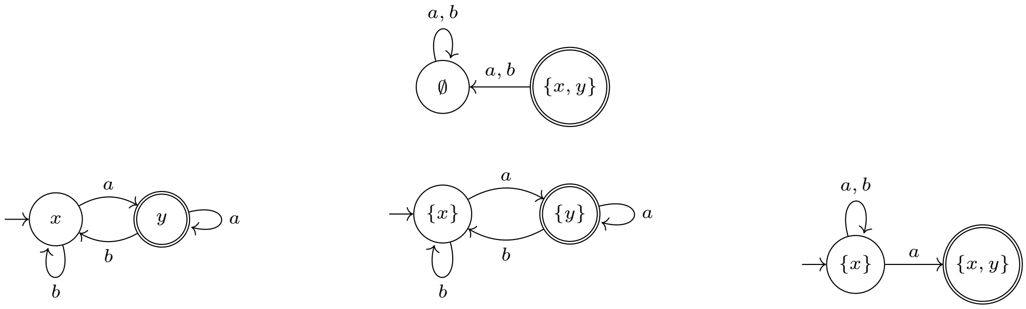

Figure 3: (a) The minimal DFA $\mathrm{M}(L)$ for $L = (a + b)^*a$; (b) The minimal $\mathbb{Z}_2$-Vect structured DFA $\mathrm{M}_{\mathbb{X}}(L)$ for $L = (a + b)^*a$; (c) The minimal xor automaton for $L = (a + b)^*a$.

□

Above result simplifies in the case one restricts to an endomorphism: the respective basis representations are *similar*. This generalises the situation for vector spaces, cf. Figure 2.

Corollary 4.7. *There exists a Kleisli isomorphism p with Kleisli inverse p^{-1} such that $f_{\alpha'\alpha'} = p^{-1} \cdot f_{\alpha\alpha} \cdot p$.*

Proof. In Proposition 4.1 let $\beta = \alpha$ and $\beta' = \alpha'$. One verifies that in the corresponding proof the definitions of the morphisms p^{-1} and q coincide. □

4.3 Bases for the underlying algebra of a bialgebra

In Proposition 3.1 we have seen that a generator for the underlying algebra of a bialgebra allows the construction of a semantically equivalent bialgebra with free state space. As the result below shows, in case one is given a basis, the respective bialgebras are not only semantically equivalent, but in fact isomorphic.

Lemma 4.10 ([31])**.** *Let (X, h, k) be a λ-bialgebra and let (Y, i, d) be a basis for the $\mathbb{T}$-algebra (X, h). Then $i^\sharp : \exp(Y, Fd \circ k \circ i) \to (X, h, k)$ is a λ-bialgebra isomorphism with inverse d.*

Above result can be exemplified by another well-known canonical acceptor for a regular language: the so-called *minimal xor automaton* [39]. While the canonical RFSA in Example 3.1 is size-minimal among residual finite state automata, the minimal xor automaton is size-minimal among $\mathbb{Z}_2$-weighted finite state automata (also called xor automata), where $\mathbb{Z}_2$ denotes the unique two element field. It is never larger than the corresponding minimal DFA and can be efficiently constructed, making it computationally interesting.

Example 4.3 (Minimal xor automaton [31])**.** Let $\mathbb{X} := \mathbb{M}_{\mathbb{Z}_2}$ be the free vector space monad of Examples 2.1 for the unique two element field $\mathbb{Z}_2$, and F the set endofunctor for deterministic automata over the alphabet A satisfying $FX = X^A \times 2$. The $\mathbb{X}$-algebra structure $h : \mathcal{X}2 \to 2$ with $h(\varphi) = \varphi(1)$ induces a canonical Eilenberg-Moore law λ between $\mathbb{X}$ and F, such that λ-bialgebras are deterministic unpointed automata in the category of $\mathbb{Z}_2$-vector spaces and linear maps.

