The Algebras for Automatic Relations

Rémi Morvan
LaBRI, Univ. Bordeaux, CNRS & Bordeaux INP, France

Abstract

We introduce "synchronous algebras", an algebraic structure tailored to recognize automatic relations (*a.k.a.* synchronous relations, or regular relations). They are the equivalent of monoids for regular languages, however they conceptually differ in two points: first, they are typed and second, they are equipped with a dependency relation expressing constraints between elements of different types.

The interest of the proposed definition is that it allows to lift, in an effective way, pseudovarieties of regular languages to that of synchronous relations, and we show how algebraic characterizations of pseudovarieties of regular languages can be lifted to the pseudovarieties of synchronous relations that they induce. Since this construction is effective, this implies that the membership problem is decidable for (infinitely) many natural classes of automatic relations. A typical example of such a pseudovariety is the class of "group relations", defined as the relations recognized by finite-state synchronous permutation automata.

In order to prove this result, we adapt two pillars of algebraic language theory to synchronous algebras: (a) any relation admits a syntactic synchronous algebra recognizing it, and moreover, the relation is synchronous if, and only if, its syntactic algebra is finite and (b) classes of synchronous relations with desirable closure properties (*i.e.* pseudovarieties) correspond to pseudovarieties of synchronous algebras.

2012 ACM Subject Classification Theory of computation → Algebraic language theory

Keywords and phrases synchronous automata, automatic relations, regular relations, transductions, synchronous algebras, Eilenberg correspondence, pseudovarieties, algebraic characterizations

Digital Object Identifier 10.4230/LIPIcs.CSL.2025.21

Related Version *Full Version*: `https://arxiv.org/abs/2404.15496`

Acknowledgements We thank Pablo Barceló, Mikołaj Bojańczyk, and Diego Figueira for helpful discussions, and some anonymous reviewers for valuable feedback.

❧ This pdf contains internal links: clicking on a notion leads to its *definition*.

1 Introduction

1.1 Background

The landscape of rationality for k-ary relations of finite words ($k \geq 2$) is far more complex than for languages – recall that languages can be seen as unary relations of finite words – as depicted in Figure 4 on page 20. Perhaps the most natural class is that of *rational relations*, defined as relations accepted by non-deterministic two-tape automata – an input (u, v) is described by writing u on the first tape and v and the second tape – that can move its two heads independently, from left to right – see [13, §2.1] for a formal definition. For instance, the suffix relation is rational.

Our paper focuses on synchronous relations, *a.k.a.* automatic relations or regular relations, defined as the rational relations that can be recognized by synchronous automata, a subclass of the machines described above obtained by keeping a single head that moves synchronously from left to right, reading one pair of letters after the other; we add padding symbols $\_$ at the end of the shorter word – see Figure 1. While the suffix relation is not synchronous, typical examples include the prefix relation, the same-length relation, etc. Synchronous relations play a central role in the definitions of automatic structures – introduced by Hodgson [23, 24, 25] and rediscovered by Khoussainov & Nerode [26], see [7, §XI, pp. 627–762]. They also have been studied in the context of graph databases [5, Definition 3.1, p.7 & Theorem 6.3, p. 13], see [18, §8, p. 17] for more context & results on *extended* conjunctive regular path queries.

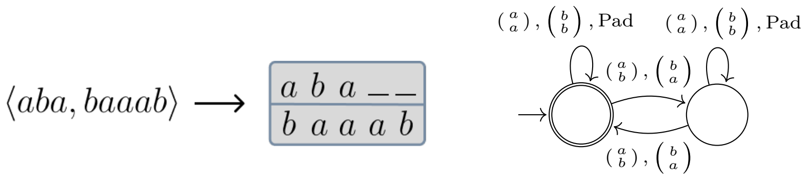


Figure 1 Encoding a pair of words of $\Sigma^* \times \Sigma^*$ into an element of $(\Sigma^2_{\_})^*$ where $\Sigma^2_{\_} \triangleq (\Sigma \times \Sigma) \cup (\Sigma \times \{\_\}) \cup (\{\_\} \times \Sigma)$ (left) and a deterministic complete synchronous automaton (right) over $\Sigma = \{a, b\}$ accepting the binary relation of pairs (u, v) such that the number of a's in $u_1 \dots u_k$ and in $v_1 \dots v_k$ are the same mod 2, where $k = \min(|u|, |v|)$. Pad denotes the set of transitions $\{\binom{a}{\_}, \binom{b}{\_}, \binom{\_}{a}, \binom{\_}{b}\}$.

▶ Remark 1.1. All our results are described for binary relations, but can be extended to k-ary synchronous relations, see Section 5.

Synchronous relations stand at the frontier between expressiveness and undecidability: for instance, Carton, Choffrut and Grigorieff showed that it is decidable whether an automatic relation is *recognizable* [13, Proposition 3.9, p. 265], meaning that it can be written as a finite union of Cartesian products of regular languages.[1,2] Synchronous relations are effectively closed under Boolean operations – see *e.g.* [7, Lemma XI.1.3, p. 627], and moreover, inclusion (and subsequent problems: universality, emptiness, equivalence…) is decidable for them, by reduction to classical automata, contrary to the equivalence problem over rational relations which is undecidable [6, Theorem 8.4, p. 81].

However, some seemingly easy problems are undecidable: Köcher showed that it is undecidable if the (infinite) graph defined by a synchronous relation is 2-colourable – [28, Proposition 6.5, p. 43], and Barceló, Figueira and Morvan showed that undecidability also holds for regular 2-colourability [3, Theorem 4.4, p. 8]. On the other hand, one can decide if said graph contains an infinite clique, see [27, Corollary 5.5, p. 32]: this is a consequence of [35, Theorem 3.20, p. 185].

1.2 Motivation

Any synchronous relation can be seen as a regular language over the alphabet $\Sigma^2_{\_} \triangleq (\Sigma \times \Sigma) \cup (\Sigma \times \{\_\}) \cup (\{\_\} \times \Sigma)$ of pairs. On the other hand any regular language L over $\Sigma^2_{\_}$ produces a synchronous relation when intersected with the language of all well-formed words – namely words where the padding symbols are consistently placed; see Section 2 for precise definitions. In fact, the semantics of synchronous automata such as the one in Figure 1 is precisely defined this way: it is the intersection of the "classical semantic" of the automaton, seen as an NFA, intersected with well-formed words.

[1] For instance, the relation "having the same length modulo 2" is recognizable, since it can be written as $(aa)^* \times (aa)^* \cup a(aa)^* \times a(aa)^*$.
[2] The problem was latter shown to be NL-complete and PSpace-complete depending on whether the input automaton is deterministic or not in [4, Theorem 1, p. 3].

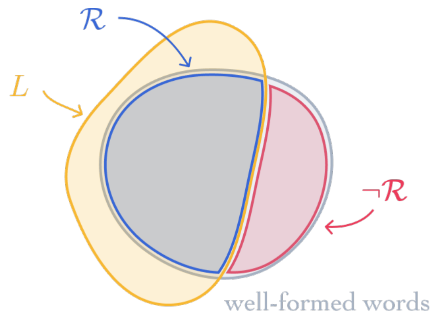


Figure 2 Drawing in $(\Sigma_{\_}^2)^*$ of a $\mathcal{V}$-relation $\mathcal{R}$ and $\neg\mathcal{R} \mathrel{\hat=} \{(u,v) \in \Sigma^* \times \Sigma^* \mid (u,v) \notin \mathcal{R}\}$, where $\mathcal{R}$ is defined as $L \cap \mathsf{WellFormed}_\Sigma$ with $L \in \mathcal{V}$.

In particular, a class $\mathcal{V}$ of regular languages over $\Sigma_{\_}^2$ (*e.g.* first-order definable languages, group languages, etc.) induces a class of so-called $\mathcal{V}$-relations, defined as the relations over Σ obtained as the intersection of some language of $\mathcal{V}$ with well-formed words, see Figure 2. For instance, the relation of Figure 1 is a $\mathcal{V}$-relation where $\mathcal{V}$ is the class of all group languages – these relations can be alternatively described as those recognized by a deterministic complete synchronous automaton whose transitions functions are permutations of states.

▶ **Question 1.2.** Given a class $\mathcal{V}$ of languages, can we characterize and decide the class of $\mathcal{V}$-relations?

As we will see in Example 2.4, for a relation to be $\mathcal{V}_{\Sigma_{\_}^2}$ is not necessary for it to be a $\mathcal{V}$-relation.

1.3 Contributions

We answer positively to this question. For this we first need to develop an algebraic theory of synchronous relations, which enables us to prove the lifting theorem. In short, the lifting theorem states that algebraic characterizations of classes of word languages can be lifted in a canonical way to algebraic characterizations of classes of word relations.

The algebraic approach usually provides more than decidability: it attaches canonical algebras to languages/relations (*e.g.* monoids for languages of finite words), and often simple ways to characterize complex properties (*e.g.* first-order definability, see *e.g.* [10, Theorem 2.6, p. 40]). Our synchronous algebras differ from monoids in two points:

- they are typed – a quite common feature in algebraic language theory, shared *e.g.* by ω-semigroups [29, §4.1, p. 91];
- they are equipped with a dependency relation, which expresses constraints between elements of different types – to our knowledge, this feature is entirely novel.[3]

Importantly, some variations are possible on the definition of synchronous algebras: in particular, one could get rid of the notion of dependency relation and Lemmas 3.11 and 4.7 would still hold. However, we show in the full version that these simplified synchronous algebras cannot characterize the property of being a $\mathcal{V}$-relation. Therefore, the notion of

[3] Note that algebras equipped with binary relations have been studied before, *e.g.* Pin's ordered ω-semigroups – see [30, §2.4, p. 7] – but the constraints (here the orderings) are always defined between elements of the *same type*.

dependency seems necessary to tackle Question 1.2. Moreover, we show that these algebras arise from a monad, but to our knowledge none of the meta-theorems developing algebraic language theories over monads apply to it, see the full version for more details.

We show that assuming that $\mathcal{V}$ is a $*$-pseudovariety of regular languages – in short, a class of regular languages with desirable closure properties – , then the algebraic characterization of $\mathcal{V}$ can be easily lifted to characterize $\mathcal{V}$-relations.

▶ **Theorem 4.2** (Lifting theorem: Elementary Formulation). *Given a relation $\mathcal{R}$ and a $*$-pseudovariety of regular languages $\mathcal{V}$ corresponding to a pseudovariety of monoids $\mathbb{V}$, the following are equivalent:*

1. *$\mathcal{R}$ is a $\mathcal{V}$-relation,*
2. *$\mathcal{R}$ is recognized by a finite synchronous algebra $\mathbf{A}$ whose underlying monoids are all in $\mathbb{V}$,*
3. *all underlying monoids of the syntactic synchronous algebras $\mathbf{A}_{\mathcal{R}}$ of $\mathcal{R}$ are in $\mathbb{V}$.*

This theorem rests on a solid algebraic theory. First, we show the existence of syntactic algebras (Lemma 3.11): each relation $\mathcal{R}$ admits a unique canonical and minimal algebra $\mathbf{A}_{\mathcal{R}}$, which is finite *iff* the relation is synchronous, and then, we exhibit a correspondence between classes of finite algebras and classes of synchronous relations (Lemma 4.7) – we assume suitable closure properties; these classes are called "pseudovarieties". While the proof structures of Lemmas 3.11 and 4.7 follow the classic proofs, see *e.g.* [31], the dependency relation has to be taken into account quite carefully, leading for instance to a surprising definition of residuals, see Definition 4.5.

