Algebraic recognizability of languages

Pascal Weil

To cite this version:

Pascal Weil. Algebraic recognizability of languages. Mathematical Foundations of Computer Science 2004, 2004, Prague, Czech Republic. pp.149-175. hal-00096619

HAL Id: hal-00096619
https://hal.science/hal-00096619

Submitted on 19 Sep 2006

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

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

ccsd-00096619, version 1 - 19 Sep 2006

Algebraic Recognizability of Languages[⋆]

Pascal Weil

LaBRI, CNRS and Université Bordeaux-1, pascal.weil@labri.fr

Abstract. Recognizable languages of finite *words* are part of every computer science cursus, and they are routinely described as a cornerstone for applications and for theory. We would like to briefly explore why that is, and how this word-related notion extends to more complex models, such as those developed for modeling distributed or timed behaviors.

Ἐν ἀρχῇ ἦν ὁ λόγος...
In the beginning was the Word...

Recognizable languages of finite *words* are part of every computer science cursus, and they are routinely described as a cornerstone for applications and for theory. We would like to briefly explore why that is, and how this word-related notion extends to more complex models, such as those developed for modeling distributed or timed behaviors.

The notion of recognizable languages is a familiar one, associated with classical theorems by Kleene, Myhill, Nerode, Elgot, Büchi, Schützenberger, etc. It can be approached from several angles: recognizability by automata, recognizability by finite monoids or finite-index congruences, rational expressions, monadic second order definability. These concepts are expressively equivalent, and this leads to a great many fundamental algorithms in the fields of compilation, text processing, software engineering, etc... Moreover, it surely indicates that the class of recognizable languages is central. These equivalence results use the specific structure of words (finite chains, labeled by the letters of the alphabet), and the monoid structure of the set of all words.

Since the beginnings of language theory, there has been an interest for other models than words – especially for the purpose of modeling distributed or timed computation (trees, traces, pomsets, graphs, timed words, etc) –, and for extending to these models the tools that were developpped for words. For many models, some of these tools may not be defined, and those who are defined, may not coincide.

In this paper, we concentrate on the algebraic notion of recognizability: that which, for finite words, exploits the monoid structure of the set of words, and relies on the consideration of monoid morphisms into finite monoids, or equivalently, of finite-index monoid congruences. Our aim is to examine why this

[⋆] This paper was prepared in part while the author was an invited Professor at the University of Nebraska-Lincoln. He also acknowledges partial support from project VERSYDIS (*ACI Sécurité Informatique*, Ministère de la Recherche).

particular approach is fruitful in the finite word case, and how it has been, or can be adapted to other models.

In Sect. 1, we explore the specific benefits of using algebraic recognizability for the study of word languages. It opens the door to a very fine classification of recognizable languages, which uses the resources of the structural theory of finite monoids. This classification of recognizable languages is not only mathematically elegant, it also allows the characterization and the decision of membership in otherwise significant classes of languages.

The emblematic example of such a class is that of star-free languages, which are exactly the first-order definable languages, those that can be defined by a formula in propositional temporal logic, those that are recognized by a counter-free automaton, and those whose syntactic monoid is aperiodic (Schützenberger, McNaughton, Pappert, Kamp). Only the latter two characterizations lead to a decision algorithm, and the algebraic approach makes this algorithm the clearest.

The example of star-free languages is however not the only one; for example, the various notions of locally testable languages are also characterized, and ultimately efficiently decided, by algebraic properties (Simon, McNaughton, Ladner). The power of finite monoid theory leads in fact to an extremely fine classification (Eilenberg), where for instance, natural hierarchies within first-order or temporal logic can be characterized as well (Brzozowski, Knast, Thomas, Thérien, Wilke).

Already in the 1960s, fundamental results appeared on notions of automata for trees and for infinite words, linking them with logical definability and rational expressions (Büchi, Doner, Mezei, Thatcher, Wright). An algebraic approach to automata-recognizable languages of infinite word was introduced in the early 1990s, and as in the case of tree languages, it requires introducing an algebraic framework different from monoid theory (Perrin, Pin, Wilke), see Sect. 4.1.

In fact, very early on, Elgot and Mezei extended the notion of recognizability to subsets of arbitrary (abstract) algebras. But the notion of logical definability for these subsets strongly depends on the combinatorial (relational) structure of the objects chosen to represent the elements of the abstract algebra under consideration. In many situations, the problem is posed in the other direction: we know which models we want to consider (they are posets, or graphs, or traces, or timed words, as arise from, say, the consideration of distributed or timed computation) and we need to identify an algebraic structure on the set of these objects, for which logical definability and algebraic recognizability will be best related. One key objective there, is to be able to decide logical specifications. Note that models of automata, while highly desirable, are not known to exist in all the interesting cases, and especially not for graphs or posets. In contrast, the algebraic and the logical points of view are universal.

A class of relational structures being fixed, Courcelle gave very loose conditions on an algebraic structure on the set of these structures, which guarantee that counting monadic second order definability implies recognizability. The converse is known to hold in a number of significant cases, but not in general.

In Sect. 4, we discuss some of the relational structures that have been studied in the literature, and the *definability vs. recognizability* results that are known for them: trees, infinite words, traces, series-parallel pomsets, message sequence charts and layered diagrams, graphs, etc. The main obstacle to the equivalence between definability and recognizability is the fact that the algebras we consider may not be finitely generated. In contrast, a good number of situations have been identified where this equivalence holds, and each time, a finiteness (or boundedness) condition is satisfied. We will try to systematically point out these finiteness conditions, and we will identify some important questions that are still pending.

When all is said and done, the central question is that of the specification and the analysis of infinite sets by finite means (finite and finitely generated algebras, finite automata, logical formulas, etc.), arguably the fundamental challenge of theoretical computer science. This paper presents a personal view on the relevance of algebraic recognizability for this purpose, beyond its original scope of application (languages of finite words), and an introduction to some of the literature and results that illustrate this view. I do not claim however that it constitutes a comprehensive survey of the said literature and results (in particular, I chose to systematically refer to books and survey papers when available), and I apologize in advance for any omission!...

1 The Finite Word Case

In the beginning were the (finite) words, and loosely following the Biblical analogy, one could say that the spirits of Kleene, Büchi and Schützenberger flew over the abyss, organising it from chaos to beauty.

Throughout this paper, A will denote an alphabet, i.e. a finite, non-empty set. We denote by A^* the set of all finite words on alphabet A.

1.1 The Classical Equivalence Results

We are all familiar with the notion of regular languages, but there are in fact several competing notions, that turn out to be equivalent for finite words. Each is interesting in its own right, as it reveals a fruitful point of view, syntactic or semantic, denotational or operational. The results of this section can be found in many books, and in particular in those of Eilenberg [33], Pin [67,68,69], Straubing [77], Sipser [75] and Sakarovitch [73].

Recognizability by Automata. One can first consider languages *recognized by finite state automata*, whether deterministic or non-deterministic. Every language recognized by a finite state automaton admits a unique *minimal deterministic automaton*, which is effectively computable.

The notion of a deterministic automaton can also be expressed in terms of a finite-index semi-congruence, and in terms of an action of the free monoid on a finite set.

Algebraic Recognizability. One can also consider languages *recognized by a finite monoid.* This exploits the monoid structure of A^*, the set of all words on alphabet A: if M is a monoid and $\varphi\colon A^* \to M$ is a morphism of monoids, we say that $L \subseteq A^*$ is recognized by φ (or by M) if $L = P\varphi^{-1}$ for some $P \subseteq M$, or equivalently, if $L = L\varphi\varphi^{-1}$. Here too, for every language L recognized by a finite monoid, there exists a least finite monoid recognizing L, called the *syntactic monoid* of L.

Rational Expressions. *Rational expressions* describe languages using the letters of the alphabet, the constant $\emptyset$, and the so-called *rational operations* of union, concatenation and Kleene star (if L is a language, L^* is the submonoid of A^* generated by L).

It should be noted that for a given rational language, there is no notion of a unique minimal rational expression describing it.

Logical Definability. *Büchi's sequential calculus* exploits the combinatorial structure of words, as A-labeled, linearly ordered finite sets: in this logical formalism, individual variables are interpreted to be positions in a word, and the predicates are $i < j$ (to say that position i is to the left of position j) and $R_a i$ (to say that position i is labeled by letter $a \in A$). *First order formulas* (FO) use only individual variables, whereas *monadic second order formulas* (MSO) also use second order variables, interpreted to be sets of positions. To a formula φ in this language, one associates the set $L(\varphi)$ of all words which satisfy φ, that is $L(\varphi)$ is the language of the finite models of φ.

The Kleene-Nerode-Myhill-Büchi Theorem. Theorems by Kleene, Nerode, Myhill and Büchi show that the notions rapidly described above coincide.

Theorem 1.1. *Let $L \subseteq A^*$. Then L is recognized by a finite state automaton, if and only if L is recognized by a finite monoid, if and only if the syntactic monoid of L is finite, if and only if L is described by a rational expression, if and only if L is defined by an MSO-formula of Büchi's calculus.*

Moreover, there are algorithms to pass from one of these specification formalisms to each other.

It is interesting to note that the many closure properties of the class of recognizable languages are easily established in an appropriate choice of one of these equivalent formalisms. For instance, closure under Boolean operations easily follows from the definition of algebraic recognizability, as does closure under inverse morphism (inverse rewriting). On the other hand, closure under concatenation, star and direct morphism is a triviality for languages described by rational expressions.

