Recognisable Languages over Monads

Mikołaj Bojańczyk[(✉)]

University of Warsaw, Warsaw, Poland
`bojan@mimuw.edu.pl`

Abstract. This paper proposes monads as a framework for algebraic language theory. Examples of monads include words and trees, finite and infinite. Each monad comes with a standard notion of an algebra, called an Eilenberg-Moore algebra, which generalises algebras studied in language theory like semigroups or ω-semigroups. On the abstract level of monads one can prove theorems like the Myhill-Nerode theorem, the Eilenberg theorem; one can also define profinite objects.

The principle behind algebraic language theory for various kinds of objects, such as words or trees, is to use a "compositional" function from the objects into a finite set. To talk about compositionality, one needs objects with some kind of substitution. It so happens that category theory has an abstract concept of objects with substitution, namely a monad. The goal of this paper is to propose monads as a unifying framework for discussing existing algebras and designing new algebras. To introduce monads and their algebras, we begin with two examples, which use a monad style to present algebras for finite and infinite words.

Example 1. Consider the following non-standard definition of a semigroup. Define a $+$-algebra $\mathbf{A}$ to be a set A called its *universe*, together with a *multiplication operation* $\mathrm{mul}_{\mathbf{A}} : A^+ \to A$, which is the identity on single letters, and which is associative in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
(A^+)^+ & \xrightarrow{\ \mu_A\ } & A^+ \\
{\scriptstyle (\mathrm{mul}_{\mathbf{A}})^+}\big\downarrow & & \big\downarrow{\scriptstyle \mathrm{mul}_{\mathbf{A}}} \\
A^+ & \xrightarrow[\ \mathrm{mul}_{\mathbf{A}}\ ]{} & A
\end{array} \ ,
$$

In the diagram, $(\mathrm{mul}_{\mathbf{A}})^+$ is the function that applies $\mathrm{mul}_{\mathbf{A}}$ to each label of a word, and μ_A is the function which flattens a word of words into a word, e.g.

$$(abc)(aa)(acaa) \qquad \mapsto \qquad abcaaacaa.$$

Restricting the multiplication operation in a +-algebra to words of length two (the semigroup binary operation) is easily seen to be a one-to-one correspondence between +-algebras and semigroups. □

The second example will be running example in the paper.

Running Example 1. Let us define an algebra for infinite words in the spirit of the previous example. Define A^∞ to be the finite and ω-words over A, i.e. $A^+ \cup A^\omega$. Define an ∞-algebra **A** to be a set A, called its *universe*, together with a *multiplication operation* $\mathrm{mul}_\mathbf{A} : A^\infty \to A$, which is the identity on single letters, and which is associative in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
(A^\infty)^\infty & \xrightarrow{\ \mu_A\ } & A^\infty \\
{\scriptstyle (\mathrm{mul}_\mathbf{A})^\infty} \downarrow & & \downarrow {\scriptstyle \mathrm{mul}_\mathbf{A}} \\
A^\infty & \xrightarrow[\ \mathrm{mul}_\mathbf{A}\ ]{} & A
\end{array}
$$

In the diagram, $(\mathrm{mul}_\mathbf{A})^\infty$ is the function that applies $\mathrm{mul}_\mathbf{A}$ to the label of every position in a word from $(A^\infty)^\infty$, and μ_A is defined in analogy to mul_{A^+}, with the following proviso: if the argument of μ_A contains an infinite word on some position, then all subsequent positions are ignored, e.g.

$$(abc)(aa)(a^\omega)(abca)(ab^\omega) \qquad \mapsto \qquad abca^\omega$$

An ∞-algebra is essentially the same thing as an ω-semigroup, see [PP04], with the difference that ω-semigroups have separate sorts for finite and infinite words. There is also a close connection with Wilke semigroups [Wil91], which will be described as the running example develops. □

The similarities in the examples suggest that the should be an abstract notion of algebra, which would cover the examples and possibly other settings, e.g. trees. A closer look at the examples reveals that concepts of algebraic language theory such as "algebra", "morphism", "language", "recognisable language" can be defined only in terms of the following four basic concepts (written below in the notation appropriate to +-algebras):

1. how a set A is transformed into a set A^+;
2. how a function $f : A \to B$ is lifted to a function $f^+ : A^+ \to B^+$;
3. a flattening operation from $(A^+)^+ \to A^+$;
4. how to represent an element of A as an element of A^+.

These four concepts are what constitutes a monad, a fundamental concept in category theory and, recently, programming languages like Haskell. However, unlike for Haskell, in this paper the key role is played by Eilenberg-Moore algebras.

