Syntactic Monoids in a Category

Jiří Adámek[1], Stefan Milius[*,2], and Henning Urbat[1]

1 **Institut für Theoretische Informatik**
Technische Universität Braunschweig, Germany
2 **Lehrstuhl für Theoretische Informatik**
Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

Abstract

The syntactic monoid of a language is generalized to the level of a symmetric monoidal closed category $\mathcal{D}$. This allows for a uniform treatment of several notions of syntactic algebras known in the literature, including the syntactic monoids of Rabin and Scott ($\mathcal{D}$ = sets), the syntactic semirings of Polák ($\mathcal{D}$ = semilattices), and the syntactic associative algebras of Reutenauer ($\mathcal{D}$ = vector spaces). Assuming that $\mathcal{D}$ is a commutative variety of algebras, we prove that the syntactic $\mathcal{D}$-monoid of a language L can be constructed as a quotient of a free $\mathcal{D}$-monoid modulo the syntactic congruence of L, and that it is isomorphic to the transition $\mathcal{D}$-monoid of the minimal automaton for L in $\mathcal{D}$. Furthermore, in the case where the variety $\mathcal{D}$ is locally finite, we characterize the regular languages as precisely the languages with finite syntactic $\mathcal{D}$-monoids.

1998 ACM Subject Classification F.4.3 Formal Languages

Keywords and phrases Syntactic monoid, transition monoid, algebraic automata theory, duality, coalgebra, algebra, symmetric monoidal closed category, commutative variety

Digital Object Identifier 10.4230/LIPIcs.CALCO.2015.1

1 Introduction

One of the successes of the theory of coalgebras is that ideas from automata theory can be developed at a level of abstraction where they apply uniformly to many different types of systems. In fact, classical deterministic automata are a standard example of coalgebras for an endofunctor. And that automata theory can be studied with coalgebraic methods rests on the observation that formal languages form the final coalgebra.

The present paper contributes to a new category-theoretic view of *algebraic* automata theory. In this theory one starts with an elegant machine-independent notion of language recognition: a language $L \subseteq X^*$ is recognized by a monoid morphism $e : X^* \to M$ if it is the preimage under e of some subset of M. Regular languages are then characterized as precisely the languages recognized by finite monoids. A key concept, introduced by Rabin and Scott [19] (and earlier in unpublished work of Myhill), is the *syntactic monoid* of a language L. It serves as a canonical algebraic recognizer of L, namely the smallest X-generated monoid recognizing L. Two standard ways to construct the syntactic monoid are:

1. as a quotient of the free monoid X^* modulo the *syntactic congruence* of L, which is a two-sided version of the well-known Myhill-Nerode equivalence, and
2. as the *transition monoid* of the minimal automaton for L.

* Stefan Milius acknowledges support by the Deutsche Forschungsgemeinschaft (DFG) under project MI 717/5-1.

6th International Conference on Algebra and Coalgebra in Computer Science (CALCO'15).
Editors: Lawrence S. Moss and Paweł Sobociński; pp. 1–16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In addition to syntactic monoids there are several related notions of syntactic algebras for (weighted) languages in the literature, most prominently the syntactic idempotent semirings of Polák [18] and the syntactic associative algebras of Reutenauer [20], both of which admit constructions similar to (1) and (2). A crucial observation is that monoids, idempotent semirings and associative algebras are precisely the monoid objects in the categories of sets, semilattices and vector spaces, respectively. Moreover, these three categories are symmetric monoidal closed w.r.t. their usual tensor product.

The main goal of our paper is thus to develop a theory of algebraic recognition in a general symmetric monoidal closed category $\mathcal{D} = (\mathcal{D}, \otimes, I)$. Following Goguen [12], a *language* in $\mathcal{D}$ is a morphism $L : X^{\circledast} \to Y$ where X is a fixed object of inputs, Y is a fixed object of outputs, and $X^{\circledast}$ denotes the free $\mathcal{D}$-monoid on X. And a $\mathcal{D}$-*automaton* is given by the picture below: it consists of an object of states Q, a morphism i representing the initial state, an output morphism f, and a transition morphism δ which may be presented in its curried form $\lambda\delta$.

$$\begin{array}{ccc} & X \otimes Q & \\ & \downarrow \delta & \\ I \xrightarrow{\;i\;} & Q & \xrightarrow{\;f\;} Y \\ & \downarrow \lambda\delta & \\ & [X, Q] & \end{array} \tag{1}$$

This means that an automaton is at the same time an *algebra* $I + X \otimes Q \xrightarrow{[i,\delta]} Q$ for the functor $FQ = I + X \otimes Q$, and a *coalgebra* $Q \xrightarrow{\langle f, \lambda\delta \rangle} Y \times [X, Q]$ for the functor $TQ = Y \times [X, Q]$. It turns out that much of the classical (co-)algebraic theory of automata in the category of sets extends to this level of generality. Thus Goguen [12] demonstrated that the initial algebra for F coincides with the free $\mathcal{D}$-monoid $X^{\circledast}$, and that every language is accepted by a unique minimal $\mathcal{D}$-automaton. We will add to this picture the observation that the final coalgebra for T is carried by the object of languages $[X^{\circledast}, Y]$, see Proposition 2.21.

In Section 3 we introduce the central concept of our paper, the *syntactic* $\mathcal{D}$-*monoid* of a language $L : X^{\circledast} \to Y$, which by definition is the smallest X-generated $\mathcal{D}$-monoid recognizing L. Assuming that $\mathcal{D}$ is a commutative variety of algebras, we will show that the above constructions (1) and (2) for the classical syntactic monoid adapt to our general setting: the syntactic $\mathcal{D}$-monoid is (1) the quotient of $X^{\circledast}$ modulo the syntactic congruence of L (Theorem 3.14), and (2) the transition $\mathcal{D}$-monoid of the minimal $\mathcal{D}$-automaton for L (Theorem 4.6). As special instances we recover the syntactic monoids of Rabin and Scott ($\mathcal{D}$ = sets), the syntactic semirings of Polák ($\mathcal{D}$ = semilattices) and the syntactic associative algebras of Reutenauer ($\mathcal{D}$ = vector spaces). Furthermore, our categorical setting yields new types of syntactic algebras "for free". For example, we will identify monoids with zero as the algebraic structures representing partial automata (the case $\mathcal{D}$ = pointed sets), which leads to the *syntactic monoid with zero* for a given language. Similarly, by taking as $\mathcal{D}$ the variety of algebras with an involutive unary operation we obtain *syntactic involution monoids*.

Most of the results of our paper apply to arbitrary languages. In Section 5 we will investigate $\mathcal{D}$-*regular languages*, that is, languages accepted by $\mathcal{D}$-automata with a finitely presentable object of states. Under suitable assumptions on $\mathcal{D}$, we will prove that a language is $\mathcal{D}$-regular iff its syntactic $\mathcal{D}$-monoid is carried by a finitely presentable object (Theorem 5.4). We will also derive a dual characterization of the syntactic $\mathcal{D}$-monoid which is new even in the "classical" case $\mathcal{D}$ = sets: if $\mathcal{D}$ is a locally finite variety, and if moreover some other locally finite variety $\mathcal{C}$ is dual to $\mathcal{D}$ on the level of finite objects, the syntactic $\mathcal{D}$-monoid of L dualizes to the local variety of languages in $\mathcal{C}$ generated by the reversed language of L.

Due to space limitations most proofs are omitted or sketched. See [1] for an extended version of this paper.

Related work. Our paper gives a uniform treatment of various notions of syntactic algebras known in the literature [18, 19, 20]. Another categorical approach to (classical) syntactic monoids appears in the work of Ballester-Bolinches, Cosme-Llopez and Rutten [5]. These authors consider automata in the category of sets specified by *equations* or dually by *coequations*, which leads to a construction of the automaton underlying the syntactic monoid of a language. The fact that it forms the transition monoid of a minimal automaton is also interpreted in that setting. In the present paper we take a more general and conceptual approach by studying algebraic recognition in a symmetric monoidal closed category $\mathcal{D}$. One important source of inspiration for our categorical setting was the work of Goguen [12].

In the recent papers [2, 4] we presented a categorical view of *varieties of languages*, another central topic of algebraic automata theory. Building on the duality-based approach of Gehrke, Grigorieff and Pin [11], we generalized Eilenberg's variety theorem and its local version to the level of an abstract (pre-)duality between algebraic categories. The idea to replace monoids by monoid objects in a commutative variety $\mathcal{D}$ originates in this work.

When revising this paper we were made aware of the ongoing work of Bojanczyk [8]. He considers, in lieu of commutative varieties, categories of Eilenberg-Moore algebras for an arbitrary monad on sorted sets, and defines syntactic congruences in this more general setting. Our Theorem 3.14 is a special case of [8, Theorem 3.1].

2 Preliminaries

Throughout this paper we work with deterministic automata in a commutative variety $\mathcal{D}$ of algebras. Recall that a *variety of algebras* is an equational class of algebras over a finitary signature. It is called *commutative* (or *entropic*) if, for any two objects A and B of $\mathcal{D}$, the set $\mathcal{D}(A, B)$ of all homomorphisms from A to B carries a subobject $[A, B] \rightarrowtail B^{|A|}$ of the product of $|A|$ copies of B. Commutative varieties are precisely the categories of Eilenberg-Moore algebras for a commutative finitary monad on the category of sets, see [13, 15]. We fix an object X (of inputs) and an object Y (of outputs) in $\mathcal{D}$.