Analogously to Example 3.1, we consider the minimal pointed λ-bialgebra $\mathrm{M}_{\mathbb{X}}(L)$ accepting L. Formally, it can be obtained as the image of the final λ-bialgebra homomorphism of $\mathrm{free}(\mathrm{M}(L))$, where $\mathrm{M}(L)$ is the minimal DFA accepting L. The $\mathbb{Z}_2$-vector space structure of $\mathrm{M}_{\mathbb{X}}(L)$ is given by the closure of the derivatives of L under symmetric-difference. For example, for $L = (a+b)^*a$ over the alphabet $\{a, b\}$, the DFA $\mathrm{M}(L)$ and the (x-pointed) bialgebra $\mathrm{M}_{\mathbb{X}}(L)$ are depicted in Figure 3.

Let $B = (Y, i, d)$ denote a basis for the underlying $\mathbb{X}$-algebra structure of $\mathrm{M}_{\mathbb{X}}(L)$. It is straight forward to verify that the ($d(x)$-pointed) $F\mathbb{X}$-coalgebra $(Y, Fd \circ k \circ i)$ in Lemma 4.10 is precisely the minimal xor automaton for L with respect to B. For example, for $L = (a+b)^*a$ one can choose the basis B with $Y = \{\{x\}, \{x, y\}\}$, i the embedding, and d defined by $d(\emptyset) := \emptyset$, $d(\{x\}) := \{\{x\}\}$, $d(\{y\}) := \{\{x\}, \{x, y\}\}$, and $d(\{x, y\}) := \{\{x, y\}\}$. The corresponding minimal xor automaton is depicted in Figure 3.

4.4 Bases for bialgebras

In the last subsection we considered bases for the underlying algebra structure of a bialgebra. This subsection is concerned with generators and bases for the *full* bialgebra. More formally, it is well-known [40] that an Eilenberg-Moore law λ between a monad $\mathbb{T}$ and an endofunctor F induces simultaneously

- a monad $\mathbb{T}_\lambda = (T_\lambda, \mu, \eta)$ on $\mathrm{Coalg}(F)$ by $T_\lambda(X,k) = (TX, \lambda_X \circ Tk)$ and $T_\lambda f = Tf$; and
- an endofunctor F_λ on $\mathrm{Alg}(\mathbb{T})$ by $F_\lambda(X,h) = (FX, Fh \circ \lambda_X)$ and $F_\lambda f = Ff$,

such that the algebras over $\mathbb{T}_\lambda$, the coalgebras of F_λ, and λ-bialgebras coincide. We will consider generators and bases for $\mathbb{T}_\lambda$-algebras, or equivalently, λ-bialgebras.

By definition, a generator for a λ-bialgebra (X,h,k) consists of an F-coalgebra (Y,k_Y) and morphisms $i : Y \to X$ and $d : X \to TY$, such that the three diagrams on the left below commute:

$$\begin{tikzcd} Y \arrow[r, "i"] \arrow[d, "k_Y"'] & X \arrow[d, "k"] \\ FY \arrow[r, "Fi"] & FX \end{tikzcd} \quad \begin{tikzcd} X \arrow[r, "d"] \arrow[d, "k"'] & TY \arrow[d, "\lambda_Y \circ Tk_Y"'] \\ FX \arrow[r, "Fd"] & FTY \end{tikzcd} \quad \begin{tikzcd} TY \arrow[r, "Ti"] & TX \arrow[d, "h"] \\ X \arrow[u, "d"] \arrow[r, "\mathrm{id}_X"] & X \end{tikzcd} \quad \begin{tikzcd} TX \arrow[r, "h"] & X \arrow[d, "d"] \\ TY \arrow[u, "Ti"] \arrow[r, "\mathrm{id}_{TY}"] & TY \end{tikzcd} . \tag{7}$$

A basis for a λ-bialgebra is a generator, such that in addition the diagram on the right above commutes.