The point of this paper is that, based on a monad one can also define things like: "syntactic algebra", "pseudovariety", "MSO logic", "profinite object", and even prove some theorems about them. Furthermore, monads as an abstraction cover practically every setting where algebraic language theory has been applied so far, including labelled scattered orderings [BR12], labelled countable total

orders [CCP11], ranked trees [Ste92], unranked trees [BW08], preclones [ÉW03]. These applications are discussed at length in the full version. The full version also shows how new algebraic settings can be easily produced using monads, as illustrated on a monad describing words with a distinguished position, where standard theorems and definitions come for free by virtue of being a monad. A related paper is [Ési10], which gives an abstract language theory for Lawvere theories, and proves that Lawvere theories admit syntactic algebras and a pseudovariety theorem. Lawvere theories can be viewed as the special case of finitary monads, e.g. finite words are Lawvere theories, but infinite words are not.

This paper shows that several results of formal language theory can be stated and proved on the abstract level of monads, including: the Myhill-Nerode theorem on syntactic algebras, the Eilenberg pseudovariety theorem, or the Reiterman theorem on profinite identities defining pseudovarieties. Another example is decidability of mso, although here monads only take care of the symbol-pushing part, leaving out the combinatorial part that is specific to individual monads, like applying the Ramsey theorem in the case of infinite words. When proving such generalisations of classical theorems, one is naturally forced to have a closer look at notions such as "derivative of a language", or "finite algebra", which are used in the assumptions of the theorems.

Much effort is also devoted to profinite constructions. It is shown that every monad has a corresponding profinite monad, which, like any monad, has its own notion of recognisability, which does not reduce to recognisability in the original monad. For example, the monad for finite words has a corresponding monad of profinite words, and recognisable languages of profinite words turn out to be a generalisation of languages of infinite words definable in the logic mso+u.

Thanks. I would like to thank Bartek Klin (who told me what a monad is), Szymon Toruńczyk and Marek Zawadowski for discussions on the subject.

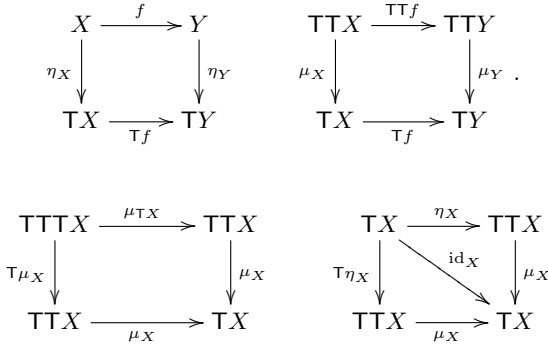## 1   Monads and Their Algebras

This paper uses only the most rudimentary notions of category theory: the definitions of a category (objects and composable morphisms between them), and of a functor (something that maps objects to objects and morphisms to morphisms in a way that is consistent with composition). All examples in this paper use the category of sets, where objects are sets and morphisms are functions; or possibly the category of sorted sets, where objects are sorted sets for some fixed set of sort names, and morphisms are sort-preserving functions.

A monad over a category is defined to be a functor T from the category to itself, and for every object X in the category, two morphisms

$$\eta_X : X \to \mathsf{T}X \qquad \text{and} \qquad \mu_X : \mathsf{T}\mathsf{T}X \to \mathsf{T}X,$$

which are called the unit and multiplication operations. The monad must satisfy the axioms given in Figure 1.

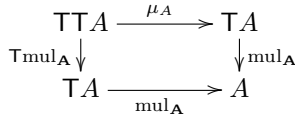We already saw two monads in Example 1 and in the running example.

$$
\begin{array}{ccc}
X \xrightarrow{\;f\;} Y & \qquad & \mathsf{TT}X \xrightarrow{\;\mathsf{TT}f\;} \mathsf{TT}Y \\
\downarrow{\eta_X} \qquad \downarrow{\eta_Y} & & \downarrow{\mu_X} \qquad \downarrow{\mu_Y}\; . \\
\mathsf{T}X \xrightarrow[\;\mathsf{T}f\;]{} \mathsf{T}Y & & \mathsf{T}X \xrightarrow[\;\mathsf{T}f\;]{} \mathsf{T}Y
\end{array}
$$

$$
\begin{array}{ccc}
\mathsf{TTT}X \xrightarrow{\;\mu_{\mathsf{T}X}\;} \mathsf{TT}X & \qquad & \mathsf{T}X \xrightarrow{\;\eta_X\;} \mathsf{TT}X \\
\downarrow{\mathsf{T}\mu_X} \qquad \downarrow{\mu_X} & & \downarrow{\mathsf{T}\eta_X}\;\; \overset{\mathrm{id}_X}{\searrow}\;\; \downarrow{\mu_X} \\
\mathsf{TT}X \xrightarrow[\;\mu_X\;]{} \mathsf{T}X & & \mathsf{TT}X \xrightarrow[\;\mu_X\;]{} \mathsf{T}X
\end{array}
$$

Fig. 1. The axioms of a monad are that these four diagrams commute for every object X in the category and every morphism $f : X \to Y$. The upper diagrams say that the unit and multiplication are natural. The lower left diagram says that multiplication is associative, and the lower right says that the unit is consistent with multiplication.