Organization. After giving preliminary results in Section 2, we introduce the synchronous algebras in Section 3 and show the existence of syntactic algebras. We then proceed to prove the lifting theorem for $*$-pseudovarieties in Section 4, and after introducing $*$-pseudovarieties of synchronous relations, we provide a more algebraic reformulation of the lifting theorem (Theorem 4.9). We conclude the paper with a short discussion in Section 5.

1.4 Related Work

The algebraic framework has been extended far beyond languages of finite words: let us cite amongst other Reutenauer's "algèbre associative syntactique" for weighted languages [33, Théorème I.2.1, p. 451] and their associated Eilenberg theorem [33, Théorème III.1.1, p. 469]; for languages of ω-words, Wilke's algebras and ω-semigroups, see [29, §II, pp. 75–131 & §VI, pp. 265–306]; more generally, for languages over countable linear orderings, see Carton, Colcombet & Puppis' "$\circledast$-monoids" and "$\circledast$-algebras" [14, §3, p. 7]. A systemic approach has been recently developed using monads, see the full version. Non-linear structures are also suited to such an approach, see *e.g.* Bojańczyk & Walukiewicz's forest algebras [11, §1.3, p. 4] [10, §5, p. 159], or Engelfriet's hyperedge replacement algebras for graph languages [15, §2.3, p. 100] [9, §6.2, p. 194]. For relations over words (*a.k.a.* transductions), recognizable relations are exactly the ones recognized by monoid morphisms $\Sigma^* \times \Sigma^* \to M$ where M is finite. This can be trivially generalized to show that a relation $\mathcal{R}$ is a finite union of Cartesian products of languages in $\mathcal{V}$ if, and only if, it is recognized by a monoid from $\mathbb{V}$, the pseudovariety of monoids corresponding to $\mathcal{V}$, see the full version. In 2023, Bojańczyk & Nguyễn managed to develop an algebraic structure called "transducer semigroups" for "regular functions" [8, Theorem 3.2, p. 6], an orthogonal class of relations to ours – see Figure 4.

The counterpart of $\mathcal{V}$-relations for rational relations – that we call here $\mathcal{V}$-rational relations – was studied by Filiot, Gauwin & Lhote [20]: they show that if $\mathcal{V}$ has decidable membership, then "$\mathcal{V}$-rational transductions" also have decidable membership [20, Theorem 4.10, p. 26]. "Rational transductions" correspond in Figure 4 to the intersection of functional relations with rational relations: this class is orthogonal to synchronous relations, but is included in the class of "regular functions". A different problem – focussing more on the semantics of the transduction – , called "$\mathcal{V}$-continuity" was studied by Cadilhac, Carton & Paperman [12, Theorem 1.3, p. 3], although it has to be noted that their results only concern a finite number of pseudovarieties.

2 Preliminaries

2.1 Automata & Relations

We assume familiarity with basic algebraic language theory over finite words, see [10, §1, 2, 4, pp. 3–66 & pp. 107–156] for a succinct and monad-driven approach, or [31, §I–XIV, pp. 3–247] for a more detailed presentation of the domain. We also refer to [36] for a presentation on pseudovarieties.[4] More precise pointers are given in the full version.

A *relation* is a subset of $\Sigma^* \times \Sigma^*$, where Σ is an alphabet – *i.e.* a non-empty finite set. We define its *complement* $\neg\mathcal{R}$ as the relation $\{(u,v) \in \Sigma^* \times \Sigma^* \mid (u,v) \notin \mathcal{R}\}$. Letting $\Sigma^2_{\_} \mathrel{\hat=} (\Sigma \times \Sigma) \cup (\Sigma \times \{\_\}) \cup (\{\_\} \times \Sigma)$, a *synchronous automaton* is a finite-state machine with initial states, final states, and non-deterministic transitions labelled by elements of $\Sigma^2_{\_}$. We denote by $\mathsf{WellFormed}_\Sigma$ the set of *well-formed* words over $\Sigma^2_{\_}$ where the padding symbols are placed consistently, namely: if some padding symbol occurs on a tape/component, then the following symbols of this tape/component must all be padding symbols. From this constraint, and since $\binom{\_}{\_} \notin \Sigma^2_{\_}$, there can never be padding symbols on both tapes.

Note that elements of $\mathsf{WellFormed}_\Sigma$ are in natural bijection with $\Sigma^* \times \Sigma^*$ – see Figure 1. The relation recognized by a synchronous automaton is the set of pairs $(u,v) \in \Sigma^* \times \Sigma^*$ such that their corresponding element in $\mathsf{WellFormed}_\Sigma$ is the label of an accepting run of the automaton. We say that a relation is *synchronous* if it is recognized by such a machine.

▶ Remark 2.1. Crucially, in the semantics of synchronous automata we *never* try to feed them inputs where the padding symbols are not consistent: for instance, while

$$\binom{aab}{b\_a}, \text{ or } \binom{aba\_}{a\_\_b}$$

are sequences in $(\Sigma^2_{\_})^*$, the behaviour of a synchronous automaton on such sequences is completely disregarded to define the relation it recognizes.

We can then reformulate the definition of the semantics of a synchronous automaton, to make the connection with $\mathcal{V}$-relations – see the next subsection – explicit.

▶ **Fact 2.2.** *Given a synchronous automaton, its semantics as a synchronous automaton can be written as the intersection of its semantics as a classical automaton over $\Sigma^2_{\_}$ with $\mathsf{WellFormed}_\Sigma$.*

In particular a relation $\mathcal{R}$ is synchronous if, and only if, it is a regular language when seen as a subset of $(\Sigma^2_{\_})^*$.

[4] "Pseudovarieties of *foo*" and "varieties of finite *foo*" – where *foo* is *e.g.* "groups" or "semigroups" – are used interchangeably in the literature.

2.2 Induced Relations

Given a class $\mathcal{V}$ of regular languages, the class of *$\mathcal{V}$-relations* over Σ consists of all relations of the form $L \cap \mathsf{WellFormed}_\Sigma$ for some $L \in \mathcal{V}_{\Sigma^2_-}$ – see Figure 2.[5]

For instance, if $\mathcal{V}$ is the class of all regular languages, then by Fact 2.2, $\mathcal{V}$-relations are exactly the regular relations, *a.k.a.* synchronous relations! However, because of Remark 2.1, the minimal automaton for a relation, seen as a language over Σ^2_-, can be significantly more complex than a deterministic complete synchronous automaton recognizing it, see Figure 3 in page 19 – while the size blow-up is only polynomial, it breaks many of the structural properties of the automaton, such as the property of being a permutation automaton.

Note that if $\mathcal{R}$ belongs to $\mathcal{V}$ when $\mathcal{R}$ is seen as a language over Σ^2_-, then $\mathcal{R}$ is a $\mathcal{V}$-relation. The converse implication holds under some strong assumption on $\mathcal{V}$ (Fact 2.3), but is not true in general (Example 2.4).

▶ **Fact 2.3.** *If $\mathcal{V}$ is a class of languages closed under intersection and that contains $\mathsf{WellFormed}_\Sigma$, then a relation $\mathcal{R}$ is a $\mathcal{V}$-relation if, and only if, it belongs to $\mathcal{V}$ when seen as a language over Σ^2_-.*

Classes of languages $\mathcal{V}$ satisfying the previous assumption (*e.g.* first-order definable languages, piecewise-testable languages, etc.) are easy to capture when it comes to $\mathcal{V}$-relations since this class reduces to $\mathcal{V}$-languages. So, in the remaining of the paper, we will focus on classes $\mathcal{V}$ which do not satisfy the assumptions of Fact 2.3, such as group languages.

▶ **Example 2.4** (Group relations). If $\mathcal{V}$ is the class of group languages, namely languages recognized by permutation automata[6] or equivalently by a finite group, then we call $\mathcal{V}$-relations "*group relations*". They can be characterized as relations recognized by permutation synchronous automata. For instance, the relation of Figure 1 is a group relation as witnessed by the permutation synchronous automaton of Figure 1. Note however that it is not a group language, when seen as a language over Σ^2_-, since its minimal automaton over Σ^2_- is not a permutation automaton, see Figure 3 on Page 19.

▶ **Fact 2.5.** *Given a relation $\mathcal{R}$ and a class $\mathcal{V}$ of languages, the following are equivalent:*

1. *$\mathcal{R}$ is a $\mathcal{V}$-relation;*
2. *$\mathcal{R}$ and $\neg\mathcal{R}$ are $\mathcal{V}$-separable as languages over Σ^2_-, i.e. there is a language in $\mathcal{V}$ which contains $\mathcal{R}$ and does not intersect $\neg\mathcal{R}$.*

Proof. By definition, see Figure 2, on page 3. ◀

And so, if the $\mathcal{V}$-separability problem is decidable, then the class of $\mathcal{V}$-relations is decidable. However, there are pseudovarieties $\mathcal{V}$ with decidable membership but undecidable separability problem [34, Corollary 1.6, p. 478].[7] Moreover, some of these classes do not contain $\mathsf{WellFormed}_\Sigma$ [34, Corollary 1.7, p. 478]. But beyond this, even when a separation algorithm exists, it can be conceptually much harder than its membership counterpart: for

---

[5] The notation $L \in \mathcal{V}_{\Sigma^2_-}$ means that L is a language over the alphabet Σ^2_-. See [31, introduction of §XIII.1] for why classes of regular languages are defined in such a way.

[6] A permutation automaton is a finite-state deterministic complete automaton whose transition functions are all permutations of states.

[7] The paper cited only claims undecidability of pointlikes, but it was noted in [21, §1, pp. 1–2] that undecidability of the 2-pointlikes also holds, which is a problem equivalent to separability by [1, Proposition 3.4, p. 6].

instance, deciding membership for group languages is trivial – it boils down to checking if a monoid is a group – , yet the decidability of the separation problem for group languages is considered to be one of the major results in semigroup theory: it follows from Ash's infamous type II theorem [2, Theorem 2.1, p. 129], see [22, Theorem 1.1, p. 3] for a presentation of the result in terms of pointlike sets, see also [32, §III, Theorem 8, p. 5] for an elegant automata-theoretic reformulation.

3 Synchronous Algebras

In this section, we introduce and study the "elementary" properties of synchronous algebras.

