On the logical definability of certain graph and poset languages*

Pascal Weil
LaBRI
Université Bordeaux-1 and CNRS
351 cours de la Libération
33405 Talence Cedex, France.
Email: pascal.weil@labri.fr

Abstract

We show that it is equivalent, for certain sets of finite graphs, to be definable in CMS (counting monadic second-order, a natural extension of monoadic second-order logic), and to be recognizable in an algebraic framework induced by the notion of modular decomposition of a finite graph.

More precisely, we consider the set $\mathcal{F}_\infty$ of composition operations on graphs which occur in the modular decomposition of finite graphs. If $\mathcal{F}$ is a subset of $\mathcal{F}_\infty$, we say that a graph is an $\mathcal{F}$-graph if it can be decomposed using only operations in $\mathcal{F}$. A set of $\mathcal{F}$-graphs is recognizable if it is a union of classes in a finite-index equivalence relation which is preserved by the operations in $\mathcal{F}$. We show that if $\mathcal{F}$ is finite and its elements enjoy only a limited amount of commutativity — a property which we call weak rigidity, then recognizability is equivalent to CMS-definability. This requirement is weak enough to be satisfied whenever all $\mathcal{F}$-graphs are posets, that is, transitive dags. In particular, our result generalizes Kuske's recent result on series-parallel poset languages.

The connection between recognizability and definability is one of the cornerstones of theoretical computer science, going back to Büchi's celebrated theorem on finite and infinite words in the 1960s (see [26]). This theorem states the equivalence between two fundamental properties of a language:

- to be definable in monadic second order logic (MS),
- to be recognizable.

In Büchi's work, recognizability is defined by means of a finite state automaton. It is well-known that recognizability by such an automaton is equivalent to *algebraic recognizability*, that is, to being the union of classes in a finite-index

*Partial support from the following sources is gratefully acknowledged: Project 2102-1 of IFCPAR/CEFIPRA; and AS *Automates, modèles distribués et temporisés* of Département STIC, CNRS.

congruence. (This is well-known for languages of finite words but it also holds, with the appropriate definitions, for languages of infinite words [22, 23]).

One direction in with this result has been extended, is the consideration of sets of other combinatorial structures than words. For instance, MS-definability and recognizability are equivalent for trace languages (see [9]). Traces can be viewed as finite posets, labeled by letters in a given alphabet A — and as such as generalizations of words, which are A-labeled linearly ordered finite sets.

A further generalization is the consideration of finite graphs. Graph languages have been widely studied for the description of complex structures or behaviors; among them, languages of partially ordered sets (*posets*) are used for modeling certain aspects of concurrency.

To handle graph languages, several definitions of algebraic recognizability can be adopted, depending on which operations on graphs (or constructors) are put forward (see Section 2). In this paper, we consider the operations on graphs given by the theory of the modular decomposition of graphs, which we call the *modular signature* (see Section 2.1). Courcelle [5] already considered this algebraic framework, and proved that CMS-definability implies recognizability (Theorem 3.1 below; see Section 1.2 about CMS *vs.* MS). Lodaya, Weil [16, 17] and Kuske [14] also considered a restriction of this algebraic framework: to a particular class of graphs, the series-parallel posets, and to the modular operations which suffice to generate them, namely the sequential and the parallel product. In particular, Kuske proved [14] the equivalence of CMS-definability and recognizability in that case. Our main result establishes the same equivalence in a wider setting: it holds for a language of finite A-labeled graphs L provided the graphs in L can be generated from one-vertex graphs using a finite number of operations in the modular signature; and provided that, apart from the parallel product (or disjoint union) of graphs, these operations enjoy only a limited amount of commutativity — a property which we call *weak rigidity* (Section 4.1). As it turns out, this requirement is weak enough and it is satisfied by all the finite subsets of the modular signature which generate only posets or even dags. In other words, our result applies to languages of finite A-labeled posets under a natural finite generation hypothesis, thus generalizing Kuske's result.

The proof, which generalizes ideas from proofs of Kuske [14] and Courcelle [5], relies on the consideration of certain tree-like normal forms for graphs (relative to the modular signature) and uses crucially the notion of an MS-definable transduction [6].

Other operations have been considered, to make the set of finite graphs into a (multi-sorted) algebra. Among the most important such signatures, we mention the VR and the HR signatures, see [6]. In a series of papers (*e.g.* [2, 3, 4, 5, 6]), Courcelle and co-authors have studied the connection between definability and recognizability with respect to these signatures. A remarkable result in that direction is the equivalence between CMS-definability (see Section 1.2 about CMS *vs.* MS) and algebraic recognizability with respect to HR, for languages of graphs of tree-width bounded by an integer k (Courcelle [3] for $k = 2$, Kaller [13] for $k = 3$ and Lapoire [15] for the general case). This result is, however, incomparable with ours.

1 Terminology and notation

In this section, we fix the notation and definitions which we will use, concerning graphs or posets, and the logical apparatus to specify their properties.

1.1 Graphs and posets

In this paper, all graphs are assumed to be finite.

We consider directed A-(vertex-)labeled graphs of the form $G = (V, E, \lambda)$ where V is the (finite) set of *vertices*, $E \subseteq V \times V$ is the *edge relation* and the *labeling function* $\lambda\colon V \to A$ is a mapping into a fixed alphabet A (a finite, non-empty set).

When the labeling is irrelevant, we omit λ in the description of G. Undirected graphs are considered as a special case of directed graphs, where the edge relation is symmetric: $(x, y) \in E$ if and only if $(y, x) \in E$.

We always assume that our graphs do not have self-loops (edges of the form (x, x)), that is, E is an anti-reflexive relation. If necessary, the presence of a self-loop at vertex x can be encoded in the letter labeling x.

At times, we view graphs up to isomorphism, and at other times, we insist on so-called concrete graphs. More precisely, when we consider graph languages, the graphs in question are up to isomorphism. When we use graphs as syntactic tools to define algebraic operations (as in Section 2.2), then the name of vertices is important, that is, distinct graphs may well be isomorphic.

A *dag*, or *directed acyclic graph* is a directed graph $G = (V, E)$ in which no path is a loop. The graph G is said to be *transitive* if the edge relation E is transitive. In particular, $G = (V, E)$ is a transitive dag if and only if E is a partial order relation on V (*minus* the reflexivity part of the relation, that is, the pairs (x, x), $x \in V$, since we are considering graphs without self-loops). When we talk of posets, we always refer to the associated transitive dags, so that a poset language is a special kind of graph language.

1.2 Logics

In this paper, we need to discuss logical properties of various kinds of relational structures, beyond labeled graphs and posets as defined above. For this purpose, we use the classical notions as in, say, [8].

In general, let $\mathcal{S}$ be a finite relational signature, that is, a finite set equipped with a mapping $\alpha\colon \mathcal{S} \to \mathbb{N}$ into the non-negative integers, called the *arity function*. An *$\mathcal{S}$-structure* is a set X (the *domain set* of the structure) equipped, for each $s \in \mathcal{S}$, with a relation s^X of arity $\alpha(s)$.

When $\mathcal{S}$ is fixed, logical formulas can be built using the usual connectives and quantifiers, and the elements of $\mathcal{S}$ as predicates (respecting the arity function α).

Example 1.1 When we discuss A-labeled graphs in this paper, the signature consists of the binary relation E (edge relation) and, for each letter $a \in A$, of a unary relation Λ_a. The graph is then, in effect, viewed as a structure with domain set its vertex set V. □

When quantification is allowed only on elements of the domain, we talk of *first-order* or *FO*-formulas. If we quantify also on sets of elements (unary relations on the domain), we talk of *monadic second order*, or *MS*-formulas.