For this paper, the most important thing about monads is that they have a natural corresponding algebras. An *Eilenberg-Moore algebra for a monad* T, or simply T-algebra, is a pair **A** consisting of a *universe* A, which is an object in the underlining category, together with a multiplication morphism
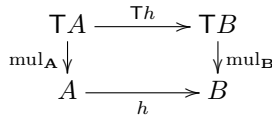
$$\mathrm{mul}_{\mathbf{A}} : \mathsf{T}A \to A,$$

such that the $\mathrm{mul}_{\mathbf{A}} \circ \eta_A$ is the identity, and which is associative in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
\mathsf{TT}A \xrightarrow{\;\mu_A\;} \mathsf{T}A \\
\downarrow{\mathsf{T}\mathrm{mul}_{\mathbf{A}}} \qquad \downarrow{\mathrm{mul}_{\mathbf{A}}} \\
\mathsf{T}A \xrightarrow[\;\mathrm{mul}_{\mathbf{A}}\;]{} A
\end{array}
$$

Observe that this associativity is similar to the lower left axiom in Figure 1. In fact, the lower left axiom in Figure 1 and the upper half of the lower right axiom say that $\mathsf{T}X$ equipped with the operation μ_X forms a T-algebra, called the free T-algebra over X.

We use the convention that an algebra is denoted by a boldface letter, while its universe is written without boldface. A T-*morphism* between two T-algebras **A** and **B** is a function h between their universes which respects their multiplication operations in the sense that the following diagram commutes.

$$
\begin{array}{ccc}
\mathsf{T}A \xrightarrow{\;\mathsf{T}h\;} \mathsf{T}B \\
\downarrow{\mathrm{mul}_{\mathbf{A}}} \qquad \downarrow{\mathrm{mul}_{\mathbf{B}}} \\
A \xrightarrow[\;h\;]{} B
\end{array}
$$

This completes the definition of monads and their algebras.

Languages and colorings. To develop the basic definitions of recognisable languages over a monad, we require the following parameters, which we call the *setting*: the underlying category, the monad, a notion of finite alphabet, and a notion of finite T-algebra. So far, we do not place any restrictions on the notions of finiteness, e.g. when considering sets with infinitely many sorts, reasonable settings will often have finite algebras whose universe is not finite in the same sense as a finite algebra. Actually, for some monads, it is not clear what a finite algebra should be, e.g. this is the case for infinite trees, and this paper sheds little new light on the question. Fix a setting, with the monad being called T, for the following definitions.

A *coloring* of a T-algebra is defined to be a morphism from its universe to some object in the underlying category. A coloring is said to be *recognised* by a T-morphism if the coloring factors through the morhpism. A coloring is called T-*recognisable* if it is recognised by some T-morphism with a finite target, according to the notion of finite T-algebra given in the setting.

When the underlying category is, possibly sorted, sets we can talk about languages, as defined below. Consider a finite alphabet, according to the notion of finite alphabet given in the setting. In all of the examples of this paper, a finite alphabet will be a possibly sorted set with finitely many elements. In particular, if there are infinitely many sorts, then a finite alphabet will use only finitely many. A T-language over a finite alphabet Σ is defined to be any subset $L \subseteq \mathsf{T}\Sigma$. Notions of recognisability are inherited from colorings, using the characteristic function of a language.

The Myhill-Nerode Theorem. We present a monad generalisation of the Myhill-Nerode theorem. That is, we give a sufficient condition for colorings, and therefore also languages, to have a syntactic (i.e. minimal) morphism. The generalisation only works in the setting of sorted sets, and therefore also in the setting of normal sets. Fix the setting of sorted sets, for some choice of, possibly infinitely many, sort names.

Define a (possibly sorted) set A to be *finitary* if for every $w \in \mathsf{T}A$, there is some finite $A_w \subseteq A$ such that $w \in \mathsf{T}A_w$. A monad is called finitary if every set is finitary, e.g. this is the case for the monad of finite words.

