INFORMATION AND COMPUTATION 85, 12–75 (1990)

The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs*

BRUNO COURCELLE

Bordeaux I University, Laboratoire d'Informatique,†
351, Cours de la Libération, 33405 Talence, France

The notion of a *recognizable set of finite graphs* is introduced. Every set of finite graphs, that is definable in monadic second-order logic is recognizable, but not vice versa. The monadic second-order theory of a context-free set of graphs is decidable.

INTRODUCTION

This paper begins an investigation of the monadic second-order logic of graphs and of sets of graphs, using techniques from universal algebra, and the theory of formal languages. (By a *graph*, we mean a finite directed hyperedge-labelled hypergraph, equipped with a sequence of distinguished vertices.) A survey of this research can be found in Courcelle [11].

An algebraic structure on the set of graphs (in the above sense) has been proposed by Bauderon and Courcelle [2, 7]. The notion of a recognizable set of finite graphs follows, as an instance of the general notion of recognizability introduced by Mezei and Wright in [25].

A graph can also be considered as a logical structure of a certain type. Hence, properties of graphs can be written in first-order logic or in second-order logic. It turns out that *monadic second-order logic*, where quantifications over sets of vertices and sets of edges are used, is a reasonably powerful logical language (in which many usual graph properties can be written), for which one can obtain decidability results. These decidability results do not hold for second-order logic, where quantifications over binary relations can also be used.

Our main theorem states that every *definable* set of finite graphs (i.e., every set that is the set of finite graphs satisfying a graph property expressible in monadic second-order logic) is recognizable.

* This work has been supported by the "Programme de Recherches Coordonnées: Mathématiques et Informatique."

† Unité de Recherche Associée au CNRS n° 726. Electronic mail: courcell@geocub.greco-prog.fr.

0890-5401/90 $3.00

It follows, in particular, that the monadic second-order theory of a context-free set of graphs is decidable. (The notion of a *context-free* set of graphs is introduced in Bauderon and Courcelle [2, 8], by means of *context-free graph-grammars*, that are essentially the *hyperedge-replacement graph-grammars* of Habel and Kreowski [21]).

It is known that a set of words, or of finite ranked trees, is definable iff it is recognizable with respect to the appropriate algebraic structure. (These results have been established respectively by Büchi [4] and Doner [16]. We also refer the reader to Thomas [29]).

In the case of graphs, some recognizable sets are not definable. But we extend the result of Doner, by proving that a set of unordered unbounded trees is recognizable iff it is definable. In this extension, the notion of definability is taken w.r.t. a strict extension of monadic second-order logic, that we call the *counting* monadic second-order logic. In this new language, special atomic formulas are introduced to test whether the cardinality of a set is equal to p modulo q. Our main theorem is actually proved for this extended logic.

We now sketch the organization of the paper, and we present its main definitions and results. Section 1 is devoted to algebraic preliminaries. The notion of a *recognizable set* in a many-sorted algebra is introduced. It is an obvious extension of the notion defined in Mezei and Wright [25] for one-sorted algebras. The notion of an *equational set* extends similarily the notion defined in [25]. The intersection of an equational and a recognizable set is equational. This result extends the classical one saying that the intersection of a context-free language with a regular one is context-free.

Section 2 defines graphs and the operations on graphs. They form the algebraic structure introduced in Bauderon and Courcelle [2, 7]. The length of the sequence of distinguished vertices of a graph is called its *type*. By means of three infinite sets of operations (defined in terms of three operation schemes), one obtains a many-sorted algebra of graphs. The set of sorts is $\mathbb{N}$, and the domain of sort n is the set of graphs of type n.

With respect to this algebraic structure, the equational sets of graphs coincide with the context-free ones (this is proved in Bauderon and Courcelle [2]). The family of recognizable sets of graphs is uncountable and is incomparable with the family of equational sets. This fact shows a major difference from the case of words.