We also make intensive use of the following extension of monadic second order logic. The formalism of *MS*-formulas is enriched with special quantifiers of the form $\exists^{\mathbf{mod}\,q}x$, where $q \geq 2$ is an integer and x is a first-order variable. A formula of the form $\exists^{\mathbf{mod}\,q}x\ \varphi(x)$ is interpreted to mean that the cardinality of the set of values of x such that $\varphi(x)$ holds is 0 mod q. The resulting logic is called *CMS* (*counting monadic second order* logic) [2].

It is well-known that *CMS* is strictly more expressive than *MS*: no *MS* formula can express the fact that an $\mathcal{S}$-structure has even cardinality [2]. On the other hand, *CMS* is designed precisely to express this type of properties.

2 Recognizability: the algebraic framework

The notion of recognizability was established in the 1960s by Mezei and Wright [20]. It makes sense with respect to a given algebraic framework, that is, in a given algebra, for a given signature.

More precisely, let $\mathcal{F}$ is a signature (finite or infinite), that is, a set equipped with a mapping $\alpha\colon \mathcal{F} \to \mathbb{N}$ into the non-negative integers, called the *arity function*. An $\mathcal{F}$*-algebra* is a set X equipped, for each $f \in \mathcal{F}$, with an $\alpha(f)$-ary operation f^X. Morphisms of $\mathcal{F}$-algebras are defined in the usual way, see [1]. A subset L of an $\mathcal{F}$-algebra X is said to be $\mathcal{F}$*-recognizable* (*recognizable* if there is no ambiguity) if there exists a morphism of $\mathcal{F}$-algebras φ from X into a finite $\mathcal{F}$-algebra such that $L = \varphi^{-1}(\varphi(L))$. Thus, the notion of a recognizable subset of X depends on the algebraic structure considered on X.

There are several natural ways to view the set of all (finite) graphs as an algebra, and hence several different notions of recognizability (see for instance Courcelle in [4, 6]). In this paper, we operate in the algebraic framework provided by the existence and the uniqueness of the so-called *modular decomposition* of finite graphs. The relevant definitions are given in the next sections.

2.1 Composition of graphs: the modular signature

If $n \geq 1$ is an integer, we denote by $[n]$ the set $\{1, \dots, n\}$.

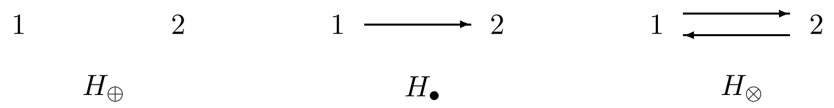
With each n-vertex graph H, we associate an n-ary operation on graphs. In order to properly define our algebraic setting, operations must have a linearly ordered set of arguments, and hence we need to view H as a concrete graph with vertex set $[n]$. In particular, distinct isomorphic graph structures on $[n]$ define different operations.

Let $H = ([n], F)$ and let $G_1, \dots, G_n$ be graphs. The graph $H\langle G_1, \dots, G_n\rangle$ is obtained by taking the disjoint union of the graphs $G_1, \dots, G_n$, and by adding, for each edge $(i, j) \in F$, an edge from every vertex of G_i to every vertex of G_j. In other words, if $G_i = (V_i, E_i)$ for $i = 1, \dots, n$, then $H\langle G_1, \dots, G_n\rangle = (V, E)$ where

$$V = V_1 \sqcup \cdots \sqcup V_n$$

$$E = E_1 \sqcup \cdots \sqcup E_n \sqcup \bigsqcup_{(i,j)\in F} V_i \times V_j$$

The following 2-vertex graphs provide particularly important examples of such operations.



The binary operation defined by $H_\oplus$, written $G_1 \oplus G_2$, is simply the disjoint union of G_1 and G_2; it is sometimes called the *parallel product* of graphs.

The binary operation defined by $H_\bullet$, written $G_1 \bullet G_2$, is called the *sequential product*, and it consists of adding to $G_1 \oplus G_2$ every edge from a vertex of G_1 to a vertex of G_2.

The binary operation defined by $H_\otimes$, written $G_1 \otimes G_2$, is called the *clique product*, and it consists of adding to $G_1 \oplus G_2$ every edge from a vertex of G_1 to a vertex of G_2 and every edge from a vertex of G_2 to a vertex of G_1.

It is immediately seen that these three operations are associative, and that the operations $\oplus$ and $\otimes$ are commutative.

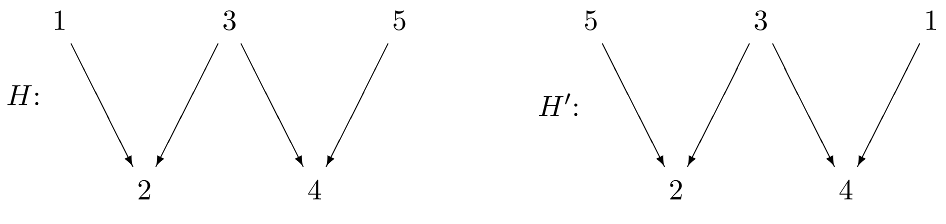
We also note the following *compositionality property*: if the graph H itself can be written as a composition, say, $H = K\langle L_1, \ldots, L_r\rangle$, then the composition $H\langle G_1, \ldots, G_n\rangle = K\langle L'_1, \ldots, L'_r\rangle$ and $L'_j = L_j\langle G_{i_{j,1}}, \ldots, G_{i_{j,r_j}}\rangle$ where $i_{j,1}, \ldots, i_{j,r_j}$ are the vertices of H in L_j.

If H cannot be written as a composition, we say that H is *prime*: the compositionality property above implies that every composition operation can be expressed in terms of operations defined by prime graphs.

Finally we note the following *commutation properties*: if $H = ([n], F)$ and $H' = ([n], F')$ are isomorphic graphs, then the corresponding composition operations differ only by the order of the arguments. More precisely, if σ is a permutation of $[n]$ which induces an isomorphism from H' into H, then

$$H\langle G_1, \ldots, G_n\rangle = H'\langle G_{\sigma(1)}, \ldots, G_{\sigma(n)}\rangle \qquad (CP)$$

Example 2.1 Let H and H' be the following concrete (prime) graphs:



The permutation (1 5) defines an isomorphism from H' to H and we have $H\langle G_1, G_2, G_3, G_4, G_5\rangle = H'\langle G_5, G_2, G_3, G_4, G_1\rangle$.

Similarly, the permutation (1 5)(2 4) defines an automorphism of H and we have $H\langle G_1, G_2, G_3, G_4, G_5\rangle = H\langle G_5, G_4, G_3, G_2, G_1\rangle$. □

In particular, we may restrict the set of concrete prime graphs defining composition operations to having at most one representative of every isomorphism class: in the above example, every H'-product can be expressed as an H-product. However, it remains necessary to retain a concrete presentation of H, in order to have a unequivocal linear order on the arguments of the corresponding operation. Note that this restriction does not eliminate the commutation properties (CP): each automorphism of a prime graph induces one.

In the sequel, we select a set $\mathcal{F}_\infty$ (the *modular signature*) consisting of the binary operations $\oplus, \otimes, \bullet$, and of the composition operations defined by a collection of graphs containing exactly one representative of each isomorphism class of prime graphs with at least three vertices. We will now view the class of finite graphs as an $\mathcal{F}_\infty$-algebra.

It is important to observe that $\mathcal{F}_\infty$ is infinite, since there are infinitely many isomorphism classes of finite prime graphs. In fact, almost all finite graphs are prime: more precisely, their relative frequency among n-vertex graphs tends to 1, see [21].