None of the equivalences in Theorem 1.1 is very difficult, but their proofs really use the different points of view on words and languages. If we compare these results with the situation that prevails for other models than words, it is in fact a very exceptional situation to have these notions be so nicely defined and be equivalent.

1.2 Classification of Recognizable Languages

With each recognizable language L, we can associate a computable canonical finite object – in fact two closely related such objects: the minimal automaton of L, and its syntactic monoid. The connection between them is tight: the syntactic monoid of L is exactly the monoid of transitions of its minimal automaton.

This paves the way for a fine classification of recognizable languages (see [67,69]). Not surprisingly, it is the syntactic monoid, with its natural algebraic structure, which offers the strongest classification tool. In this section, we give a few instances of this classification: some are well-known and open up important applications (star-free languages, locally testable languages), some are more specific, and demonstrate the degree of refinement allowed by this method.

Star-Free Languages. The most illuminating example of a significant subclass of recognizable languages is given by the *star-free languages.* These are the languages which can be described by star-free expressions, *i.e.*, using the letters of the alphabets, the constants $\emptyset$ and 1 (the empty word), the Boolean operations and concatenation (but no star).

The characterization of star-free languages requires the following definitions: a deterministic automaton is said to be *counter-free* if whenever a non-trivial power u^n ($u \neq 1$, $n \neq 0$) labels a loop at some state q, then the word u also labels a loop at the same state. A finite monoid M is said to be *aperiodic* if it contains no non-trivial group, if and only if for each $x \in M$, $x^n = x^{n+1}$ for all large enough n. Finally, PTL (*propositional temporal logic*) is a modal logic, interpreted on positions in words, with modalities next, eventually and until.

The following statement combines results by Schützenberger, McNaughton, Pappert and Kamp, see [33,49,67,77].

Theorem 1.2. *Let $L \subseteq A^*$. Then L is star-free, if and only if L is recognized by a finite aperiodic monoid, if and only if the syntactic monoid of L is finite and aperiodic, if and only if L is recognized by a counter-free automaton, if and only if L is defined by an FO-formula of Büchi's calculus, if and only if L is defined by a PTL-formula.*

Moreover, there are algorithms to pass from one of these specification formalisms to each other.

Thus, the class of star-free languages, with its natural definition in terms of generalized rational expressions, ends up having natural characterizations in terms of all the formalisms used in Theorem 1.1 – to which we can add PTL, a logical formalism considered to be very useful to specify the behavior of complex systems.

The historically first side of this result is the algebraic one, which links star-free languages and aperiodic monoids. It is of particular interest for two reasons.

First because it offers an algorithm to decide whether a language is star-free; and second, because it shows that the algebraic structure of the syntactic monoid of a recognizable language (not just its finiteness) reflects the combinatorial properties of that language. This gave the first hint of Eilenberg's theorem, discussed further in this section.

Variants of FO-Definability. Several refinements and generalizations further reinforce the significance of Theorem 1.2.

Consider the extension FO+MOD of FO, where we also allow *modulo quantification* of the form $\exists^{\bmod q} x\ \varphi(x)$ $(q \geq 1)$. Such a quantification is interpreted to mean that the set of values x for which $\varphi(x)$ holds, has cardinality a multiple of q. Straubing, Thérien and Thomas showed that a language is FO+MOD-definable if and only if the subgroups of its syntactic monoid are solvable [77].

Considering now subclasses of FO, it turns out that every star-free language can be defined by a FO-formula using only 3 variables. Let FO_2 be the class of star-free languages defined by FO-formulas with only 2 variables. Let also **DA** be the class of finite monoids in which every regular element is idempotent (if $xyx = x$ for some y, then $x^2 = x$). Then a combination of results of Etessami, Pin, Schützenberger, Thérien, Vardi, Weil, Wilke [74,70,37,80] shows that a language L is in FO_2, if and only if it is defined by a Σ_2- and by a Π_2-formula, if and only if its syntactic monoid is in **DA**, if and only if L is defined by a PTL formula which does not use the until modality, if and only if L can be obtained from the letters using only disjoint unions and unambiguous products (and the constants $\emptyset$ and A^*).

It is well-known that every FO-formula is equivalent to one in prenex normal form (consisting of a sequence of quantifications, followed by a quantifier-free formula). This gives rise to the classical *quantifier-alternation hierarchy*, based on counting the number of alternated blocks of existential and universal quantifiers. Another natural hierarchy, seen from the point of view of star-free expressions, defines its n+1-st level as the Boolean closure of products of level n languages (and level 0 consists of $\emptyset$ and A^*). This is the so-called *dot-depth hierarchy.* Thomas showed that these two hierarchies coincide, that is, a language can be defined by an FO-formula in prenex normal form with n alternating blocks of quantifiers, if and only if it is in the n-th level of the dot-depth hierarchy [83]. Decidable algebraic characterizations were given for level 1 of these hierarchies, but the decidability of level 2 and the further levels is still an open question. It was however showed (Brzozowski, Knast, Simon, Straubing, see [68]) that the hierarchy is infinite (if $|A| \geq 2$), and that each level is characterized by an algebraic property, in the following sense: if two languages have the same syntactic monoid and one is at level n, then so is the other one.

There is also a natural hierarchy on PTL-formulas, based on the number of nested usage of the until modality. Thérien and Wilke showed that the levels of this infinite hierarchy are characterized by the algebraic properties of the syntactic monoid, and that each is decidable [81].

Communication complexity. The *communication complexity* of a language L is a measure of the amount of communication that is necessary for two partners, each holding part of a word, to determine whether the word lies in L, see [51]. Tesson and Thérien showed that the communication complexity of a recognizable language is entirely determined by its syntactic monoid, and that it can be computed on this basis [78].

Piecewise and Locally Testable Languages. A word v is a *subword* of a word u if $v = a_1 \cdots a_n$ and $u = u_0 a_1 u_1 \cdots a_n u_n$ for some $u_0, \ldots, u_n \in A^*$. It is a *factor* of u if $u = xvy$ for some $x, y \in A^*$.

A language L is said to be n-piecewise testable if whenever u and v have the same subwords of length at most n and $u \in L$, then $v \in L$. The language L is *piecewise testable* if it is n-piecewise testable for some n.

A language L is said to be n-locally testable if whenever u and v have the same factors of length at most n and the same prefix and suffix of length $n - 1$, and $u \in L$, then $v \in L$. The language L is *locally testable* if it is n-locally testable for some n. Locally testable languages are widely used in the fields of learning and pattern matching, whereas piecewise testable languages form the first level of the dot-depth hierarchy.

Results of Simon and McNaughton, Ladner (see [33,67,69]) show that both these properties are characterized by algebraic properties of syntactic monoids. More precisely, a language L is piecewise testable if and only if every principal two-sided ideal of its syntactic monoid $S(L)$, admits a single generator. The language L is locally testable if and only if $S(L)$ is aperiodic and $eS(L)e$ is an idempotent commutative monoid, for each idempotent $e \neq 1$ in $S(L)$.

Varieties of Languages. Many more examples can be found in the literature, where natural algebraic properties of finite monoids match natural combinatorial or logical properties of languages (see for instance [1,33,67,69]). The scope of this matching is described in Eilenberg's variety theorem; the latter identifies the closure properties on classes of recognizable languages and on classes of finite monoids, that characterize the classes that can occur in this correspondence. These classes are called, respectively, *varieties of recognizable languages* and *pseudovarieties of finite monoids.*

Decision Procedures. The varieties of recognizable languages thus identified by algebraic means are all the more interesting if they are decidable. Since the syntactic monoid of a recognizable language is computable, this reduces to deciding the membership of a finite monoid in certain pseudovarieties. In fact, in the examples surveyed above, this is the only path known to a decision algorithm.

Let us now assume that we are considering a decidable pseudovariety of monoids (and hence a decidable variety of languages). The syntactic monoid of a language L, which is the transition monoid of the minimal automaton of L, may have a size exponential in the number of states of that automaton. Thus deciding whether a recognizable language given by a deterministic finite state automaton lies in a given variety, seems to require exponential time and space.

In view of the connection between syntactic monoid and minimal automaton, it is possible to translate the relevant algebraic property of finite monoids to a property of automata, and to check this property on the minimal automaton. This possibility is explicitly stated in Theorem 1.2, but it is also the underlying reason for the decision procedures concerning piecewise testable (Stern [76]) and locally testable languages (Kim, McNaughton, McCloskey [50]).

In many important situations, this leads to polynomial time membership algorithms: piecewise and locally testable languages, FO_2, certain varieties related to the dot-depth hierarchy [70], etc. One major exception though, is the class of star-free languages, for which the membership problem is PSPACE-complete (Cho, Huynh [14]). In other words, given a deterministic automaton, there is no fundamentally better algorithm to decide whether the corresponding language is star-free, than to verify whether the syntactic monoid is aperiodic.

It must be stressed that even in the cases where we have polynomial membership algorithms, these algorithms are a translation to automata of algebraic properties of the syntactic monoid, they were not discovered until after the corresponding pseudovariety of monoids was identified, and their natural justification is *via* monoid-theoretic considerations.