It is easy to verify that by forgetting the F-coalgebra structure, every generator for a bialgebra in particular provides a generator for its underlying algebra. By Proposition 3.1 it thus follows that there exists a λ-bialgebra homomorphism $i^\sharp : \exp(Y, Fd \circ k \circ i) \to (X,h,k)$. The next result establishes that there exists a second equivalent free bialgebra with a different coalgebra structure.

Lemma 4.11. *Let (Y, k_Y, i, d) be a generator for (X,h,k). Then $i^\sharp : TY \to X$ is a λ-bialgebra homomorphism $i^\sharp : \mathrm{free}(Y, k_Y) \to (X,h,k)$.*

Proof. By definition we have $\mathrm{free}(Y, k_Y) = (TY, \mu_Y, \lambda_Y \circ Tk_Y)$. Clearly the lifting $i^\sharp$ is a homomorphism between the underlying $\mathbb{T}$-algebra structures. It is an F-coalgebra homomorphism since the diagram below commutes:

$$\begin{tikzcd} TY \arrow[r, "Ti"] \arrow[d, "Tk_Y"'] & TX \arrow[r, "h"] \arrow[d, "Tk"] & X \arrow[dd, "k"] \\ TFY \arrow[r, "TFi"] \arrow[d, "\lambda_Y"'] & TFX \arrow[d, "\lambda_X"] & \\ FTY \arrow[r, "FTi"] & FTX \arrow[r, "Fh"] & FX \end{tikzcd} .$$

□

If one moves from generators for bialgebras to bases for bialgebras, both structures coincide.

Lemma 4.12. *Let (Y, k_Y, i, d) be a basis for (X,h,k), then $\mathrm{free}(Y, k_Y) = \exp(Y, Fd \circ k \circ i)$.*

Proof. Using Lemma 4.2 we establish the commutativity of the diagram below:

$$\begin{tikzcd} TY \arrow[rr, "Tk_Y"] \arrow[dd, "\mathrm{id}_{TY}"'] & & TFY \arrow[rr, "\lambda_Y"] \arrow[dd, "TFi"] & & FTY \arrow[rr, "\mathrm{id}_{FTY}"] \arrow[d, "FTi"] & & FTY \arrow[dd, "\mathrm{id}_{FTY}"] \\ & & & & FTX \arrow[r, "Fh"] \arrow[d, "FTd"] & FX \arrow[dr, "Fd"] & \\ TY \arrow[r, "Ti"] & TX \arrow[r, "Tk"] & TFX \arrow[r, "TFd"] & TFTY \arrow[r, "\lambda_{TY}"] & FT^2Y \arrow[rr, "F\mu_Y"] & & FTY \end{tikzcd} .$$

□

Example 4.4 (Canonical RFSA)**.** In Example 3.1 we considered the generator $(\mathcal{J}(M_{\mathbb{P}}(L)), i, d)$ with $i(y) = y$ and $d(x) = \{y \in \mathcal{J}(M_{\mathbb{P}}(L)) \mid y \leq x\}$ for the underlying algebraic part of $\mathrm{M}_{\mathbb{P}}(L) = (X, h, k)$ to recover the canonical RFSA for $L = (a+b)^*a$ as the coalgebra $Fd \circ k \circ i$. Figure 1 shows that the coalgebraic part k restricts to $\mathcal{J}(M_{\mathbb{P}}(L))$, suggesting $\alpha = (\mathcal{J}(M_{\mathbb{P}}(L)), k, i, d)$ as a possible generator for the full bialgebra $\mathrm{M}_{\mathbb{P}}(L)$. However, as one easily verifies, the a-action on $[\{y\}]$ implies the non-commutativity of the second diagram on the left of (7). The issue can be fixed by modifying the definition of d by $d([\{y\}]) := \{[\{y\}]\}$. In consequence $\mathrm{free}(\mathcal{J}(M_{\mathbb{P}}(L)), k)$ and $\exp(\mathcal{J}(M_{\mathbb{P}}(L)), Fd \circ k \circ i)$ coincide (even though the modification does not yield a basis).