3.1 Types & dependent Sets

Motivation. The axiomatization of a semigroup reflects the algebraic structure of finite words: these objects can be concatenated, in an associative way – reflecting the linearity of words. Now observe that elements of $\mathsf{WellFormed}_\Sigma$ are still linear, but not all words can be concatenated together: for instance, $\binom{a}{\_}$ cannot be followed by $\binom{a}{b}$. Formally, given two words $u, v \in \mathsf{WellFormed}_\Sigma$, to decide if $uv \in \mathsf{WellFormed}_\Sigma$ it is necessary and sufficient to know if the last pair of u and first pair of v consists of a pair of proper letters (denoted by L/L), a pair of a proper letter and a blank/padding symbol (L/B) or a pair of a blank/padding symbol and a proper letter (B/L). This information is called the *letter-type* of an element of $\Sigma^2_{\_}$.

We then define the *type* of a word of $(\Sigma^2_{\_})^+$ as the pair (α, β), usually written $\alpha \to \beta$, of the letter-types of its first and last letters. It is then routine to check that the possible types of well-formed words are

$$\mathcal{T} \triangleq \left\{\mathrm{L}/\mathrm{L} \to \mathrm{L}/\mathrm{L},\ \mathrm{L}/\mathrm{L} \to \mathrm{L}/\mathrm{B},\ \mathrm{L}/\mathrm{B} \to \mathrm{L}/\mathrm{B},\ \mathrm{L}/\mathrm{L} \to \mathrm{B}/\mathrm{L},\ \mathrm{B}/\mathrm{L} \to \mathrm{B}/\mathrm{L}\right\}.$$

For the sake of readability, we will write α instead of $\alpha \to \alpha$ for $\alpha \in \{\mathrm{L}/\mathrm{L}, \mathrm{L}/\mathrm{B}, \mathrm{B}/\mathrm{L}\}$.

One non-trivial point lies in the following innocuous question: what is the type of the empty word? Any type of $\mathcal{T}$ sounds like an acceptable answer. But then it would be natural to say that the concatenation of $\binom{aaa}{aaa}$ of type L/L with the empty word of type $\mathrm{L}/\mathrm{L} \to \mathrm{L}/\mathrm{B}$ should be $\binom{aaa}{aaa}$ of type $\mathrm{L}/\mathrm{L} \to \mathrm{L}/\mathrm{B}$. Automata-wise, this would represent a sequence of transitions $\binom{a}{a}, \binom{a}{a}, \binom{a}{a}$ together with the promise that the next transition would have a padding symbol on its second tape. But then, one has to formalize the idea that the two elements $\binom{aaa}{aaa}$ of type L/L and $\mathrm{L}/\mathrm{L} \to \mathrm{L}/\mathrm{B}$ represent the same underlying pair of words of $\Sigma^* \times \Sigma^*$: this idea will be captured by what we call a dependency relation. A more natural solution would be to simply introduce a new type for the empty word (or to forbid it), but we show in the full version that the resulting notion of algebras cannot capture the property of being a $\mathcal{V}$-relation.

A *$\mathcal{T}$-typed set* (or *typed set* for short) consists of a tuple $\mathbf{X} = (X_\tau)_{\tau \in \mathcal{T}}$, where each X_τ is a set. Instead of $x \in X_\tau$, we will often write $x_\tau \in \mathbf{X}$. A *map between typed sets* $\mathbf{X}$ and $\mathbf{Y}$ is a collection of functions $X_\tau \to Y_\tau$ for each type τ. Similarly, a subset of $\mathbf{X}$ is a tuple of subsets of X_τ for each type τ. To make the notations less heavy, we will often think of typed sets as sets with type annotations rather than tuples, and ask that all operators/constructions should preserve this type.

▶ **Definition 3.1.** *A dependency relation over a typed set* $\mathbf{X}$ *consists of a reflexive and symmetric relation* $\asymp$ *over* $\biguplus \mathbf{X} \triangleq \bigcup_{\tau \in \mathcal{T}} X_\tau \times \{\tau\}$*, such that for all* $x_\sigma, y_\sigma \in \mathbf{X}$*, if* $x_\sigma \asymp y_\sigma$*, then* $x_\sigma = y_\tau$*.*

Crucially, we do not ask for this relation to be transitive – in some examples the dependency relation will be an equivalence relation, but not always (see the full version), and this non-transitivity is actually an important feature, motivated amongst other by the syntactic congruence and Corollary 3.14.

A dependent set is a $\mathcal{T}$*-typed set together with a dependency relation over it. A closed subset of a dependent set* $\langle \mathbf{X}, \asymp \rangle$ *is a subset* $C \subseteq \mathbf{X}$ *such that for all* $x, x' \in \mathbf{X}$*, if* $x \asymp x'$ *then* $x \in C \iff x' \in C$*.*[8]

▶ **Example 3.2.** Given a finite alphabet Σ, let $\mathbf{S}_2\Sigma$ be[9] the dependent set of *synchronous words* defined by:

- $(\mathbf{S}_2\Sigma)_{\text{L/L}} \triangleq (\Sigma \times \Sigma)^*$,
- $(\mathbf{S}_2\Sigma)_{\text{L/L}\to\text{L/B}} \triangleq (\Sigma \times \Sigma)^*(\Sigma \times \_)^*$,
- $(\mathbf{S}_2\Sigma)_{\text{L/B}} \triangleq (\Sigma \times \_)^*$,
- $(\mathbf{S}_2\Sigma)_{\text{L/L}\to\text{B/L}} \triangleq (\Sigma \times \Sigma)^*(\_ \times \Sigma)^*$,
- $(\mathbf{S}_2\Sigma)_{\text{B/L}} \triangleq (\_ \times \Sigma)^*$.

Moreover, $\asymp$ is the reflexive and symmetric closure of the relation that identifies $u_{\text{L/L}}$ with $u_{\text{L/L}\to\beta}$ for all $u \in (\Sigma \times \Sigma^*)$ and $\beta \in \{\text{L/B}, \text{B/L}\}$, and $u_{\text{L/L}\to\text{L/B}}$ with $u_{\text{L/B}}$ for $u \in (\Sigma \times \_)^*$, and $u_{\text{L/L}\to\text{B/L}}$ with $u_{\text{B/L}}$ for $u \in (\_ \times \Sigma)^*$. This structure is depicted in Figure 5.

Given a relation $\mathcal{R} \subseteq \Sigma^* \times \Sigma^*$, we denote by $\underline{\mathcal{R}} = \{(u,v)_\tau \mid (u,v)_\tau \in \mathbf{S}_2\Sigma \text{ and } (u,v) \in \mathcal{R}\}$ the closed subset of $\mathbf{S}_2\Sigma$ induced by $\mathcal{R}$.

▶ **Fact 3.3.** *The map* $\mathcal{R} \mapsto \underline{\mathcal{R}}$ *is a bijection between relations and closed subsets of* $\mathbf{S}_2\Sigma$*.*

Proof. Let f be the function which maps a closed subset C of $\mathbf{S}_2\Sigma$ to $\{(u,v) \in \Sigma^* \times \Sigma^* \mid (u,v)_\tau \in C \text{ for some } \tau \in \mathcal{T}\}$. It then follows that $f \circ \underline{-}$ (resp. $\underline{f(-)}$) is the identity on subsets of $\Sigma^* \times \Sigma^*$ (resp. closed subsets of $\mathbf{S}_2\Sigma$). ◀

3.2 Synchronous Algebras

One key property of types is that some of them can be concatenated to produce other types. We say that two types $\sigma, \tau \in \mathcal{T}$ are *compatible* when there exists non-empty words $u, v \in \mathsf{WellFormed}_\Sigma$ of type σ and τ, respectively, such that uv is well-formed. Said otherwise, $\alpha \to \beta$ is compatible with $\beta' \to \gamma$ if either $\beta = \beta'$ or $\beta = \text{L/L}$ – indeed, for this last case note that *e.g.* the concatenation of $\binom{aaa}{aaa}$ of type L/L with $\binom{\_\_}{aa}$ of type B/L is well-formed. Lastly, if $\alpha \to \beta$ is compatible with $\beta' \to \gamma$, we define their product as $(\alpha \to \beta)\cdot(\beta' \to \gamma) \triangleq \alpha \to \gamma$. Note that this partial operation is associative, in the following sense: for $\rho, \sigma, \tau \in \mathcal{T}$, $(\rho\cdot\sigma)\cdot\tau$ is well-defined if and only if $\rho\cdot(\sigma\cdot\tau)$ is well-defined, in which case both types are equal. This implies that the notion of compatibility of types can be unambiguously lifted to finite lists of types $\tau_1, \ldots, \tau_n$.

---

[8] In other words, C is a union of equivalence classes of the transitive closure of $\asymp$.
[9] The index refers to the arity of the relations we are considering: here we focus on binary relations, but all constructions can be generalized to higher arities.

▶ **Definition 3.4.** *A synchronous algebra $\langle \mathbf{A}, \cdot, \asymp \rangle$ consists of a dependent set $\langle \mathbf{A}, \asymp \rangle$ together with a partial binary operation $\cdot$ on $\mathbf{A}$, called product such that:*

- *for $x_\sigma, y_\tau \in \mathbf{A}$, $x_\sigma \cdot y_\tau$ is defined iff σ and τ are compatible,*
- associativity: *for all $x_\rho, y_\sigma, z_\tau \in \mathbf{A}$, if ρ, σ, τ are compatible:*

$$(x_\rho \cdot y_\sigma) \cdot z_\tau = x_\rho \cdot (y_\sigma \cdot z_\tau),$$

- "monotonicity": *for all $x_\sigma, x'_{\sigma'}, y_\tau \in \mathbf{A}$, if $x_\sigma \asymp x'_{\sigma'}$ and both σ, τ and σ', τ are compatible, then $x_\sigma \cdot y_\tau \asymp x'_{\sigma'} \cdot y_\tau$, and dually if τ, σ and τ, σ' are compatible, then $y_\tau \cdot x_\sigma \asymp y_\tau \cdot x'_{\sigma'}$,*
- units: *for each type τ there is an element $1_\tau \in \mathbf{A}$ such that for any $x_\sigma \in \mathbf{A}$, then $1_\tau \cdot x_\sigma \asymp x_\sigma$ if τ and σ are compatible, and $x_\sigma \cdot 1_\tau \asymp x_\sigma$ if σ and τ are compatible, and moreover, $1_{\mathrm{L/L}\to\beta} = 1_{\mathrm{L/L}} \cdot 1_\beta$ for $\beta \in \{\mathrm{L/B}, \mathrm{B/L}\}$.*

Note in particular that for any type $\tau \in \{\mathrm{L/L}, \mathrm{L/B}, \mathrm{B/L}\}$, then $1_\tau \cdot x_\tau \asymp x_\tau$ but since $1_\tau \cdot x_\tau$ has type τ and $\asymp$ is a dependency relation, then $1_\tau \cdot x_\tau = x_\tau$. This implies in particular that restricting $\langle \mathbf{A}, \cdot \rangle$ to a type $\mathrm{L/L}$, $\mathrm{L/B}$ or $\mathrm{B/L}$ yields a monoid. These are called the three *underlying monoids* of $\mathbf{A}$. The canonical example of synchronous algebras is synchronous words $\mathbf{S}_2\Sigma$ under concatenation. Its underlying monoids are $(\Sigma \times \Sigma)^*$, $(\Sigma \times \{\_\})^*$ and $(\{\_\} \times \Sigma)^*$.