2.2 Modular decomposition

Let $G = (V, E)$ be a graph. A *module* in G is a subset X of V which interacts uniformly with its complement $V \setminus X$: more precisely, if $v \in V \setminus X$ and E contains a pair (v, x) with $x \in X$, then $\{v\} \times X \subseteq E$; and dually, if $(x, v) \in E$ for some $x \in X$, the $X \times \{v\} \subseteq E$.

We say that a module X is *prime* if $X \neq V$ and for every module Y, either $Y \subseteq X$ or $X \subseteq Y$ or $X \cap Y = \emptyset$. One can show that the prime modules of a prime module X of G are prime modules of G. In addition, if V is finite, then the maximal prime modules of G form a partition of V. Let $\equiv$ be the corresponding equivalence relation on V and let H be the quotient graph $H = G/\!\equiv$: its vertex set is $V/\!\equiv$ and its edge relation is the image of E in the projection from $V \times V$ onto $(V/\equiv) \times (V/\equiv)$. Then one can show that H is either a prime graph $H = ([n], F)$ with $n \geq 3$, or it is the transitive closure of one of the three following graphs (for $n \geq 2$):

$$1 \longrightarrow 2 \longrightarrow \cdots \longrightarrow n \qquad \textit{(n-element linear order)}$$

$$1 \qquad 2 \qquad \cdots \qquad n \qquad \textit{(n-element set)}$$

$$1 \rightleftarrows 2 \rightleftarrows \cdots \rightleftarrows n \qquad \textit{(n-element clique)}$$

In particular, if the maximal prime modules of G are $G_1, \ldots, G_n$, then exactly one of the following holds:

$$\begin{aligned}
G &= H\langle G_1, \ldots, G_n \rangle \\
G &= G_1 \bullet G_2 \bullet \cdots \bullet G_n \\
G &= G_1 \oplus G_2 \oplus \cdots \oplus G_n \\
G &= G_1 \otimes G_2 \otimes \cdots \otimes G_n
\end{aligned}$$

It follows that each finite graph can be constructed from singleton graphs, using operations from the modular signature. Such a description of a graph is called its *modular decomposition*. In other words, the class of all finite graphs is the $\mathcal{F}_\infty$-algebra generated by a single element.

Moreover, the modular decomposition of a finite graph is unique up to the associativity of $\bullet, \oplus, \otimes$, the commutativity of $\oplus, \otimes$, and the commutation properties (CP), based on the non-trivial automorphisms of prime graphs.

The above discussion has been entirely concerned with unlabeled graphs. The generators of the $\mathcal{F}_\infty$-algebra of A-labeled graphs are simply the A-labeled one-vertex graphs: in other words, the $\mathcal{F}_\infty$-algebra of A-labeled graphs is generated by A.

Finally, if $\mathcal{F} \subseteq \mathcal{F}_\infty$, we say that a (labeled) graph is an *$\mathcal{F}$-graph* if it is in the $\mathcal{F}$-algebra generated by the singleton graphs.

Remark 2.2 The idea of the modular decomposition of a graph has been rediscovered a number of times in the context of graph theory and of other fields using graph-theoretic representations. We refer to [21] for a historical survey of this question, and to [19] for a concise presentation.

The modular decomposition of a graph can be computed in linear time [18, 19, 7]. □

Remark 2.3 If $\mathcal{F}$ contains only dags, then the $\mathcal{F}$-graphs are dags. If in addition, $\mathcal{F}$ consists only of transitive dags (that is, posets), then the $\mathcal{F}$-graphs are posets. Conversely, every prime graph occurring in the modular decomposition of a dag (resp. a poset) is a dag (resp. a poset).

Similarly, if $\mathcal{F}$ contains only undirected graphs (graphs with a symmetric edge relation), then the $\mathcal{F}$-graphs are all undirected. Conversely, every prime graph occurring in the modular decomposition of an undirected graph is undirected. □

2.3 Tree-like representations

We will use the following tree-like representations of an A-labeled graph to account for its modular decomposition.

We first consider the tree $\mathbf{mdec}(G)$ (Courcelle [5, Sec. 6]), whose set of nodes is the set of prime modules of G, and such that a node x is the parent of a node y if and only if y is a maximal prime module of x. Moreover, each leaf x of $\mathbf{mdec}(G)$ (necessarily a single vertex) is labeled by $\lambda(x) \in A$, and each inner node of $\mathbf{mdec}(G)$ is labeled H (a prime graph in $\mathcal{F}_\infty$ with at least three vertices), $\bullet$, $\oplus$ or $\otimes$, according to the fact that x is an H-product, a $\bullet$-product, a $\oplus$-product, or a $\otimes$-product of its maximal prime modules.

In particular, each $\bullet$-labeled node has at least 2 children, none of which is $\bullet$-labeled; the analogous property holds for each $\oplus$-labeled node and for each $\otimes$-labeled node. Each H-labeled node has n children if H has n vertices.

In addition to this tree structure, $\mathbf{mdec}(G)$ also encodes the following information. First, there is a linear order on the children of a $\bullet$-labeled node x, which comes from the modular decomposition of x. There is no such order on the children of $\oplus$- or $\otimes$-labeled nodes. The case of an H-labeled node x (where H is an n-vertex prime graph in $\mathcal{F}_\infty$, $n \geq 3$) is intermediary: the modular decomposition of x provides an enumeration (that is, a linear order) of the n children of x, which is defined up to the action of $\mathrm{Aut}(H)$; equivalently, the modular decomposition of x provides a collection of linear orders on the children of x, such that any of these order relations can be mapped to any other one by some permutation $\sigma \in \mathrm{Aut}(H)$.

In view of the discussion in Section 2.2, this enriched tree structure uniquely defines G.

Technically, we view $\mathbf{mdec}(G)$ as a relational structure whose domain are the prime modules of G, together with the following (interpreted) predicates:

> $\mathbf{child}(x,y)$ *if* y *is a maximal prime module of* x,
> $\mathbf{label}_a(x)$ *(*$a \in A$*) if* x *is an* a*-labeled vertex of* G,
> $\mathbf{label}_\oplus(x)$ *if* x *is an* $\oplus$*-product of its maximal prime modules,*
> $\mathbf{label}_\otimes(x)$ *if* x *is an* $\otimes$*-product of its maximal prime modules,*
> $\mathbf{label}_\bullet(x)$ *if* x *is a* $\bullet$*-product of its maximal prime modules,*
> $x < y$ *if there exists a prime module* z*, with maximal prime modules* $z_1, \ldots, z_n$*, such that* $z = z_1 \bullet \cdots \bullet z_n$*,* $x = z_i$ *and* $y = z_j$ *for some* $1 \leq i < j \leq n$*.*
> $\mathbf{label}_H(x)$ *(with* $H \in \mathcal{F}_\infty$ *a graph with* $n \geq 3$ *vertices) if* x *is an* H*-product of its maximal prime modules,*
> $\mathbf{children}_H(x, y_1, \ldots, y_n)$ *if* $H \in \mathcal{F}_\infty$ *has* $n \geq 3$ *vertices and* $x = H\langle y_1, \ldots, y_n\rangle$*.*

Note that if $\mathbf{label}_H(x)$, $\sigma \in \mathrm{Aut}(H)$ and $y_1, \ldots, y_n$ are the children of H, then $\mathbf{children}_H(x, y_1, \ldots, y_n)$ if and only if $\mathbf{children}_H(x, y_{\sigma(1)}, \ldots, y_{\sigma(n)})$.