We close this section by observing that a basis for the underlying algebra of a bialgebra is sufficient for constructing a generator for the full bialgebra.

Lemma 4.13. *Let (X, h, k) be a λ-bialgebra and (Y, i, d) a basis for the $\mathbb{T}$-algebra (X, h). Then $(TY, (Fd \circ k \circ i)^\sharp, i^\sharp, \eta_{TY} \circ d)$ is a generator for (X, h, k).*

Proof. In the following we abbreviate $k_{TY} := (Fd \circ k \circ i)^\sharp = F\mu_Y \circ \lambda_{TY} \circ T(Fd \circ k \circ i)$. By Proposition 3.1 the morphism $i^\sharp$ is an F-coalgebra homomorphism $i^\sharp : (TY, k_{TY}) \to (X, k)$. This shows the commutativity of the diagram on the left of (7). By Lemma 4.10 the morphism d is an F-coalgebra homomorphism in the reverse direction. Together with the commutativity of the diagram on the left below this implies the commutativity of the second diagram to the left of (7):

$$\begin{tikzcd}
TY \arrow[r, "\eta_{TY}"] \arrow[dd, "k_{TY}"'] & T^2Y \arrow[d, "Tk_{TY}"] \\
 & TFTY \arrow[d, "\lambda_{TY}"] \\
FTY \arrow[ur, "\eta_{FTY}"] \arrow[r, "F\eta_{TY}"] & FT^2Y
\end{tikzcd}
\qquad
\begin{tikzcd}
T^2Y \arrow[r, "T^2i"] & T^2X \arrow[rr, "Th"] \arrow[dr, "\mu_X"] & & TX \arrow[dd, "h"] \\
TY \arrow[u, "\eta_{TY}"] \arrow[r, "Ti"] & TX \arrow[u, "\eta_{TX}"] \arrow[r, "\mathrm{id}_{TX}"] & TX \arrow[dr, "h"] & \\
X \arrow[u, "d"] \arrow[rrr, "\mathrm{id}_X"] & & & X.
\end{tikzcd}$$

Similarly, the commutativity of third diagram to the left of (7) follows from the commutativity of the diagram on the right above. ☐

4.5 Bases as coalgebras

This section is concerned with an alternative approach to the generalisation of bases. More specifically, we are interested in the work of Jacobs [36], where a basis for an algebra over a monad is defined as a coalgebra for the comonad on the category of Eilenberg-Moore algebras induced by the free algebra adjunction. Explicitly, a basis for a $\mathbb{T}$-algebra (X, h) in the former sense consists of a T-coalgebra (X, k) such that the following three diagrams commute:

$$\begin{tikzcd}
TX \arrow[r, "Tk"] \arrow[d, "h"'] & T^2X \arrow[d, "\mu_X"] \\
X \arrow[r, "k"] & TX
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "k"] \arrow[dr, "\mathrm{id}_X"'] & TX \arrow[d, "h"] \\
 & X
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[r, "k"] \arrow[d, "k"'] & TX \arrow[d, "T\eta_X"] \\
TX \arrow[r, "Tk"] & T^2X
\end{tikzcd} . \tag{8}$$

The next result shows that a basis as in Definition 4.1 induces a basis as in [36].