▶ **Example 2.1.**
1. **Set** is a commutative variety with $[A, B] = B^A$.
2. A *pointed set* $(A, \bot)$ is a set A together with a chosen point $\bot \in A$. The category $\mathbf{Set}_\bot$ of pointed sets and point-preserving functions is a commutative variety. The point of $[(A, \bot_A), (B, \bot_B)]$ is the constant function with value $\bot_B$.
3. An *involution algebra* is a set with an involutive unary operation $x \mapsto \widetilde{x}$, i.e. $\widetilde{\widetilde{x}} = x$. We call $\widetilde{x}$ the *complement* of x. Morphisms are functions f with $f(\widetilde{x}) = \widetilde{f(x)}$. The variety **Inv** of involution algebras is commutative. Indeed, the set $[A, B]$ of all homomorphisms is an involution algebra with pointwise complementation: $\widetilde{f}$ sends x to $\widetilde{f(x)}$.
4. All other examples we treat in our paper are varieties of modules over a semiring. Given a semiring $\mathbb{S}$ (with 0 and 1) we denote by $\mathbf{Mod}(\mathbb{S})$ the category of all $\mathbb{S}$-modules and module homomorphisms (i.e. $\mathbb{S}$-linear maps). Three interesting special cases of $\mathbf{Mod}(\mathbb{S})$ are:
   a. $\mathbb{S} = \{0, 1\}$, the boolean semiring with $1 + 1 = 1$: the category $\mathbf{JSL}_0$ of join-semilattices with 0, and homomorphisms preserving joins and 0;
   b. $\mathbb{S} = \mathbb{Z}$: the category **Ab** of abelian groups and group homomorphisms;
   c. $\mathbb{S} = \mathbb{K}$ (a field): the category $\mathbf{Vec}(\mathbb{K})$ of vector spaces over $\mathbb{K}$ and linear maps.

▶ **Notation 2.2.** We denote by $\Psi : \mathbf{Set} \to \mathcal{D}$ the left adjoint to the forgetful functor $|-| : \mathcal{D} \to \mathbf{Set}$. Thus ΨX_0 is the free object of $\mathcal{D}$ on the set X_0.

▶ **Example 2.3.**

1. We have $\Psi X_0 = X_0$ for $\mathcal{D} = \mathbf{Set}$ and $\Psi X_0 = X_0 + \{\bot\}$ for $\mathcal{D} = \mathbf{Set}_\bot$.
2. For $\mathcal{D} = \mathbf{Inv}$ the free involution algebra on X_0 is $\Psi X_0 = X_0 + \widetilde{X_0}$ where $\widetilde{X_0}$ is a copy of X_0 (whose elements are denoted $\widetilde{x}$ for $x \in X_0$). The involution swaps the copies of X_0, and the universal arrow $X_0 \to X_0 + \widetilde{X_0}$ is the left coproduct injection.
3. For $\mathcal{D} = \mathbf{Mod}(\mathbb{S})$ the free module ΨX_0 is the submodule of $\mathbb{S}^{X_0}$ on all functions $X_0 \to \mathbb{S}$ with finite support. Equivalently, ΨX_0 consists of formal linear combinations $\sum_{i=1}^n s_i x_i$ with $s_i \in \mathbb{S}$ and $x_i \in X_0$. In particular, $\Psi X_0 = \mathcal{P}_f X_0$ (finite subsets of X_0) for $\mathcal{D} = \mathbf{JSL}_0$, and ΨX_0 is the vector space with basis X_0 for $\mathcal{D} = \mathbf{Vec}(\mathbb{K})$.

▶ **Definition 2.4.** Given objects A, B and C of $\mathcal{D}$, a *bimorphism* from A, B to C is a function $f : |A| \times |B| \to |C|$ such that the maps $f(a, -) : |B| \to |C|$ and $f(-, b) : |A| \to |C|$ carry morphisms of $\mathcal{D}$ for every $a \in |A|$ and $b \in |B|$. A *tensor product* of A and B is a universal bimorphism $t : |A| \times |B| \to |A \otimes B|$, which means that for every bimorphism $f : |A| \times |B| \to |C|$ there is a unique morphism $f' : A \otimes B \to C$ in $\mathcal{D}$ with $f' \cdot t = f$.

▶ **Theorem 2.5** (Banaschweski and Nelson [6]). *Every commutative variety $\mathcal{D}$ has tensor products, making $\mathcal{D} = (\mathcal{D}, \otimes, I)$ with $I = \Psi 1$ a symmetric monoidal closed category. That is, we have the following bijective correspondence of morphisms, natural in $A, B, C \in \mathcal{D}$:*

$$\frac{f : \quad A \otimes B \to C}{\lambda f : \quad A \to [B, C]}$$

▶ **Remark 2.6.** Recall that a *monoid* (M, m, i) in a monoidal category $(\mathcal{D}, \otimes, I)$ (with tensor product $\otimes : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$ and tensor unit $I \in \mathcal{D}$) is an object M equipped with a multiplication $m : M \otimes M \to M$ and unit $i : I \to M$ satisfying the usual associative and unit laws. Due to $\otimes$ and $I = \Psi 1$ representing bimorphisms, this categorical definition is equivalent to the following algebraic one in our setting: a $\mathcal{D}$-*monoid* is a triple $(M, \bullet, i)$ where M is an object of $\mathcal{D}$ and $(|M|, \bullet, i)$ is a monoid in $\mathbf{Set}$ with $\bullet : |M| \times |M| \to |M|$ a bimorphism of $\mathcal{D}$. A *morphism* $h : (M, \bullet, i) \to (M', \bullet', i')$ of $\mathcal{D}$-monoids is a morphism $h : M \to M'$ of $\mathcal{D}$ such that $|h| : |M| \to |M'|$ is a monoid morphism in $\mathbf{Set}$. We denote by $\mathbf{Mon}(\mathcal{D})$ the category of $\mathcal{D}$-monoids and their homomorphisms. In the following we will freely work with $\mathcal{D}$-monoids in both categorical and algebraic disguise.

▶ **Example 2.7.**
1. In $\mathbf{Set}$ the tensor product is the cartesian product, $I = \{*\}$, and $\mathbf{Set}$-monoids are ordinary monoids.
2. In $\mathbf{Set}_\bot$ we have $I = \{\bot, *\}$, and the tensor product of pointed sets $(A, \bot_A)$ and $(B, \bot_A)$ is $A \otimes B = (A \backslash \{\bot_A\}) \times (B \backslash \{\bot_B\}) + \{\bot\}$. $\mathbf{Set}_\bot$-monoids are precisely monoids with zero. Indeed, given a $\mathbf{Set}_\bot$-monoid structure on $(A, \bot)$ we have $x \bullet \bot = \bot = \bot \bullet x$ for all x because $\bullet$ is a bimorphism, i.e. $\bot$ is a zero element. Morphisms of $\mathbf{Mon}(\mathbf{Set}_\bot)$ are zero-preserving monoid morphisms.
3. An $\mathbf{Inv}$-monoid (also called an *involution monoid*) is a monoid equipped with an involution $x \mapsto \widetilde{x}$ such that $x \bullet \widetilde{y} = \widetilde{x} \bullet y = \widetilde{x \bullet y}$. For example, for any set A the power set $\mathcal{P}A$ naturally carries the structure of an involution monoid: the involution takes complements, $\widetilde{S} = A \backslash S$, and the monoid multiplication is the symmetric difference $S \oplus T = (S \backslash T) \cup (T \backslash S)$.
4. $\mathbf{JSL}_0$-monoids are precisely idempotent semirings (with 0 and 1). Indeed, a $\mathbf{JSL}_0$-monoid on a semilattice (i.e. a commutative idempotent monoid) $(D, +, 0)$ is given by a unit 1 and a monoid multiplication that, being a bimorphism, distributes over $+$ and 0.

5. More generally, a $\mathbf{Mod}(\mathbb{S})$-monoid is precisely an associative algebra over $\mathbb{S}$: it consists of an $\mathbb{S}$-module together with a unit 1 and a monoid multiplication that distributes over $+$ and 0 and moreover preserves scalar multiplication in both components.

▶ **Notation 2.8.** We denote by $X^{\otimes n}$ $(n < \omega)$ the n-fold tensor power of X, recursively defined by $X^{\otimes 0} = I$ and $X^{\otimes(n+1)} = X \otimes X^{\otimes n}$.

▶ **Proposition 2.9** (see Mac Lane [14]). *The forgetful functor* $\mathbf{Mon}(\mathcal{D}) \to \mathcal{D}$ *has a left adjoint assigning to every object* X *the free* $\mathcal{D}$*-monoid* $X^{\circledast} = \coprod_{n<\omega} X^{\otimes n}$. *The monoid structure* $(X^{\circledast}, m_X, i_X)$ *is given by the coproduct injection* $i_X : I = X^{\otimes 0} \to X^{\circledast}$ *and* $m_X : X^{\circledast} \otimes X^{\circledast} \to X^{\circledast}$, *where* $X^{\circledast} \otimes X^{\circledast} = \coprod_{n,k<\omega} X^{\otimes n} \otimes X^{\otimes k}$ *and* m_X *has as its* (n,k)*-component the* $(n+k)$*-th coproduct injection. The universal arrow* $\eta_X : X \to X^{\circledast}$ *is the first coproduct injection.*

▶ **Proposition 2.10.** *The free* $\mathcal{D}$*-monoid on* $X = \Psi X_0$ *is* $X^{\circledast} = \Psi X_0^*$. *Its monoid multiplication extends the concatenation of words in* X_0^*, *and its unit is the empty word* ε.