In Section 3, graphs are considered as logical structures. The *counting monadic second-order logic* and the associated definable sets of graphs are introduced.

The main result of this paper is proved in Section 4. It says that every definable set of graphs is recognizable. It follows that, for every graph property expressible in counting monadic second-order logic, the set of graphs satisfying this property, and belonging to a given context-free set of graphs forms a context-free set. One can decide whether such a property holds for all graphs of a given context-free set.

Section 5 deals with *unordered unbounded finite trees*. These trees should be contrasted with the finite ordered ranked trees, classically introduced as graph representations of terms. We prove that a set of finite unordered unbounded trees is recognizable iff it is definable (in counting monadic second-order logic).

In Section 6, we prove that the counting monadic second-order logic is strictly more powerful than the "ordinary" one, in arbitrary logical structures. The two languages are equally powerful for classes of finite logical structures where linear orders are definable in monadic second-order logic. Since such orders are definable in the structures representing words and ranked trees, the "counting feature" is unnecessary in the proofs of the afore-mentioned results by Büchi [4] and Doner [16]. On the other hand *it is necessary* in the analogous result for unbounded unordered trees, that we give in Section 5.

These algebraic and logical investigations are extended in Courcelle [13, 15] to countable graphs. Applications are given in Courcelle [12, 14] concerning finite and countable graphs. Applications to the analysis of recursive definitions are given in Courcelle [9]. The monadic second-order theory of the sets of graphs defined by *context-free node labeled controlled graph grammars* (a restriction of a class originally defined by Janssens and Rozenberg) is proved to be decidable by a similar technique by Courcelle [6].

1. Algebraic Preliminaries

We first review a few general mathematical notations.

We denote by $\mathbb{N}$ the set of non-negative integers, and by $\mathbb{N}_+$, the set of positive ones. We denote by $[n]$ the interval $\{1, 2, 3, ..., n\}$ for $n \geqslant 0$ (with $[0] = \emptyset$). We denote by $[i, j]$ the set $\{k \in \mathbb{N}/i \leqslant k \leqslant j\}$. We write $p = n$ **mod** q if $p = n + kq$, where $0 \leqslant n < q$, $k \in \mathbb{N}$.

The domain of a partial mapping $f: A \to B$ is denoted by $\mathbf{Dom}(f)$. The restriction of f to a subset A' of A is denoted by $f \restriction A'$. The partial mapping with an empty domain is denoted by $\emptyset$, as the empty set. If two partial mappings $f: A \to B$ and $f': A' \to B$ coincide on $\mathbf{Dom}(f) \cap \mathbf{Dom}(f')$, we denote by $f \cup f'$ their common extension into a partial mapping: $A \cup A' \to B$ with domain $\mathbf{Dom}(f) \cup \mathbf{Dom}(f')$.

The cardinality of a set A is denoted by $\mathbf{Card}(A)$. The powerset of A is denoted by $\mathscr{P}(A)$. An equivalence relation is *finite* if it has finitely many classes.

The set of nonempty sequences of elements of a set A is denoted by A^+, and sequences are denoted by $(a_1, ..., a_n)$ with commas and parentheses. The empty sequence is denoted by ( ), and A^* is $A^+ \cup \{(\ )\}$. When A is an alphabet, i.e., when its elements are letters, then a sequence $(a_1, ..., a_n)$ in A^+ can be written unambiguously $a_1 a_2 \cdots a_n$. The empty sequence is denoted by ε, a special symbol that is reserved for this purpose. The elements of A^* are called words. The length of a sequence μ is denoted by $|\mu|$.

A set A is *effectively given* if it is given together with a recursive subset $\|A\|$ of $\mathbb{N}$, and a bijection $c_A: A \to \|A\|$. From this assumptions, one cannot decide whether A is finite, but if A is given as a finite list of elements, then it is effectively given.

When we say: "let A be a finite set," we mean that A is given as a list of elements.