Theorem 1.1. *[Syntactic Morphism Theorem] Consider a monad* T *in the setting of sorted sets. Let* f *be a coloring of a* T-*algebra* **A***, which is recognised by a* T-*morphism* h *into some* T-*algebra with finitary universe. There exists a surjective* T-*morphism into a* T-*algebra*

$$\mathsf{synt}f : \mathbf{A} \to \mathbf{A}_f,$$

called the syntactic morphism of f, *which recognises* f *and which factors through every surjective* T-*morphism recognising* f. *Furthermore,* $\mathsf{synt}f$ *is unique up to isomorphisms on* $\mathbf{A}_f$.

If **A** itself has finitary universe, then f is recognised by the identity T-morphism on **A**. Therefore, if the monad is finitary, then every T-language has

a syntactic morphism. This covers monads for objects such as finite words, ultimately periodic words, or various kinds of finite trees. In monads describing truly infinite objects, e.g. the monad for ∞-words used in the running example, a syntactic morphism might not exist.

Running Example 2. Consider the following ∞-language

$$L = \{a^{n_1} b a^{n_2} b \cdots : \text{the sequence } n_i \text{ is unbounded, i.e. } \limsup n_i = \infty.\}$$

One can show that this language does not have a syntactic morphism, not even if the target algebra is allowed to have infinite universe. The idea is that for every n, there is a recognising morphism which identifies all words in $\{a^1, a^2, \ldots, a^n\}$, but there is no morphism which identifies all words in a^+. $\square$

The Eilenberg Theorem. Another result which can be stated and proved on the level of monads is Eilenberg's pseudovariety theorem. This theorem says that, in the case of semigroups, language pseudovarieties and algebra pseudovarieties, which will be defined below, are in bijective correspondence. The theorem implies that if $\mathbb{L}$ is a language pseudovariety, then the membership problem $L \in \mathbb{L}$ can be decided only by looking at the syntactic semigroup of L, and one need not look at the accepting set, nor at the information about which letters are mapped to which elements of the semigroup.

Surely Eilenberg must have known that the pseudovariety works for monads in general and not just for monoids and semigroups, since he invented both the pseudovariety theorem and algebras in abstract monads. However, such a generalisation is not in his book [Eil74]. The generalisation subsumes pseudovariety theorems for: finite words in both monoid and semigroup variants [Eil74], ∞-words [Wil91], scattered linear orderings [BR12], finite trees [Ste92].

Fix a setting where the category is sets with finitely many sorts. Assume that the notion of finite T-algebra is simply that the universe is finite, and a finite alphabet is one with finitely many elements. A T-algebra with a finite universe is finitary, and therefore every T-recognisable language has a syntactic algebra by the Syntactic Morphism Theorem. Define a *derivative*[1] of a T-recognisable T-language $L \subseteq \mathsf{T}\Sigma$ to be any other subset of $\mathsf{T}\Sigma$ that is recognised by the syntactic algebra of L. Define a T-*language pseudovariety* to be a class of T-recognisable T-languages which is closed under Boolean combinations, derivatives, and pre-images under T-morphisms. Since complementation is a form of derivative, Boolean combinations could be replaced by unions in the definition of a T-language pseudovariety. As usual for pseudovarieties, a T-language is formally treated as its characteristic function, which means that a language comes with a description of its input alphabet. Define a T-*algebra pseudovariety* to

---

[1] This notion of derivative is nonstandard. For example, in the case of finite words, the more accepted notion is that a derivative of L is any language of the form $w^{-1}Lv^{-1}$. Because of this nonstandard definition, Theorem 1.2 does not directly generalise the classical Pseudovariety Theorem by Eilenberg. This is discussed in the full version, which contains a proper generalisation of the classical Pseudovariety Theorem.

be a class of finite T-algebras which is closed under finite products, images of surjective T-morphisms, and subalgebras.

For a class $\mathbb{L}$ of recognisable T-languages, define $\mathsf{Alg}\,\mathbb{L}$ to be the class of finite T-algebras that only recognise T-languages from $\mathbb{L}$. For a class $\mathbb{A}$ of finite T-algebras, define $\mathsf{Lan}\,\mathbb{A}$ to be the T-languages recognised by T-algebras from $\mathbb{A}$. The Pseudovariety theorem says that these mappings are mutual inverses.