As in Courcelle [5] and Kuske [14], we also use the following representation, written $\mathbf{mdec}'(G)$, built from $\mathbf{mdec}(G)$ by adding internal nodes in such a way that every $\bullet$-labeled node has exactly two children, the first of which is not $\bullet$-labeled. More precisely, for each $\bullet$-labeled node u of $\mathbf{mdec}(G)$ with children $v_1 < \ldots < v_n$ ($n \geq 3$), we add $\bullet$-labeled nodes $u_2, \ldots, u_{n-1}$ in such a way that the children of u are $v_1 < u_2$, the children of u_i are $v_i < u_{i+1}$ for $2 \leq i \leq n-2$, and the children of u_{n-1} are $v_{n-1} < v_n$. All other nodes (that is, all nodes that are not $\bullet$-labeled) and relations are left unchanged.

The nodes of $\mathbf{mdec}'(G)$ can also be identified with subsets of the vertex set of G, but not necessarily with prime modules of G. More precisely, with the above notation, the new vertex u_i can be identified with the union $\bigcup_{h=i}^{n} v_h$.

As a relational structure, $\mathbf{mdec}'(G)$ has domain the set of its nodes, and it is equipped with the following predicates, inherited from $\mathbf{mdec}(G)$:

> $\mathbf{child}(x,y)$, $\mathbf{label}_a(x)$ *(*$a \in A$*)*, $\mathbf{label}_\oplus(x)$, $\mathbf{label}_\otimes(x)$, $\mathbf{label}_\bullet(x)$, $\mathbf{label}_H(x)$ *and* $\mathbf{children}_H(x, y_1, \ldots, y_n)$ *if* $H \in \mathcal{F}_\infty$ *is a graph with* $n \geq 3$ *vertices.*

Instead of the relation $x < y$ between distinct children of a $\bullet$-labeled node, $\mathbf{mdec}'(G)$ is equipped with the binary predicate

> $\mathbf{first\text{-}child}(x,y)$ if $\mathbf{label}_\bullet(x)$, $\mathbf{child}(x,y)$ and y is the first (minimal, left-most) child of x.

We say that a labeled tree of the form $\mathbf{mdec}'(G)$, for some A-labeled graph G, is an $\mathbf{mdec}'$-*tree*.

3 Recognizability vs. definability: known results

The connection between definability and $\mathcal{F}_\infty$-recognizability was first studied by Courcelle [5]. A slight modification of [5, Theorem 6.11] shows the following

Theorem 3.1 *Let $\mathcal{F}$ be a finite subset of $\mathcal{F}_\infty$ and let L be a language of A-labeled $\mathcal{F}$-graphs. Then the following are equivalent:*

- *L is $\mathcal{F}_\infty$-recognizable;*
- *L is $\mathcal{F}$-recognizable;*
- *the tree language $\mathbf{mdec}'(L)$ is CMS-definable*

Moreover, if L is CMS-definable, then L is $\mathcal{F}$-recognizable.

Remark 3.2 Courcelle's result [5, Theorem 6.11] is actually more precise: it also proves the equivalence between $\mathcal{F}$-recognizability and definability in a certain extension of CMS-logic, called MS_{lin}, which is well-adapted to this situation but lacks the good algorithmic properties of MS and CMS logic. For our purpose, we do not need to get into the definition of MS_{lin}, and it suffices to know that CMS-definability implies MS_{lin}-definability.

Another difference between the above statement and Courcelle's result is that the latter is given for unlabeled graphs and for particular values of $\mathcal{F}$: namely the (finite) subset $\mathcal{F}_n$ of all graphs in $\mathcal{F}_\infty$ with at most n vertices for some n. It is a routine verification that the same proof applies to A-labeled graphs and to any finite subset of $\mathcal{F}_\infty$ – which is necessarily contained in some $\mathcal{F}_n$. $\square$

Courcelle shows the equivalence between MS-definability, CMS-definability and $\mathcal{F}$-recognizability when the tree language $\mathbf{mdec}'(L)$ satisfies certain combinatorial properties [5, Theorem 6.12], and especially when the out-degree of the internal nodes of the elements of $\mathbf{mdec}'(L)$ is uniformly bounded.

This equivalence is also known to hold without restriction on the shape of the trees in $\mathbf{mdec}'(L)$ for certain small values of $\mathcal{F}$.

If $\mathcal{F} = \{\bullet\}$ Since $H_\bullet$ is a poset, the $\mathcal{F}$-algebra consists of posets, and it is easily seen that these posets are of the form $[n]$, equipped with the usual linear order. The A-generated $\mathcal{F}$-algebra is then naturally identified with the free semigroup A^+, i.e., the set of all finite words on alphabet A and the setting of classical language theory. Theorems 3.1 (together with [5, Theorem 6.12], see above) reduces to Büchi's theorem on the equivalence between recognizability and MS- (and hence CMS-) definability.

If $\mathcal{F} = \{\oplus\}$ The A-generated $\mathcal{F}$-algebra consists of the finite A-labeled discrete graphs (graphs without any edges). This algebra is naturally identified with $A^\oplus$, the free commutative semigroup on A. It is known (Courcelle [2]) that, for languages of discrete graphs, recognizability is equivalent to CMS-definability, and not to MS-definability. In fact, MS-definability allows only the description of finite or cofinite discrete graph languages [2].

If $\mathcal{F} = \{\otimes\}$ The A-generated $\mathcal{F}$-algebra consists of the finite A-labeled cliques. As this is the dual situation of discrete graphs (by edge-complementation), the same results hold.

If $\mathcal{F} = \{\oplus, \bullet\}$ The graphs $H_\bullet$ and $H_\oplus$ are posets, and hence every $\mathcal{F}$-graph is a poset. These posets, called the *series-parallel posets*, are exactly those whose graph is N-free [11, 27].

The languages of A-labeled $\mathcal{F}$-graphs, also called *series-parallel languages* or *sp-languages*, were studied by Lodaya and Weil [16, 17] and by Kuske [14]. In particular, Kuske showed that for *sp*-languages, $\mathcal{F}$-recognizability is equivalent to CMS-definability [14, Theorem 6.15].

4 Weakly rigid signatures and CMS-definability

Our main result, Theorem 4.3 below, generalizes the results of the previous section: it asserts the equivalence between $\mathcal{F}$-recognizability and CMS-definability for more general finite subsignatures $\mathcal{F}$ of the modular signature, and in particular for every finite subsignature consisting only of dags.

4.1 Weakly rigid signatures

Let $H = ([n], F)$ be a prime graph ($n \geq 2$). We say that H is *weakly rigid* if the automorphism group $\mathrm{Aut}(H)$ does not act transitively on $[n]$. That is: there are vertices $i \neq j$ of H such that no automorphism of H maps i to j.

Example 4.1 For each $n \geq 2$, the directed cycle of length n, C_n, is not weakly rigid. Indeed, every cyclic permutation of $[n]$ defines an automorphism of C_n. The same holds for D_n, the undirected cycle of length n (for $n = 2$ or $n \geq 5$: D_3 and D_4 are not prime...). Note that $D_2 = H_\otimes$.

The graph $H_\bullet$ is weakly rigid. The graph H from Example 2.1 is weakly rigid since it has a single non-trivial automorphism, namely (1 5)(2 4). In particular, no automorphism of H can map vertex 1 to vertex 2.

This graph H is a particular case of more general situation: every prime dag is weakly rigid, except for $H_{\oplus}$. Indeed, in such a dag there are maximal elements (for the partial order relation obtained by taking the reflexive transitive closure of the edge relation) and not every vertex is maximal. The weak rigidity follows from the simple observation that every automorphism of a dag preserves the maximal elements.