▶ **Example 2.11.**
1. In **Set** we have $X^{\circledast} = X^*$. In $\mathbf{Set}_\perp$ with $X = \Psi X_0 = X_0 + \{\perp\}$ we get $X^{\circledast} = X_0^* + \{\perp\}$. The product $x \bullet y$ is concatenation for $x, y \in X_0^*$, and otherwise $\perp$.
2. In **Inv** with $X = \Psi X_0 = X_0 + \widetilde{X_0}$ we have $X^{\circledast} = X_0^* + \widetilde{X_0^*}$. The multiplication restricted to X_0^* is concatenation, and is otherwise determined by $\widetilde{u} \bullet v = \widetilde{uv} = u \bullet \widetilde{v}$ for $u, v \in X_0^*$.
3. In $\mathbf{JSL}_0$ with $X = \Psi X_0 = \mathcal{P}_f X_0$ we have $X^{\circledast} = \mathcal{P}_f X_0^*$, the semiring of all finite languages over X_0. Its addition is union and its multiplication is the concatentation of languages.
4. More generally, in $\mathbf{Mod}(\mathbb{S})$ with $X = \Psi X_0$ we get $X^{\circledast} = \Psi X_0^* = \mathbb{S}[X_0]$, the module of all finite $\mathbb{S}$-weighted languages over the alphabet X_0. Hence the elements of $\mathbb{S}[X_0]$ are functions $c : X_0^* \to \mathbb{S}$ with finite support, which may be expressed as polynomials $\sum_{i=1}^{n} c(w_i) w_i$ with $w_i \in X_0^*$ and $c(w_i) \in \mathbb{S}$. The $\mathbb{S}$-algebraic structure of $\mathbb{S}[X_0]$ is given by the usual addition, scalar multiplication and product of polynomials.

▶ **Definition 2.12** (Goguen [12]). A $\mathcal{D}$*-automaton* (Q, δ, i, f) consists of an object Q (of states) and morphisms $\delta : X \otimes Q \to Q$, $i : I \to Q$ and $f : Q \to Y$; see Diagram (1). An *automata homomorphism* $h : (Q, \delta, i, f) \to (Q', \delta', i', f')$ is a morphism $h : Q \to Q'$ preserving transitions as well as initial states and outputs, i.e. making the following diagrams commute:

$$
\begin{array}{ccc}
X \otimes Q & \xrightarrow{\delta} & Q \\
{\scriptstyle X\otimes h}\downarrow & & \downarrow{\scriptstyle h} \\
X \otimes Q' & \xrightarrow[\delta']{} & Q'
\end{array}
\qquad
\begin{array}{ccccc}
I & \xrightarrow{i} & Q & \xrightarrow{f} & Y \\
 & {\scriptstyle i'}\searrow & \downarrow{\scriptstyle h} & \nearrow{\scriptstyle f'} & \\
 & & Q' & &
\end{array}
$$

The above definition makes sense in any monoidal category $\mathcal{D}$. In our setting, since $I = \Psi 1$, the morphism i chooses an *initial state* in $|Q|$. Moreover, if $X = \Psi X_0$ for some set X_0 (of inputs), the morphism δ amounts to a choice of endomorphisms $\delta_a : Q \to Q$ for $a \in X_0$, representing transitions. This follows from the bijections

$$
\begin{array}{ll}
\Psi X_0 \otimes Q \to Q & \text{in } \mathcal{D} \\
\hline
\Psi X_0 \to [Q, Q] & \text{in } \mathcal{D} \\
\hline
X_0 \to \mathcal{D}(Q, Q) & \text{in } \mathbf{Set}
\end{array}
$$

▶ **Example 2.13.**
1. The classical deterministic automata are the case $\mathcal{D} = \mathbf{Set}$ and $Y = \{0, 1\}$. Here $f : Q \to \{0, 1\}$ defines the set $F = f^{-1}[1] \subseteq Q$ of final states. For general Y we get deterministic Moore automata with outputs in Y.

2. The setting $\mathcal{D} = \mathbf{Set}_\bot$ with $X = X_0 + \{\bot\}$ and $Y = \{\bot, 1\}$ gives *partial* deterministic automata. Indeed, the state object $(Q, \bot)$ has transitions $\delta_a : (Q, \bot) \to (Q, \bot)$ for $a \in X_0$ preserving $\bot$, that is, $\bot$ is a sink state. Equivalently, we may consider δ_a as a partial transition map on the state set $Q \backslash \{\bot\}$. The morphism $f : (Q, \bot) \to \{\bot, 1\}$ again determines a set of final states $F = f^{-1}[1]$ (in particular, $\bot$ is non-final). And the morphism $i : \{\bot, *\} \to (Q, \bot)$ determines a partial initial state: either $i(*)$ lies in $Q \backslash \{\bot\}$, or no initial state is defined.
3. In $\mathcal{D} = \mathbf{Inv}$ let us choose $X = X_0 + \widetilde{X_0}$ and $Y = \{0, 1\}$ with $\widetilde{0} = 1$. An $\mathbf{Inv}$-automaton is a deterministic automaton with complementary states $x \mapsto \widetilde{x}$ such that (i) for every transition $p \xrightarrow{a} q$ there is a complementary transition $\widetilde{p} \xrightarrow{a} \widetilde{q}$ and (ii) a state q is final iff $\widetilde{q}$ is non-final.
4. For $\mathcal{D} = \mathbf{JSL}_0$ with $X = \mathcal{P}_f X_0$ and $Y = \{0, 1\}$ (the two-chain) an automaton consists of a semilattice Q of states, transitions $\delta_a : Q \to Q$ for $a \in X_0$ preserving finite joins (including 0), an initial state $i \in Q$ and a homomorphism $f : Q \to \{0, 1\}$ which defines a *prime upset* $F = f^{-1}[1] \subseteq Q$ of final states. The latter means that a finite join of states is final iff one of the states is. In particular, 0 is non-final.
5. More generally, automata in $\mathcal{D} = \mathbf{Mod}(\mathbb{S})$ with $X = \Psi X_0$ and $Y = \mathbb{S}$ are $\mathbb{S}$-*weighted automata*. Such an automaton consists of an $\mathbb{S}$-module Q of states, linear transitions $\delta_a : Q \to Q$ for $a \in X_0$, an initial state $i \in Q$ and a linear output map $f : Q \to \mathbb{S}$.

▶ **Remark 2.14.**
1. An *algebra* for an endofunctor F of $\mathcal{D}$ is a pair (Q, α) of an object Q and a morphism $\alpha : FQ \to Q$. A *homomorphism* $h : (Q, \alpha) \to (Q', \alpha')$ of F-algebras is a morphism $h : Q \to Q'$ with $h \cdot \alpha = \alpha' \cdot Fh$. Throughout this paper we work with the endofunctor $FQ = I + X \otimes Q$; its algebras are denoted as triples (Q, δ, i) with $\delta : X \otimes Q \to Q$ and $i : I \to Q$. Hence $\mathcal{D}$-automata are precisely F-algebras equipped with an output morphism $f : Q \to Y$. Moreover, automata homomorphisms are precisely F-algebra homomorphisms preserving outputs.
2. Analogously, a *coalgebra* for an endofunctor T of $\mathcal{D}$ is a pair (Q, γ) of an object Q and a morphism $\gamma : Q \to TQ$. Throughout this paper we work with the endofunctor $TQ = Y \times [X, Q]$; its coalgebras are denoted as triples (Q, τ, f) with $\tau : Q \to [X, Q]$ and $f : Q \to Y$. Hence $\mathcal{D}$-automata are precisely *pointed* T-coalgebras, i.e. T-coalgebras equipped with a morphism $i : I \to Q$. Indeed, given a pointed coalgebra $I \xrightarrow{i} Q \xrightarrow{\langle f, \tau \rangle} Y \times [X, Q]$, the morphism $Q \xrightarrow{\tau} [X, Q]$ is the curried form of a morphism $Q \otimes X \xrightarrow{\cong} X \otimes Q \xrightarrow{\delta} Q$. Automata homomorphisms are T-coalgebra homomorphisms preserving initial states.

▶ **Definition 2.15.** Given a $\mathcal{D}$-monoid (M, m, i) and a morphism $e : X \to M$ of $\mathcal{D}$, the *F-algebra associated to M and e* has carrier M and structure

$$[i, \delta] = (I + X \otimes M \xrightarrow{I + e \otimes M} I + M \otimes M \xrightarrow{[i, m]} M).$$

In particular, the F-algebra associated to the free monoid $X^\circledast$ (and its universal arrow η_X) is

$$[i_X, \delta_X] = (I + X \otimes X^\circledast \xrightarrow{I + \eta_X \otimes X^\circledast} I + X^\circledast \otimes X^\circledast \xrightarrow{[i_X, m_X]} X^\circledast).$$

▶ **Example 2.16.** In **Set** every monoid M together with an "input" map $e : X \to M$ determines an F-algebra with initial state i and transitions $\delta_a = - \bullet e(a)$ for all $a \in X$. The F-algebra associated to X^* is the usual automaton of words: its initial state is ε and the transitions are given by $w \xrightarrow{a} wa$ for $a \in X$.