▶ **Fact 3.5.** *Any closed subset of $\mathbf{A}$ either contains all units, or none of them.*

Proof. From $1_{\mathrm{L/L}\to\mathrm{L/B}} = 1_{\mathrm{L/L}} \cdot 1_{\mathrm{L/B}}$ we have $1_{\mathrm{L/L}} \asymp 1_{\mathrm{L/L}\to\mathrm{L/B}}$ and $1_{\mathrm{L/L}\to\mathrm{L/B}} \asymp 1_{\mathrm{L/B}}$. By symmetry between $\mathrm{L/B}$ and $\mathrm{B/L}$, we also have $1_{\mathrm{L/L}} \asymp 1_{\mathrm{L/L}\to\mathrm{B/L}}$ and $1_{\mathrm{L/L}\to\mathrm{B/L}} \asymp 1_{\mathrm{B/L}}$. Hence, if a closed subset of $\mathbf{A}$ contains at least one unit, then it must contain them all. ◀

Note that the product induces a monoid left (resp. right) action of the underlying monoid $\mathbf{A}_{\mathrm{L/L}}$ (resp. $\mathbf{A}_{\mathrm{L/B}}$) on the set $\mathbf{A}_{\mathrm{L/L}\to\mathrm{L/B}}$. Moreover, $x_{\mathrm{L/L}} \mapsto x_{\mathrm{L/L}} \cdot 1_{\mathrm{L/B}}$ identifies any element of type $\mathrm{L/L}$ with an element of type $\mathrm{L/L} \to \mathrm{L/B}$. Over $\mathbf{S}_2\Sigma$, these identifications are injective, but it need not be the case in general. Note also that in general, $x_{\mathrm{L/L}} \cdot 1_{\mathrm{L/L}\to\mathrm{L/B}} = x_{\mathrm{L/L}} \cdot 1_{\mathrm{L/L}} \cdot 1_{\mathrm{L/B}} = x_{\mathrm{L/L}} \cdot 1_{\mathrm{L/B}}$.

▶ Remark 3.6. There exists a monad over the category of dependent sets whose Eilenberg-Moore algebras exactly correspond to synchronous algebras, see the full version.

Morphisms of synchronous algebras are defined naturally as maps that preserve the type, units, the product and the dependency relation.

Free algebras. $\mathbf{S}_2\Sigma$ is free in the sense that for any synchronous algebra $\mathbf{A}$, there is a natural bijection between synchronous algebra morphisms $\mathbf{S}_2\Sigma \to \mathbf{A}$ and maps of typed sets $\Sigma^2_\_ \to \mathbf{A}$. Said otherwise, synchronous algebra morphisms are uniquely defined by their value on $\Sigma^2_\_$.

3.3 Recognizability

Given a synchronous algebra $\mathbf{A}$, a morphism $\varphi\colon \mathbf{S}_2\Sigma \to \mathbf{A}$ and a closed subset $\mathrm{Acc} \subseteq \mathbf{A}$ called "accepting set", we say that $\langle \varphi, \mathbf{A}, \mathrm{Acc} \rangle$ *recognizes* a relation $\mathcal{R} \subseteq \Sigma^* \times \Sigma^*$ when $\mathcal{R} = \varphi^{-1}[\mathrm{Acc}]$. We extend the notion of recognizability to $\langle \varphi, \mathbf{A} \rangle$ or to simply $\mathbf{A}$ by existential quantification over the missing elements in the tuple $\langle \varphi, \mathbf{A}, \mathrm{Acc} \rangle$.

Synchronous algebra induced by a monoid. A monoid morphism $\varphi\colon (\Sigma^2_{\_})^* \to M$ naturally *induces* a synchronous algebra morphism $\tilde{\varphi}\colon \mathbf{S}_2\Sigma \to \mathbf{A}_M$, where:

- $\mathbf{A}_M$ has for every type τ a copy of M, and $\asymp$ is $\{(x_\sigma, x_\tau) \mid x \in M, \sigma, \tau \in \mathcal{T}\}$,
- for all $x_\sigma, y_\tau \in \mathbf{A}_M$ with compatible type, $x_\sigma \cdot y_\tau \mathrel{\hat{=}} (x \cdot y)_{\sigma\cdot\tau}$,
- $\tilde{\varphi}\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) \mathrel{\hat{=}} \left(\varphi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right)\right)_{\mathrm{L/L}}$, $\tilde{\varphi}\left(\begin{smallmatrix} a \\ \_ \end{smallmatrix}\right) \mathrel{\hat{=}} \left(\varphi\left(\begin{smallmatrix} a \\ \_ \end{smallmatrix}\right)\right)_{\mathrm{L/B}}$, and $\tilde{\varphi}\left(\begin{smallmatrix} \_ \\ a \end{smallmatrix}\right) \mathrel{\hat{=}} \left(\varphi\left(\begin{smallmatrix} \_ \\ a \end{smallmatrix}\right)\right)_{\mathrm{B/L}}$.

The algebra simply duplicates M as many times as needed and identifies two elements together when they originated from the same element of M.

▶ **Fact 3.7.** *If φ recognizes $\mathcal{R}$ for some relation $\mathcal{R} \subseteq \Sigma^* \times \Sigma^*$ seen as a language over $\Sigma^2_{\_}$, then $\tilde{\varphi}$ recognizes $\mathcal{R}$.*

Consolidation of a synchronous algebra. Given a synchronous algebra morphism $\varphi\colon \mathbf{S}_2\Sigma \to \mathbf{A}$, define its *consolidation*[10] as the semigroup morphism $\varphi^0\colon (\Sigma^2_{\_})^* \to \mathbf{A}^0$, where $\mathbf{A}^0$ is the monoid obtained from $\biguplus \mathbf{A}$ by first merging units, by adding a zero (denoted by 0), and extending $\cdot$ to be a total function by letting all missing products equal 0, and φ^0 sends a word $u \in (\Sigma^2_{\_})^*$ to

- 0 if u is not well-formed,
- $\varphi(u_{\mathrm{L/L}})$ if $u \in (\Sigma \times \Sigma)^*$,
- $\varphi(u_{\mathrm{L/B}})$ if $u \in (\Sigma \times \_)^+$,
- $\varphi(u_{\mathrm{B/L}})$ if $u \in (\_ \times \Sigma)^+$,
- $\varphi(u_{\mathrm{L/L}\to\mathrm{L/B}})$ if $u \in (\Sigma \times \Sigma)^+(\Sigma \times \_)^+$,
- $\varphi(u_{\mathrm{L/L}\to\mathrm{B/L}})$ if $u \in (\Sigma \times \Sigma)^+(\_ \times \Sigma)^+$.

Note that this operation disregards the dependency relation of $\mathbf{A}$.

▶ **Fact 3.8.** *If φ recognizes some relation $\underline{\mathcal{R}}$, then φ^0 recognizes $\mathcal{R}$, when seen as a language over $\Sigma^2_{\_}$.*

The following result follows from Facts 2.2, 3.7, and 3.8.

▶ **Proposition 3.9.** *A relation is synchronous if and only if it is recognized by a finite synchronous algebra.*

Let us continue with a slightly less trivial example of algebra.

▶ **Example 3.10** (Group relations: Example 2.4, cont'd.)**.** Fix $p, q \in \mathbb{N}_{>0}$. Let $\mathbf{Z}_{p,q}$ denote the algebra whose underlying monoids are:

- the trivial monoid $(0, +)$ for type L/L,
- the cyclic monoid $(\mathbb{Z}/p\mathbb{Z}, +)$ for type L/B,
- the cyclic monoid $(\mathbb{Z}/q\mathbb{Z}, +)$ for type B/L.

Moreover, the sets $Z_{\mathrm{L/L}\to\mathrm{L/B}}$ and $Z_{\mathrm{L/L}\to\mathrm{B/L}}$ are defined as $\mathbb{Z}/p\mathbb{Z}$ and $\mathbb{Z}/q\mathbb{Z}$, respectively. The product is addition – we identify $0_{\mathrm{L/L}}$ with the zero of $\mathbb{Z}/p\mathbb{Z}$ and of $\mathbb{Z}/q\mathbb{Z}$. We denote by $\bar{k}$ the equivalence class of $k \in \mathbb{Z}$ in $\mathbb{Z}/n\mathbb{Z}$ when n is clear from context. The dependency relation identifies (1) all units together and (2) x_σ with $1_\tau \cdot x_\sigma$ and $x_\sigma \cdot 1_\tau$ when the types are compatible.

Let $\varphi\colon \mathbf{S}_2\Sigma \to \mathbf{Z}_{p,q}$ be the synchronous algebra morphism defined by

$$\varphi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) \mathrel{\hat{=}} \bar{0}_{\mathrm{L/L}}, \quad \varphi\left(\begin{smallmatrix} a \\ \_ \end{smallmatrix}\right) \mathrel{\hat{=}} \bar{1}_{\mathrm{L/B}}, \quad \varphi\left(\begin{smallmatrix} \_ \\ a \end{smallmatrix}\right) \mathrel{\hat{=}} \bar{1}_{\mathrm{B/L}} \quad \text{and} \quad \varphi(\varepsilon_\tau) \mathrel{\hat{=}} \bar{0}_\tau \text{ for } \tau \in \mathcal{T}.$$

[10] Named by analogy with Tilson's construction [37, §3, p. 102].

This morphism recognizes any relation of the form

$$\mathcal{R}^{I,J} \mathrel{\hat=} \big\{(u,v) \bigm| |u| > |v| \text{ and } (|u| - |v| \bmod p) \in I, \text{ or} \\ |u| < |v| \text{ and } (|v| - |u| \bmod q) \in J. \quad \big\},$$

where $I \subseteq \mathbb{Z}/p\mathbb{Z}$ and $J \subseteq \mathbb{Z}/q\mathbb{Z}$ are such that $\bar{0} \notin I$ and $\bar{0} \notin J$. This last condition is necessary because the accepting set has to be a closed subset of $\mathbf{Z}_{p,q}$: if $\bar{0}$ was in I, then we would need $\bar{0} \in J$, but also to add $\bar{0}_{\mathrm{L}/\mathrm{L}}$ to the accepting set: this would recognize

$$\big\{(u,v) \bigm| |u| > |v| \text{ and } (|u| - |v| \bmod p) \in I, \text{ or} \\ |u| < |v| \text{ and } (|v| - |u| \bmod q) \in J, \text{ or } |u| = |v|\big\}.$$

Note also that all relations $\mathcal{R}^{I,J}$ with $\bar{0} \notin I$ and $\bar{0} \notin J$ are group relations: letting G be the group $\mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$, $\mathcal{R}$ can be written as $\mathsf{WellFormed}_\Sigma \cap \psi^{-1}[I \times \{0\} \cup \{0\} \times J]$ where $\psi\colon (\Sigma^2_-)^* \to G$ is the monoid morphism defined by $\psi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}\right) \mathrel{\hat=} (\bar{0},\bar{0})$, $\psi\left(\begin{smallmatrix} a \\ - \end{smallmatrix}\right) \mathrel{\hat=} (\bar{1},\bar{0})$ and $\psi\left(\begin{smallmatrix} - \\ a \end{smallmatrix}\right) \mathrel{\hat=} (\bar{0},\bar{1})$.