1.3 Recognizable and Context-Free Languages

Recognizable languages form but the lowest level of the Chomsky hierarchy, where the next level consists of the context-free languages. *Context-free languages* are defined by context-free grammars, which can be viewed, with a more algebraic mindset, as finite systems of polynomial equations of the form $x_i = \sum p(\boldsymbol{x})$ $(1 \leq i \leq n)$ where $\boldsymbol{x} = (x_1, \ldots, x_n)$ is the vector of variables, the summations are finite and each p is a word over the letters of A and the variables (see [11]). A solution of such a system is a vector of languages $\boldsymbol{L} = (L_1, \ldots, L_n)$, and the context-free languages arise as the components of maximal solutions of such systems. Accordingly, context-free languages are also called *equational*, or *algebraic*.

Recognizable languages are components of maximal solutions of certain simpler systems, where each $p(\boldsymbol{x})$ is a word of the form $x_j u$, $1 \leq j \leq n$, $u \in A^*$ (right-linear equation). In particular, not all context-free languages are recognizable. The class of context-free languages is not closed under intersection, but it is closed under intersection with recognizable languages.

2 Almost as Established: the Finite Tree Case

Tree languages were considered in the early 1960s, see [39]. Here we use a *ranked alphabet*, that is, a set Σ, equipped with an arity function $\sigma\colon \Sigma \to \mathbb{N}$. A Σ-term is defined recursively as follows: every letter of arity 0 (a constant) is a Σ-term, and if $a \in \Sigma$ has arity n and $t_1, \ldots, t_n$ are Σ-terms, then $a(t_1, \ldots, t_n)$ is a Σ-term. Terms are naturally (and unequivocally) represented by Σ-labeled trees, where an a-labeled node has $\sigma(a)$ linearly ordered children. We let T_Σ be the set of all Σ-terms.

Thatcher and Wright introduced a model of automata for Σ-labeled trees, the so-called *bottom-up automata* [39,79]. To describe their expressiveness, they used the natural algebraic structure on the set of Σ-terms: each element $a \in \Sigma$ is an operation, of arity $\sigma(a)$, and no relation is assumed to hold between these operations. Now, let a Σ-algebra be any set S equipped with a $\sigma(a)$-ary operation a^S for each $a \in \Sigma$. If we use this algebraic notion, we can define recognizable and equational sets of Σ-terms: a subset $L \subseteq T_\Sigma$ is said to be recognizable if there exists a morphism (of Σ-algebras) φ from T_Σ to a finite Σ-algebra, such that $L = L\varphi\varphi^{-1}$; and L is equational if it is a component of a vector of maximal solutions of a system of polynomial equations. These systems are defined as in Sect. 1.3, except that the parameters p are now taken to be terms rather than words. Note that again, given $L \subseteq T_\Sigma$, there exists a unique least Σ-algebra recognizing it, called the *syntactic Σ-algebra* of L.

Thatcher and Wright also described subsets of T_Σ by *generalized rational expressions*, involving the letters, unions, the Σ-operations, and an appropriate notion of iteration.

Finally, Doner considered a logical formalism to be applied to the trees representing Σ-terms [29]: the individual variables are interpreted as nodes in a finite tree and the predicates are interpreted to express the labeling function and the parent-child relation.

Results of Doner, Thatcher and Wright [29,39,79] prove the following statement.

Theorem 2.1. *Let $L \subseteq T_\Sigma$. Then L is recognized by a bottom-up automaton, if and only if L is recognized by a finite Σ-algebra, if and only if the syntactic Σ-algebra of L is finite, if and only if L is described by a generalized rational expression, if and only if L is defined by an* MSO *formula, if and only if L is equational.*

Moreover, there are algorithms to pass from one of these specification formalisms to each other.

Note that the particularity of this setting is that equational sets are recognizable. Another important remark is that deterministic bottom-up automata are really Σ-algebras, so the notions of automata-theoretic and algebraic recognizability are not really distinct. This last point makes Theorem 2.1 a little less satisfying than its word counterpart. Another (subjective) cause of dissatisfaction is that the generalized rational expression are rather awfully complex. Finally, this result has not made it easy to classify term languages in the spirit of Sect. 1.2. This is maybe due to a less long history of investigating the structural properties of finite Σ-algebras. Some interesting related results on binary trees, expressed in terms of certain context-free languages of words were proved by Beaudry, Lemieux, Thérien [5,6,7,8]. Nevertheless, it is fair to say that no structural theory of Σ-algebras clearly emerges.

An open question which may serve as a benchmark in this direction is the following: given a recognizable tree language, can one decide whether it is FO-definable?

3 The General Notion of Recognizability

Adapting the discussion in Sect. 2, one can easily define *recognizable* and *equational* subsets in any algebra. Recognizable sets are defined in terms of morphisms into finite algebras of the same type (or in terms of finite index congruences), and equational sets in terms of systems of polynomial equations (Mezei, Wright [60], Courcelle [17]). With those definitions, recognizable sets form a Boolean algebra, equational sets are closed under union, recognizable sets are always equational, finite sets are equational (even though they may fail to be recognizable), products of equational sets (using the operations in the algebra under consideration) are equational, but the analogous statement for recognizable sets is not always true, and the intersection of a recognizable and an equational set is equational. Finally, if $\varphi\colon S \to T$ is a morphism between algebras of the same type, then $L\varphi^{-1}$ is recognizable if $L \subseteq T$ is recognizable, and $L\varphi$ is equational if $L \subseteq S$ is equational.

3.1 Choosing an Algebraic Structure

As discussed above, if the sets we consider are naturally contained in an algebra, the notion of recognizability is straightforward. Sometimes however (frequently maybe), we want to discuss sets of relational structures, and we then design an algebraic signature to combine these structures.

For instance, it is one such abstract construction that has us see trees as terms. Consider even finite words: the interest of the model maybe lies simply in the notion of a totally ordered A-labeled finite set. We chose to view the set of words as a monoid under concatenation, and this gave rise to the notion of algebraically recognizable languages discussed in Sect. 1. We could also consider the following algebraic structure on the set of words: each letter $a \in A$ defines a unary operation $u \mapsto ua$. Then the set of all finite words is the algebra generated by A and the constant 1 (this amounts to considering the set of words as the algebra of $(A \cup \{1\})$-terms). One can verify that the notion of recognizable language is not modified – another sign of the robustness of the model of words. In fact, finite $(A \cup \{1\})$-algebras are naturally identified with deterministic finite state automata, and the equivalence between this notion of algebraic recognizability and the monoid-based one is a rephrasing of Kleene's theorem.

Relational structures, in the sense of this paper, are sets equipped with relations from a given relational signature. For instance, as mentioned earlier, words are A-labeled totally ordered sets: the relational signature consists of the binary order relation and one labeling unary relation R_a for each letter $a \in A$. In trees, the relations are the labeling relations and the parent-child binary relation (or the predecessor relation, or the parent and the sibling relations, etc, – these choices are equivalent when it comes to expressing properties in monadic second order logic, see Sect. 3.3).

The choice of an algebraic structure can be guided by the natural constructions generating the finite relational structures under consideration (concatenation of words; construction of terms; construction of a word letter by letter), but there is really nothing canonical or unique about the algebraic structure.

Suppose for instance that we consider very few operations: then there are many more recognizable set, maybe to the extent that every set is recognizable (for exemple, consider the set of words with no operations at all: every finite partition, say L and L^c, is a finite index congruence). If on the other hand we have too many operations, then there will be less recognizable sets. For instance, let sh (for *shift*) be the unary operations on words that fixes 1, and maps ua to au ($a \in A$, $u \in A^*$). One can verify that the set a^*b is not recognizable for the algebra whose signature consists of the concatenation product and the shift operation. Another extreme example is given by $\mathbb{N}$, equipped with the constant 0 and the unary predecessor and successor operations: then the only recognizable sets are $\emptyset$ and $\mathbb{N}$.

It may also happen that adding certain operations does not change the class of recognizable sets. For instance, adding the mirror operation (defined inductively by $\tilde{1} = 1$ and $\widetilde{ua} = a\tilde{u}$) to the concatenation product, does not alter the notion of recognizability. See also Sect. 4.3.

In Sect. 4, we discuss a number of relational structures for which very interesting notions of recognizability have emerged in the literature.

3.2 Multi-Sorted Algebras, Ordered Algebras, etc

Sometimes, algebras are too constrained: the domain and the range of certain natural operations may consist of certain kinds of elements only. This is taken care of by the definition of *multi-sorted algebras*, see [17,25].

A typical example is provided by the study of languages of infinite words (more details are given in Sect. 4.1). It turns out that the best algebraic framework consists of considering simultaneously the finite and infinite words. One relevant operation is the concatenation product: between two finite words, it is the usual, fundamental operation, yielding a finite word; the product uv where u is a finite word and v is infinite, is an infinite word; and while it is possible to define the product of two infinite words, the outcome of such a product carries no significant information ($uv = u$) and the operation can be discarded. So we find that we need to consider two *sorts* of elements, finite and $\mathsf{infinite}$, and two binary product operations, of type $\mathsf{finite} \times \mathsf{finite} \to \mathsf{finite}$ and $\mathsf{finite} \times \mathsf{infinite} \to \mathsf{infinite}$. We also need to consider the ω-power, a unary operation of type $\mathsf{finite} \to \mathsf{infinite}$ (since it turns a finite word into an infinite one).