▶ **Proposition 2.17** (Goguen [12]). *For any symmetric monoidal closed category $\mathcal{D}$ with countable coproducts, $X^{\circledast}$ is the initial algebra for F.*

▶ **Remark 2.18.** Given any F-algebra (Q, δ, i) the unique F-algebra homomorphism $e_Q : X^{\circledast} \to Q$ is constructed as follows: extend the morphism $\lambda\delta : X \to [Q, Q]$ to a $\mathcal{D}$-monoid morphism $(\lambda\delta)^+ : X^{\circledast} \to [Q, Q]$. Then

$$e_Q = (X^{\circledast} \cong X^{\circledast} \otimes I \xrightarrow{(\lambda\delta)^+ \otimes i} [Q, Q] \otimes Q \xrightarrow{\mathsf{ev}} Q), \tag{2}$$

where ev is the 'evaluation morphism', i.e. the counit of the adjunction $- \otimes Q \dashv [Q, -]$.

▶ **Notation 2.19.** $\delta^{\circledast} : X^{\circledast} \otimes Q \to Q$ denotes the uncurried form of $(\lambda\delta)^+ : X^{\circledast} \to [Q, Q]$.

▶ **Remark 2.20.** Recall from Rutten [21] that the final coalgebra for the functor $TQ = \{0, 1\} \times Q^X$ on **Set** is the coalgebra $\mathcal{P}X^* \cong [X^*, \{0, 1\}]$ of all languages over X. Given any coalgebra Q, the unique coalgebra homomorphism from Q to $\mathcal{P}\Sigma^*$ assigns to every state q the language accepted by q (as an initial state). These observations generalize to our present setting. The object $[X^{\circledast}, Y]$ of $\mathcal{D}$ carries the following T-coalgebra structure: its transition morphism $\tau_{[X^{\circledast}, Y]} : [X^{\circledast}, Y] \to [X, [X^{\circledast}, Y]]$ is the two-fold curryfication of

$$[X^{\circledast}, Y] \otimes X \otimes X^{\circledast} \xrightarrow{[X^{\circledast}, Y] \otimes \eta_X \otimes X^{\circledast}} [X^{\circledast}, Y] \otimes X^{\circledast} \otimes X^{\circledast} \xrightarrow{[X^{\circledast}, Y] \otimes m_X} [X^{\circledast}, Y] \otimes X^{\circledast} \xrightarrow{\mathsf{ev}} Y,$$

and its output morphism $f_{[X^{\circledast}, Y]} : [X^{\circledast}, Y] \to Y$ is

$$f_{[X^{\circledast}, Y]} = ([X^{\circledast}, Y] \cong [X^{\circledast}, Y] \otimes I \xrightarrow{[X^{\circledast}, Y] \otimes i_X} [X^{\circledast}, Y] \otimes X^{\circledast} \xrightarrow{\mathsf{ev}} Y).$$

▶ **Proposition 2.21.** *$[X^{\circledast}, Y]$ is the final coalgebra for T.*

Proof sketch. Given any coalgebra (Q, τ, f), let $\delta : X \otimes Q \to Q$ be the uncurried version of $\tau : Q \to [X, Q]$, see Remark 2.14. Then the unique coalgebra homomorphism into $[X^{\circledast}, Y]$ is $\lambda h : Q \to [X^{\circledast}, Y]$, where $h = (Q \otimes X^{\circledast} \cong X^{\circledast} \otimes Q \xrightarrow{\delta^{\circledast}} Q \xrightarrow{f} Y)$. ◀

▶ **Definition 2.22** (Goguen [12]). A *language* in $\mathcal{D}$ is a morphism $L : X^{\circledast} \to Y$.

Note that if $X = \Psi X_0$ (and hence $X^{\circledast} = \Psi X_0^*$) for some set X_0, one can identify a language $L : X^{\circledast} = \Psi X_0^* \to Y$ in $\mathcal{D}$ with its adjoint transpose $\widetilde{L} : X_0^* \to |Y|$, via the adjunction $\Psi \dashv |-| : \mathcal{D} \to \mathbf{Set}$. In the case where $|Y|$ is a two-element set, $\widetilde{L}$ is the characteristic function of a "classical" language $L_0 \subseteq X_0^*$.

▶ **Example 2.23.**
1. In $\mathcal{D} = \mathbf{Set}$ (with $X^{\circledast} = X^*$ and $Y = \{0, 1\}$) one represents $L_0 \subseteq X^*$ by its characteristic function $L : X^* \to \{0, 1\}$.
2. In $\mathcal{D} = \mathbf{Set}_\perp$ (with $X = X_0 + \{\perp\}$, $X^{\circledast} = X_0^* + \{\perp\}$ and $Y = \{\perp, 1\}$) one represents $L_0 \subseteq X_0^*$ by its extended characteristic function $L : X_0^* + \{\perp\} \to \{\perp, 1\}$ where $L(\perp) = \perp$.
3. In $\mathcal{D} = \mathbf{Inv}$ (with $X = X_0 + \widetilde{X_0}$, $X^{\circledast} = X_0^* + \widetilde{X_0^*}$ and $Y = \{0, 1\}$) one represents $L_0 \subseteq X_0^*$ by $L : X_0^* + \widetilde{X_0^*} \to \{0, 1\}$ where $L(w) = 1$ iff $w \in L_0$ and $L(\widetilde{w}) = 1$ iff $w \notin L_0$ for all words $w \in X_0^*$.
4. In $\mathcal{D} = \mathbf{JSL}_0$ (with $X = \mathcal{P}_f X_0$, $X^{\circledast} = \mathcal{P}_f X_0^*$ and $Y = \{0, 1\}$) one represents $L_0 \subseteq X_0^*$ by $L : \mathcal{P}_f X_0^* \to \{0, 1\}$ where $L(U) = 1$ iff $U \cap L_0 \neq \varnothing$.
5. In $\mathcal{D} = \mathbf{Mod}(\mathbb{S})$ (with $X = \Psi X_0$, $X^{\circledast} = \mathbb{S}[X_0]$ and $Y = \mathbb{S}$) an $\mathbb{S}$-weighted language $L_0 : X_0^* \to \mathbb{S}$ is represented by its free extension to a module homomorphism

$$L : \mathbb{S}[X_0^*] \to \mathbb{S}, \quad L\left(\sum_{i=1}^{n} c(w_i) w_i\right) = \sum_{i=1}^{n} c(w_i) L_0(w_i).$$

▶ **Definition 2.24** (Goguen [12]). The language *accepted* by a $\mathcal{D}$-automaton (Q, δ, i, f) is $L_Q = (X^{\circledast} \xrightarrow{e_Q} Q \xrightarrow{f} Y)$, where e_Q is the F-algebra homomorphism of Remark 2.18.

▶ **Example 2.25.**

1. In $\mathcal{D} = \mathbf{Set}$ with $Y = \{0, 1\}$, the homomorphism $e_Q : X^* \to Q$ assigns to every word w the state it computes in Q, i.e. the state the automaton reaches on input w. Thus $L_Q(w) = 1$ iff Q terminates in a final state on input w, which is precisely the standard definition of the accepted language of an automaton. For general Y, the function $L_Q : X^* \to Y$ is the behavior of the Moore automaton Q, i.e. $L_Q(w)$ is the output of the last state in the computation of w.
2. For $\mathcal{D} = \mathbf{Set}_\perp$ with $X = X_0 + \{\perp\}$ and $Y = \{\perp, 1\}$, we have $e_Q : X_0^* + \{\perp\} \to (Q, \perp)$ sending $\perp$ to $\perp$, and sending a word in X_0^* to the state it computes (if any), and to $\perp$ otherwise. Hence $L_Q : X_0^* + \{\perp\} \to \{\perp, 1\}$ defines (via the preimage of 1) the usual language accepted by a partial automaton.
3. In $\mathcal{D} = \mathbf{Inv}$ with $X = X_0 + \widetilde{X_0}$ and $Y = \{0, 1\}$, the map $L_Q : X_0^* + \widetilde{X_0^*} \to \{0, 1\}$ sends $w \in X_0^*$ to 1 iff w computes a final state, and it sends $\widetilde{w} \in \widetilde{X_0^*}$ to 1 iff w computes a non-final state.
4. In $\mathcal{D} = \mathbf{JSL}_0$ with $X = \mathcal{P}_f X_0$ and $Y = \{0, 1\}$, the map $L_Q : \mathcal{P} X_0^* \to \{0, 1\}$ assigns to $U \in \mathcal{P}_f X_0^*$ the value 1 iff the computation of at least one word in U ends in a final state.
5. In $\mathcal{D} = \mathbf{Mod}(\mathbb{S})$ with $X = \Psi X_0$ and $Y = \mathbb{S}$, the map $L_Q : \mathbb{S}[X_0^*] \to \mathbb{S}$ assigns to $\sum_{i=1}^n c(w_i) w_i$ the value $\sum_{i=1}^n c(w_i) y_i$, where y_i is the output of the state Q reaches on input w_i. Taking $Q = \mathbb{S}^n$ for some natural number n yields a classical n-state weighted automaton, and in this case one can show that the restriction of L_Q to X_0^* is is the usual language of a weighted automaton.