3.4 Syntactic Morphisms & Algebras

▶ **Lemma 3.11** (Syntactic morphism theorem)**.** *For each relation $\mathcal{R}$, there exists a surjective synchronous algebra morphism*

$$\eta_{\mathcal{R}}\colon \mathbf{S}_2\Sigma \twoheadrightarrow \mathbf{A}_{\mathcal{R}}$$

that recognizes $\mathcal{R}$ and is such that for any other surjective synchronous algebra morphism $\varphi\colon \mathbf{S}_2\Sigma \twoheadrightarrow \mathbf{B}$ recognizing $\mathcal{R}$, there exists a synchronous algebra morphism $\psi\colon \mathbf{B} \twoheadrightarrow \mathbf{A}_{\mathcal{R}}$ such that the diagram

$$\begin{array}{ccc} \mathbf{S}_2\Sigma & \xrightarrow{\eta_{\mathcal{R}}} & \mathbf{A}_{\mathcal{R}} \\ & \searrow^{\varphi} & \uparrow \psi \\ & & \mathbf{B}, \end{array}$$

commutes. The objects $\eta_{\mathcal{R}}$ and $\mathbf{A}_{\mathcal{R}}$ are called the syntactic synchronous algebra morphism *and* syntactic synchronous algebra *of $\mathcal{R}$, respectively. Moreover, these objects are unique up to isomorphisms of the algebra.*

▶ **Corollary 3.12** (of Proposition 3.9 and Lemma 3.11)**.** *A relation is synchronous if and only if its syntactic synchronous algebra is finite.*

The proof of Lemma 3.11 – see the full version – relies, as in the case of monoids, on the notion of congruence.

Given a synchronous algebra $\langle \mathbf{A}, \asymp, \cdot \rangle$, a *congruence* is any reflexive, symmetric relation $\curlyeqprec$ over $\mathbf{A}$ which is coarser than $\asymp$, and which is *locally transitive*, meaning that for all $x_\sigma, x'_\sigma, y_\tau, y'_\tau \in \mathbf{X}$, if $x'_\sigma \curlyeqprec x_\sigma$, $x_\sigma \curlyeqprec y_\tau$ and $y_\tau \curlyeqprec y'_\tau$, then $x'_\sigma \curlyeqprec y'_\tau$.[11]

The *quotient structure* $\mathbf{A}/\curlyeqprec$ of $\mathbf{A}$ by a congruence $\curlyeqprec$ is defined as follows:

- its underlying typed set consists of the equivalence classes of $\mathbf{A}$ under the equivalence relation $\{(x_\sigma, y_\sigma) \mid x_\sigma \curlyeqprec y_\sigma\}$, such a class being abusively denoted by $[x]^{\curlyeqprec}$,
- its product is the product induced by $\mathbf{A}$, in the sense that $[x]^{\curlyeqprec} \cdot [y]^{\curlyeqprec} \mathrel{\hat=} [xy]^{\curlyeqprec}$, and

[11] In particular, it implies that $\curlyeqprec$ is transitive when restricted to elements of the same type.

- its dependency relation is the relation induced by $\bowtie$, *i.e.* $[x]^{\bowtie} \asymp [y]^{\bowtie}$ whenever $x \bowtie y$,
- its units are defined as the equivalence classes of the units of $\mathbf{A}$.

Moreover, $x \mapsto [x]^{\bowtie}$ defines a surjective morphism of synchronous algebras from $\mathbf{A}$ to $\mathbf{A}/{\bowtie}$.

Given a synchronous algebra $\langle \mathbf{A}, \asymp, \cdot \rangle$ and a closed subset $C \subseteq \mathbf{A}$, we define a congruence $\bowtie_C$, called *syntactic congruence* of C over $\mathbf{A}$ by letting $a_\sigma \bowtie_C b_\tau$ when for all $x, y \in \mathbf{A}$

- if both $xa_\sigma y$ and $xb_\tau y$ are defined, then $xa_\sigma y \in C$ *iff* $xb_\tau y \in C$, and
- if both xa_σ and xb_τ are defined, then $xa_\sigma \in C$ *iff* $xb_\tau \in C$, and
- if both $a_\sigma y$ and $b_\tau y$ are defined, then $a_\sigma y \in C$ *iff* $b_\tau y \in C$.

It is routine to check that the syntactic congruence is indeed a congruence. For instance, to prove that $\bowtie_C$ is coarser than $\asymp$, observe that if $a_\sigma \asymp b_\tau$, then for all x, y *s.t.* both $xa_\sigma y$ and $xb_\tau y$ are defined, then $xa_\sigma y \asymp xb_\tau y$, and since C is a closed subset of $\mathbf{A}$, $xa_\sigma y \in C$ *iff* $xb_\tau y \in C$. The other two conditions are proven in the same fashion. Note however that while the relation is locally transitive, it is not transitive in general.

When $\mathcal{R} \subseteq \Sigma^* \times \Sigma^*$ is a relation, we abuse the notation and write $\bowtie_{\mathcal{R}}$ to denote the syntactic congruence $\bowtie_{\underline{\mathcal{R}}}$ of $\underline{\mathcal{R}}$ in $\mathbf{S}_2\Sigma$. The existence of the syntactic morphism then follows from the next proposition, proven in the full version.

▶ **Proposition 3.13.** *Let $\varphi\colon \mathbf{S}_2\Sigma \twoheadrightarrow \mathbf{A}$ be a surjective synchronous algebra morphism that recognizes $\mathcal{R}$, say $\underline{\mathcal{R}} = \varphi^{-1}[Acc]$ for some closed subset $Acc \subseteq \mathbf{A}$, then*

$$\begin{array}{rccc} \varphi/{\bowtie_{Acc}}\colon & \mathbf{S}_2\Sigma & \twoheadrightarrow & \mathbf{A}/{\bowtie_{Acc}} \\ & u & \mapsto & [\varphi(u)]^{\bowtie_{Acc}} \end{array}$$

is the syntactic morphism of $\mathcal{R}$.

▶ **Corollary 3.14.** *In the syntactic synchronous algebra $\mathbf{A}_{\mathcal{R}}$, the syntactic congruence $\bowtie_{Acc}$ and the dependency relation $\asymp$ coincide.*

Proof. By Proposition 3.13 applied to the syntactic morphism, $x \mapsto [x]^{\bowtie_{\mathrm{Acc}}}$ is an isomorphism from $\mathbf{A}_{\mathcal{R}}$ to $\mathbf{A}_{\mathcal{R}}/{\bowtie_{\mathrm{Acc}}}$. Hence, $[x]^{\bowtie_{\mathrm{Acc}}} \asymp [y]^{\bowtie_{\mathrm{Acc}}}$ in $\mathbf{A}_{\mathcal{R}}/{\bowtie_{\mathrm{Acc}}}$ *iff* $x \asymp y$ in $\mathbf{A}_{\mathcal{R}}$, for all $x, y \in \mathbf{A}_{\mathcal{R}}$. But then, the dependency relation $\asymp$ of $\mathbf{A}_{\mathcal{R}}/{\bowtie_{\mathrm{Acc}}}$ is, by definition, such that $[x]^{\bowtie_{\mathrm{Acc}}} \asymp [y]^{\bowtie_{\mathrm{Acc}}}$ *iff* $x \bowtie_{\mathrm{Acc}} y$. Putting both equivalences together, we get that $x \bowtie_{\mathrm{Acc}} y$ *iff* $x \asymp y$ for all $x, y \in \mathbf{A}_{\mathcal{R}}$. ◀

We provide in the full version a simple example of syntactic synchronous algebra whose dependency relation is not an equivalence relation.

Boolean operations. Given two synchronous algebras $\mathbf{A}$ and $\mathbf{B}$, define their *Cartesian product* $\mathbf{A} \times \mathbf{B}$ by taking, for each type τ, the Cartesian product $A_\tau \times B_\tau$. Units, product are defined naturally, and the dependency relation is defined by taking the conjunction over each component. Then $\neg\mathcal{R}$ is recognized by $\mathbf{A}$, and $\mathcal{R} \cup \mathcal{S}$ and $\mathcal{R} \cap \mathcal{S}$ are recognized by $\mathbf{A} \times \mathbf{B}$.

4 The Lifting Theorem & Pseudovarieties

4.1 Elementary Formulation

▶ **Example 4.1** (Group relations: Example 3.10 cont'd)**.** We want to decide when the relation

$$\mathcal{R}^{I,J} \triangleq \big\{(u,v) \;\big|\; |u| > |v| \text{ and } (|u| - |v| \bmod p) \in I, \text{ or} \\ |u| < |v| \text{ and } (|v| - |u| \bmod q) \in J. \quad \big\}$$

from Example 3.10 is a group relation. By definition this happens if and only if there exists a finite group G, together with a monoid morphism $\varphi\colon (\Sigma_-^2)^* \to G$ and a subset $\mathrm{Acc} \subseteq G$ *s.t.* $\forall u \in \mathsf{WellFormed}_\Sigma$, $u \in \mathcal{R}^{I,J}$ *iff* $\varphi(u) \in \mathrm{Acc}$. We claim:

$$\mathcal{R}^{I,J} \text{ is a group relation} \quad \textit{iff} \quad (\bar{0} \notin I \text{ and } \bar{0} \notin J). \tag{✻}$$

The right-to-left implication was shown in Example 3.10. We prove the implication from left to right: let n be the order of G so that $x^n = 1$ for all $x \in G$. In particular, we have: $\varphi\big(\binom{a}{\_}^{pqn}\big) = 1 = \varphi\big(\binom{a}{a}^{pqn}\big)$. Since $\varphi\big(\binom{a}{a}^{pqn}\big) \notin \mathcal{R}^{I,J}$, it follows that $\binom{a}{\_}^{pqn} \notin \mathcal{R}^{I,J}$ *i.e.* $\bar{0} \notin I$. Also, $\bar{0} \notin J$ by symmetry, which concludes the proof.

Even more generally, we can decide if a relation $\mathcal{R}$ is a group relation by simply looking at the syntactic synchronous algebra of $\mathcal{R}$.

▶ **Theorem 4.2** (Lifting theorem: Elementary Formulation)**.** *Given a relation $\mathcal{R}$ and a $*$-pseudovariety of regular languages $\mathcal{V}$ corresponding to a pseudovariety of monoids $\mathbb{V}$, the following are equivalent:*

1. *$\mathcal{R}$ is a $\mathcal{V}$-relation,*
2. *$\mathcal{R}$ is recognized by a finite synchronous algebra $\mathbf{A}$ whose underlying monoids are all in $\mathbb{V}$,*
3. *all underlying monoids of the syntactic synchronous algebras $\mathbf{A}_{\mathcal{R}}$ of $\mathcal{R}$ are in $\mathbb{V}$.*

See the proof in the full version.