More generally, every prime graph in which the in-degree or the out-degree is not uniform, is weakly rigid. $\square$

We say that a subset $\mathcal{F}$ of $\mathcal{F}_\infty$ is a *weakly rigid signature* if $\mathcal{F}$ is finite, $\mathcal{F}$ contains at most one of the operations $\otimes$ and $\oplus$, and every other operation in $\mathcal{F}$ is associated with a weakly rigid prime graph.

Example 4.2 In view of Remark 2.3 and Example 4.1, every finite subsignature of $\mathcal{F}_\infty$ consisting only of dags is weakly rigid. $\square$

We now state our main theorem.

Theorem 4.3 *Let $\mathcal{F}$ be a weakly rigid signature and let L be a language of A-labeled $\mathcal{F}$-graphs. Then L is CMS-definable if and only if L is $\mathcal{F}$-recognizable.*

Remark 4.4 Theorem 4.3 generalizes Kuske's result on *sp*-languages [14], see Section 3. It constitutes a refinement of Courcelle's theorem [5, Theorem 6.11] (see Remark 3.2), which only asserts the equivalence between $\mathcal{F}$-recognizability and MS_{lin}-definability. However, Courcelle's result does not assume that $\mathcal{F}$ is weakly rigid. $\square$

In view of the importance of poset languages, it is worth stating the following particular case (see Example 4.2) of Theorem 4.3.

Corollary 4.5 *Let $\mathcal{F}$ be a finite subset of $\mathcal{F}_\infty$ such that every $\mathcal{F}$-graph is a poset (resp. a dag). A language of A-labeled $\mathcal{F}$-posets is $\mathcal{F}$-recognizable if and only if it is CMS-definable.*

4.2 Proof of Theorem 4.3

The proof of Theorem 4.3, given below, uses the notion of an MS-definable transduction introduced by Courcelle [3, Section 2]. The definition of these transductions is given in Section 4.3 together with the proof of the following theorem.

Theorem 4.6 *Let $\mathcal{F}$ be a weakly rigid signature. The mapping which assigns to each A-labeled $\mathcal{F}$-graph G the tree $\mathbf{mdec}'(G)$ is MS-definable.*

Note that Courcelle shows that if $\mathcal{F}$ is any finite subset of $\mathcal{F}_\infty$, then the mapping which assigns to each linearly ordered A-labeled $\mathcal{F}$-graph G the tree $\mathbf{mdec}'(G)$ is an MS-transduction [5, Corollary 6.9]. With our extra assumption on $\mathcal{F}$, we are able to dispense with the heavy requirement of considering only linearly ordered graphs.

Proof of Theorem 4.3. Theorem 3.1 proves half of the equivalence; namely, it asserts that every CMS-definable language of A-labeled $\mathcal{F}$-graphs is $\mathcal{F}$-recognizable (without assuming that $\mathcal{F}$ is weakly rigid).

In order to prove the converse, we assume that L is an $\mathcal{F}$-recognizable language of A-labeled $\mathcal{F}$-graphs. By Theorem 3.1, $\mathbf{mdec}'(L)$ is CMS-definable in the language of $\mathbf{mdec}'$-trees.

Now, the inverse image of a CMS-definable subset by an MS-transduction is CMS-definable [3, Corollary 2.7]. Thus Theorem 4.3 is an immediate consequence of Theorem 4.6. □

4.3 Proof of Theorem 4.6

We now fix a finite weakly rigid signature $\mathcal{F}$. For convenience, we assume that $\otimes \notin \mathcal{F}$; the proof would be completely similar if we assumed that $\oplus \notin \mathcal{F}$.

Let us first explain how we use the hypothesis that $\mathcal{F}$ is weakly rigid: for each (concrete) prime graph $H = ([n], F) \in \mathcal{F}$ with $n \geq 3$ vertices, we fix a proper, non-empty subset $\mathbf{dist}(H)$ of $[n]$ of so-called *distinguished vertices*, which is preserved under the action of $\mathrm{Aut}(H)$. Such a set exists by assumption: we can choose an orbit of $[n]$ under the action of $\mathrm{Aut}(H)$, or in the case of a dag the set of maximal vertices, etc. With this choice of $\mathbf{dist}(H)$, we define in an $\mathbf{mdec}'$-tree a new binary predicate $\mathbf{dist\text{-}child}_H(x, y)$, interpreted to mean that x is H-labeled, there exist $y_1, \ldots, y_n$ such that $\mathbf{children}_H(x, y_1, \ldots, y_n)$, and $y = y_i$ for some $i \in \mathbf{dist}(H)$.

Thus, if G is an $\mathcal{F}$-graph, each node of $\mathbf{mdec}'(G)$ that is neither a leaf nor is labeled $\oplus$ has some distinguished children and some non-distinguished ones. By convention, the distinguished vertex of $H_\bullet$ is the origin of the single edge — so that the distinguished child of a $\bullet$-labeled node is its first child.

Remark 4.7 We have seen that the order of children of an H-labeled node is defined only up to the action of $\mathrm{Aut}(H)$: the notion of distinguished children is devised precisely to take into account this flexibility. Weakly rigid operations are precisely those for which some children can be designated unambiguously as *distinguished.* □

Remark 4.8 Note that, the set $\mathbf{dist}(H)$ being fixed, $\mathbf{dist\text{-}child}_H$ is not truly a new predicate to be added in the signature of $\mathbf{mdec}'$-trees, but rather an abbreviation for a first-order formula in the language of $\mathbf{mdec}'$-trees.

Since we are dealing only with finite signatures, we do not bother with a formal mechanism to choose the sets $\mathbf{dist}(H)$. If we had to work with an infinite signature, it would be important to introduce a more formal definition of distinguished children, for instance based on a logical formula (on H) describing these distinguished children. For instance, in the case of dags, one could always consider the maximal elements (or equivalently, the vertices of in-degree zero). □

Now we need to show that if G is an A-labeled $\mathcal{F}$-graph, then $\mathbf{mdec}'(G)$ can be represented in G, its domain can be specified in the monadic second-order logic of graphs, and the predicates of the language of $\mathbf{mdec}'$-trees can be specified in the same language.

More precisely, following the definition in [3, Section 2], we need to verify that there exist integers $k, n \geq 0$ and MS-formulas in the language of graphs $\varphi(\vec{X})$, $\psi_1(x, \vec{X}), \ldots, \psi_k(x, \vec{X})$ and $\theta_{q,\vec{\imath}}(x_1, \ldots, x_\ell, \vec{X})$ for each ℓ-ary predicate q in the language of $\mathbf{mdec}'$-trees and each length ℓ vector $\vec{\imath}$ of integers in $[0, k]$ (where $\vec{X} = (X_1, \ldots, X_n)$ is a vector of second-order variables called *parameters*), such that, for each A-labeled $\mathcal{F}$-graph G,

- for each assignment $\vec{\gamma}$ of values to the vector of variables $\vec{X}$ such that G satisfies $\varphi(\vec{\gamma})$ define the structure $\mathbf{repr}_{\vec{\gamma}}(G)$ by letting:
  - the domain of $\mathbf{repr}_{\vec{\gamma}}(G)$ consists of the pairs (v, i) such that $v \in V$, $0 \leq i \leq k$ and G satisfies $\psi_i(v, \vec{\gamma})$;
  - if $\vec{\imath} = (i_1, \ldots, i_\ell)$ is a vector of integers in $[0, k]$, q is an ℓ-ary predicate and $(v_1, i_1), \ldots, (v_\ell, i_\ell)$ are elements of the domain of $\mathbf{repr}_{\vec{\gamma}}(G)$, then $\mathbf{repr}_{\vec{\gamma}}(G)$ satisfies $q((v_1, i_1), \ldots, (v_\ell, i_\ell))$ if and only if G satisfies $\theta_{q,\vec{\imath}}(v_1, \ldots, v_\ell, \vec{\gamma})$.
- there exists an assignment $\vec{\gamma}$ such that $\mathbf{repr}_{\vec{\gamma}}(G)$ isomorphic to $\mathbf{mdec}'(G)$.