▶ **Remark 2.26.** By Remark 2.14 every $\mathcal{D}$-automaton (Q, δ, i, f) is an F-algebra as well as a T-coalgebra. Our above definition of L_Q was purely algebraic. The corresponding coalgebraic definition uses the unique coalgebra homomorphism $c_Q : Q \to [X^{\circledast}, Y]$ into the final T-coalgebra and precomposes with $i : I \to Q$ to get a morphism $c_Q \cdot i : I \to [X^{\circledast}, Y]$ (choosing a language, i.e. an element of $[X^{\circledast}, Y]$). Unsurprisingly, the results are equal:

▶ **Proposition 2.27.** *The language $L_Q : X^{\circledast} \to Y$ of an automaton (Q, δ, i, f) is the uncurried form of the morphism $c_Q \cdot i : I \to [X^{\circledast}, Y]$.*

3 Algebraic Recognition and Syntactic $\mathcal{D}$-Monoids

In classical algebraic automata theory one considers recognition of languages by (ordinary) monoids in lieu of automata. One key concept is the *syntactic monoid* which is characterized as the smallest monoid recognizing a given language. There are also related concepts of canonical algebraic recognizers in the literature, e.g. the syntactic idempotent semiring and the syntactic associative algebra. In this section we will give a uniform account of algebraic language recognition in our categorical setting. Our main result is the definition and construction of a minimal algebraic recognizer, the *syntactic $\mathcal{D}$-monoid* of a language.

▶ **Definition 3.1.** A $\mathcal{D}$-monoid morphism $e : X^{\circledast} \to M$ *recognizes* the language $L : X^{\circledast} \to Y$ if there exists a morphism $f : M \to Y$ of $\mathcal{D}$ with $L = f \cdot e$.

▶ **Example 3.2.** We use the notation of Example 2.23.

1. $\mathcal{D} = \mathbf{Set}$ with $X^{\circledast} = X^*$ and $Y = \{0, 1\}$: given a monoid M, a function $f : M \to \{0, 1\}$ defines a subset $F = f^{-1}[1] \subseteq M$. Hence a monoid morphism $e : X^* \to M$ recognizes

L via f (i.e. $L = f \cdot e$) iff $L_0 = e^{-1}[F]$. This is the classical notion of recognition of a language $L_0 \subseteq X^*$ by a monoid, see e.g. Pin [17].

2. $\mathcal{D} = \mathbf{Set}_\bot$ with $X = X_0 + \{\bot\}$, $X^\circledast = X_0^* + \{\bot\}$ and $Y = \{\bot, 1\}$: given a monoid with zero M, a $\mathbf{Set}_\bot$-morphism $f : M \to \{\bot, 1\}$ defines a subset $F = f^{-1}[1]$ of $M\backslash\{0\}$. A zero-preserving monoid morphism $e : X_0^* + \{\bot\} \to M$ recognizes L via f iff $L_0 = e^{-1}[F]$.
3. $\mathcal{D} = \mathbf{Inv}$ with $X = X_0 + \widetilde{X_0}$, $X^\circledast = X_0^* + \widetilde{X_0^*}$ and $Y = \{0, 1\}$: for an involution monoid M to give a morphism $f : M \to \{0, 1\}$ means to give a subset $F = f^{-1}[1] \subseteq M$ satisfying $m \in F$ iff $\widetilde{m} \notin F$. Then L is recognized by $e : X_0^* + \widetilde{X_0^*} \to M$ via f iff $L_0 = X_0^* \cap e^{-1}[F]$.
4. $\mathcal{D} = \mathbf{JSL}_0$ with $X = \mathcal{P}_f X_0$, $X^\circledast = \mathcal{P}_f X_0^*$ and $Y = \{0, 1\}$: for an idempotent semiring M a morphism $f : M \to Y$ defines a prime upset $F = f^{-1}[1]$, see Example 2.13. Hence L is recognized by a semiring homomorphism $e : \mathcal{P}_f X_0^* \to M$ via f iff $L_0 = X_0^* \cap e^{-1}[F]$. Here we identify X_0^* with the set of all singleton languages $\{w\}$, $w \in X_0^*$. This is the concept of language recognition introduced by Polák [18] (except that he puts $F = f^{-1}[0]$, so 0 and 1 must be swapped, as well as F and $M\backslash F$).
5. $\mathcal{D} = \mathbf{Mod}(\mathbb{S})$ with $X = \Psi X_0$, $X^\circledast = \mathbb{S}[X_0]$ and $Y = \mathbb{S}$: given an associative algebra M, the language L is recognized by $e : \mathbb{S}[X_0] \to M$ via $f : M \to \mathbb{S}$ iff $L = f \cdot e$. For the case where the semiring $\mathbb{S}$ is a ring, this notion of recognition is due to Reutenauer [20].

▶ **Remark 3.3.**
1. Since $\mathcal{D}$ and $\mathbf{Mon}(\mathcal{D})$ are varieties, we have the usual factorization system of regular epimorphisms (= surjective homomorphisms) and monomorphisms (= injective homomorphisms). *Quotients* and *subobjects* are understood w.r.t. this system.
2. By an *X-generated $\mathcal{D}$-monoid* we mean a quotient $e : X^\circledast \twoheadrightarrow M$ in $\mathbf{Mon}(\mathcal{D})$. For two such quotients $e_i : X^\circledast \twoheadrightarrow M_i$, $i = 1, 2$, we say, as usual, that e_1 is *smaller or equal to* e_2 (notation: $e_1 \leqslant e_2$) if e_1 factorizes through e_2. Note that if $X = \Psi X_0$, the free $\mathcal{D}$-monoid $X^\circledast = \Psi X_0^*$ on X is also the free $\mathcal{D}$-monoid on the set X_0 (w.r.t. the forgetful functor $\mathbf{Mon}(\mathcal{D}) \to \mathbf{Set}$), see Proposition 2.10. In this case, to give a quotient $e : X^\circledast \twoheadrightarrow M$ is equivalent to giving a set of generators for the $\mathcal{D}$-monoid M indexed by X_0 – which is why M may also be called an *X_0-generated $\mathcal{D}$-monoid*.
3. Let $e : X^\circledast \twoheadrightarrow M$ be an X-generated $\mathcal{D}$-monoid with unit $i : I \to M$ and multiplication $m : M \otimes M \to M$. Recall that $\eta_X : X \to X^\circledast$ denotes the universal morphism of the free $\mathcal{D}$-monoid on X and consider the F-algebra associated to M and $X \xrightarrow{\eta_X} X^\circledast \xrightarrow{e} M$ (see Definition 2.15). Thus, together with a given $f : M \to Y$ an X-generated $\mathcal{D}$-monoid induces an automaton (M, δ, i, f) called the *derived* automaton.

▶ **Lemma 3.4.** *The language recognized by an X-generated $\mathcal{D}$-monoid $e : X^\circledast \twoheadrightarrow M$ via $f : M \to Y$ is the language accepted by its derived automaton.*

We are now ready to give an abstract account of syntactic algebras in our setting. In classical algebraic automata theory the syntactic monoid of a language is characterized as the smallest monoid recognizing that language. We will use this property as our definition of the syntactic $\mathcal{D}$-monoid.

▶ **Definition 3.5.** The *syntactic $\mathcal{D}$-monoid* of language $L : X^\circledast \to Y$, denoted by $\mathsf{Syn}(L)$, is the smallest X-generated monoid recognizing L.