▶ Remark 4.3. In light of Theorem 4.2, one can wonder whether the notion of synchronous algebra is necessary to characterize $\mathcal{V}$-relations, or if it is enough to look at the languages corresponding to the underlying monoids. Said otherwise, is the membership of $\mathcal{R}$ in the class of $\mathcal{V}$-relations uniquely determined by the regular languages $\mathcal{R} \cap (\Sigma \times \Sigma)^*$, $\mathcal{R} \cap (\Sigma \times \{\_\})^*$ and $\mathcal{R} \cap (\{\_\} \times \Sigma)^*$? Unsurprisingly, synchronous algebras are indeed necessary, as there are relations $\mathcal{R}$ such that:

$$\mathcal{R} \cap (\Sigma \times \Sigma)^* \in \mathcal{V}_{\Sigma\times\Sigma}, \quad \mathcal{R} \cap (\Sigma \times \_)^* \in \mathcal{V}_{\Sigma\times\_} \quad \text{and} \quad \mathcal{R} \cap (\_ \times \Sigma)^* \in \mathcal{V}_{\_\times\Sigma}, \tag{❁}$$

but $\mathcal{R}$ is *not* a $\mathcal{V}$-relation. This can happen even if $\mathcal{V}$ is the $*$-pseudovariety of all regular languages: for instance for the relation

$$\mathcal{R} \mathrel{\hat{=}} \{(u,v) \mid |u| > |v| > 0 \text{ and } |u| - |v| \text{ is prime}\}.$$

Notice that there is a subtle but crucially important difference between (❁) and the second item of the Lifting Theorem: while the underlying monoids of a synchronous algebra $\mathbf{A}$ recognizing $\mathcal{R}$ only accept words of the form $(\Sigma \times \Sigma)^*$, $(\Sigma \times \_)^*$ or $(\_ \times \Sigma)^*$, elements of $(\Sigma \times \Sigma)^+(\Sigma \times \_)^+$ or $(\Sigma \times \Sigma)^+(\_ \times \Sigma)^+$ influence the underlying monoids of $\mathbf{A}$ via the axioms of synchronous algebras.

Also, note that the existence the Lifting Theorem follows from the careful definition of synchronous algebras: more naive definitions of these algebras simply cannot characterize $\mathcal{V}$-relations, see the full version.

From Theorem 4.2 and the implicit fact that all our constructions are effective, we obtain a decidability (meta-)result for $\mathcal{V}$-relations.

▶ **Corollary 4.4.** *The class of $\mathcal{V}$-relations has decidable membership if, and only if, $\mathcal{V}$ has decidable membership.*

For instance, a relation is a group relation if, and only if, all underlying monoids of its syntactic synchronous algebra are groups.

4.2 Pseudovarieties of Synchronous Relations

We introduce the notion of pseudovariety of synchronous algebras and $*$-pseudovariety of synchronous relations. We show an Eilenberg correspondence between these two notions. We then reformulate the Lifting Theorem to show that any Eilenberg correspondence between monoids and regular languages lifts to an Eilenberg correspondence between synchronous algebras and synchronous relations.

Say that a synchronous algebra $\mathbf{A}$ is a *quotient* of $\mathbf{B}$ when there exists a surjective synchronous algebra morphism from $\mathbf{B}$ to $\mathbf{A}$. A *subalgebra* of $\mathbf{B}$ is any closed subset of $\mathbf{B}$ closed under product and containing the units. We then say that synchronous algebra $\mathbf{A}$ *divides* $\mathbf{B}$ when $\mathbf{A}$ is a quotient of a subalgebra of $\mathbf{B}$.

Observe that $\mathbf{S}_2\Sigma$ admits the following property: elements of type ${}^{\mathrm{L}}/_{\mathrm{L}} \to {}^{\mathrm{L}}/_{\mathrm{B}}$ and ${}^{\mathrm{L}}/_{\mathrm{L}} \to {}^{\mathrm{B}}/_{\mathrm{L}}$ are generated by the underlying monoids. Since syntactic synchronous algebras are homomorphic images of $\mathbf{S}_2\Sigma$, they also satisfy this property. In general, we say that a synchronous algebra $\mathbf{A}$ is *locally generated* if every element of type ${}^{\mathrm{L}}/_{\mathrm{L}} \to {}^{\mathrm{L}}/_{\mathrm{B}}$ (resp. ${}^{\mathrm{L}}/_{\mathrm{L}} \to {}^{\mathrm{B}}/_{\mathrm{L}}$) can be written as the product of an element of type ${}^{\mathrm{L}}/_{\mathrm{L}}$ with an element of type ${}^{\mathrm{L}}/_{\mathrm{B}}$ (resp. ${}^{\mathrm{B}}/_{\mathrm{L}}$).

A *pseudovariety of synchronous algebras* is any class $\mathbb{V}$ of locally generated finite synchronous algebras closed under

- *finite product:* if $\mathbf{A}, \mathbf{B} \in \mathbb{V}$ then $\mathbf{A} \times \mathbf{B} \in \mathbb{V}$,
- *division:* if some finite locally generated algebra $\mathbf{A}$ divides $\mathbf{B}$ for some $\mathbf{B} \in \mathbb{V}$, then $\mathbf{A} \in \mathbb{V}$.

Because of Lemma 3.11, a synchronous relation is recognized by a finite synchronous algebra of a pseudovariety $\mathbb{V}$ *iff* its syntactic synchronous algebra belongs to $\mathbb{V}$.

A *$*$-pseudovariety of synchronous relations* is a function $\mathcal{V}\colon \Sigma \mapsto \mathcal{V}_\Sigma$ such that for any finite alphabet Σ, $\mathcal{V}_\Sigma$ is a set of synchronous relations over Σ such that $\mathcal{V}$ is closed under

- *Boolean combinations:* if $\mathcal{R}, \mathcal{S} \in \mathcal{V}_\Sigma$, then $\neg\mathcal{R}$, $\mathcal{R} \cup \mathcal{S}$ and $\mathcal{R} \cap \mathcal{S}$ belong to $\mathcal{V}_\Sigma$ too,
- *Syntactic derivatives:* if $\mathcal{R} \in \mathcal{V}_\Sigma$, then any relation recognized by the syntactic synchronous algebra morphism of $\mathcal{R}$ also belongs to $\mathcal{V}_\Sigma$.
- *Inverse morphisms:* if $\varphi\colon \mathbf{S}_2\Gamma \to \mathbf{S}_2\Sigma$ is a synchronous algebra morphism and $\mathcal{R} \in \mathcal{V}_\Sigma$ then $\varphi^{-1}[\mathcal{R}] \in \mathcal{V}_\Gamma$.

To recover a more traditional definition (of the form "closure under Boolean operations, residuals[12] and inverse morphisms"), we need to properly define what are the residuals of a relation. It turns out that the answer is quite surprising and less trivial than what one would expect.

▶ **Definition 4.5** (Residuals). *Let $\mathbf{A}$ be a synchronous algebra, $x_\sigma \in \mathbf{A}$, and $C \subseteq \mathbf{A}$ be a closed subset. The* left residual *and* right residual *of C by x_σ are defined by*

$$x_\sigma^{-1}C \triangleq \{y_\tau \in \mathbf{A} \mid \exists y'_{\tau'} \preccurlyeq_C y_\tau,\ x_\sigma y'_{\tau'} \in C\}, \text{ and}$$
$$Cx_\sigma^{-1} \triangleq \{y_\tau \in \mathbf{A} \mid \exists y'_{\tau'} \preccurlyeq_C y_\tau,\ y'_{\tau'} x_\sigma \in C\},$$

respectively. We refer indiscriminately to both these notions as residuals. *We extend these notions to sets, by letting $X^{-1}C \triangleq \bigcup_{x\in X} x^{-1}C$ and $CX^{-1} \triangleq \bigcup_{x \in X} Cx^{-1}$.*

[12] Also called "quotient" *e.g.* in [31, §III.1.3, p. 39], or "polynomial derivative" in [9, §4, p. 19].

For the sake of readability, we will sometimes drop the type of elements when dealing with residuals. It is routine to check that residuals are always closed subsets (since $\preccurlyeq_C$ is coarser than the dependency relation), or that $(x^{-1}C)y^{-1} = x^{-1}(Cy^{-1})$. Equivalently, Cx_σ^{-1} can be defined as the smallest closed subset containing the "naive residual" $\{y_\tau \in \mathbf{A} \mid y_\tau x_\sigma \in C\}$. This latter set is always contained in Cx_σ^{-1} (by reflexivity of $\preccurlyeq_C$), and moreover, if it is empty, then so is Cx_σ^{-1}.

As an example, consider the relation $\mathcal{R}$ from the full version. Then the "naive right residual" of $\underline{\mathcal{R}}$ by $\binom{a}{-}_{\mathrm{L/B}}$ consists of $\varepsilon_{\mathrm{L/L}}$ and all elements of type $\mathrm{L/B}$ and $\mathrm{L/L} \to \mathrm{L/B}$. But it does not contain any element of type $\mathrm{B/L}$ or $\mathrm{L/L} \to \mathrm{B/L}$ because such elements cannot be concatenated with $\binom{a}{-}_{\mathrm{L/B}}$ on the right. Yet, the residual $\underline{\mathcal{R}}\binom{a}{-}_{\mathrm{L/B}}^{-1}$ contains all elements of type $\mathrm{B/L}$ (and also $\mathrm{L/L} \to \mathrm{B/L}$): for instance, $\binom{-}{a}_{\mathrm{B/L}} \in \underline{\mathcal{R}}\binom{a}{-}_{\mathrm{L/B}}^{-1}$ since $\binom{-}{a}_{\mathrm{B/L}} \preccurlyeq_{\mathcal{R}} \binom{a}{-}_{\mathrm{L/B}}$ and $\binom{a}{-}_{\mathrm{L/B}}\binom{a}{-}_{\mathrm{L/B}} \in \mathcal{R}$.

On the other hand, in the algebra $\mathbf{S}_2a$ consider the relation $\mathcal{S} = (aa)^* \times a(aa)^*$. Then $\underline{\mathcal{S}}\binom{a}{a}_{\mathrm{L/L}}^{-1}$ is empty since its "naive residual" $\{y_\tau \in \mathbf{S}_2a \mid y_\tau \cdot \binom{a}{a} \in \mathcal{S}\}$ is empty. Indeed, for $y_\tau \cdot \binom{a}{a}_{\mathrm{L/L}}$ to be well-defined, one needs τ to be $\mathrm{L/L}$, *i.e.* y encodes a pair of two words (u, v) of the same length. But then $(ua, va) \notin \mathcal{S}$.

▶ **Lemma 4.6.** *A class $\mathcal{V}\colon \Sigma \mapsto \mathcal{V}_\Sigma$ is a $*$-pseudovariety of synchronous relations if, and only if, it is closed under Boolean combinations, residuals and inverse morphisms.*

See the proof in the full version.