Another example is discussed in Sect. 4.3, where algebras with infinitely many sorts are considered. We do not want to give here a detailed discussion of congruences in multi-sorted algebra, only pointing out that such congruences can only identify elements of the same sort. If there are finitely many sorts, recognizability is defined by considering morphisms into finite algebras, or finite-index congruences. If the algebraic signature under consideration has infinitely many sorts, non-trivial algebras are usually not finite, and we consider *locally finite* algebras (in which each sort has a finite number of elements) and *locally finite index* congruences (with a finite number of classes in each sort).

In the mid-1990s, Pin introduced the usage of *ordered semigroups* to refine the classification of recognizable languages [69]. The same idea can as naturally be used in any algebra (and has been for instance in [55]), but we will keep it outside the discussion in this paper to avoid increased complexity.

3.3 Definability vs. Recognizability

Based on the examples of words and trees, the natural language for logical definability of recognizable sets would seem to be MSO, *monadic second order logic.* It is actually more natural to use CMSO, *counting monadic second order logic.* CMSO [16,19] is monadic second order logic, enriched with the modulo quantifiers $\exists^{\bmod q} x$ introduced in Sect. 1. In the case of words, CMSO is equivalent to MSO. In fact, this holds for any relational structure that comes equipped with a linear order, or for which a linear order can be defined by a MSO-formula (*e.g.* A-labeled trees as in Sect. 2, or traces as in Sect. 4.2), but it is not true in general.

For instance, when discussing multisets (subsets of A with multiplicity), we can view them as A-labeled finite discrete graphs (graphs without edges). Then, MSO can only define finite and cofinite sets, and it is strictly weaker than CMSO. Note that, algebraically, the multisets on A under union, form the free commutative monoid on A. The same monoid can be interpreted in terms of traces (with a commutative alphabet), its elements are then viewed as certain directed acyclic graphs with one connected component per letter (see Sect. 4.2 on traces), and MSO is equivalent to CMSO in this context. The recognizable subsets are the same in both interpretations, since their definition is given in terms of the same algebraic structure, that of the free commutative monoid over A, but recognizability is equivalent to CMSO-definability in one interpretation, and to MSO-definability in the other.

Say that a map $\varphi\colon S \to T$ between sets of relational structures is a MS-transduction if there exist MSO-formulas (in the language of the relational structures in S) that express each $s\varphi$ (its domain and its relations) as a subset of a direct product of a fixed number of copies of s (see Courcelle [19] for a precise definition). For instance, if A_0 is the subset of constants in an alphabet A, the word in A_0^* formed by the leaves of an A-labeled tree t can be easily described by MSO-formulas inside the set of nodes of t.

Now consider a set of relational structures M, equipped with an algebraic structure with signature Σ. A simple example is given by $M = A^*$, the set of words on alphabet A, seen as a monoid: the signature Σ consists of a binary operation (interpreted in A^* as concatenation) and of $|A|$ constant symbols (interpreted in A^* as the letters of A). The valuation morphism val maps every

Σ-term (a Σ-labeled tree) to its interpretation in M. The following result is due to Courcelle [16,19].

Theorem 3.1. *If the valuation morphism is surjective and is an* MS*-transduction, then every* CMSO*-definable subset of* M *is recognizable.*

The mechanism of the proof is worth sketching: let $L \subseteq M$ be CMSO-definable. The inverse image of a CMSO-definable set by an MS-transduction is CMSO-definable, so $\mathsf{val}^{-1}(L)$ is CMSO-definable in T_Σ. But in the set of Σ-labeled trees, CMSO-definability is equivalent to MSO-definability, and hence to recognizability. And it is easy to show that if $\mathsf{val}^{-1}(L)$ is recognizable, then so is L.

Examining this sketch of proof also sheds light on the decidability of CMSO-defined sets, and on the complexity of such a decision problem. Suppose we have a parsing algorithm, which maps a given relational structure $x \in M$ to a Σ-term $\mathsf{parse}(x)$ describing it. Let $L \subseteq M$ be described by a CMSO-formula φ and let $x \in M$: we want to decide whether $x \in L$. An MSO-formula ψ describing $\mathsf{val}^{-1}(L)$ can be computed from φ and the formulas describing the MS-transduction val. The problem then reduces to deciding whether $\mathsf{parse}(x)$ satisfies ψ, and by Theorem 2.1, this can be solved (efficiently) by running $\mathsf{parse}(x)$ through a bottom-up tree automaton.

The converse of Theorem 3.1 does not always hold: there are situations, in particular in the discussion of languages of graphs, where some recognizable sets are not CMSO-definable. However, the two notions are known to be equivalent in important cases: we have already seen it for words or trees; other interesting situations are discussed in Sect. 4.2 and 4.3. It is interesting to note that a common feature of those situations where the notions of definability and recognizability coincide, is that we are able to describe a parsing function parse as an MS-transductions, and this is possible only because some finite generation condition is assumed to hold (which cannot be assumed for the class of all finite graphs).

For some of the specific relational structures discussed in the sequel, there is a notion of automaton that matches the definition of recognizability – but in many other situations, especially when dealing with graphs or posets, no such notion is known. In those cases, the algebraic approach is really the only tool we have to characterize logical definability, and to hope to bring about decision algorithms.

4 Recognizable Sets of Discrete Structures

For the discrete structures discussed in this section, fruitful algebraic structures have been introduced in the literature. The first measure of the interest of such algebraic structures, is whether the corresponding notion of recognizability matches some natural notion of logical definability, or some natural notion of recognizability by automata. A second measure of interest is whether the algebraic theory thus introduced allows us to characterize – and if possible decide

– significant classes of recognizable sets. Typically, deciding FO-definability is a key problem, but other classes may arise naturally depending on the type of discrete structures we consider.

4.1 Infinite Words

We start with infinite words because it is an area where the theory has been developed for a long time (Büchi's theorem goes back to the early 1960s), and has a strong algebraic flavor. Here we are talking of *one-way infinite words*, or *ω-words*, that is, A-labeled infinite chains, or elements of $A^{\mathbb{N}}$. For a detailed presentation of the results surveyed in this section, we refer the readers to Perrin and Pin's book [66] and to the survey papers [65,68].

The notions of *Büchi* and (deterministic) *Muller* or *Rabin automata* were evolved in the 1960s, and they were proved to have the same expressive power as MSO-formulas (on A-labeled infinite chains), and as *ω-rational expressions.* The latter describe every MSO-definable language of ω-words as finite unions of products of the form KL^ω, where K, L are recognizable languages of finite words and the ω-power denotes infinite iteration. In particular, this indicates that the sets of ω-words that can be accessed by MSO or automata-theoretic specifications are in a sense ultimately infinite iterations of a recognizable set of finite words.

An algebraic approach to ω-rational languages took longer to evolve. Early work of Arnold, Pécuchet, Perrin emphasized the necessary interplay of relations on $A^{\mathbb{N}}$ (concerning infinite words) and ordinary monoid congruences on A^* (concerning finite words). It also emphasized that nothing much could be expected from the monoid structure of $A^{\mathbb{N}}$, in which every product is equal to its first factor. Eventually, it was recognized that finite and infinite words cannot be considered separately, but they form a two-sorted algebra, as explained in Sect. 3.2. The definition of the binary concatenation product does not pose any problem, but must be split in one operation of type $\mathsf{finite}^2 \to \mathsf{finite}$ and one operation of type $\mathsf{finite} \times \mathsf{infinite} \to \mathsf{infinite}$. But the generation of infinite words from finite one can be envisaged in two fashions: we can consider an ω-ary product, of type $\mathsf{finite}^\omega \to \mathsf{infinite}$, or the unary ω-power operation, of type $\mathsf{finite} \to \mathsf{infinite}$. The first choice is termed an *ω-semigroup* (Perrin, Pin), and $A^\infty = A^+ \cup A^{\mathbb{N}}$ is (freely) generated by A as an ω-semigroup. The second choice is termed a *Wilke algebra* (Wilke), and the sub-Wilke algebra of A^∞ generated by A consists in the finite and ultimately periodic ω-words only. A Ramsey theorem shows however that on a finite set, a Wilke algebra structure can be canonically extended to an ω-semigroup structure, so that the consideration of these two algebraic structures yields the same class of recognizable languages.

The robustness of this algebraic approach to recognizable subsets of A^∞ (and not $A^{\mathbb{N}}$!) is such that an Eilenberg-style theory of varieties was developped (see Sect. 1), and that a good number of combinatorially or logically interesting classes of recognizable sets have been characterized algebraically (Perrin and Pin [66], Carton [22]).

From the algorithmic point of view, note that passing from a Büchi automaton to a deterministic Muller or Rabin automaton (say, for the purpose of complementation) is notoriously difficult, see Safra's exponential time algorithm, but no significantly better algorithm is possible [85].

Elegant results generalize this discussion to transfinite words, that is, A-labeled ordinals longer than ω, see the work of Bedon, Bruyère, Carton, Choueka [15,9,10,13].

For infinite trees, we know models of automata that are equivalent to MSO-decidability (Rabin, see [85]), but the extension of the algebraic ideas sketched above remains to be done. The finiteness results implied by Ramsey's theory seem much harder to obtain for trees.

4.2 Poset-Related Models

A pomset (partially ordered multiset) is an A-labeled poset. The first example, of course, is that of words, which are A-labeled chains. Other examples were considered, and first of all the case of traces.