In more detail, the syntactic $\mathcal{D}$-monoid is an X-generated $\mathcal{D}$-monoid $e_L : X^\circledast \twoheadrightarrow \mathsf{Syn}(L)$ together with a morphism $f_L : \mathsf{Syn}(L) \to Y$ of $\mathcal{D}$ such that (i) e_L recognizes L via f_L, and (ii) for every X-generated $\mathcal{D}$-monoid $e : X^\circledast \twoheadrightarrow M$ recognizing L via $f : M \to Y$ we have

$e_L \leqslant e$, that is, the left-hand triangle below commutes for some $\mathcal{D}$-monoid morphism h:

$$\begin{array}{ccccc} X^{\circledast} & \xrightarrow{e} & M & \xrightarrow{f} & Y \\ & \searrow^{e_L} & \downarrow h & \nearrow_{f_L} & \\ & & \mathsf{Syn}(L) & & \end{array}$$

Note that the right-hand triangle also commutes since e is epimorphic and $f \cdot e = L = f_L \cdot e_L$. The universal property determines $\mathsf{Syn}(L)$, e_L and f_L uniquely up to isomorphism. A construction of $\mathsf{Syn}(L)$ is given below (Construction 3.13). We first consider a special case:

▶ **Example 3.6.** In $\mathcal{D} = \mathbf{Set}$ with $Y = \{0, 1\}$, the syntactic monoid of a language $L \subseteq X^*$ can be constructed as the quotient of X^* modulo the *syntactic congruence*, see e.g. [17]:

$$\mathsf{Syn}(L) = X^*/{\sim}, \qquad \text{where } u \sim v \text{ iff for all } x, y \in X^*\colon\ xuy \in L \iff xvy \in L.$$

We aim to generalize this construction to our categorical setting. First note the following

▶ **Lemma 3.7.** *Let $\mathcal{D}$ be any symmetric monoidal closed category with countable coproducts. Then the forgetful functor $\mathbf{Mon}(\mathcal{D}) \to \mathcal{D}$ preserves reflexive coequalizers.*

▶ **Notation 3.8.** Let (M, m, i) be a $\mathcal{D}$-monoid and $x : I \to M$. We write $x \bullet -$ and $- \bullet x$ for the following morphisms, respectively:

$$M \cong I \otimes M \xrightarrow{x \otimes M} M \otimes M \xrightarrow{m} M \qquad \text{and} \qquad M \cong M \otimes I \xrightarrow{M \otimes x} M \otimes M \xrightarrow{m} M.$$

Recall that in our setting, where $\mathcal{D}$ is a commutative variety, we have $I = \Psi 1$ and so the morphism x is the adjoint transpose of an element of M (see Remark 2.6). In the following we shall often write $x \bullet y$, identifying $x, y : I \to M$ with their corresponding elements of M.

▶ **Definition 3.9.** The *syntactic congruence* of a language $L : X^{\circledast} \to Y$ is the following relation on the underlying set of $X^{\circledast}$:

$$E = \{(u, v) \in X^{\circledast} \times X^{\circledast} \mid \forall x, y \in X^{\circledast} : L(x \bullet u \bullet y) = L(x \bullet v \bullet y)\}$$

The projection maps are denoted by $l, r : E \to X^{\circledast}$.

▶ **Lemma 3.10.** *The set E carries a canonical $\mathcal{D}$-algebraic structure making it a $\mathcal{D}$-object.*

Proof sketch. Just observe that $E = \bigcap E_{x,y}$ where for fixed $x, y \in X^{\circledast}$ the object $E_{x,y}$ is the kernel of the $\mathcal{D}$-morphism $X^{\circledast} \xrightarrow{x \bullet -} X^{\circledast} \xrightarrow{- \bullet y} X^{\circledast} \xrightarrow{L} Y$. ◀

That the name syntactic *congruence* makes sense follows from Lemma 3.11 below. First recall that a $\mathcal{D}$-*monoid congruence* on a given $\mathcal{D}$-monoid M is an equivalence relation in $\mathbf{Mon}(\mathcal{D})$, that is, a jointly monic pair $c_1, c_2 : C \to M$ of $\mathcal{D}$-monoid morphisms (equivalently a $\mathcal{D}$-submonoid $\langle c_1, c_2 \rangle : C \rightarrowtail M \times M$) which is reflexive, symmetric and transitive. Congruences on M are ordered as subobjects of $M \times M$, i.e. via inclusion.

▶ **Lemma 3.11.** *E is a $\mathcal{D}$-monoid congruence on $X^{\circledast}$.*

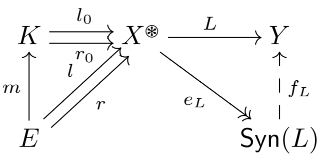
We can give an alternative, more conceptual, description of E:

▶ **Lemma 3.12.** *Let $l_0, r_0 : K \to X^{\circledast}$ be the kernel pair of $L : X^{\circledast} \to Y$ in $\mathcal{D}$. Then $l, r : E \to X^{\circledast}$ is the* largest *$\mathcal{D}$-monoid congruence contained in K.*

▶ **Construction 3.13.** Let $L : X^{\circledast} \to Y$ be a language and $l, r : E \to X^{\circledast}$ its syntactic congruence. We construct the $\mathcal{D}$-monoid $\mathsf{Syn}(L)$ as the coequalizer of l and r in $\mathbf{Mon}(\mathcal{D})$:

$$E \overset{l}{\underset{r}{\rightrightarrows}} X^{\circledast} \xrightarrow{e_L} \mathsf{Syn}(L).$$

We need to show that $\mathsf{Syn}(L)$ has the universal property of Definition 3.5, which first requires to define the morphism $f_L : \mathsf{Syn}(L) \to Y$ with $L = f_L \cdot e_L$. To this end consider the diagram below, where l_0, r_0 is the kernel pair of L and m witnesses that E is contained in K, i.e. $l = l_0 \cdot m$ and $r = r_0 \cdot m$ (see Lemma 3.12).



By Lemma 3.7 the morphism e_L is also a coequalizer of l and r in $\mathcal{D}$. Since $L \cdot l = L \cdot r$ by the above diagram, this yields a unique $f_L : \mathsf{Syn}(L) \to Y$ with $L = f_L \cdot e_L$. In other words, $\mathsf{Syn}(L)$ recognizes L via f_L.

▶ **Theorem 3.14.** $\mathsf{Syn}(L)$ *together with* e_L *and* f_L *forms the syntactic* $\mathcal{D}$*-monoid of* L*.*

Proof sketch. This follows from the correspondence between kernel pairs and regular quotients: since $l, r : E \to X^{\circledast}$ is the largest congruence contained in the kernel pair of L by Lemma 3.12, the coequalizer e_L of l, r is the smallest quotient of $X^{\circledast}$ recognizing L. ◀

▶ **Remark 3.15.** Our proof of Theorem 3.14 is quite conceptual and works in a general symmetric monoidal closed category $\mathcal{D}$ with enough structure. On this level of generality one would use Lemma 3.12 to *define* the syntactic congruence E as the largest $\mathcal{D}$-monoid congruence contained in the kernel of $L : X^{\circledast} \to Y$. However, it is unclear whether such a congruence exists in this generality and so its existence might have to be taken as an assumption. Hence we restricted ourselves to the setting of a commutative variety $\mathcal{D}$.

▶ **Example 3.16.** Using the notation of Example 2.23 we obtain the following concrete syntactic algebras:

1. In $\mathbf{Set}_\perp$ with $X = X_0 + \{\perp\}$ and $Y = \{\perp, 1\}$ the *syntactic monoid with zero* of a language $L_0 \subseteq X_0^*$ is $(X_0^* + \{\perp\})/{\sim}$ where, for all $u, v \in X_0^* + \{\perp\}$,

   $$u \sim v \quad \text{iff} \quad \text{for all } x, y \in X_0^* : xuy \in L_0 \Leftrightarrow xvy \in L_0.$$

   The zero element is the congruence class of $\perp$.
2. In $\mathbf{Inv}$ with $X = X_0 + \widetilde{X_0}$ and $Y = \{0, 1\}$ the *syntactic involution monoid* of a language $L_0 \subseteq X_0^*$ is the quotient of $X_0 + \widetilde{X_0^*}$ modulo the congruence $\sim$ defined for words $u, v \in X_0^*$ as follows:
   (i) $u \sim v$ iff $\widetilde{u} \sim \widetilde{v}$ iff for all $x, y \in X_0^* : xuy \in L_0 \iff xvy \in L_0$;
   (ii) $u \sim \widetilde{v}$ iff $\widetilde{u} \sim v$ iff for all $x, y \in X_0^* : xuy \in L_0 \iff xvy \notin L_0$.
3. In $\mathbf{Mod}(\mathbb{S})$ with $X = \Psi X_0$ and $Y = \mathbb{S}$ the *syntactic associative* $\mathbb{S}$*-algebra* of a weighted language $L_0 : X_0^* \to \mathbb{S}$ is the quotient of $\mathbb{S}[X_0]$ modulo the congruence defined for $U, V \in \mathbb{S}[X_0]$ as follows:

   $$U \sim V \quad \text{iff} \quad \text{for all } x, y \in X_0^* : L(xUy) = L(xVy) \tag{3}$$

   Indeed, since $L : \mathbb{S}[X_0] \to \mathbb{S}$ is linear, (3) implies $L(PUQ) = L(PVQ)$ for all $P, Q \in \mathbb{S}[X_0]$, which is the syntactic congruence of Definition 3.9.

4. In particular, for $\mathcal{D} = \mathbf{JSL}_0$ with $X = \mathcal{P}_f X_0$ and $Y = \{0, 1\}$, we get the *syntactic (idempotent) semiring* of a language $L_0 \subseteq X_0^*$ introduced by Polák [18]: it is the quotient $\mathcal{P}_f X_0^*/\!\sim$ where for $U, V \in \mathcal{P}_f X_0^*$ we have

$$U \sim V \quad \text{iff} \quad \text{for all } x, y \in X_0^* : (xUy) \cap L_0 \neq \varnothing \iff xVy \cap L_0 \neq \varnothing.$$

5. For $\mathcal{D} = \mathbf{Vec}(\mathbb{K})$ with $X = \Psi X_0$ and $Y = \mathbb{K}$, the *syntactic* $\mathbb{K}$*-algebra* of a $\mathbb{K}$-weighted language $L_0 : X_0^* \to \mathbb{K}$ is the quotient $\mathbb{K}[X_0]/I$ of the $\mathbb{K}$-algebra of finite weighted languages modulo the ideal

$$I = \{V \in \mathbb{K}[X_0] \mid \text{for all } x, y \in X_0^* : L(xVy) = 0\}.$$

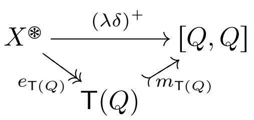
Indeed, the congruence this ideal I generates ($U \sim V$ iff $U - V \in I$) is precisely (3). Syntactic $\mathbb{K}$-algebras were studied by Reutenauer [20].