Lemma 4.14. *Let (Y, i, d) be a basis for a $\mathbb{T}$-algebra (X, h). Then (8) commutes for $k := Ti \circ d$.*

Proof. The commutativity of the diagram on the left of (8) follows directly from Corollary 4.3. The diagram in the middle of (8) commutes by the definition of a generator. The commutativity of the diagram on the right of (8) is a consequence of Lemma 4.2:

$$\begin{tikzcd}
X \arrow[r, "d"] \arrow[d, "d"'] & TY \arrow[r, "Ti"] & TX \arrow[dd, "T\eta_X"] \\
TY \arrow[ur, "\mathrm{id}_{TY}"] \arrow[dr, "T\eta_Y"] \arrow[d, "Ti"'] & & \\
TX \arrow[r, "Td"] & T^2Y \arrow[r, "T^2i"] & T^2X
\end{tikzcd} .$$

☐

Conversely, assume (X, k) is a T-coalgebra structure satisfying (8) and $i_k : Y_k \to X$ an equaliser of k and η_X. If the underlying category is the category of sets and functions, and Y_k non-empty, one can show that the equaliser is preserved under T, that is, Ti_k is an equaliser of Tk and $T\eta_X$ [36]. Since it holds $Tk \circ k = T\eta_X \circ k$ by (8), there thus exists a unique morphism $d_k : X \to TY_k$, which can be shown to be the inverse of $h \circ Ti_k$ [36]. In other words, $G(X, k) := (Y_k, i_k, d_k)$ is a basis for (X, h) in the sense of Definition 4.1. In the following let $F(Y, i, d) := (X, Ti \circ d)$ for an arbitrary basis of (X, h).

Lemma 4.15. *Let (Y, i, d) be a basis for a $\mathbb{T}$-algebra (X, h) and $k := Ti \circ d$. Then $\eta_X \circ i = k \circ i$ and $Tk \circ (\eta_X \circ i) = T\eta_X \circ (\eta_X \circ i)$.*

Proof. The statement follows from Lemma 4.2:

$$
\begin{tikzcd}
Y \arrow[rr, "i"] \arrow[d, "i"] \arrow[dr, "\eta_Y"] & & X \arrow[d, "\eta_X"] \\
X \arrow[r, "d"] & TY \arrow[r, "Ti"] & TX
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[r, "i"] \arrow[dr, "\eta_Y"] \arrow[dd, "i"] & X \arrow[r, "\eta_X"] \arrow[d, "d"] & TX \arrow[r, "Td"] & T^2Y \arrow[dd, "T^2 i"] \\
& TY \arrow[urr, "\eta_{TY}"] \arrow[d, "Ti"] & & \\
X \arrow[r, "\eta_X"] \arrow[rrr, bend right, "T\eta_X \circ \eta_X"] & TX \arrow[rr, "\eta_{TX}"] & & T^2X
\end{tikzcd}
.
$$

□

Corollary 4.8. *Let $\alpha := (Y, i, d)$ be a non-empty basis for a set-based $\mathbb{T}$-algebra (X, h) and $k := Ti \circ d$. Then $(\mathrm{id}_{(X,h)})_{\alpha, GF\alpha} : Y \to TY_k$ is the unique morphism making the diagram below commute:*

$$
\begin{tikzcd}
TY_k \arrow[r, "Ti_k"] & TX \arrow[r, shift left, "Tk"] \arrow[r, shift right, "T\eta_X"'] & T^2X \\
Y \arrow[u, dashed] \arrow[ur, "\eta_X \circ i"'] & &
\end{tikzcd}
.
$$

Proof. Since i_k is an equaliser of k and η_X [36], it follows from Lemma 4.15 that there exists a unique morphism $\varphi : Y \to Y_k$ such that $i_k \circ \varphi = i$. Since Ti_k is an equaliser of Tk and $T\eta_X$ [36], it follows from Lemma 4.15 that there exists a unique morphism $\psi : Y \to TY_k$ such that $Ti_k \circ \psi = \eta_X \circ i$. It is not hard to see that $\psi = \eta_{Y_k} \circ \varphi$. The statement thus follows from $(\mathrm{id}_{(X,h)})_{\alpha, GF\alpha} = d_k \circ i = d_k \circ i_k \circ \varphi = \eta_{Y_k} \circ \varphi = \psi$.