Traces. There the alphabet is equipped with a structure – which can be viewed as an independence relation, or a dependence relation, or a distributed structure. A *trace* can then be viewed in several fashions: as an equivalence class of words in the free monoid A^*, in the congruence induced by the commutation of independent letters (so traces form a monoid); or as a so-called *dependence graph*, that is, an A-labeled poset where the order is constrained by the distributed structure of the alphabet, see Diekert and Rozenberg's book [27]. The latter is the more significant model, from the point of view of the original motivation of traces as a model of distributed computation.

The power of MSO-definability – interpreted on the dependence graph model – was proved to be equivalent to the power of *Zielonka's automata* (a model of automata which which incorporates information on the distributed structure of the alphabet), and to algebraic recognizability in the trace monoid [27].

Note that, as discussed in Sect. 3.3, in the particular case where the letters are independent from one another, the trace monoid is the free commutative monoid. When elements of this monoid are represented by trace dependence graphs, where for each letter $a \in A$, the set of A-labeled elements is a chain, then antichains have bounded cardinality (that of A), and a linearization of the poset can be defined by a MSO-formula, so MSO-definability is equivalent to CMSO-definability. When the elements of the same monoid are represented by finite discrete A-labeled graphs, without any edges, then MSO-definability is strictly weaker than CMSO-definability. In both cases however, recognizability is equivalent to CMSO-definability.

Good results are also known for FO-definable trace languages: they are characterized by star-free rational expressions, and by the aperiodicity of their syntactic monoid (Guaiana, Restivo, Salemi [42]), and important temporal logics with the same expressive power have been developed (see Thiagarajan and Walukiewicz [82] and Diekert and Gastin [26]).

There is a large body of literature on recognizable trace languages, and the results summarized above point to a rather well understood situation. Some questions however are not solved in a completely satisfactory fashion. For instance, the question of rational expressions for trace languages remains unclear (see the star problem): the difficulty comes from the fact that the star of a recognizable trace language may not be recognizable; the notion of concurrent star, which takes care of that obstacle, retains an *ad hoc* flavor [27]. Similarly, with the remarkable exception of FO-definable trace languages, the task of identifying, characterizing and deciding interesting subclasses of recognizable languages has eluded efforts.

One can argue that this is due to the loss of information that occurs if we consider the set of traces as a monoid – which we must do if the algebraic structure on the set of traces is that of a monoid: in the resulting definition of recognizability, a set of traces is recognizable if and only if its set of linearizations (in A^*) is recognizable. From an algebraic point of view, this puts the emphasis on commutation, but two traces may commute because they are independent, or because they are powers of a third one, in which case they are deeply dependent. From a more algorithmic point of view, what is done there is to reduce the study of a trace language to the study of the language of all its linearizations.

On the other hand, Zielonka's automata succeed in taking into account the distributed structure of the computation model, and are well-adapted to traces. Since they match monoid recognizability all the same, this points to the following problem: to find an alternative algebraic structure on the set of traces, which does not change the family of recognizable sets, yet better accounts for the distributed nature of that model, and hence (hopefully) naturally connects with Zielonka's automata (*i.e.*, provides an algebraic proof of Zielonka's theorem) and allows the identification and characterization of structurally significant subclasses of recognizable trace languages.

Infinite traces exhibit interesting properties, from the point of view of automata-recognizability and logical definability, see [27,30].

Message Sequence Charts and Communication Diagrams. *Message sequence charts* (*MSC*s) form a specification language for the design of communication protocols, that has attracted a lot of attention in the past few years. They can also be considered as specifications of particular pomsets, that are disjoint unions of k chains. An abstraction of this model is given by *Lamport Diagrams* (LDs) and by *Layered Lamport Diagrams* (LLDs), which are LDs subject to a boundedness condition.

Henriksen, Kumar, Mukund, Sohoni, Thiagarajan [43,44,62] considered the class of *bounded* finite MSC languages, defined by so-called *bounded* (Alur, Yannakakis [3]) or *locally synchronised* (Muscholl, Peled [63]) MSC-graphs. For bounded MSC languages, MSO-definability is equivalent to rationality of the language of all linearizations, and to recognizability by deterministic (resp. nondeterministic) *message-passing automata*. Kuske extended these results to FO-definable MSC languages, and to infinite bounded MSCs [53].

The restriction to classes of posets with a rational language of linearizations is rather severe, but little work so far has discussed definability or recognizability outside this hypothesis. Meenakshi and Ramanujam [59] and Peled [64] investigated decidable logics for MSCs and LLDs, that are structural, *i.e.*, not defined on the language of linearizations. There does not seem yet to exist an algebraic approach of (a subclass of LDs) that would match the power of MSO-definability.

Series-Parallel Pomsets. Sets of series-parallel pomsets (or *sp-languages*) were investigated by Lodaya, Weil, Kuske [52,55]. A poset is *series-parallel* if it can be obtained from singletons by using the operations of sequential and parallel product. There is a combinatorial characterization of these posets (N-free posets [41,87]), but the definition above naturally leads to the consideration of the so-called *series-parallel algebras* [55], that is, sets equipped with two binary associative operations, one of which is commutative. Kuske showed that an *sp*-language is recognizable if and only if it is CMSO-definable [52]. Lodaya and Weil introduced a model of branching automata and a notion of rational expressions, which they proved had the same expressive power [55]. However these automata accept not only the recognizable *sp*-languages, but also some non-recognizable ones.

The *bounded-width* condition is a natural constraint on *sp*-languages: a set L of series-parallel pomsets has bounded-width if there is a uniform upper bound on the cardinality of an anti-chain in the element so L. Results of Kuske, Lodaya and Weil [52,55] show that when we consider only bounded-width *sp*-languages, then recognizability is equivalent to automata-recognizability, to MSO-decidability, and to expressibility by a so-called *series-rational expression.* Under the bounded-width hypothesis, FO-definable *sp*-languages are characterized by a notion of star-free rational expressions, and by an algebraic condition on the syntactic *sp*-algebra which is analogous to the aperiodicity of monoids [52].

Texts and n-Pomsets. An *A-labeled text* is a finite A-labeled set, equipped with 2 linear orders. Texts form a particular class of the *2-structures* studied by Ehrenfeucht, Engelfriet, Harju, Proskurowski and Rozenberg [31,32,34]. Hoogeboom and ten Pas introduced an algebraic structure on the set of all texts [47]. This algebra has an infinite signature, but within any finitely generated subalgebra (generated by A and any finite subset of the signature, the hypothesis of *bounded primitivity* in [47,48]), recognizability is equivalent to MSO-definability.

The class of texts generated by the alphabet and the two arity 2 operations on texts (*alternating texts*) is of particular interest, as we now discuss.

A pair of linear orders $(\leq_1, \leq_2)$ on a finite set specifies and can be specified by a pair of partial orders $(\sqsubseteq_1, \sqsubseteq_2)$ such that every pair of distinct elements is comparable in exactly one of these partial orders (this defines a *2-poset*): it suffices to take $\leq_1 = \sqsubseteq_1 \cup \sqsubseteq_2$ and $\leq_2 = \sqsubseteq_1 \cup \sqsupseteq_2$; and conversely $\sqsubseteq_1 = \leq_1 \cap \leq_2$ and $\sqsubseteq_2 = \leq_1 \cap \geq_2$. Since the translation between texts and 2-posets is described by quantifier-free formulas, MSO-definability is preserved under this translation. On 2-posets, one can consider two natural operations: one behaves like a sequential product on $\sqsubseteq_1$ and a parallel product on $\sqsubseteq_2$; and the other is defined dually, exchanging the roles of the two partial orders. Let $SPB(A)$ be the algebra of 2-posets generated by A and these two operations. Ésik and Németh observed that these two operations on 2-posets translate to the two arity 2 operations of the text algebra; moreover, they introduced a simple model of automata for subsets of $SPB(A)$, whose power is equivalent to recognizability and to MSO-definability [35].

Ésik and Németh's automata can also be defined for n-posets, where they are also equivalent to recognizability and to MSO-definability.

Pomsets in General. There does not seem to be a natural model of automaton that makes sense on all pomsets. However, since pomsets can be represented by A-labeled directed acyclic graphs (*dags*), they are directly concerned by the discussion in the next section. In particular, and getting ahead of ourselves, let us observe that the subsignature of the modular signature consisting of the operations defined by graphs that are posets (resp. dags), generates the class of all finite posets (resp. dags) - and the results on CMSO-definability discussed in Sect. 4.3 therefore apply to pomset languages.

4.3 Graphs and Relational Structures

Graphs (edge- or vertex-labeled, colored, with designated vertices, etc), and beyond them, relational structures (*i.e.*, hypergraphs) are the next step, and they occur indeed in many modeling problems. The notion of logical definability is rather straightforward, although it may depend on the logical structure we consider on graphs (whether a graph is a set of vertices with a binary edge predicate, or two sets of vertices and edges with incidence predicates). From the algebraic point of view, there is no prominent choice for a signature to describe graphs. However, three signatures emerge from the literature. One of them, the modular signature, arises from the theory of modular decomposition of graphs, the other two (the HR- and the VR-signature) arise from the theory of graph grammars. We will also consider a fourth signature, on the wider class of relational structures. We will see that under suitable finiteness conditions, the resulting notions of recognizability are equivalent.