Theorem 1.2. *The mappings* Lan *and* Alg *are mutual inverses, when restricted to pseudovarieties.*

Running Example 3. Call an ∞-language *definite* if there is some $n \in \mathbb{N}$ such that membership in the language depends only on the first n letters. Examples of definite languages include: "words that begin with a", or "words of length at least two". Call an ∞-algebra **A** *definite* if there is some $n \in \mathbb{N}$ such that

$$\mathrm{mul}_{\mathbf{A}}(x_1 \cdots x_n x_{n+1}) = \mathrm{mul}_{\mathbf{A}}(x_1 \cdots x_n)$$

holds for every $x_1, \ldots, x_{n+1}$ in the universe of the algebra. It is not difficult to show that a recognisable ∞-language is definite if and only if its syntactic algebra is definite, definite ∞-languages form a language pseudovariety, definite ∞-algebras form an algebra pseudovariety, and the two are in correspondence as in the Pseudovariety Theorem. $\square$

2 Deciding Monadic Second-Order Logic

An important part of language theory is the equivalence of recognisability and definability in monadic second-order logic MSO. Examples where this equivalence holds include finite words and trees, and more interestingly from a combinatorial perspective, infinite words and trees. There are common parts in all of the proofs, and parts that are specific to each domain. We show that the common parts can be stated and proved on the abstract level of monads.

Representing an algebra. To give an algorithm for MSO satisfiability that uses algebras, one needs a representation of finite algebras so that they can be manipulated by algorithms. We propose such a representation; this is the main part of this section. Fix a setting for the rest of this section, with the category being sets, possibly sorted, but with finitely many sorts.

In most interesting cases, the monad T produces infinite sets, even on finite arguments. Therefore, the finiteness of the universe of a T-algebra **A** does not, on its own, imply that the algebra has a finite representation, because one needs to also represent its multiplication operation, whose type is $\mathsf{T}A \to A$. To represent algebras, we will use an assumption, to made more precise below, which roughly says that if an algebra has finite universe A, then the the multiplication is determined by its values on a small finite subset of $\mathsf{T}A$. For instance, in the monad of finite words from Example 1, one chooses from A^+ only the length

two words, because a +-algebra, i.e. a semigroup, is uniquely determined by its binary multiplication. We now describe these notions in more detail.

Define a *subfunctor* of T to be a mapping T_0 which takes each set X to a subset $\mathsf{T}_0 X \subseteq \mathsf{T} X$. A subfunctor on its own is not a monad, however it can be used to generate a monad as follows. For a set X, define $\mathsf{T}_0^* X$ to be the least set which contains the units of X, and which also contains any multiplication (flattening) of an element in $\mathsf{T}_0 \mathsf{T}_0^* X$. It is not difficult to show that T_0^* is a submonad of T, i.e. a subfunctor with a monad structure as inherited from T. A subfunctor T_0 is said to *span* a T-algebra $\mathbf{A}$ if for every subset X of the universe, $\mathrm{mul}_\mathbf{A}$ has the same image over $\mathsf{T}_0^* X$ and over $\mathsf{T} X$, i.e.

$$\mathrm{mul}_\mathbf{A} \mathsf{T}_0^* X = \mathrm{mul}_\mathbf{A} \mathsf{T} X.$$

A subfunctor is *complete* if it spans every T-algebra, and *finitely complete* if it spans every finite T-algebra; the latter depends on the notion of finite T-algebra.

Consider a subfunctor T_0 that is finitely complete for a monad T. For a finite T-algebra $\mathbf{A}$, define its T_0-*reduct* to be the pair consisting of the universe A of $\mathbf{A}$, and the restriction of the multiplication operation from $\mathbf{A}$ to the subfunctor:

$$\mathrm{mul}_\mathbf{A}|_{\mathsf{T}_0 A} : \mathsf{T}_0 A \to A$$

It is easy to show, using associativity, that if T_0 spans $\mathbf{A}$, then $\mathbf{A}$ is uniquely determined by its T_0-reduct. In particular, if T_0 is complete, then every operation $\mathsf{T}_0 A \to A$ extends to at most one T-algebra with universe A. Note the "at most one", e.g. not every binary operation extends to a semigroup operation, for this associativity is needed. The same holds for finite completeness and finite algebras. The point of using T_0-reducts is that sometimes T_0 can be chosen so that it preserves finiteness, and therefore finite T-algebras can be represented in a finite way as functions $\mathsf{T}_0 A \to A$.

Running Example 4. As in [Wil91], one can use the Ramsey theorem to show that a finite ∞-algebra is uniquely determined by the values of its multiplication operation on arguments of the form xy and x^ω. Stated differently,

$$\mathsf{W} X \quad \stackrel{\mathrm{def}}{=} \quad \{xy, x^\omega : x, y \in X\} \subseteq X^\infty.$$

is finitely complete subfunctor of the ∞-functor. The submonad W^* maps an alphabet X to the finite and ultimately periodic words over X. A W-reduct of a finite ∞-algebra is essentially the same thing as a Wilke semigroup, modulo the difference that Wilke semigroups are two-sorted. In [Wil91], Wilke shows axioms which describe when a W-algebra extends to an ∞-algebra. $\square$

A subfunctor T_0 is called *effective* if it satisfies the following two conditions.