A mapping $f: A_1 \times \cdots \times A_n \to B$ is *computable* if $A_1, ..., A_n, B$ are effectively given and $f(a_1, ..., a_n) = c_B^{-1}(\|f\|(c_{A_1}(a_1), ..., c_{A_n}(a_n)))$ for all $a_1 \in A_1, ..., a_n \in A_n$, where $\|f\|$ is a given total recursive mapping: $\|A_1\| \times \cdots \times \|A_n\| \to \|B\|$.

We shall use := for "equal by definition," i.e., for introducing a new notation, or a definition. The notation :⇔ will be used similarly for defining logical conditions.

(1.1) DEFINITION. **Many-sorted magmas.** As in many other works, we use the term *magma* for what is usually called an *algebra.* The words "algebra" and "algebraic" are used in many different situations with different meanings. We prefer to avoid them completely and use fresh words. For a set we shall use the term "equational" introduced by Mezei and Wright [25] rather than the term "algebraic" introduced by Eilenberg and Wright [18].

Many-sorted notions are studied in detail in Ehrig and Mahr [17] and Wirsing [31]. We mainly review the notations. We shall use *infinite* sets of sorts and *infinite* signatures. For this reason, we need to pay a certain attention to effectivity questions.

Let $\mathscr{S}$ be a set called the set of *sorts*. An $\mathscr{S}$*-signature* is a set F given with two mappings $\alpha: F \to \mathscr{S}^*$ (the *arity* mapping), and $\sigma: F \to \mathscr{S}$ (the *sort* mapping). The length of $\alpha(f)$ is called the *rank* of f, and is denoted by $\rho(f)$. The profile of f in F is the pair $(\alpha(f), \sigma(f))$ written $s_1 \times s_2 \times \cdots \times s_n \to \sigma(f)$, where $\alpha(f) = (s_1, ..., s_n)$.

An *F-magma* (i.e., an *F-algebra* in the sense of [17] and [31]) is an object $\mathbf{M} = \langle (\mathbf{M}_s)_{s \in \mathscr{S}}, (f_{\mathbf{M}})_{f \in F} \rangle$, where $\mathbf{M}_s$ is a nonempty set, for each s in $\mathscr{S}$, called the *domain of sort* s of $\mathbf{M}$, and $f_{\mathbf{M}}$ is a total mapping:

$\mathbf{M}_{\alpha(f)} \to \mathbf{M}_{\sigma(f)}$ for each $f \in F$. (For a sequence $\mu = (s_1, ..., s_n)$ in $\mathscr{S}^+$, we let $\mathbf{M}_\mu := \mathbf{M}_{s_1} \times \mathbf{M}_{s_2} \times \cdots \times \mathbf{M}_{s_n}$.)

It is *effectively given* if $\mathscr{S}$, F, and the sets $\mathbf{M}_s$ are effectively given, and if the mappings α, σ, and the mapping associating $f_\mathbf{M}(d_1, ..., d_k)$ with every $(f, d_1, ..., d_k)$ in $F \times (\bigcup\{\mathbf{M}_s / s \in \mathscr{S}\})^*$ such that $k = \rho(f)$ and $d_i \in \mathbf{M}_{s_i}$ for all $i = 1, ..., k$ are computable.

If $\mathbf{M}$ and $\mathbf{M}'$ are two F-magmas, a homomorphism $h: \mathbf{M} \to \mathbf{M}'$ is a family of mappings $(h_s)_{s \in \mathscr{S}}$ such that h_s maps $\mathbf{M}_s$ into $\mathbf{M}'_s$, and the operations of F are preserved in a well-known way.

We denote by $\mathbf{M}(F)$ the initial F-magma, and by $\mathbf{M}(F)_s$ its domain of sort s. This set can be identified with the set of well-formed ground terms over F. It is effectively given if $\mathscr{S}$ and F are effectively given, and if α and σ are computable.