After briefly describing these signatures, and comparing the notions of recognizability which they induce, we rapidly survey known definability results. We conclude with the discussion of a couple of situations where automata-theoretic models have been introduced.

The Modular Signature. A concrete graph H with vertices $\{1, \dots, n\}$ and edge set E_H, induces an n-ary operation on graphs as follows: the vertex set of the graph $H\langle G_1, \dots, G_n\rangle$ is the disjoint union of the vertex sets of the G_i, it contains all the edges of the G_i, and for each edge $(i, j) \in E_H$, it also has all the edges from a vertex of G_i to a vertex of G_j. A graph is said to be *prime* if it cannot be decomposed non-trivially by such an operation. The *modular signature*

$\mathcal{F}_\infty$ consists of the set of all prime graphs, or rather, of one representative of each isomorphism class of prime graphs. It is an infinite signature. In particular, $\mathcal{F}_\infty$ contains a finite number of operations of each arity. The operations of arity 2 are the parallel product, the sequential product and the clique product: they are defined by the graphs with 2 vertices and no edge, 1 edge and 2 edges, respectively [18,88].

The theory of modular decomposition of graphs [57,61] shows that the class of all finite graphs is generated by the singleton graph and $\mathcal{F}_\infty$, and describes the relations between the operations in $\mathcal{F}_\infty$. Finite A-labeled graphs are generated by A and $\mathcal{F}_\infty$. If $\mathcal{F}$ is a finite subset of $\mathcal{F}_\infty$, the algebra generated by A and $\mathcal{F}$ is called the class of A-labeled $\mathcal{F}$-graphs.

For instance, if $\mathcal{F}$ consists of the sequential product, the $\mathcal{F}$-graphs are the finite words. If $\mathcal{F}$ consists of the parallel (resp. clique) product, they are the discrete graphs (resp. cliques). If $\mathcal{F}$ consists of the parallel and the sequential products, we get the series-parallel posets (see Sect. 4.2), and if it consists of the parallel and clique products, we get the cographs (see below).

The Signature *HR*. Here, graphs are considered as sets equipped with a binary edge predicate, and a finite number of constants (*i.e.*, designated vertices), called *sources*. Each finite set of source names defines a sort in the HR-algebra $\mathcal{GS}$ of *graphs with sources*. The operations in the algebra are the disjoint union of graphs with disjoint sets of source names, the renaming of sources, forgetting sources, and the fusion of two sources [19]. A number of variants can be considered, which do not affect the class of HR-recognizable subsets, see Courcelle and Weil [21]: the disjoint union can be replaced with parallel composition (source name sets need not be disjoint, and sources with the same name get identified), the sources may be assumed to be pairwise distinct (*source separated graphs*), the source renaming operations can be dropped, or the source forgetting operations, etc.

The signature HR emerged from the literature on graph grammars, and the acronym HR stands for *Hyperedge Replacement*. More precisely, the equational sets of graphs, relative to the signature HR, are known to enjoy good closure properties, and can be elegantly characterized in terms of recognizable tree languages and MS-transductions where both vertex and edge sets can be quantified (see Courcelle [19]).

The Signature *VR*. Now graphs are considered as sets equipped with a binary edge predicate, and a finite number of unary predicates (*i.e.*, colors on the set of vertices), called *ports*. Each finite set of port names defines a sort in the VR-algebra $\mathcal{GP}$ of *graphs with ports*. The operations in the algebra are the disjoint union, the edge adding operation (adding an edge from each p-port to each q-port for designated port names p, q), and the renaming and forgetting of port names. Again, the class of VR-recognizable subsets is not affected by variants such as the consideration of graphs where ports must cover the vertex set, or must partition it. It also coincides with $\mathcal{NLC}$-recognizable graphs, see Courcelle and Weil [21].

The signature VR (standing for *Vertex Replacement*) also emerged from the literature on graph grammars, and the equational sets of graphs, relative to the signature VR, enjoy good closure properties, and are characterized in terms of recognizable tree languages and MS-transductions where only vertex sets can be quantified (see Courcelle [19]).

The Signature $\mathcal{S}$ on Relational Structures with Sources. Subsuming the algebras of graphs with sources and with ports, we can consider the class of *relational structures with sources* $\mathcal{S}t\mathcal{S}$. These are sets equipped with a finite relational structures, and a finite number of constants (*sources*). Each pair consisting of a relational signature and a set of source names defines a sort, and the operations in the signature $\mathcal{S}$ are the disjoint union between sorts with disjoint sets of source names, and all the unary operations that can be defined on a given sort using quantifier-free formulas, see [21]. The operations in the signatures VR and HR are particular examples of such *quantifier-free definable operations*. The notion of $\mathcal{S}$-recognizability is not affected if we consider parallel composition instead of disjoint union (as for the signature HR), nor if we consider only structures where sources are separated [21].

Comparing the Notions of Recognizability. Combining a number of results of Courcelle and Weil [21], we find that a set of graphs is VR-recognizable if and only if it is $\mathcal{S}$-recognizable. Moreover, a VR-recognizable set of graphs is $\mathcal{F}_\infty$-recognizable, and it is also HR-recognizable (the implication for HR- and VR-equational sets goes in the other direction). Finally, $\mathcal{F}_\infty$-, HR- and VR-recognizability (resp. equationality) are equivalent under certain boundedness conditions.

In particular if we consider a set L of graphs without $\boldsymbol{K}_{n,n}$ for some n ($\boldsymbol{K}_{n,n}$ is the complete bipartite directed graph with $n+n$ vertices), then L is VR-recognizable if and only if it is HR-recognizable [21]. This sufficient condition is implied by the following boundedness properties (in increasingly general order): the graphs in L have uniformly bounded degree, they have bounded tree-width, they are sparse. The notion of bounded tree-width can be seen as a finite generation property, relative to the signature HR [19].

If $\mathcal{F}$ is a finite subset of the modular signature $\mathcal{F}_\infty$ and L is a set of $\mathcal{F}$-graphs, then L is $\mathcal{F}$-recognizable if and only if it is $\mathcal{F}_\infty$-recognizable (resp. VR-recognizable, $\mathcal{S}$-recognizable) [18,21].

Monadic Second-Order Definability. From the logical definability point of view, graphs can be seen as sets (of vertices) equipped with an edge predicate, or as pairs of sets (of vertices and edges respectively) with incidence predicates. Let us denote by CMSO[E] the CMSO emerging from the first point of view, and by CMSO[inc] the second one. It is easily verified that CMSO[E]-definable sets of graphs are also $\mathsf{CMSO[inc]}$-definable. Moreover $\mathsf{CMSO[E]}$-definability implies VR-recognizability, and $\mathsf{CMSO[inc]}$-definability implies HR-recognizability, see Courcelle [19].

Lapoire showed that if L is a set of graphs with bounded tree-width, then $\mathsf{CMSO[inc]}$-definability is equivalent to HR-recognizability [54]. Moreover, if L is uniformly sparse, then $\mathsf{CMSO[E]}$- and $\mathsf{CMSO[inc]}$-definability are equivalent (Courcelle [20]). In view of the equivalence result between HR- and VR-recognizability mentioned above, it would be interesting to find out whether both definabilities are also equivalent if L is without $\boldsymbol{K}_{n,n}$ for some n.

Returning to the modular signature, $\mathsf{CMSO[E]}$-definability implies $\mathcal{F}_\infty$-recognizability, by general reasons (see Sect. 3.3). Weil showed that the converse holds for sets of $\mathcal{F}$-graphs, provided $\mathcal{F}$ is finite and the operations of $\mathcal{F}$ enjoy a limited amount of commutativity (*weakly rigid signatur*e, see [88] for details). This assumption is rather general, and covers in particular all the cases where $\mathcal{F}$-graphs are dags or posets, and notably the case of *sp*-languages.

A typical example of a subsignature of $\mathcal{F}_\infty$ which is not weakly rigid, consists of the parallel and the clique products, two binary commutative, associative operations which generate the cographs. *Cographs* form a class of undirected graphs, closely related with comparability graphs (Corneil, Lerchs, Stewart [23]) and can be characterized as follows: an undirected graph is a cograph if and only if it does not contain P_4 (P_4 has vertex set $\{1, \ldots, 5\}$ and edges between i and $i+1$, $1 \leq i \leq 4$). The arguments that show that $\mathcal{F}$-recognizability is equivalent to $\mathsf{CMSO[E]}$-definability when $\mathcal{F}$ is weakly rigid, fail for cographs. Courcelle [18] asks whether CMSO-definability is strictly weaker than $\mathcal{F}$-recognizability for a general finite subsignature $\mathcal{F} \subseteq \mathcal{F}_\infty$: the first place to look for a counter-example seems to be cographs.

Series-$\boldsymbol{\Sigma}$ Algebras. In their investigation of *sp*-languages, Lodaya and Weil introduced series-Σ-algebras and their subsets ($s\Sigma$-languages): here Σ is a ranked alphabet (as in the study of tree languages, Sect. 2) and $\bullet$ is a binary associative operation not in Σ. The *$s\Sigma$-terms*, that is, the elements of the algebra freely generated by A, Σ and $\bullet$ (called the *free series-Σ-algebra over A*), can be viewed as finite sequences of Σ-trees, where each child of the root is in fact a smaller $s\Sigma$-term. Lodaya and Weil introduced a model of automata and a notion of rational expressions, both of which are equivalent to recognizability [56] – a result which generalizes the characterization of recognizability by finite automata for both words and trees. They showed that their result could be adapted if some of the operations in Σ were assumed to be commutative, but not if some amount of associativity was introduced (e.g. *sp*-languages, cographs). The logical dimension of $s\Sigma$-languages was not developped.