□

5 Related work

One of the main motivations for the present paper has been our broad interest in active learning algorithms for state-based models [41], in particular automata for NetKAT [42], a formal system for the verification of networks based on Kleene Algebra with Tests [43]. One of the main challenges in learning non-deterministic models such as NetKAT automata is the common lack of a unique minimal acceptor for a given language [30]. The problem has been independently approached for different variants of non-determinism, often with the common idea of finding a subclass admitting a unique representative [10, 11]. A more general and unifying perspective has been given in [12] by van Heerdt, see also [27, 28].

One of the central notions in the work of van Heerdt is the concept of a scoop, originally introduced by Arbib and Manes [13]. Scoops coincide with what we call a generator in Definition 3.1. They have primarily been used as a tool for constructing minimal realisations of automata, similarly to Proposition 3.1. Strengthening the definition of Arbib and Manes to the notion of a basis in Definition 4.1 allows us to further extend such automata-theoretical results, e.g. Lemma 4.10, but also uncovers ramifications with universal algebra, leading for instance to a representation theory of algebra homomorphisms in the same framework.

A generalisation of the notion of a basis to algebras of arbitrary monads has been approached before. For instance, Jacobs [36] defines a basis for an algebra as a coalgebra for the comonad on the category of algebras induced by the free algebra adjunction. In Section 4.5 we showed that a basis in the sense of Definition 4.1 always induces a basis in the sense of [36]. Conversely, given certain assumptions about the existence and preservation of equaliser, it is possible to recover a basis in the sense of Definition 4.1 from a basis in the sense of [36]. As equaliser do not necessarily exist and are not necessarily preserved, our approach carries additional data and thus can be seen as finer.

6 Discussion and future work

We have presented the notions of generators and bases for an algebra over a monad on an arbitrary category which subsume the familiar notions for algebraic theories. We have covered questions about the existence and uniqueness of bases, their product, and established a representation theory for homomorphisms between algebras over a monad in the spirit of linear algebra by substituting Kleisli morphisms for matrices. Building on foundations in the work of Arbib and Manes [13], we have considered generators and bases for the underlying algebra of a bialgebra, and for the full bialgebra. Finally, we have compared our work to the coalgebraic generalisation of bases by Jacobs [36], shortly explored the case in which a monad is induced by a variety, and briefly related our notion to finitely generated objects in finitely presentable categories.

For the future we are particularly interested in using the present work for a unified view on the theory of canonical acceptors of regular languages, e.g. residual finite state automata [30] (RFSA) and variations of it, for instance the theories of residual probabilistic automata [10] (which can be modelled with the distribution monad), residual alternating automata [11], and the átomaton [44] (which can be described with the neighbourhood monad). RFSA are non-deterministic finite state automata that share with deterministic finite state automata two important properties: for any regular language they admit a unique minimal acceptor, and the language of each state is a residual of the language of its initial state. In Example 3.1 we have demonstrated that the so-called canonical RFSA can be recovered as the bialgebra with free state space induced by a generator of join-irreducibles for the underlying algebra of a particular bialgebra. An analogous reasoning has further allowed us to recover the minimal xor automaton in Example 4.3. We believe we can uncover similar correspondences for other variations of non-determinism. Similar ideas have already served as motivation in the work of Arbib and Manes [13] and have recently come up again in [29]. We are also interested in insights into the formulation of active learning algorithms along the lines of [41] for different classes of residual automata, as sketched in the related work section.

7 Acknowledgements

We are grateful to the anonymous reviewers for their feedback and help in improving this paper, in particular for suggesting Lemma 3.4 and Lemma 3.5. This research has been supported by GCHQ via the VeTSS grant "Automated black-box verification of networking systems" (4207703/RFA 15845).

References

[1] Serge Lang. Algebra, volume 211 of. *Graduate Texts in Mathematics*, 2004.