We denote by $h_\mathbf{M}$ the unique homomorphism: $\mathbf{M}(F) \to \mathbf{M}$, where $\mathbf{M}$ is an F-magma. If $t \in \mathbf{M}(F)_s$, then the image of t under $h_\mathbf{M}$ is an element of $\mathbf{M}_s$, also denoted by $t_\mathbf{M}$. One considers t as an *expression denoting* $t_\mathbf{M}$, and $t_\mathbf{M}$ as *the value of t* in $\mathbf{M}$. We say that F *generates* $\mathbf{M}$ if every element of $\mathbf{M}$ is the value $t_\mathbf{M}$ of some term t in $\mathbf{M}(F)$.

If $\mathbf{M}$ is effectively given, then $h_\mathbf{M}$ is computable. If, furthermore, $\mathbf{M}$ is generated by F, then a computable mapping $k_\mathbf{M}: \mathbf{M} \to \mathbf{M}(F)$ defining, for every element of $\mathbf{M}$ a term denoting it, can be defined by the following algorithm: given d in $\mathbf{M}$, one enumerates $\mathbf{M}(F)$, and for every term t, one computes $t_\mathbf{M}$. The term $k_\mathbf{M}(d)$ is the first one such that $t_\mathbf{M} = d$.

An $\mathscr{S}$*-sorted set of variables* is a pair $(\mathscr{X}, \sigma)$ consisting of a set $\mathscr{X}$, and a sort mapping $\sigma: \mathscr{X} \to \mathscr{S}$. It is more simply denoted by $\mathscr{X}$. We denote by $\mathbf{M}(F, \mathscr{X})_s$ the set of well-formed terms of sort s, written with $F \cup \mathscr{X}$. Hence, $\mathbf{M}(F, \mathscr{X})_s = \mathbf{M}(F \cup \mathscr{X})_s$.

When $\mathscr{X}$ is the set $\{x_1, x_2, ..., x_n, ...\}$, we denote by $\mathscr{X}_n$ the subset $\{x_1, x_2, ..., x_n\}$ of $\mathscr{X}$, *ordered in this way*. If $t \in \mathbf{M}(F, \mathscr{X}_k)_s$, we denote by $t_{M,k}$ the mapping: $\mathbf{M}_\mu \to \mathbf{M}_s$ (where $\mu = (\sigma(x_1), ..., \sigma(x_k))$), associated with t in a classical way. We call it a *derived operation of* $\mathbf{M}$. If k is known from the context, we write $t_\mathbf{M}$ instead of $t_{\mathbf{M},k}$.

If $t \in \mathbf{M}(F, \mathscr{X}_k)$, $t_1, ..., t_k \in \mathbf{M}(F, \mathscr{X})$ with $\sigma(t_i) = \sigma(x_i)$ for $i = 1, ..., k$, then $t[t_1/x_1, ..., t_k/x_k]$ denotes the result of the simultaneous substitution of t_1 for $x_1, ..., t_k$ for x_k in t. We also use the notation $t[t_1, ..., t_k]$ if the sequence $x_1, ..., x_k$ is clear from the context. If $t_1, ..., t_k \in \mathbf{M}(F, \mathscr{X}_n)$, then, for every F-magma $\mathbf{M}$, we have

$$t[t_1, ..., t_k]_{\mathbf{M},n} = t_{\mathbf{M},k} \circ (t_{1\mathbf{M},n}, ..., t_{k\mathbf{M},n}).$$

For $s, r \in \mathscr{S}$, we let $\mathbf{Ctxt}(F)_{s,r}$ denote the set of elements of $\mathbf{M}(F, \{u\})_r$ having one and only one occurrence of u, where u is a variable of sort s. If $c \in \mathbf{Ctxt}(F)_{s,r}$ and $t \in \mathbf{M}(F, \{x_1, ..., x_k\})_s$ then $c[t] := c[t/u]$ is an element

t' of $\mathbf{M}(F, \{x_1, \ldots, x_k\})_r$. We say that c is *a context of* t *in* t'. If $\mathbf{M}$ is an F-magma and $c \in \mathbf{Ctxt}(F)_{s,r}$, then $c_{\mathbf{M}}$ is a mapping $\mathbf{M}_s \to \mathbf{M}_r$ and $c[t]_{\mathbf{M}} = c_{\mathbf{M}} \circ t_{\mathbf{M}}$. The specific variable u is irrelevant, and the notation $\mathbf{Ctxt}(F)_{s,r}$ avoids mentioning it explicitly.