For this, we first encode the inner nodes of an $\mathbf{mdec}'$-tree in its leaves. This idea was first introduced by Potthoff and Thomas [25], and used also in [5, Section 5]. In fact, we cannot use a single encoding as in the works cited, and we construct a collection of four such encodings as in Kuske's [14]. As it turns out, it is more convenient to define the inverse of these encodings: this is done in Section 4.3.1.

This construction allows us to consider a structure isomorphic to $\mathbf{mdec}'(G)$, and defined within G in the form required by the definition of MS-transductions (Section 4.3.2). It then suffices to verify that the domain of $\mathbf{repr}(G)$ and the predicates in this structure are expressible by means of MS-formulas on the graph G, which is done in Section 4.3.3.

We strongly rely on the fact that the nodes of the tree $\mathbf{mdec}'(G)$ are naturally viewed as subsets of V (see Section 2.3), and that V is both the vertex set of G and the set of leaves of $\mathbf{mdec}'(G)$. In particular, in the encodings we construct, each inner node is represented by a leaf of $\mathbf{mdec}'(G)$ and not by a pair of leaves as in Potthoff and Thomas [25], Courcelle [5] or Kuske [14].

4.3.1 Encoding the nodes of an mdec′-tree

Let T be an $\mathbf{mdec}'$-tree. We partition its set N of nodes as follows: we let N_0 be the set of leaves; N_1 be the set of $\oplus$-labeled nodes all of whose children are leaves; N_2 be the set of $\oplus$-labeled nodes not in N_1; and N_3 be the complement of $N_0 \cup N_1 \cup N_2$. That is, N_3 consists of the $\bullet$-labeled and the H-labeled nodes, where $H \in \mathcal{F}$ has arity at least 3; in particular, the nodes in N_3 have distinguished and non-distinguished children.

Next we define mappings ν (resp. μ_0, μ_1, μ_2, μ_3) from N (resp. N_0, N_1, N_2, N_3) to the powerset of N_0 as follows.

> *If $x \in N_0$, we let $\nu(x) = \mu_0(x) = \{x\}$.*
> *If $x \in N_1$, we let $\nu(x) = \mu_1(x)$ be the set of children of x.*
> *If $x \in N_2$, we let $\nu(x) = \bigcup \nu(y)$ where the union runs over the children y of x; and we let $\mu_2(x) = \bigcup \mu_3(y)$ where the union runs over the children of x which are not leaves, and hence which are in N_3.*
> *If $x \in N_3$, we let $\nu(x) = \bigcup \nu(y)$ where the union runs over the non-distinguished children y of x; and we let $\mu_3(x) = \bigcup \nu(y)$ where the union runs over the distinguished children y of x.*

It is easily verified that these mappings are well-defined and that, for each $x \in N_i$ $(i = 0, 1, 2, 3)$, $\nu(x)$ and $\mu_i(x)$ are non-empty sets of leaves.

For each leaf x of T we denote by $\rho(x)$ the unique path from x to the root of T. The following lemma is a simple rewriting of the definition of ν and the μ_i.

Lemma 4.9 *Let $i \in \{0, 1, 2, 3\}$, let $y \in N_i$ be a node of T and let $x \in N_0$ be a leaf. The following are equivalent:*

- *if $i = 0$, then $x \in \nu(y)$ iff $x \in \mu_0(y)$ iff $x = y$;*
- *if $i = 1$, then $x \in \nu(y)$ iff $x \in \mu_1(y)$ iff y is the parent node of x, y is labeled $\oplus$ and all its children are leaves;*
- *if $i = 2$ or $i = 3$, then $x \in \nu(y)$ iff y sits along $\rho(x)$ and every node in N_3 between x and y along $\rho(x)$ is reached from one of its non-distinguished children;*
- *if $i = 2$, then $x \in \mu_2(y)$ iff y sits along $\rho(x)$, y is reached from one of its children in N_3, say z, z is reached from one of its distinguished children, and every node in N_3 along $\rho(x)$ and before z is reached from one of its non-distinguished children;*
- *if $i = 3$, then $x \in \mu_3(y)$ iff y sits along $\rho(x)$, y is reached from one of its distinguished children, and every other node in N_3 along $\rho(x)$ and before y is reached from one of its non-distinguished children;*
- *if $i = 2$, then $x \in \mu_2(y)$ iff $x \in \mu_3(z)$ for some child z of y in N_3.*

It follows from this lemma that for each i, μ_i is the inverse image of a partial onto mapping $\kappa_i \colon N_0 \to N_i$. More precisely, we have:

- $\kappa_0(x) = x$;
- $\kappa_1(x)$ is the parent node of x — if that node is labeled $\oplus$ and all its children are leaves;
- $\kappa_3(x)$ is the first node $y \in N_3$ along $\rho(x)$ (starting from the leaf x) reached from one of its distinguished children — if there is such a node y;

- $\kappa_2(x)$ is the parent node of $\kappa_3(x)$ — if $\kappa_3(x)$ exists and its parent node is labeled $\oplus$.

4.3.2 Representing an mdec′-tree in its leaves

Let T be an **mdec′**-tree as above. Let $\mathbf{repr}_0(T)$ be the following structure, with the same signature as **mdec′**-trees. The domain of $\mathbf{repr}_0(T)$ is the set

$$\{(x,i) \mid x \in N_0,\ 0 \leq i \leq 3,\ \kappa_i(x) \text{ is defined}\}.$$

We let:

- $\mathbf{label}_a((x,i))$ if and only if $i=0$ and $\mathbf{label}_a(x)$ in T (for each $a \in A$);
- $\mathbf{label}_\oplus((x,i))$ if $i=1$ or $i=2$;
- $\mathbf{label}_\bullet((x,i))$ if and only if $i=3$ and $\mathbf{label}_\bullet(\kappa_3(x))$ in T;
- $\mathbf{label}_H((x,i))$ if and only if $i=3$ and $\mathbf{label}_H(\kappa_3(x))$ in T (where H is a prime graph in $\mathcal{F}$ with at least 3 vertices).
- $\mathbf{child}((x,i),(y,j))$ if $\mathbf{child}(\kappa_i(x),\kappa_j(y))$ in T;
- $\mathbf{first\text{-}child}((x,i),(y,j))$ if $\mathbf{first\text{-}child}(\kappa_i(x),\kappa_j(y))$ in T;
- $\mathbf{children}_H((x,i),(y_1,j_1),\ldots,(y_r,j_r))$ if $\mathbf{children}_H(\kappa_i(x),\kappa_{j_1}(y_1),\ldots,\kappa_{j_r}(y_r))$;
- $\mathbf{dist\text{-}child}_H((x,i),(y,j))$ if $\mathbf{dist\text{-}child}_H(\kappa_i(x),\kappa_j(y))$ in T.

Note that the mappings κ_i are usually many-to-one, so that $\mathbf{repr}_0(T)$ is not isomorphic to T (and it is not an **mdec′**-tree).