[2] Samuel Eilenberg, John C Moore, et al. Adjoint functors and triples. *Illinois Journal of Mathematics*, 9(3):381–398, 1965. doi:10.1215/ijm/1256068141.

[3] Fred EJ Linton. Some aspects of equational categories. In *Proceedings of the Conference on Categorical Algebra*, pages 84–94. Springer, 1966. doi:10.1007/978-3-642-99902-4_3.

[4] Dion Coumans and Bart Jacobs. Scalars, monads, and categories. *arXiv preprint arXiv:1003.0585*, 2010.

[5] Roger Godement. Topologie algébrique et théorie des faisceaux. *Publications de l'Institut de Mathématique de l'Université de Strasbourg*, 1958.

[6] Eugenio Moggi. *Computational lambda-calculus and monads.* University of Edinburgh, Department of Computer Science, Laboratory for Foundations of Computer Science, 1988.

[7] Eugenio Moggi. *An abstract view of programming languages.* University of Edinburgh, Department of Computer Science, Laboratory for Foundations of Computer Science, 1990.

[8] Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991. doi:10.1016/0890-5401(91)90052-4.

[9] Jan JMM Rutten. Automata and coinduction (an exercise in coalgebra). In *International Conference on Concurrency Theory*, pages 194–218. Springer, 1998. doi:10.1007/BFb0055624.

[10] Yann Esposito, Aurélien Lemay, François Denis, and Pierre Dupont. Learning probabilistic residual finite state automata. In *International Colloquium on Grammatical Inference*, pages 77–91. Springer, 2002. doi:10.1007/3-540-45790-9_7.

[11] Sebastian Berndt, Maciej Liśkiewicz, Matthias Lutter, and Rüdiger Reischuk. Learning residual alternating automata. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[12] Gerco van Heerdt, Matteo Sammartino, and Alexandra Silva. Learning automata with side-effects. *arXiv preprint arXiv:1704.08055*, 2017.

[13] Michael A Arbib and Ernest G Manes. Fuzzy machines in a category. *Bulletin of the Australian Mathematical Society*, 13(2):169–210, 1975. doi:10.1017/S0004972700024412.

[14] Martin Hyland and John Power. The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electronic Notes in Theoretical Computer Science*, 172:437–458, 2007. doi:10.1016/j.entcs.2007.02.019.

[15] Bart Jacobs. A recipe for state-and-effect triangles. *arXiv preprint arXiv:1703.09034*, 2017.

[16] Andrew M Pitts. *Nominal sets: Names and symmetry in computer science.* Cambridge University Press, 2013.

[17] Nick Bezhanishvili, Marcello Bonsangue, Helle Hvid Hansen, Dexter Kozen, Clemens Kupke, Prakash Panangaden, and Alexandra Silva. Minimisation in logical form. *arXiv preprint arXiv:2005.11551*, 2020.

[18] Ieke Moerdijk. Monads on tensor categories. *Journal of Pure and Applied Algebra*, 168(2-3):189–208, 2002.

[19] James Robert Cheney. *Nominal Logic Programming.* PhD thesis, Cornell University. URL: https://homepages.inf.ed.ac.uk/jcheney/publications/thesis-informal.pdf.

[20] Gavin J Seal. Order-adjoint monads and injective objects. *Journal of Pure and Applied Algebra*, 214(6):778–796, 2010.

[21] Jan JMM Rutten. Universal coalgebra: a theory of systems. *Theoretical computer science*, 249(1):3–80, 2000. doi:10.1016/S0304-3975(00)00056-6.

[22] Jon Beck. Distributive laws. In *Seminar on triples and categorical homology theory*, pages 119–140. Springer, 1969. doi:10.1007/BFb0083084.

[23] Ross Street. Weak distributive laws. *Theory and Applications of Categories [electronic only]*, 22:313–320, 2009.

[24] Bart Jacobs, Alexandra Silva, and Ana Sokolova. Trace semantics via determinization. *Journal of Computer and System Sciences*, 81(5):859–879, 2015.