A term is *linear* if each variable occurs at most once.

When writing terms, we shall use the prefix notation with parentheses and commas, but we shall frequently omit the parentheses surrounding the unique argument of a monadic function symbol. Hence we shall use the simplified notation $fgfh(x, fx)$ for $f(g(f(h(x, f(x)))))$.

(1.2) Definition. **Polynomial systems and equational sets.** Polynomial systems have been introduced (under the name of "systems") in Mezei and Wright [25]. Let $\mathcal{S}$, F be as above. We augment F into F_+ by adding, for every sort s in $\mathcal{S}$, a new symbol $+_s$ of profile: $s \times s \to s$, and a new constant Ω_s of sort s.

With $\mathbf{M}$ as above we associate its power-set magma:

$$\mathcal{P}(\mathbf{M}) := \langle (\mathcal{P}(\mathbf{M}_s))_{s \in \mathcal{S}}, (f_{\mathcal{P}(\mathbf{M})})_{f \in F_+} \rangle,$$

where for $A_1, \ldots, A_k \subseteq \mathbf{M}_{s_1}, \ldots, \mathbf{M}_{s_k}$:

$$A_1 +_{s\mathcal{P}(\mathbf{M})} A_2 := A_1 \cup A_2 \qquad \text{(where } s = s_1 = s_2\text{)},$$

$$f_{\mathcal{P}(\mathbf{M})}(A_1, \ldots, A_k) := \{f_{\mathbf{M}}(a_1, \ldots, a_k)/a_1 \in A_1, \ldots, a_k \in A_k\}$$

(where $\alpha(f) = (s_1, \ldots, s_k)$), and

$$\Omega_{s\mathcal{P}(\mathbf{M})} = \varnothing$$

Hence $\mathcal{P}(\mathbf{M})$ is an F_+-magma.

A *polynomial system* over F is a sequence of equations $S = \langle u_1 = p_1, \ldots, u_n = p_n \rangle$, where $U = \{u_1, \ldots, u_n\}$ is the $\mathcal{S}$-sorted set of *unknowns*. Each p_i is a *polynomial*, i.e., a term of the form Ω_s or

$$t_1 +_s t_2 +_s \cdots +_s t_m$$

where the t_j's (called monomials) belong to $\mathbf{M}(F \cup U)_s$, with $s = \sigma(u_i)$. The subscript s is usually omitted in $+_s$ and in Ω_s.

A mapping $S_{\mathcal{P}(\mathbf{M})}$: $\mathcal{P}(\mathbf{M}_{\sigma(u_1)}) \times \cdots \times \mathcal{P}(\mathbf{M}_{\sigma(u_n)})$ into itself is associated with S and $\mathbf{M}$ as follows: for $A_1 \subseteq \mathbf{M}_{\sigma(u_1)}, \ldots, A_n \subseteq \mathbf{M}_{\sigma(u_n)}$,

$$S_{\mathcal{P}(\mathbf{M})}(A_1, \ldots, A_n) = (A'_1, \ldots, A'_n),$$

where $A'_i = p_{i\mathcal{P}(\mathbf{M})}(A_1, \ldots, A_n)$ for $i = 1, \ldots, n$.