6. Analogously, for $\mathcal{D} = \mathbf{Ab}$ with $X = \Psi X_0$ and $Y = \mathbb{Z}$, the *syntactic ring* of a $\mathbb{Z}$-weighted language $L_0 : X_0^* \to \mathbb{Z}$ is the quotient $\mathbb{Z}[X_0]/I$, where I is the ideal of all $V \in \mathbb{Z}[X_0]$ with $L(xVy) = 0$ for all $x, y \in X_0^*$.

4 Transition $\mathcal{D}$-Monoids

Here we present another construction of the syntactic $\mathcal{D}$-monoid of a language: it is the transition $\mathcal{D}$-monoid of the minimal $\mathcal{D}$-automaton for this language. Recall that for any object Q of a closed monoidal category $\mathcal{D}$, the object $[Q, Q]$ forms a $\mathcal{D}$-monoid w.r.t. composition.

▶ **Definition 4.1.** The *transition* $\mathcal{D}$*-monoid* $\mathsf{T}(Q)$ of an F-algebra (Q, δ, i) is the image of the $\mathcal{D}$-monoid morphism $(\lambda\delta)^+ : X^\circledast \to [Q, Q]$ extending $\lambda\delta : X \to [Q, Q]$:



▶ **Example 4.2.**

1. In **Set** the transition monoid of an F-algebra Q (i.e. an automaton without final states) is the monoid of all extended transition maps $\delta_w = \delta_{a_n} \cdot \dots \cdot \delta_{a_1} : Q \to Q$ for $w = a_1 \cdots a_n \in X^*$, with unit $\mathsf{id}_Q = \delta_\varepsilon$ and composition as multiplication.
2. In $\mathbf{Set}_\perp$ with $X = X_0 + \{\perp\}$ (the setting for partial automata) this is completely analogous, except that we add the constant endomap of Q with value $\perp$.
3. In **Inv** with $X = X_0 + \widetilde{X_0}$ we get the involution monoid of all δ_w and $\widetilde{\delta_w}$. Again the unit is δ_ε, and the multiplication is determined by composition plus the equations $x\widetilde{y} = \widetilde{xy} = \widetilde{x}y$.
4. In $\mathbf{JSL}_0$ with $X = \mathcal{P}_f X_0$ the transition semiring consists of all finite joins of extended transitions, i.e. all semilattice homomorphisms of the form $\delta_{w_1} \vee \cdots \vee \delta_{w_n}$ for $\{w_1, \dots, w_n\} \in \mathcal{P}_f X_0^*$. The transition semiring was introduced by Polák [18].
5. In $\mathbf{Mod}(\mathbb{S})$ with $X = \Psi X_0$ the associative transition algebra consists of all linear maps of the form $\sum_{i=1}^n s_i \delta_{w_i}$ with $s_i \in \mathbb{S}$ and $w_i \in X_0^*$.

Recall from Definition 2.12 that a $\mathcal{D}$-automaton is an F-algebra Q together with an output morphism $f : Q \to Y$. Hence we can speak of the transition $\mathcal{D}$-monoid of a $\mathcal{D}$-automaton.

▶ **Proposition 4.3.** *The language accepted by a* $\mathcal{D}$*-automaton* (Q, δ, f, i) *is recognized by the* $\mathcal{D}$*-monoid morphism* $e_{\mathsf{T}(Q)} : X^\circledast \twoheadrightarrow \mathsf{T}(Q)$.

Proof sketch. The desired morphism $f_{\mathsf{T}(Q)} : \mathsf{T}(Q) \to Y$ with $L_Q = f_{\mathsf{T}(Q)} \cdot e_{\mathsf{T}(Q)}$ is

$$f_{\mathsf{T}(Q)} = (\mathsf{T}(Q) \xrightarrow{m_{\mathsf{T}(Q)}} [Q,Q] \cong [Q,Q] \otimes I \xrightarrow{[Q,Q]\otimes i} [Q,Q] \otimes Q \xrightarrow{\mathsf{ev}} Q \xrightarrow{f} Y). \blacktriangleleft$$

▶ **Definition 4.4.** A $\mathcal{D}$-automaton (Q, δ, i, f) is called *minimal* iff it is

(a) *reachable*: the unique F-algebra homomorphism $X^{\circledast} \to Q$ is surjective;

(b) *simple*: the unique T-coalgebra homomorphism $Q \to [X^{\circledast}, Y]$ is injective.

▶ **Theorem 4.5** (Goguen [12])**.** *Every language $L : X^{\circledast} \to Y$ is accepted by a minimal $\mathcal{D}$-automaton* $\mathsf{Min}(L)$*, unique up to isomorphism. Given any reachable automaton Q accepting L, there is a unique surjective automata homomorphism from Q into* $\mathsf{Min}(L)$*.*

This leads to the announced construction of syntactic $\mathcal{D}$-monoids via transition $\mathcal{D}$-monoids. The case $\mathcal{D} = \mathbf{Set}$ is a standard result of algebraic automata theory (see e.g. Pin [17]), and the case $\mathcal{D} = \mathbf{JSL}_0$ is due to Polák [18].

▶ **Theorem 4.6.** *The syntactic $\mathcal{D}$-monoid of a language $L : X^{\circledast} \to Y$ is isomorphic to the transition $\mathcal{D}$-monoid of its minimal $\mathcal{D}$-automaton:*

$$\mathsf{Syn}(L) \cong \mathsf{T}(\mathsf{Min}(L)).$$

Proof sketch. Using reachability and simplicity of $\mathsf{Min}(L)$, one proves that the quotients $e_L : X^{\circledast} \twoheadrightarrow \mathsf{Syn}(L)$ and $e_{\mathsf{T}(\mathsf{Min}(L))} : X^{\circledast} \twoheadrightarrow \mathsf{T}(\mathsf{Min}(L))$ have the same kernel pair, namely the syntactic congruence of L. This implies the statement of the theorem. ◀

5 $\mathcal{D}$-Regular Languages

Our results so far apply to arbitrary languages in $\mathcal{D}$. In the present section we focus on *regular languages*, which in $\mathcal{D} = \mathbf{Set}$ are the languages accepted by finite automata, or equivalently the languages recognized by finite monoids. For arbitrary $\mathcal{D}$ the role of finite sets is taken over by finitely presentable objects. Recall that an object D of $\mathcal{D}$ is *finitely presentable* if the hom-functor $\mathcal{D}(D, -) : \mathcal{D} \to \mathbf{Set}$ preserves filtered colimits. Equivalently, D is an algebra presentable with finitely many generators and relations.

▶ **Definition 5.1.** A language $L : X^{\circledast} \to Y$ is called *$\mathcal{D}$-regular* if it is accepted by some $\mathcal{D}$-automaton with a finitely presentable object of states.

To work with this definition, we need the following

▶ **Assumptions 5.2.** We assume that the full subcategory $\mathcal{D}_f$ of finitely presentable objects of $\mathcal{D}$ is closed under subobjects, strong quotients and finite products.

▶ **Example 5.3.**

1. Recall that a variety is *locally finite* if all finitely presentable algebras (equivalently all finitely generated free algebras) are finite. Every locally finite variety satisfies the above assumptions. This includes our examples $\mathbf{Set}$, $\mathbf{Set}_\perp$, $\mathbf{Inv}$ and $\mathbf{JSL}_0$.
2. A semiring $\mathbb{S}$ is called *Noetherian* if all submodules of finitely generated $\mathbb{S}$-modules are finitely generated. In this case, as shown in [10], the category $\mathbf{Mod}(\mathbb{S})$ satisfies our assumptions. Every field is Noetherian, as is every finitely generated commutative ring, so $\mathbf{Vec}(\mathbb{K})$ and $\mathbf{Ab} = \mathbf{Mod}(\mathbb{Z})$ are special instances.

▶ **Theorem 5.4.** *For any language $L : X^{\circledast} \to Y$ the following statements are equivalent:*
(a) *L is $\mathcal{D}$-regular.*
(b) *The minimal $\mathcal{D}$-automaton $\mathsf{Min}(L)$ has finitely presentable carrier.*
(c) *L is recognized by some $\mathcal{D}$-monoid with finitely presentable carrier.*
(d) *The syntactic $\mathcal{D}$-monoid $\mathsf{Syn}(L)$ has finitely presentable carrier.*

Proof sketch. This follows immediately from the universal properties of $\mathsf{Syn}(L)$ and $\mathsf{Min}(L)$ and the assumed closure properties of $\mathcal{D}_f$. ◀

Just as the collection of all languages is internalized by the final coalgebra $[X^{\circledast}, Y]$, see Proposition 2.21, we can internalize the regular languages by means of the *rational coalgebra*.

▶ **Definition 5.5.** The *rational coalgebra* ϱT for T is the colimit (taken in the category of T-coalgebras and homomorphisms) of all T-coalgebras with finitely presentable carrier.