Let us say that two elements (x,i) and (y,j) of the domain of $\mathbf{repr}_0(T)$ are $\equiv$-equivalent if $i=j$ and $\kappa_i(x)=\kappa_i(y)$. It is easily verified that if $X_0,\ldots,X_3 \subseteq N_0$ are such that $X = \bigcup_{i=0}^{3}(X_i \times \{i\})$ is a set of representatives of the $\equiv$-classes, then the restriction of $\mathbf{repr}_0(T)$ to X is isomorphic to T. This substructure of $\mathbf{repr}_0(T)$ (which depends on the choice of the X_i, but is unique up to isomorphism), is denoted — abusively — by $\mathbf{repr}(T)$.

4.3.3 mdec′(G) is MS-definable

We now consider the case where the tree T arises from the modular decomposition of an A-labeled $\mathcal{F}$-graph $G=(V,E,\lambda)$, $T=\mathbf{mdec}'(G)$.

Recall that the set N_0 of leaves of T is equal to V and that, more generally, the nodes of T are particular subsets of V. In view of their definition, the mappings κ_i can be described as follows.

Lemma 4.10 *Let $x \in V$.*

- $\kappa_0(x)=\{x\}$.
- *If the least disconnected prime module P containing x is discrete, then $\kappa_1(x)=P$; otherwise $\kappa_1(x)$ is not defined.*

- *If there exists a connected node P containing x such that x lies in a distinguished child of P and, for every non-trivial connected node Q containing x and properly contained in P, x lies in a non-distinguished child of Q, then $\kappa_3(x) = P$; otherwise $\kappa_3(x)$ is not defined.*

- *If $\kappa_3(x)$ is defined and the least node P properly containing it is disconnected, then $\kappa_2(x) = P$; otherwise $\kappa_2(x)$ is not defined.*

We use the following collection of MS-definable properties of an A-labeled $\mathcal{F}$-graph $G = (V, E, \lambda)$. Upper-case letters $X, Y, \dots$ represent subsets of V or second-order variables, and lower-case letters $x, y, \dots$ represent elements of V or first-order variables. $H = ([n], F)$ is a prime graph in $\mathcal{F}$ with $n \geq 3$.

singleton(X, x) *if* $X = \{x\}$.

label$_a(X)$, *where* $a \in A$, *if* X *is an* A*-labeled node of* $\mathbf{mdec}'(G)$, *that is,* $X = \{x\}$ *and* $\lambda_a(x)$ *for some* x.

partition$(X, X_1, \dots, X_n)$ *if* $(X_1, \dots, X_n)$ *is a partition of* X, *that is,* X *is the disjoint union of the* X_i *and each* X_i *is non-empty.* (To be completely correct, this predicate should be replaced by $(n+1)$-ary predicates **partition**$_n$, for $n = 2$ and for each n such that an n-vertex graph lies in $\mathcal{F}$; furthermore, everyone of these predicates can be expressed in terms of **partition**$_2$.)

module(X) *if* X *is a module of* G. *This is equivalent to*

$$\forall y \notin X \quad \exists x \in X\ E(x, y) \Rightarrow \forall x \in X\ E(x, y) \wedge \exists x \in X\ E(y, x) \Rightarrow \forall x \in X\ E(y, x).$$

pmodule(X) *if* X *is a prime module of* G, *that is,*

$$\mathbf{module}(X) \wedge (\forall Y\ \mathbf{module}(Y) \Longrightarrow (Y \subseteq X) \vee (X \subseteq Y) \vee (X \cap Y = \emptyset)).$$

disconnected(X) *if* X *is disconnected, that is,*

$$\exists Y\ \exists Z\ \mathbf{partition}(X, Y, Z) \wedge (\forall y \in Y\ \forall z \in Z\ \neg E(y, z) \wedge \neg E(z, y)).$$

connected(X) *if* X *is connected, that is* $\neg$**disconnected**(X).

label$_\oplus(X)$ *if* X *is an* $\oplus$*-labeled node of* $\mathbf{mdec}'(G)$, *that is, a disconnected prime module.*

children$_H(X, X_1, \dots, X_n)$ *if* $X, X_1, \dots, X_n$ *are prime modules and* $X = H\langle X_1, \dots, X_n \rangle$. *The latter assertion is equivalent to*

$$\begin{array}{ll} \mathbf{partition} & (X, X_1, \dots, X_n) \\ \wedge & \bigwedge_{(i,j) \in F} \forall x_i \in X_i\ \forall x_j \in X_j\ E(x_i, x_j) \\ \wedge & \bigwedge_{(i,j) \notin F} \forall x_i \in X_i\ \forall x_j \in X_j\ \neg E(x_i, x_j) \end{array}$$

label$_H(X)$ *if* X *is an* H*-labeled node of* $\mathbf{mdec}'(G)$, *that is,*

$$\exists X_1\ \dots\ \exists X_n\ \mathbf{children}_H(X, X_1, \dots, X_n).$$

suffix(X, Z) *if* Z *is a non-trivial (sequential) suffix of* X, *that is,*

$$\exists Y\ \mathbf{partition}(X, Y, Z) \wedge (\forall y \in Y\ \forall z \in Z\ E(y, z) \wedge \neg E(z, y)).$$

sequential(X) *if X is a sequential product, that is, it has a non-trivial suffix.*

initial(X, Y) *if Y is the first (least) prefix of X, that is, $X \setminus Y$ is a suffix of X and Y itself is not sequential.*

label$_\bullet(X)$ *if X is a $\bullet$-labeled node of* $\mathbf{mdec}'(G)$*, that is, X is sequential and either it is prime module, or it is a suffix of a sequential prime module.*

node(X) *if X is a node of* $\mathbf{mdec}'(X)$*, that is, X is either a singleton, or a $\bullet$-labeled node, or an $\oplus$-labeled node, or an H-labeled node for some prime graph $H \in \mathcal{F}$ with at least three vertices.*

child$^*(X, Y)$ *if X and Y are nodes and X is an ancestor of Y in* $\mathbf{mdec}'(X)$*, that is, Y is properly contained in X.*

child(X, Y) *if X and Y are nodes and X is a (the) minimal ancestor of Y.*

dist-child(X, Y) *if Y is a distinguished child of X, that is* **child**(X, Y) *and, either X is $\bullet$-labeled and Y is the first prefix of X, or X is H-labeled (for some $H \in \mathcal{F}$ with at least three vertices) and*

$$\exists X_1 \ \ldots \ \exists X_n \ \mathbf{children}_H(X, X_1, \ldots, X_n) \wedge \bigvee_{i \in \mathbf{dist}(H)} Y = X_i.$$

Together with Lemma 4.10, this list of definable properties shows that the formulas $X = \kappa_0(x)$, $X = \kappa_1(x)$, $X = \kappa_2(x)$ and $X = \kappa_3(x)$ can be expressed in monadic second-order formulas.

We now verify formally that the mapping $G \mapsto \mathbf{mdec}'(G)$ is MS-definable. As established in Section 4.3.2, it suffices to study the mapping $G \mapsto \mathbf{repr}(T)$ where $T = \mathbf{mdec}'(G)$. With reference to the definition given at the beginning of Section 4.3, we let $k = 3$ and $n = 4$, that is, the definition makes use of 4 parameter second-order variables $X_0, \ldots, X_3$, which will stand for sets of representatives of the $\equiv$-classes among the domain elements of the form $(x, 0), \ldots, (x, 3)$.

Since $\mathbf{repr}(T)$ is defined for every G, the role of formula $\varphi(X_0, \ldots, X_3)$ is solely to make sure that the assignment of values to the parameter variables is correct. It is chosen to express, for $i = 0, 1, 2, 3$, that X_i is contained in the domain of κ_i, it does not contain distinct elements with the same κ_i-image, and for each x, if $\kappa_i(x)$ is defined then there exists $y \in X_i$ such that $\kappa_i(x) = \kappa_i(y)$.