Automata for Graph Languages. We have seen some automata, designed for specific situations (finite and infinite words, traces, series-parallel pomsets, MSC languages, n-pomsets), see Sect. 4.2. As discussed there, these automata models match the expressiveness of recognizability and MSO-definability, sometimes under additional boundedness hypothesis.

For general graph languages, Thomas introduced the notion of *graph acceptor* [84,86], generalizing the *tiling systems* introduced earlier by Giammarresi and Restivo [40] for *pictures* (A-labeled rectangular grids). Recognizability by a graph acceptor was shown to be equivalent to EMSO-definability, where EMSO is the extension of FO by existential quantification of monadic second order variables.

4.4 Revisiting Trees

As mentioned in Sect. 2, the problem of deciding whether a given recognizable set of Σ-trees is FO-definable is still open, and various attempts to use the structure of Σ-algebras described in Sect. 2 to solve it in the spirit of Schützenberger's theorem (Theorem 1.2), have failed [46,71,72]. Recently, Ésik and Weil introduced a new algebraic framework to investigate this particular problem on tree languages [36]. The point was to enrich the algebraic framework, without modifying the notion of a recognizable subset, but introducing additional algebraic structure.

Ésik and Weil's algebras, called *preclones*, are multi-sorted algebras with one sort for each integer n, and Σ-trees form the 0-sort of the free Σ-generated preclone. As indicated, a set of Σ-trees is preclone-recognizable if and only if it is recognizable with respect to Σ-algebras; moreover, if L is a recognizable Σ-tree language, its syntactic preclone (which is finitary but not finite due to the infinite number of sorts) admits a finite presentation, encoded in the minimal bottom-up automaton of L. This is naturally important if we want to use syntactic preclones in algorithms.

The main result of [36] states that L is FO-definable if and only if its syntactic preclone lies in the least pseudovariety of preclones closed under *two-sided wreath products*, and containing a certain very simple 1-generated preclone. The two-sided wreath product is a generalization of the operation of the same name on monoids (Rhodes, Tilson, see [77]), and this result generalizes Schützenberger's theorem on finite words. It is the first algebraic characterization of FO-definable tree languages, but unfortunately, it is not clear at this point whether this characterization can be used to derive a decision algorithm.

The approach in [36] also applies to FO+MOD-definable tree languages, and other similarly defined languages.

4.5 Timed Models

Timed automata appeared in the 1990s, to represent the behavior of finite state systems subjected to explicit time constraints (Alur, Dill [2]). While they are already widely used, the foundations of the corresponding theory are still under development. There are several variants of these automata, such as event-clock automata, and of the models of timed computations (timed words, clock words, etc). There have also been several attempts to develop appropriate notions of rational expressions, that would be equivalent to the expressive power of timed automata, see Henzinger, Raskin, Schobbens [45], Asarin, Maler, Caspi [4], Dima [28], Maler, Pnueli [58] among others. At the same time, timed automata and timed languages may exhibit paradoxical behaviors, due to the continuous nature of time, so the central ideas and techniques from the classical theory cannot simply be enriched with timed constraints to account for the behavior of timed automata.

The development of an algebraic apparatus and of a logical formalism is also still in its infancy. One should mention however the recent work of Maler and Pnueli [58], and the results of Francez and Kaminski [38] and Bouyer, Petit and Thérien [12] on generalizations of timed languages and automata, to *automata on infinite alphabets* and to *data languages*, respectively. In both cases, an interesting notion of algebraic recognizability is introduced, that is at least as powerful as timed automata. languages. Moreover, several logics have been introduced (see for instance Demri, D'Souza [24]), but none is completely satisfactory with respect to the motivation of formulating and solving the controller synthesis problem, and none is connected in a robust way to an algebraic approach of recognizability.

The development of a complete theory of timed systems, incorporating automata-theoretic, algebraic and logical aspects, appears to be one of the more difficult challenges of the moment.

References

1. J. Almeida. *Finite semigroups and universal algebra*. Series in Algebra **3**, World Scientific, Singapore (1994).
2. R. Alur, D. Dill. A Theory of Timed Automata, *Theoret. Computer Science* **126** (1994) 183-235.
3. R. Alur, M. Yannakakis. Model Checking of Message Sequence Charts. In: *Proc. CONCUR 99*, Lect. Notes in Computer Science **1664**, 114-129, Springer, 1999.
4. E. Asarin, O. Maler, P. Caspi Timed Regular Expressions, *Journal of the ACM* **49** (2002) 172-206.
5. M. Beaudry. Finite idempotent groupoids and regular languages, *RAIRO Informatique théorique* **32** (1998) 127-140.
6. M. Beaudry. Languages recognized by finite aperiodic groupoids, *Theoret. Computer Science* **209** (1998) 299-317.
7. M. Beaudry, F. Lemieux. D. Thérien. Finite loops recognize exactly the regular open languages. In *Proc. ICALP 97*, Lect. Notes in Computer Science (1997), pp. 110-120.
8. M. Beaudry, F. Lemieux. D. Thérien. Star-free open languages and aperiodic loops. In *Proc. STACS 01*, Lect. Notes in Computer Science **2010**, 87-98, Springer, 2001.
9. N. Bedon. Logic over words on denumerable ordinals, *J. Computer and System Science* **63** (2001) 394-431.
10. N. Bedon. Automata, semigroups and recognizability of words on ordinals, *Intern. J. Algebra and Computation* **8** (1998) 1-21.
11. J. Berstel. *Transductions and context-free languages*, Teubner, Stuttgart (1979).
12. P. Bouyer, A. Petit, D. Thérien. An algebraic characterization of data and timed languages, *Information and Computation* **182** (2003) 137-162.

13. V. Bruyère, O. Carton. Automata on linear orderings. In: *Proc. DLT 02* (M. Ito, M. Toyama, eds.), Lect. Notes in Computer Science **2450**, 103-115, Springer 2002.
14. S. Cho, D.T. Huynh. Finite automaton aperiodicity is PSAPCE-complete. *Theoret. Computer Science* **88** (1991) 99-116.
15. Y. Choueka. Finite automata, definable sets, and regular expressions over ω^n-tapes, *J. Computer and System Sciences* **17** (1978) 81-97.
16. B. Courcelle. The monadic second-order logic of graphs I: recognizable sets of finite graphs, *Information and Computation* **85** (1990) 12-75.
17. B. Courcelle. Basic notions of universal algebra for language theory and graph grammars, *Theoret. Computer Science* **163** (1996) 1-54.
18. B. Courcelle. The monadic second-order logic of graphs X: Linear orders, *Theoret. Computer Science* **160** (1996) 87-143.
19. B. Courcelle. The expression of graph properties and graph transformations in monadic second order logic. In: *Handbook of Graph Grammars and Computing by Graph Transformations* (G. Rozenberg, ed.), vol. 1, chapter 5. World Scientific, 1997, 313-400.
20. B. Courcelle. The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications, *Theoret. Computer Science* **299** (2003) 1-36.
21. B. Courcelle, P. Weil. The recognizability of sets of graphs is a robust property, to appear.
22. O. Carton. Wreath product and infinite words, *J. Pure and Applied Algebra* **153** (2000) 129-150.
23. D. Corneil, H. Lerchs, L. Stewart. Complement reducible graphs, *Discrete Applied Mathematics* **3** (1981) 163-174.
24. S. Demri, D. D'Souza. An Automata-Theoretic Approach to Constraint LTL. In: *Proc. FSTTCS 02*, Lect. Notes in Computer Science **2256**, 121-132, Springer, 2002.
25. K. Denecke, S. L. Wismath. *Universal Algebra and Applications in Theoretical Computer Science*, Chapman and Hall (2002).
26. V. Diekert, P. Gastin. Pure future local temporal logics are expressively complete for Mazurkiewicz traces. In: *Proceedings of LATIN'04*, Lect. Notes in Computer Science **2976**, 232-241, Springer, 2004.
27. V. Diekert, G. Rozenberg (eds.). *The book of traces*, World Scientific (1995).
28. C. Dima. Real-time automata and the Kleene algebra of sets of real numbers, *J. Automata, Languages and Computation* **6** (2001) 3-23.
29. J. Doner. Tree acceptors and some of their applications, *J. Comput. System Sci.* **4** (1970) 406-451.
30. W. Ebinger, A. Muscholl. On logical definability of omega-trace languages. *Theoret. Computer Science* **154** (1996) 67-84.
31. A. Ehrenfeucht, G. Rozenberg. Theory of 2-structures, Part 1: clans, basic subclasses and morphism. Part 2: representation through labeled tree families, *Theoret. Computer Science* **70** (1990) 277-303, 305-342.
32. A. Ehrenfeucht, G. Rozenberg. Angular 2-structures, *Theoret. Computer Science* **92** (1992) 227-248.
33. S. Eilenberg. *Automata, languages and machines*, vol. A (1974) and vol. B (1976), Academic Press, New York.
34. J. Engelfriet, T. Harju, A. Proskurowski, G. Rozenberg. Characterization and complexity of uniformly nonprimitive labeled 2-structures, *Theoret. Computer Science* **154** (1996) 247-282.
35. Z. Ésik, Z. Németh. Higher dimensional automata, to appear.