1. If X is a finite set then $\mathsf{T}_0 X$ is finite and can be computed. This means that a finite T-algebra with universe A be represented as a function $\mathsf{T}_0 A \to A$.
2. For every finite set Σ and every $w \in \mathsf{T}_0 \Sigma$, one can compute a T-morphism into a finite T-algebra that recognises $\{w\}$, with the T-algebra represented as in item 1, and the T-morphism represented by its values on units of Σ.

The second condition is maybe less natural, it will be used in deciding MSO.

Running Example 5. We claim that the functor W in the running example is effective. For the first condition, the set WX is isomorphic to the disjoint union $X^2 + X$, and can therefore clearly be computed. For the second condition, one needs to show that for every ultimately periodic ∞-word w, there is a finite ∞-algebra that recognises the singleton $\{w\}$, and its W-reduct can be computed. The universe of this algebra consists of suffixes of w, finite infixes modulo repetitions, and an error element. $\square$

Monadic second-order logic. To establish the connection between MSO and recognisability on the level of monads, we use a definition of MSO which does not talk about "positions" or "sets of positions" of a structure, but which is defined in purely language theoretic terms. In the abstract version, predicates, such as the successor predicate, are modelled by languages. For a set $\mathcal{L}$ of T-languages, define MSO$_\mathsf{T}(\mathcal{L})$ to be the smallest class of T-languages which contains $\mathcal{L}$, is closed under Booolean operations, images and inverse images of T-morphisms.

Running Example 6. The standard notion of MSO for ∞-words, as studied by Büchi, is equivalent to MSO$_\mathsf{T}(\mathcal{L})$ where $\mathcal{L}$ contains only two recognisable ∞-languages over alphabet $\{0, 1\}$, namely the language of words which contain only zeros, and the language of words where every one is followed only by ones. $\square$

One can show that if $\mathcal{L}$ contains only T-recognisable T-languages, then so does MSO$_\mathsf{T}(\mathcal{L})$. This uses the assumptions on the setting being sorted sets, and finite algebras being ones with finite universes. A non-example is the category of nominal sets with orbit-finite sets, where powerset does not preserve orbit-finiteness, and also MSO contains non-recognisable languages, see [Boj13].

A language in MSO$_\mathsf{T}(\mathcal{L})$ can be represented as a tree where leaves are languages from $\mathcal{L}$, binary nodes are labelled by union or intersection, complementation is represented by unary nodes, and for every T-morphism $h : \mathsf{T}\Sigma \to \mathsf{T}\Gamma$ there are two kinds of unary nodes, one for image and the other for inverse image. Therefore, if $\mathcal{L}$ is finite (or has some fixed enumeration) then it makes sense to consider the following decision problem, called MSO$_\mathsf{T}(\mathcal{L})$ *satisfiability*: an instance is a tree representing a language from MSO$_\mathsf{T}(\mathcal{L})$, and the question is whether the corresponding language is nonempty. We provide below a sufficient criterion for the decidability the problem.

Theorem 2.1. *Consider a setting with finitely sorted sets, and let $\mathcal{L}$ be all recognisable languages. If there is a subfunctor that is effective and finitely complete, then* MSO$_\mathsf{T}(\mathcal{L})$ *satisfiability is decidable.*

As mentioned at the beginning of the section, the theorem takes care of the symbol pushing part in deciding satisfiability of MSO, and leaves only the combinatorial part. The combinatorial part is finding an effective and finitely complete subfunctor, this is typically done using some kind of Ramsey theorem.

Running Example 7. Applying Theorem 2.1 to the observations made so far in the running example, we conclude that MSO satisfiability over ∞-words is decidable. The proof obtained this way follows the same lines as Büchi's original proof. $\square$

3   Profinite Monads

Profinite constructions are an important tool in the study of recognisable languages. Example applications include: lattices of word languages correspond to implications on profinite words [GGP08], pseudovarieties in universal algebra correspond to profinite identities [Rei82], recognisable word languages are the only ones which give a uniformly continuous multiplication in the profinite extension [GGP10]. In this section we show that these results can be stated and proved on the abstract level of monads, thus covering cases like profinite trees or profinite ∞-words. Furthermore, we show that profinite objects, e.g. profinite words, form a monad as well, which has its own notion of recognisable language, a notion that seems interesting and worthy of further study.