Let $\mathbb{V} \to \mathcal{V}$ denote the map (called *correspondence*) that takes a pseudovariety of synchronous algebras and maps it to

$$\mathcal{V}\colon \Sigma \mapsto \{\mathcal{R} \subseteq \Sigma^* \times \Sigma^* \mid \mathbf{A}_{\mathcal{R}} \in \mathbb{V}\}.$$

Dually, let $\mathcal{V} \to \mathbb{V}$ denote the *correspondence* that takes a $*$-pseudovariety of synchronous relations $\mathcal{V}$ and maps it to the pseudovariety of synchronous algebras generated by all $\mathbf{A}_{\mathcal{R}}$ for some $\mathcal{R} \in \mathcal{V}_\Sigma$. Here, the *pseudovariety generated* by a class C of finite locally generated synchronous algebras is the smallest pseudovariety containing all finite locally generated algebras of C, or equivalently,[13] the class of all finite locally generated synchronous algebras that divide a finite product of algebras of C.[14]

▶ **Lemma 4.7** (An Eilenberg theorem for synchronous relations)**.** *The correspondences $\mathbb{V} \to \mathcal{V}$ and $\mathcal{V} \to \mathbb{V}$ define mutually inverse bijections between pseudovarieties of synchronous algebras and $*$-pseudovarieties of synchronous relations.*

See the proof in the full version.

As consequence of Lemma 4.7, if $\mathcal{V}$ is a $*$-pseudovariety of synchronous relations and $\mathbb{V}$ is a pseudovariety of synchronous algebras, we write $\mathcal{V} \leftrightarrow \mathbb{V}$ to mean that either $\mathcal{V} \to \mathbb{V}$ or, equivalently, $\mathbb{V} \to \mathcal{V}$. This relation is called an *Eilenberg-Schützenberger correspondence*.

[13] The proof is straightforward, see *e.g.* [31, Proposition XI.1.1, p. 190] for a proof in the context of semigroups.

[14] Note that "being locally generated" is not preserved by taking subalgebras, but this is not an issue: we restrict the construction to (finite) locally generated algebras.

▶ **Proposition 4.8.** *If* $\mathbb{V}$ *is a pseudovariety of monoids, then*

$$\mathbb{V}^{sync} \mathrel{\hat{=}} \{\mathbf{A} \text{ locally generated finite synchronous algebra} \\ \text{s.t. all underlying monoids of } \mathbf{A} \text{ are in } \mathbb{V}\}$$

is a pseudovariety of synchronous algebras. Similarly, if $\mathcal{V}$ *is an* $*$*-pseudovariety of regular languages, then the class of* $\mathcal{V}$*-relations, namely*

$$\mathcal{V}^{sync}\colon \Sigma \mapsto \{\mathcal{R} \subseteq \Sigma^* \times \Sigma^* \mid \exists L \in \mathcal{V}_{\Sigma^2_{\perp}},\ \mathcal{R} = L \cap \mathsf{WellFormed}_\Sigma\},$$

is a $*$*-pseudovariety of synchronous relations.*

Proof. The first point is straightforward. The second one follows from it and Lemma 4.7 and Theorem 4.2. ◀

Finally, Theorem 4.2 can be elegantly rephrased by saying that correspondences between pseudovarieties of monoids and $*$-pseudovarieties of regular languages lift to correspondences between pseudovarieties of synchronous algebras and $*$-pseudovarieties of synchronous relations.

▶ **Theorem 4.9** (Lifting Theorem: Pseudovariety Formulation)**.** *If, in the Eilenberg correspondence between pseudovarieties of monoids and* $*$*-pseudovarieties of regular languages we have* $\mathcal{V} \leftrightarrow \mathbb{V}$*, then in the Eilenberg correspondence between the pseudovariety of synchronous algebras* $\mathbb{V}^{sync}$ *and the* $*$*-pseudovariety of synchronous relations* $\mathcal{V}^{sync}$*, we have* $\mathcal{V}^{sync} \leftrightarrow \mathbb{V}^{sync}$.

5 Discussion

A natural next step is to generalize Question 1.2 by replacing $\mathsf{WellFormed}_\Sigma$ by a fixed regular language Ω.

▶ **Question 5.1.** Given a class of regular languages $\mathcal{V}$, can we characterize (and decide) all languages of the form $L \cap \Omega$ for some $L \in \mathcal{V}$?

We claim that the construction of synchronous algebras can be generalized for any Ω, giving rise to the notion of "path algebras". The lifting theorem for monoids can be shown to hold for some Ω, including well-formed words for n-ary relations with $n \geq 3$, and that it cannot effectively hold for all Ω.

A natural next step would then be to study the relationship between "path algebras" and Figueira & Libkin's L-controlled relations [19, §3], see also [16].

Lastly, it would be interesting to extend the results on algebras to automata: for instance, can we adapt our proof to show the existence of a minimal *synchronous* automaton for each relation?

References

1 Jorge Almeida. Some algorithmic problems for pseudovarieties. *Publ. Math. Debrecen*, 52(1):531–552, 1999. Consulted version: `https://www.researchgate.net/profile/Jorge-Almeida-14/publication/2510507_Some_Algorithmic_Problems_for_Pseudovarieties/links/02e7e531d968b4fe8f000000/Some-Algorithmic-Problems-for-Pseudovarieties.pdf`.

2 C. J. Ash. Inevitable graphs: a proof of the type II conjecture and some related decision procedures. *International Journal of Algebra and Computation*, 01(01):127–146, March 1991. doi:10.1142/S0218196791000079.

3 Pablo Barceló, Diego Figueira, and Rémi Morvan. Separating Automatic Relations. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:15, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Consulted version: `https://arxiv.org/abs/2305.08727v2`. `doi:10.4230/LIPIcs.MFCS.2023.17`.

4 Pablo Barceló, Chih-Duo Hong, Xuan-Bach Le, Anthony W. Lin, and Reino Niskanen. Monadic Decomposability of Regular Relations. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 103:1–103:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Consulted version: `https://arxiv.org/abs/1903.00728v1`. `doi:10.4230/LIPIcs.ICALP.2019.103`.

5 Pablo Barceló, Leonid Libkin, Anthony W. Lin, and Peter T. Wood. Expressive languages for path queries over graph-structured data. *ACM Trans. Database Syst.*, 37(4), December 2012. Consulted version: `https://homepages.inf.ed.ac.uk/libkin/papers/pods10-tods.pdf` (saved on `http://web.archive.org`). `doi:10.1145/2389241.2389250`.

6 Jean Berstel. *Transductions and Context-Free Languages*. Vieweg+Teubner Verlag, Wiesbaden, 1979. Consulted version: `http://www-igm.univ-mlv.fr/~berstel/LivreTransductions/LivreTransductions.pdf` (saved on `http://web.archive.org/`). URL: `http://link.springer.com/10.1007/978-3-663-09367-1`.

7 Achim Blumensath. Monadic Second-Order Model Theory. Version of 2023-12-19 (saved on `http://web.archive.org/`), 2023. URL: `https://www.fi.muni.cz/~blumens/MSO.pdf`.

8 Mikołaj Bojańczyk and Lê Thành Dũng (Tito) Nguyễn. Algebraic Recognition of Regular Functions. In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 117:1–117:19, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Consulted version: `https://hal.science/hal-03985883v2`. `doi:10.4230/LIPIcs.ICALP.2023.117`.

9 Mikołaj Bojańczyk. Recognisable Languages over Monads. In Igor Potapov, editor, *Developments in Language Theory*, Lecture Notes in Computer Science, pages 1–13. Springer International Publishing, 2015. Consulted version: `https://arxiv.org/abs/1502.04898v1`. `doi:10.1007/978-3-319-21500-6_1`.

10 Mikołaj Bojańczyk. Languages recognised by finite semigroups, and their generalisations to objects such as trees and graphs, with an emphasis on definability in monadic second-order logic, August 2020. Lecture notes. `arXiv:2008.11635`, `doi:10.48550/arXiv.2008.11635`.

11 Mikołaj Bojańczyk and Igor Walukiewicz. Forest algebras. In Jörg Flum, Erich Grädel, and Thomas Wilke, editors, *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, volume 2 of *Texts in Logic and Games*, pages 107–132. Amsterdam University Press, 2008. Consulted version: `https://hal.science/hal-00105796v1`.

12 Michaël Cadilhac, Olivier Carton, and Charles Paperman. Continuity of Functional Transducers: A Profinite Study of Rational Functions. *Logical Methods in Computer Science*, Volume 16, Issue 1, February 2020. `doi:10.23638/LMCS-16(1:24)2020`.

13 Olivier Carton, Christian Choffrut, and Serge Grigorieff. Decision problems among the main subfamilies of rational relations. *RAIRO - Theoretical Informatics and Applications*, 40(2):255–275, April 2006. Consulted version: `http://www.numdam.org/item/10.1051/ita:2006005.pdf`. `doi:10.1051/ita:2006005`.

14 Olivier Carton, Thomas Colcombet, and Gabriele Puppis. An algebraic approach to MSO-definability on countable linear orderings. *The Journal of Symbolic Logic*, 83(3):1147–1189, September 2018. Consulted version: `https://arxiv.org/abs/1702.05342v2`. `doi:10.1017/jsl.2018.7`.

15 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2012. Consulted version: https://hal.science/hal-00646514v1. doi:10.1017/CBO9780511977619.

16 María Emilia Descotte, Diego Figueira, and Gabriele Puppis. Resynchronizing Classes of Word Relations. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 123:1–123:13, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Consulted version: https://hal.science/hal-01721046v2. doi:10.4230/LIPIcs.ICALP.2018.123.

17 Joost Engelfriet and Hendrik Jan Hoogeboom. Mso definable string transductions and two-way finite-state transducers. *ACM Trans. Comput. Logic*, 2(2):216–254, April 2001. Consulted version: SciHub. doi:10.1145/371316.371512.

18 Diego Figueira. Foundations of Graph Path Query Languages (Course Notes). In *Reasoning Web Summer School 2021*, volume 13100 of *Reasoning Web. Declarative Artificial Intelligence - 17th International Summer School 2021, Leuven, Belgium, September 8-15, 2021, Tutorial Lectures*, pages 1–21, Leuven, Belgium, September 2021. Springer. Consulted version: https://hal.science/hal-03349901. doi:10.1007/978-3-030-95481-9_1.

19 Diego Figueira and Leonid Libkin. Synchronizing Relations on Words. *Theory of Computing Systems*, 57(2):287–318, August 2015. Consulted version: https://hal.science/hal-01793633v1/. doi:10.1007/s00224-014-9584-2.

20 Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. Logical and Algebraic Characterizations of Rational Transductions. *Logical Methods in Computer Science*, Volume 15, Issue 4, December 2019. doi:10.23638/LMCS-15(4:16)2019.

21 S. J. v. Gool and B. Steinberg. Pointlike sets for varieties determined by groups. *Advances in Mathematics*, 348:18–50, May 2019. Consulted version: https://arxiv.org/abs/1801.04638v1. doi:10.1016/j.aim.2019.03.020.

22 Karsten Henckell, Stuart W. Margolis, Jean-Éric Pin, and John Rhodes. Ash's type II theorem, profinite topology and Malcev products: part I. *International Journal of Algebra and Computation*, 01(04):411–436, December 1991. Consulted version: https://www.irif.fr/~jep/PDF/HMPR.pdf (saved on http://web.archive.org/). doi:10.1142/S0218196791000298.