[25] Jan Rutten, Marcello Bonsangue, Filippo Bonchi, and Alexandra Silva. Generalizing determinization from automata to coalgebras. *Logical Methods in Computer Science*, 9, 2013. doi:10.2168/LMCS-9(1:9)2013.

[26] Michael O Rabin. Probabilistic automata. *Information and control*, 6(3):230–245, 1963. doi:10.1016/S0019-9958(63)90290-0.

[27] Gerco van Heerdt. An abstract automata learning framework. Master's thesis, Radboud University Nijmegen, 2016.

[28] Gerco van Heerdt. *CALF: Categorical Automata Learning Framework*. PhD thesis, University College London, 2020.

[29] Robert SR Myers, Jiří Adámek, Stefan Milius, and Henning Urbat. Coalgebraic constructions of canonical nondeterministic automata. *Theoretical Computer Science*, 604:81–101, 2015. doi:10.1016/j.tcs.2015.03.035.

[30] François Denis, Aurélien Lemay, and Alain Terlutte. Residual finite state automata. *Fundamenta Informaticae*, 51(4):339–368, 2002.

[31] Stefan Zetzsche, Gerco van Heerdt, Alexandra Silva, and Matteo Sammartino. Canonical automata via distributive law homomorphisms. *arXiv preprint arXiv:2104.13421*, 2021.

[32] Daniele Turi. *Functorial operational semantics*. PhD thesis, PhD thesis, Free University, Amsterdam, 1996.

[33] Jiří Adámek, J Adamek, J Rosicky, et al. *Locally presentable and accessible categories*, volume 189. Cambridge University Press, 1994.

[34] Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 2013.

[35] Jiří Adámek, Stefan Milius, Lurdes Sousa, and Thorsten Wißmann. Finitely presentable algebras for finitary monads. *arXiv preprint arXiv:1909.02524*, 2019.

[36] Bart Jacobs. Bases as coalgebras. In *International conference on algebra and coalgebra in computer science*, pages 237–252. Springer, 2011. doi:10.1007/978-3-642-22944-2_17.

[37] Gavin J Seal. Tensors, monads and actions. *Theory and Applications of Categories*, 28(15):403–433, 2013.

[38] Louis Parlant, Jurriaan Rot, Alexandra Silva, and Bas Westerbaan. Preservation of Equations by Monoidal Monads. In Javier Esparza and Daniel Kráľ, editors, *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*, volume 170 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 77:1–77:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12746, doi:10.4230/LIPIcs.MFCS.2020.77.

[39] Jean Vuillemin and Nicolas Gama. Efficient equivalence and minimization for non deterministic xor automata. Research Report ffinria-00487031f, Ecole Normale Supèrieure, Paris, France, 2010.

[40] Daniele Turi and Gordon Plotkin. Towards a mathematical operational semantics. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 280–291. IEEE, 1997. doi:10.1109/LICS.1997.614955.

[41] Dana Angluin. Learning regular sets from queries and counterexamples. *Information and computation*, 75(2):87–106, 1987. doi:10.1016/0890-5401(87)90052-6.

[42] Carolyn Jane Anderson, Nate Foster, Arjun Guha, Jean-Baptiste Jeannin, Dexter Kozen, Cole Schlesinger, and David Walker. NetKAT: Semantic foundations for networks. In *ACM SIGPLAN Notices*, volume 49, pages 113–126. ACM, 2014. doi:10.1145/2578855.2535862.

[43] Dexter Kozen and Frederick Smith. Kleene algebra with tests: Completeness and decidability. In *International Workshop on Computer Science Logic*, pages 244–259. Springer, 1996. doi:10.1007/3-540-63172-0_43.

[44] Janusz A. Brzozowski and Hellis Tamm. Theory of átomata. *Theor. Comput. Sci.*, 539:13–27, 2014. doi:10.1016/j.tcs.2014.04.016.