The Stone dual. A short definition of the profinite object uses Stone duality. Define an *ultrafilter* in a Boolean algebra to be a subset of its universe which is closed under $\wedge$, and which contains every element of the Boolean algebra or its complement, but not both. The set of ultrafilters is called the *Stone dual* of the Boolean algebra. The Stone dual also comes with a topological structure, but since we do not use it here, we do not describe it.

From the monad point of view, we are interested in the special case of the Boolean algebra of recognisable languages in a T-algebra. Let **A** be a T-algebra, not necessarily finite. Define the Stone dual of **A**, denoted by StoneA, to be the Stone dual of the Boolean algebra of those subsets of **A** that are recognised by T-morphisms from **A** into finite T-algebras. This definition generalises the well-known space of profinite words, i.e. if the monad is the monad of finite words. Note how the definition depends on the notion of finite algebra.

Running Example 8. For an alphabet Σ and $w \in \Sigma^+$, define $w^\#$ to be the set of languages $L \subseteq \Sigma^\infty$ which are recognised by some finite ∞-algebra and which contain $w^{n!}$ for all but finitely many n. This set is clearly closed under intersection, and a standard pumping argument shows that for every recognisable language L, it contains L or $\Sigma^+ - L$. Therefore $w^\#$ is an ultrafilter, i.e. a profinite ∞-word. A common notation in profinite words would be to use ω, but this would conflict with the infinite power in the context of ∞-words. □

Defining pseudovarieties by identities. Our first result on Stone duals is a monad generalisation of the Reiterman Theorem [Rei82]. The original Reiterman Theorem, which uses terminology of universal algebra, says that pseudovarieties can be characterised by profinite identities. In the full version, we present a notion of profinite identity, and prove the following theorem.

Theorem 3.1. *Let $\mathbb{L}$ be class of recognisable T-languages. Then $\mathbb{L}$ is a pseudovariety if and only if it is defined by a set of profinite identities.*

Running Example 9. Recall the pseudovariety of definite ∞-languages. This pseudovariety is defined by a single profinite identity, namely

$$x^\omega = x^\#.$$

One proves this the same way as in the classical result on definite languages of finite words, which are characterised by the identity $x^\# y = x^\#$. The latter identity is implied by the former, because $x^\omega y = x^\omega$ is true in all ∞-words. □

The full version contains other results on Stone duals, in particular monad generalisations of results from [GGP08] and [GGP10].

A profinite monad. We now explain how to convert a monad T into another monad, called $\overline{\mathsf{T}}$, that describes profinite objects over T. The functor $\overline{\mathsf{T}}$ maps a set Σ to the Stone dual of the T-algebra $\mathsf{T}\Sigma$. For a function $f : \Sigma \to \Gamma$, the mapping $\overline{\mathsf{T}}f$ takes an ultrafilter $U \in \overline{\mathsf{T}}\Sigma$ to the set

$$V = \{L \subseteq \mathsf{T}\Gamma : (\mathsf{T}f)^{-1}(L) \in U\}$$

which is easily seen to be an ultrafilter, and therefore an element of $\overline{\mathsf{T}}\Gamma$. This definition is a special case of the functor in the classical theorem on duality of Boolean algebras and Stone spaces; in particular $\overline{\mathsf{T}}$ is a functor. It remains to provide the monad structure, namely the multiplication and unit. These will be given in Theorem 3.2, using the following notion of profinite completion. Define the *profinite completion* of a T-morphism $h : \mathsf{T}\Sigma \to \mathbf{A}$ into a finite T-algebra to be the mapping

$$\bar{h} : \overline{\mathsf{T}}\Sigma \to A$$

defined as follows: an ultrafilter U is mapped to the unique element a in the universe of $\mathbf{A}$ such that the ultrafilter contains the language $h^{-1}(a)$.

Theorem 3.2. *For every set Σ, there are unique operations*

$$\bar{\eta}_\Sigma : \Sigma \to \overline{\mathsf{T}}\Sigma \qquad \bar{\mu}_\Sigma : \overline{\mathsf{T}}\,\overline{\mathsf{T}}\Sigma \to \overline{\mathsf{T}}\Sigma$$

such that for every finite T-algebra $\mathbf{A}$ and every T-morphism $h : \mathsf{T}\Sigma \to \mathbf{A}$, the following diagrams commute

Furthermore, equipped with the above operations, $\overline{\mathsf{T}}$ is a monad.