▶ **Proposition 5.6.** *There is a one-to-one correspondence between $\mathcal{D}$-regular languages and elements $I \to \varrho T$ of the rational coalgebra.*

We conclude this section with an interesting dual perspective on syntactic monoids, based on our previous work [2, 4]. For lack of space we restrict to the case $\mathcal{D} = \mathbf{Set}$. This category is *predual* to the category $\mathbf{BA}$ of boolean algebras in the sense that the full subcategories of finite sets and finite boolean algebras are dually equivalent. Indeed, this is a restriction of the well-known Stone duality: the dual equivalence functor assigns to a finite boolean algebra B the set $\mathsf{At}(B)$ of its atoms, and to a boolean homomorphism $h : A \to B$ the map $\mathsf{At}(h) : \mathsf{At}(B) \to \mathsf{At}(A)$ sending $b \in \mathsf{At}(B)$ to the unique atom $a \in \mathsf{At}(A)$ with $ha \geqslant b$.

How do the concepts we investigated in $\mathbf{Set}$ – languages, automata and monoids – dualize to $\mathbf{BA}$? Observe that $\mathsf{Reg}(X)$, the boolean algebra of regular languages over the alphabet X, can be viewed as a deterministic automaton: its final states are the regular languages containing the empty word, and the transitions are given by $L \xrightarrow{a} a^{-1}L$ for $a \in X$, where $a^{-1}L = \{w \in X^* : aw \in L\}$ is the *left derivative* of L w.r.t. the letter a. (Similarly, the *right derivative* of L w.r.t. a is $La^{-1} = \{w \in X^* : wa \in L\}$.) This makes $\mathsf{Reg}(X)$ a coalgebra for the endofunctor $\overline{T} = \{0,1\} \times \mathsf{Id}^X$ on $\mathbf{BA}$. Since the two-chain $\{0,1\}$ is dual to the singleton set 1, finite coalgebras for $\overline{T}$ dualize to finite *algebras* for the functor $F = 1 + X \times \mathsf{Id} \cong 1 + \coprod_X \mathsf{Id}$ on $\mathbf{Set}$. Based on this, we proved in [2] that further (i) finite $\overline{T}$-subcoalgebras of $\mathsf{Reg}(X)$ dualize to finite quotient algebras of the initial F-algebra X^*, and (ii) finite *local varieties of languages* (i.e. finite $\overline{T}$-subcoalgebras of $\mathsf{Reg}(X)$ closed under right derivatives) dualize to those F-algebras associated to X-generated monoids, see Definition 2.15. For a regular language $L \subseteq X^*$ the F-algebras associated to the minimal automaton $\mathsf{Min}(L)$ and the syntactic monoid $\mathsf{Syn}(L)$ are finite. Their dual $\overline{T}$-coalgebras are characterized as follows:

▶ **Theorem 5.7.** *Let $L \subseteq X^*$ be a regular language, and L^{rev} its reversed language.*
(a) *$\mathsf{Min}(L)$ is dual to the smallest subcoalgebra of $\mathsf{Reg}(X)$ containing L^{rev}.*
(b) *$\mathsf{Syn}(L)$ is dual to the smallest local variety of languages containing L^{rev}.*

Part (a) of this theorem adds to the recently developed dual view of minimal automata, see [7] and also [16, 3]. All the above considerations generalize from $\mathbf{BA}/\mathbf{Set}$ to arbitrary pairs $\mathcal{C}/\mathcal{D}$ of predual locally finite varieties of algebras. Examples include the self-predual varieties $\mathcal{C} = \mathcal{D} = \mathbf{JSL}_0$ and $\mathcal{C} = \mathcal{D} = \mathbf{Vec}(\mathbb{K})$ for a finite field $\mathbb{K}$.

6 Conclusions and Future Work

We proposed the first steps of a categorical theory of algebraic language recognition. Despite our assumption that $\mathcal{D}$ is a commutative variety, the bulk of our definitions, constructions and proofs works in any symmetric monoidal closed category with enough structure. However, the construction of the syntactic monoid via the syntactic congruence, and the proof that it coincides with a transition monoid, required the concrete algebraic setting. It remains an open problem to develop a genuinely abstract framework for our theory. In particular, such a generalized setting should provide the means for incorporating *ordered* algebras, e.g. the syntactic ordered monoids of Pin [17]. We expect this can be achieved by working with (order-)enriched categories, where the coequalizer in our construction of the syntactic monoid is replaced by a coinserter. A more general theory of recognition might also open the door to treating algebraic recognizers for additional types of behaviors, including Wilke algebras [22] (representing ω-languages) and forest algebras [9] (representing tree and forest languages).

One of the leading themes of algebraic automata theory is the classification of languages in terms of their syntactic algebras. For instance, by Schützenberger's theorem a language is star-free iff its syntactic monoid is aperiodic. We hope that our conceptual view of syntactic monoids (notably their dual characterization in Theorem 5.7) can contribute to a duality-based approach to such results, leading to generalizations and new proof techniques.

References

1 Jiří Adámek, Stefan Milius, , and Henning Urbat. Syntactic monoids in a category. Extended version. `http://arxiv.org/abs/1504.02694`, 2015.

2 Jiří Adámek, Stefan Milius, Robert S. R. Myers, and Henning Urbat. Generalized Eilenberg Theorem I: Local Varieties of Languages. In Anca Muscholl, editor, *Proc. Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 8412 of *Lecture Notes Comput. Sci.*, pages 366–380. Springer, 2014.

3 Jiří Adámek, Stefan Milius, Robert S. R. Myers, and Henning Urbat. On continuous nondeterminism and state minimality. In Bart Jacobs, Alexandra Silva, and Sam Staton, editors, *Proc. Mathematical Foundations of Programming Science (MFPS XXX)*, volume 308 of *Electron. Notes Theor. Comput. Sci.*, pages 3–23. Elsevier, 2014.

4 Jiří Adámek, Stefan Milius, Robert S. R. Myers, and Henning Urbat. Varieties of Languages in a Category. Accepted for LICS 2015. `http://arxiv.org/abs/1501.05180`, 2015.

5 A. Ballester-Bolinches, E. Cosme-Llopez, and J.J.M.M. Rutten. The dual equivalence of equations and coequations for automata. Technical report, CWI, 2014.

6 Bernhard Banaschweski and Evelyn Nelson. Tensor products and bimorphisms. *Canad. Math. Bull.*, 19:385–402, 1976.

7 Nick Bezhanishvili, Clemens Kupke, and Prakash Panangaden. Minimization via duality. In Luke Ong and Ruy de Queiroz, editors, *Logic, Language, Information and Computation*, volume 7456 of *Lecture Notes in Computer Science*, pages 191–205. Springer Berlin Heidelberg, 2012.

8 Mikołaj Bojánczyk. Recognisable languages over monads. Preprint: `http://arxiv.org/abs/1502.04898`, 2015.

9 Mikołaj Bojánczyk and Igor Walukiewicz. Forest algebras. In *Automata and Logic: History and Perspectives*, pages 107–132, 2006.

10 Marcello M. Bonsangue, Stefan Milius, and Alexandra Silva. Sound and complete axiomatizations of coalgebraic language equivalence. *ACM Trans. Comput. Log.*, 14(1:7), 2013.

11 Mai Gehrke, Serge Grigorieff, and Jean-Éric Pin. Duality and equational theory of regular languages. In *Proc. ICALP 2008, Part II*, volume 5126 of *Lecture Notes Comput. Sci.*, pages 246–257. Springer, 2008.

12 Joseph A. Goguen. Discrete-time machines in closed monoidal categories. I. *J. Comput. Syst. Sci.*, 10(1):1–43, 1975.

13 Anders Kock. Monads on symmetric monoidal closed categories. *Arch. Math.*, 21:1–10, 1970.

14 Saunders Mac Lane. *Categories for the working mathematician.* Springer, 2nd edition, 1998.

15 Fred Linton. Autonomous equational categories. *J. Math. Mech.*, 15:637–642, 166.

16 Robert S. R. Myers, Jiří Adámek, Stefan Milius, and Henning Urbat. Canonical nondeterministic automata. In Marcello M. Bonsangue, editor, *Proc. Coalgebraic Methods in Computer Science (CMCS'14)*, volume 8446 of *Lecture Notes Comput. Sci.*, pages 189–210. Springer, 2014.

17 Jean-Éric Pin. Mathematical foundations of automata theory. available at `http://www.liafa.jussieu.fr/~jep/PDF/MPRI/MPRI.pdf`, January 2015.

18 Libor Polák. Syntactic semiring of a language. In Jiří Sgall, Aleš Pultr, and Petr Kolman, editors, *Proc. International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 2136 of *Lecture Notes Comput. Sci.*, pages 611–620. Springer, 2001.

19 Michael O. Rabin and Dana S. Scott. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125, April 1959.

20 Christophe Reutenauer. Séries formelles et algèbres syntactiques. *J. Algebra*, 66:448–483, 1980.

21 Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoret. Comput. Sci.*, 249(1):3–80, 2000.

22 Thomas Wilke. An Eilenberg Theorem for Infinity-Languages. In *Proc. ICALP 91*, pages 588–599. Springer, 1991.