For $i \in [0, 3]$, the formula $\psi_i(x, \vec{X})$ is $\psi_i = (x \in X_i)$.

Finally, each relation q of arity r in the description of $\mathbf{repr}(T)$ is as in $\mathbf{repr}_0(T)$, and it can be MS-defined using the list of properties given above.

This concludes the proof of Theorem 4.6.

Conclusions

We have proved the equivalence between $\mathcal{F}$-recognizability and CMS-definability for a large class of finite subsignatures $\mathcal{F}$ of the modular signature $\mathcal{F}_\infty$. We have not however proved that this equivalence does not hold for the other subsignatures! In fact, Courcelle conjectured that CMS-definability is strictly weaker than MS_{lin}-definability for general graphs [5, Conjecture 7.3]. One closely related, yet stronger question is to find out whether there exists a finite subset $\mathcal{F} \subseteq \mathcal{F}_\infty$ such that, for sets of $\mathcal{F}$-graphs, CMS-definability is strictly weaker than $\mathcal{F}$-recognizability (or than $\mathcal{F}_\infty$-recognizability, see Theorem 3.1). Theorem 4.3 does not solve this problem, it only designates a large class of finite signatures for which the two notions are equivalent. As pointed out by Courcelle, an archetypal setting to discuss this conjecture is given by cographs, that is, the $\mathcal{F}$-graphs for $\mathcal{F} = \{\otimes, \oplus\}$.

One can also investigate which natural $\mathcal{F}$-recognizable classes of $\mathcal{F}$-graphs are characterized by algebraic properties of the finite $\mathcal{F}$-algebras recognizing them. This type of investigation is highly developed in the field of word languages (see [24]), but also of trace languages [12, 10, 9], infinite word languages [23]. Lodaya and Weil showed, in this fashion, that the recognizable languages of series-parallel posets of bounded width are characterized algebraically (and in an effective fashion) [16]. Kuske studied the first-order definable languages of series-parallel posets, and gave an algebraic characterization for them in the bounded width case [14]. The general (arbitrary-width) case remains open.

Finally, one could ask for a model of automata to handle $\mathcal{F}$-recognizable sets of $\mathcal{F}$-graphs. Let us mention that [17] proposes a model of automata which can be used to process $\mathcal{F}$-graphs if $\mathcal{F}$ contains neither $\oplus$ nor $\otimes$. To be precise, the input for these automata is an **mdec**-tree or an **mdec**$'$-tree, but this distinction is not algorithmically crucial if we remember that such a tree can be computed in linear time from the graph itself (see Remark 2.2). The accepting power of these automata matches exactly that of $\mathcal{F}$-recognizability. On the other hand, if $\oplus \in \mathcal{F}$, then the same automaton model can be used but it is strictly more powerful than $\mathcal{F}$-recognizability. Eliminating in this way the use of an associative commutative operation reduces the interest of the construction, and the question remains open to propose a different automaton model for $\mathcal{F}$-graphs in general — or for series-parallel posets in particular, that is for the situation where $\mathcal{F} = \{\bullet, \oplus\}$, studied especially in [16, 17].

References

[1] S. Burris, H.P. Sankappanavar. *A course in Universal Algebra*, Springer 1981.

[2] B. Courcelle. The monadic second-order logic of graphs I: recognizable sets of finite graphs, *Information and Computation* **85** (1990) 12-75.

[3] B. Courcelle. The monadic second-order logic of graphs V: on closing the gap between definability and recognizability, *Theoretical Computer Science* **80** (1991) 153-202.

[4] B. Courcelle. Recognizable sets of graphs: equivalent definitions and closure properties, *Mathematical Structures in Computer Science* **4** (1994) 1-32.

[5] B. Courcelle. The monadic second-order logic of graphs X: Linear orders, *Theoretical Computer Science* **160** (1996) 87-143.

[6] B. Courcelle. The expression of graph properties and graph transformations in monadic second-order logic, Chapter 5 in *Handbook of graph grammars and com-*

puting by graph transformations, vol. 1 (G. Rozenberg ed.), World Scientific 1997, 313-400.

[7] A. Cournier, M. Habib. A new linear algorithm for modular decomposition, in *CAAP 1994* (S. Tison ed.), Lecture Notes in Computer Science **787**, Springer 1994, 68-84.

[8] H.-D. Ebbinghaus, J. Flum, W. Thomas. *Mathematical Logic*, Springer 1994.

[9] W. Ebinger. Logical definability of trace languages, Chapter 10.10 in *The book of traces* (V. Diekert, G. Rozenberg eds.), World Scientific 1995, 382-390.

[10] W. Ebinger, A. Muscholl. On logical definability of omega-trace languages, *Theoretical Computer Science* **154** (1996) 67-84.

[11] J. Grabowski. On partial languages, *Fundamenta Informaticæ* **4** (1981) 427-498.

[12] G. Guaiana, A. Restivo, S. Salemi. Star-free trace languages, *Theoretical Computer Science* **97** (1992)301-311.

[13] D. Kaller. Definability equals recognizability for partial 3-trees, in *WG '96* (F. d'Amore, P.G. Franciosa, A. Marchetti-Spaccamela eds.), Lecture Notes in Computer Science **1197**, Springer 1996, 239-253.

[14] D. Kuske. Towards a language theory for infinite N-free pomsets, *Theoretical Computer Science*, to appear.

[15] D. Lapoire. Recognizability equals Monadic Second-Order definability, for sets of graphs of bounded tree-width, in *STACS'98*, Lecture Notes in Computer Science **1373**, Springer 1998, 618-628.

[16] K. Lodaya, P. Weil. Series-parallel languages and the bounded-width property, *Theoretical Computer Science* **237** (2000) 347-380.

[17] K. Lodaya, P. Weil. Rationality in algebras with a series operation, *Information and Computation* **171** (2001) 269-293.

[18] R. McConnell, J. Spinrad. Linear-time modular decomposition and efficient transitive orientation of comparability graphs, in *Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM 1994, 536-545.

[19] R. McConnell, J. Spinrad. Modular decomposition and transitive orientation, *Discrete Mathematics* **201** (1999) 189-241.

[20] J. Mezei, J. Wright. Algebraic automata and context-free sets, *Information and Control* **11** (1967) 3-29.

[21] R.H. Möhring, F.J. Radermacher. Substitution decomposition for discrete structures and connections with combinatorial optimization, *Annals of Discrete Mathematics* **19** (1984) 257-356.

[22] D. Perrin, J.-E. Pin. Semigroups and automata on infinite words, in *NATO Advanced Study Institute Semigroups, Formal Languages and Groups* (J. Fountain ed.), Kluwer 1995, 49-72.

[23] D. Perrin, J.-E. Pin. *Infinite words*, Academic Press, to appear.

[24] J.-E. Pin. Logic, Semigroups and Automata on Words, *Annals of Mathematics and Artificial Intelligence* **16** (1996) 343-384.

[25] A. Potthoff, W. Thomas. Regular tree languages without unary symbols are star-free, in *FCT 1993* (Z. Ésik ed.), Lecture Notes in Computer Science **710**, Springer 1993, 396-405.

[26] W. Thomas. Languages, Automata, and Logic, in *Handbook of Formal Language Theory* (G. Rozenberg, A. Salomaa, eds.), vol. III, Springer 1997, 389-455.

[27] J. Valdes, R.E. Tarjan, E.L. Lawler. The recognition of series parallel digraphs. *SIAM Journal on Computing* **11** (1982) 298-313.