What is the benefit of seeing $\overline{\mathsf{T}}$ as a monad? Since $\overline{\mathsf{T}}$ is itself a monad, it can have its own notion of finite algebra and recognisable language. We illustrate how these notions can be interesting, using the special case of profinite words. Consider the monad $\Sigma \mapsto \Sigma^+$ for finite words. Let us denote by $\Sigma \mapsto \Sigma^{\bar{+}}$ the profinite version of this monad. An element of $\Sigma^{\bar{+}}$ is a simply profinite word over the alphabet Σ. One way of creating a finite $\bar{+}$-algebra is to take a finite $+$-algebra, i.e. a finite semigroup, and extend its multiplication operation to

profinite words using profinite completion. More interesting $\bar{+}$-algebras are not obtained this way, here is one example.

Example 2. We say that a profinite word $w \in \{0,1\}^{\bar{+}}$ has *exactly n ones* if, when seen as an ultrafilter, it contains the recognisable language of words with at least n ones. If a profinite word has exactly n ones for some n, then we say that it has *a bounded number* of ones. For example, $1^{\#}$ does not have a bounded number of ones. A certain amount of calculations shows that the set of profinite words in $\{0,1\}^{\bar{+}}$ which have a bounded number of ones is recognised by a $\bar{+}$-morphism into a finite $\bar{+}$-algebra. The finite $\bar{+}$-algebra has three elements in its universe, standing for: "only zeros", "a bounded number of ones", and "not a bounded number of ones".

Let us define MSO+inf by applying the abstract notion of MSO defined in Section 2, with the predicates being the following languages of profinite words over alphabet $\{0,1\}$: "a bounded number of ones", "only zeros", "every one is followed by only zeros". Under a different terminology, this class of languages was considered in [Tor12]. Since the operators of MSO preserve recognisability, it follows that MSO+inf contains only $\bar{+}$-recognisable languages. It is not clear if MSO+inf contains all $\bar{+}$-recognisable languages, but Corollary 2 from [Tor12] and a new undecidability result from [MB15] imply that MSO+inf has undecidable satisfiability. $\square$

References

[Boj13] Bojanczyk, M.: Nominal monoids. Theory Comput. Syst. **53**(2), 194–222 (2013)

[BR12] Bedon, N., Rispal, C.: Schützenberger and Eilenberg theorems for words on linear orderings. J. Comput. Syst. Sci. **78**(2), 517–536 (2012)

[BW08] Bojanczyk, M., Walukiewicz, I.: Forest algebras. In: Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas], pp. 107–132 (2008)

[CCP11] Carton, O., Colcombet, T., Puppis, G.: Regular languages of words over countable linear orderings. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part II. LNCS, vol. 6756, pp. 125–136. Springer, Heidelberg (2011)

[Eil74] Eilenberg, S.: Automata, languages, and machines, vol. A (1974)

[Ési10] Ésik, Z.: Axiomatizing the equational theory of regular tree languages. The Journal of Logic and Algebraic Programming **79**(2), 189–213 (2010)

[ÉW03] Ésik, Z., Weil, P.: On logically defined recognizable tree languages. In: Pandya, P.K., Radhakrishnan, J. (eds.) FSTTCS 2003. LNCS, vol. 2914, pp. 195–207. Springer, Heidelberg (2003)

[GGP08] Gehrke, M., Grigorieff, S., Pin, J.É.: Duality and equational theory of regular languages. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 246–257. Springer, Heidelberg (2008)

[GGP10] Gehrke, M., Grigorieff, S., Pin, J.É.: A topological approach to recognition. In: Abramsky, S., Gavoille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP 2010. LNCS, vol. 6199, pp. 151–162. Springer, Heidelberg (2010)

[MB15] Toruńczyk, S., Bojańczyk, M., Parys, P.: The MSO+U theory of $(\mathbb{N}, <)$ is undecidable (2015). CoRR, arXiv:1502.04578

[PP04] Perrin, D., Pin, J.-É.: Infinite Words: Automata, Semigroups, Logic and Games. Elsevier (2004)

[Rei82] Reiterman, J.: The Birkhoff theorem for finite algebras. Algebra Universalis **14**(1), 1–10 (1982)

[Ste92] Steinby, M.: A theory of tree language varieties. In: Tree Automata and Languages, pp. 57–82 (1992)

[Tor12] Toruńczyk, S.: Languages of profinite words and the limitedness problem. In: Czumaj, A., Mehlhorn, K., Pitts, A., Wattenhofer, R. (eds.) ICALP 2012, Part II. LNCS, vol. 7392, pp. 377–389. Springer, Heidelberg (2012)

[Wil91] Wilke, T.: An Eilenberg theorem for infinity-languages. In: Leach Albert, J., Monien, B., Rodríguez-Artalejo, M. (eds.) ICALP 1991. LNCS, vol. 510, pp. 588–599. Springer, Heidelberg (1991)