23 Bernard R. Hodgson. *Théories décidables par automate fini*. PhD thesis, Université de Montréal, 1976. Not available online.

24 Bernard R. Hodgson. On direct products of automaton decidable theories. *Theoretical Computer Science*, 19(3):331–335, September 1982. doi:10.1016/0304-3975(82)90042-1.

25 Bernard R. Hodgson. Décidabilité par automate fini. *Annales des Sciences Mathématiques du Québec*, 7(1):39–57, 1983. Consulted version: https://www.mat.ulaval.ca/fileadmin/mat/documents/bhodgson/Hodgson_ASMQ_1983.pdf (saved on http://web.archive.org/).

26 Bakhadyr Khoussainov and Anil Nerode. Automatic presentations of structures. In Gerhard Goos, Juris Hartmanis, Jan Leeuwen, and Daniel Leivant, editors, *Logic and Computational Complexity*, volume 960, pages 367–392. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995. doi:10.1007/3-540-60178-3_93.

27 Dietrich Kuske and Markus Lohrey. Some natural decision problems in automatic graphs. *The Journal of Symbolic Logic*, 75(2):678–710, June 2010. Consulted version: https://www.eti.uni-siegen.de/ti/veroeffentlichungen/08-euler-hamilton.pdf (saved on http://web.archive.org/). doi:10.2178/jsl/1268917499.

28 Chris Köcher. *Analyse der Entscheidbarkeit diverser Probleme in automatischen Graphen*. PhD thesis, Technische Universität Ilmenau, Ilmenau, 2014. (Saved on http://web.archive.org/). URL: https://people.mpi-sws.org/~ckoecher/files/theses/bsc-thesis.pdf.

29 Dominique Perrin and Jean-Éric Pin. *Infinite Words, Automata, Semigroups, Logic and Games*, volume 141. Elsevier, 2004. Consulted version: Libgen.

30 Jean-Éric Pin. Positive varieties and infinite words. In Cláudio L. Lucchesi and Arnaldo V. Moura, editors, *LATIN'98: Theoretical Informatics*, Lecture Notes in Computer Science, pages 76–87, Berlin, Heidelberg, 1998. Springer. Consulted version: https://hal.science/hal-00113768v1. doi:10.1007/BFb0054312.

31 Jean-Éric Pin. Mathematical Foundations of Automata Theory, 2022. Version of February 18, 2022 (saved on http://web.archive.org/); MPRI lecture notes. URL: https://www.irif.fr/~jep/PDF/MPRI/MPRI.pdf.

32 Thomas Place and Marc Zeitoun. Group separation strikes back. In *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13, 2023. Consulted version: https://arxiv.org/abs/2205.01632v2. doi:10.1109/LICS56636.2023.10175683.

33 Christophe Reutenauer. Séries formelles et algèbres syntactiques. *Journal of Algebra*, 66(2):448–483, October 1980. doi:10.1016/0021-8693(80)90097-6.

34 John Rhodes and Benjamin Steinberg. Pointlike sets, hyperdecidability and the identity problem for finite semigroups. *International Journal of Algebra and Computation*, November 2011. Consulted version: SciHub. doi:10.1142/S021819679900028X.

35 Sasha Rubin. Automata presenting structures: A survey of the finite string case. *Bulletin of Symbolic Logic*, 14(2):169–209, 2008. Consulted version: SciHub. doi:10.2178/bsl/1208442827.

36 Howard Straubing and Pascal Weil. Varieties. In Jean Éric Pin, editor, *Handbook of Automata Theory, volume I: Theoretical Foundations*, pages Chapter 16, pp. 569–614. European Mathematical Society Publishing House, September 2021. doi:10.4171/Automata.

37 Bret Tilson. Categories as algebra: An essential ingredient in the theory of monoids. *Journal of Pure and Applied Algebra*, 48(1):83–198, 1987. doi:10.1016/0022-4049(87)90108-3.

Appendix

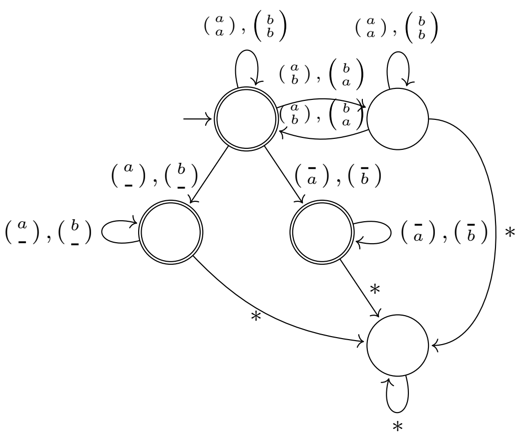


Figure 3 Minimal (deterministic complete) "classical" automaton for the binary relation of pairs (u, v) such that the number of a's in $u_1 \ldots u_k$ and in $v_1 \ldots v_k$ are the same mod 2, where $k = \min(|u|, |v|)$, seen as a language over Σ^2_-. Said otherwise, this is automaton rejects exactly all words in $(\Sigma^2_-)^*$ which (1) are not the valid encoding of a pair of words and (2) are the encoding of a pair which does not satisfy the property above. Each label $*$ is defined so that the automaton is deterministic and complete.

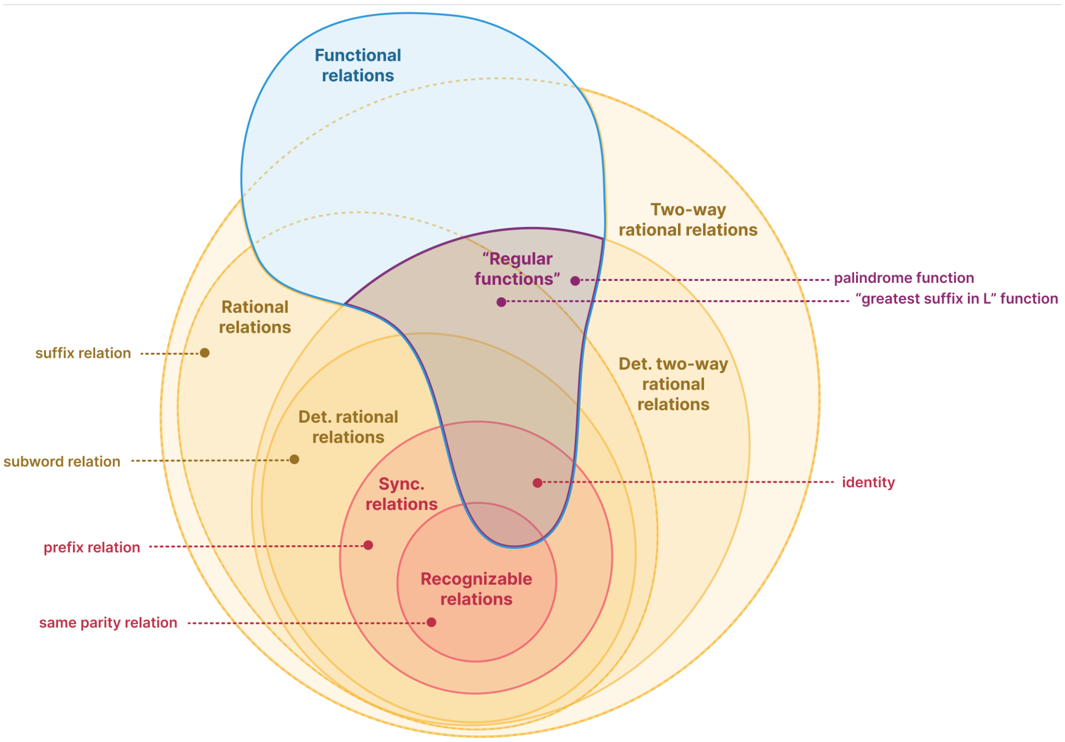
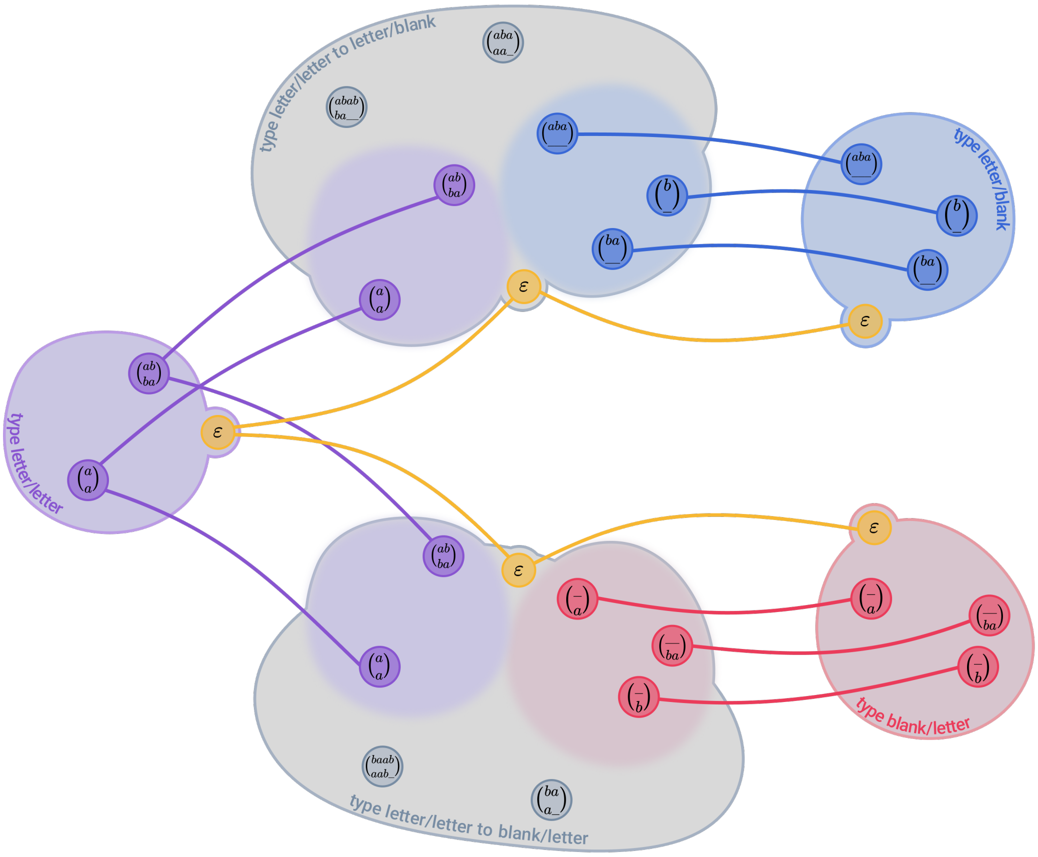


Figure 4 The landscape of rationality for binary relations. Dashed regions are empty: the intersection of functional relations and two-way rational relations collapses to regular functions by [17, Theorem 22, p. 243].



■ **Figure 5** Representation of the dependent set $\mathbf{S}_2\Sigma$ of synchronous words. Coloured edges represent the dependency relation, and self-loops are not drawn.