36. Z. Ésik, P. Weil. On logically defined recognizable tree languages. In: *Proc. FST TCS 2003* (P. K. Pandya, J. Radhakrishnan eds.), Lect. Notes in Computer Science **2914** (2003), Springer, 195-207.
37. K. Etessami, M. Y. Vardi, Th. Wilke. First-order logic with two variables and unary temporal logic. *Inform. and Computation* **179** (2002) 279-295.
38. N. Francez, M. Kaminski. An algebraic characterization of deterministic regular languages over infinite alphabets, *Theoret. Computer Science* **306** (2003) 155-175.
39. F. Gécseg, M. Steinby. *Tree automata*, Akadémiai Kiadó, Budapest (1994).
40. D. Giammarresi, A. Restivo. Two-dimensional languages. In: Handbook of Formal Languages (G. Rozenberg and A. Salomaa, eds), volume III, Springer, New York, 1997.
41. J. Grabowski. On partial languages, *Fundamenta Informatica* **4** (1981) 427-498.
42. G. Guaiana, A. Restivo, S. Salemi. Star-Free Trace Languages, *Theoret. Computer Science* **97** (1992) 301-311.
43. J.G. Henriksen, M. Mukund, K. Narayan Kumar, P.S. Thiagarajan. Regular Collections of Message Sequence Charts. In: *Proc. MFCS 00*, Lect. Notes in Computer Science **1893**, 405-414, Springer, 2000.
44. J.G. Henriksen, M. Mukund, K. Narayan Kumar, P.S. Thiagarajan. On Message Sequence Graphs and Finitely Generated Regular MSC Languages. In: *Proc. ICALP 00*, Lect. Notes in Computer Science **1853**, 675-686, Springer, 2000.
45. T. Henzinger, J.-F. Raskin, P.-Y. Schobbens. The Regular Real-Time Languages. In: *Proc. ICALP 98*, Lect. Notes in Computer Science **1443**, 580-591, Springer, 1998.
46. U. Heuter. First-order properties of trees, star-free expressions, and aperiodicity. In: *Proc. STACS 88*, Lect. Notes in Computer Science **294**, 136-148, Springer, 1988.
47. H.J. Hoogeboom, P. ten Pas. Text languages in an algebraic framework, *Fundamenta Informatica* **25** (1996) 353-380.
48. H.J. Hoogeboom, P. ten Pas. Monadic second order definable text languages, *Theory Comput. Syst.* **30** (1997) 335-354.
49. J. A. Kamp. *Tense logic and the theory of linear order*, Ph. D. Thesis, UCLA, 1968.
50. S. Kim, R. McNaughton, R. McCloskey. A polynomial time algorithm for the local testability problem of deterministic finite automata, *IEEE Trans. Comput.* **40** (1991) 1087-1093.
51. E. Kushilevitz, N. Nisan. *Communication Complexity*, Cambridge University Press (1997).
52. D. Kuske. Towards a language theory for infinite N-free pomsets, *Theoret. Computer Science* **299** (2003) 347-386.
53. D. Kuske. Regular sets of infinite message sequence charts, *Information and Computation* **187** (2003) 90-109.
54. D. Lapoire. Recognizability equals Monadic Second-Order definability, for sets of graphs of bounded tree-width. In: *Proc. STACS 98.* Lect. Notes in Computer Science **1373**, 618-628, Springer, 1998.
55. K. Lodaya, P. Weil. Series-parallel languages and the bounded-width property, *Theoret. Computer Science* **237** (2000) 347-380.
56. K. Lodaya, P. Weil. Rationality in algebras with a series operation, *Information and Computation* **171** (2001) 269-293.
57. R. McConnell, J. Spinrad. Modular decomposition and transitive orientation, *Discrete Mathematics* **201** (1999) 189-241.

58. O. Maler, A. Pnueli. On Recognizable Timed Languages. In: *Proc. FOSSACS 04*, Lect. Notes in Computer Science **2987**, 348-362, Springer, 2004.
59. B. Meenakshi and R. Ramanujam. Reasoning about layered message-passing systems. In: *Proc. VMCAI 03*, of Lect. Notes in Computer Science **2575**, 268-282, Springer, 2003.
60. J. Mezei and J. B. Wright. Algebraic automata and context-free sets, *Information and Control* **11** (1967) 3-29.
61. R.H. Möhring, F.J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization, *Annals of Discrete Mathematics* **19** (1984) 257-356.
62. M. Mukund, K. Narayan Kumar, M. Sohoni. Synthesizing distributed finite-state systems from MSCs. In: *Proc. CONCUR 00*, Lect. Notes in Computer Science **1877**, 521-535, Springer, 2000.
63. A. Muscholl, D. Peled. Message sequence graphs and decision problems on Mazurkiewicz traces. In: *Proc. MFCS 99* (M. Kutylowski, L. Pacholski, eds.), Lect. Notes in Computer Science **1672**, 81-91, Springer, 1999.
64. D. Peled. Specification and Verification of Message Sequence Charts. In: *Proc. FORTE 2000*, IFIP Conference Proceedings **183**, 139-154, Kluwer, 2000.
65. D. Perrin, J.-E. Pin. Semigroups and automata on infinite words. In: *NATO Advanced Study Institute Semigroups, Formal Languages and Groups* (J. Fountain, ed.), pp. 49-72, Kluwer, 1995.
66. D. Perrin, J.-E. Pin. *Infinite words*, Pure and Applied Mathematics Vol. 141, Elsevier (2004).
67. J.-E. Pin. *Variétés de langages formels*, Masson, Paris (1984); English translation: *Varieties of formal languages*, Plenum, New-York (1986).
68. J.-E. Pin. Logic, Semigroups and Automata on Words. *Annals of Mathematics and Artificial Intelligence* **16** (1996) 343–384.
69. J.-E. Pin. Syntactic semigroups. Chap. 10 in: *Handbook of formal languages*, Vol. I (G. Rozenberg et A. Salomaa, eds.), Springer Verlag, 1997, 679-746.
70. J.-E. Pin, P. Weil. Polynomial closure and unambiguous product. *Theory Comput. Systems* **30** (1997) 383-422.
71. A. Potthoff. Modulo counting quantifiers over finite trees. In: *Proc. CAAP '92*, Lect. Notes in Computer Science **581**, 265-278, Springer, 1992.
72. A. Potthoff. First order logic on finite trees. In: *Proc. TAPSOFT '95*, Lect. Notes in Computer Science **915**, Springer, 1995.
73. J. Sakarovitch. *Élements de théorie des automates*, Vuibert, Paris (2003).
74. M. P. Schützenberger. Sur le produit de concaténation non ambigu. *Semigroup Forum* **13** (1976) 47-75.
75. M. Sipser. *Introduction to the theory of computation*, International Thompson Publishing (1996).
76. J. Stern. Characterization of some classes of regular events. *Theoret. Computer Science* **35** (1985) 17-42.
77. H. Straubing. *Finite automata, formal logic, and circuit complexity*, Birkhaüser (1994).
78. P. Tesson, D. Thérien. Complete classifications for the communication complexity of regular languages. In *STACS 2003*, H. Alt, M. Habib (Eds.), Lect. Notes in Computer Science **2607**, pp. 62-73, Springer (2003).
79. J. W. Thatcher, J. B. Wright. Generalized finite automata theory with an application to a decision problem of second-order logic. *Math. Systems Theory* **2** (1968) 57-81.

80. D. Thérien, Th. Wilke. Over words, two variables are as powerful as one quantifier alternation: $\mathsf{FO}^2 = \Sigma_2 \cap \Pi_2$. In *Proc. 13th ACM SToC*, 41-47, 1998.
81. D. Thérien, Th. Wilke. Temporal logic and semidirect products: An effective characterization of the until hierarchy. *SIAM Journal on Computing* **31** (2002) 777-798.
82. P.S. Thiagarajan and I. Walukiewicz. An expressively complete linear time temporal logic for Mazurkiewicz traces. In *Proc. LICS 97*, 183-194, 1997.
83. W. Thomas. Classifying regular events in symbolic logic. *J. Comput. Syst. Sci.* **25** (1982) 360–375.
84. W. Thomas. Automata theory on trees and partial orders. In: *Proc. TAPSOFT 97* (M. Bidoit, M. Dauchet, eds), Lect. Notes in Computer Science **1214**, 20-38, Springer, 1997.
85. W. Thomas. Automata on infinite objects. In: *Handbook of Theoretical Computer Science* (J. van Leeuwen ed.), vol. B. Elsevier, 1990, 133-192.
86. W. Thomas. Languages, automata, and logic. In: *Handbook of Formal Languages* (G. Rozenberg, A. Salomaa, eds), volume III, 389-455, Springer, New York, 1997.
87. J. Valdes, R.E. Tarjan and E.L. Lawler. The recognition of series-parallel digraphs, *SIAM J. Comput.* **11** (1981) 298-313.
88. P. Weil. On the logical definability of certain graph and poset languages, *J. Automata, Languages and Computation* **9** (2004) 147-165.