A *solution* of S in $\mathcal{P}(\mathbf{M})$ is an n-tuple $(A_1, \ldots, A_n)$ such that

$(A_1, ..., A_n) = S_{\mathscr{P}(\mathbf{M})}(A_1, ..., A_n)$. Such a system has a least solution in $\mathscr{P}(\mathbf{M})$ w.r.t. set inclusion, denoted by $(L((S, \mathbf{M}), u_1), ..., L((S, \mathbf{M}), u_n))$. The components of the least solution in $\mathscr{P}(\mathbf{M})$ of a polymonial system are the **M**-*equational* sets. We denote by **Equat(M)** the family of **M** equational sets.

Every set of the form $L((S, \mathbf{M}), U') := \bigcup\{L((S, \mathbf{M}), u)/u \in U'\}$ where U' is a set of unknowns all of the same sort, is **M**-equational. We write $L(S, u_i)$ and $L(S, U')$ instead of $L((S, \mathbf{M}(F)), u_i)$ and $L((S, \mathbf{M}(F)), U')$, respectively. Furthermore, $L((S, \mathbf{M}), u_i) = \emptyset$ iff $L(S, u_i) = \emptyset$, and this property is decidable. We refer the reader to Courcelle [5] for a thorough study of polynomial systems.

(1.3) DEFINITION. **Recognizable sets.** The notion of a recognizable set is due to Mezei and Wright [25]. Let F and $\mathscr{S}$ be as above. An F-magma **A** is *locally finite* if every domain $\mathbf{A}_s$, $s \in \mathscr{S}$, is finite.

Let **M** be an F-magma and $s \in \mathscr{S}$. A subset B of $\mathbf{M}_s$ is **M**-*recognizable* if there exists a locally finite F-magma **A**, a homomorphism $h: \mathbf{M} \to \mathbf{A}$, and a (finite) subset C of $\mathbf{A}_s$ such that $B = h^{-1}(C)$. The pair $(h, \mathbf{A})$ is called a *semi-automaton*, and the triple $(h, \mathbf{A}, C)$ is called an *automaton*. Intuitively, C is the set of "final states" of a deterministic automaton.

A set $B \subseteq \mathbf{M}_s$ is *effectively* **M**-*recognizable* if **M** is effectively given, and if it is defined by an *effectively given automaton*, i.e., an automaton $(h, \mathbf{A}, C)$, where **A** and C are effectively given, and h is computable. (These conditions imply that one can decide whether an element of $\mathbf{M}_s$ belongs to B).

We denote by $\mathbf{Rec}(\mathbf{M})_s$ the family of **M**-recognizable subsets of $\mathbf{M}_s$.

The recognizable subsets of $\mathbf{M}(F)$, where F is a finite signature, can be characterized by tree-automata of various types (top–down or bottom-up, deterministic or not; see Gecseg and Steinby [20]). The classical identification of terms with finite ordered ranked trees explains the qualification of "tree"-automaton. But there are several other notions of trees. More precisely there are several *theories of trees* as shown in Courcelle [10]. Appropriate notions of tree automata are defined in [10].

Recognizable sets can also be characterized in terms of congruences. A *congruence* on **M** is a family $\sim = (\sim_s)_{s \in \mathscr{S}}$, where $\sim_s$ is an equivalence relation on $\mathbf{M}_s$ for every $s \in \mathscr{S}$, and such that, for every $f \in F$ of profile $s_1 \times s_2 \times \cdots \times s_n \to t$, if $d_1 \sim_{s_1} d'_1, ..., d_n \sim_{s_n} d'_n$, then $f_{\mathbf{M}}(d_1, ..., d_n) \sim_t f_{\mathbf{M}}(d'_1, ..., d'_n)$.

A classical construction associates with **M** and $\sim$ as above, a *quotient-magma* $\mathbf{M}/\sim$, and a surjective homomorphism $h: \mathbf{M} \to \mathbf{M}/\sim$.

A congruence $\sim$ on **M** is *locally finite*, if each equivalence relation $\sim_s$ is finite.

A subset L of $\mathbf{M}_s$ is *saturated w.r.t.* $\sim$ (or $\sim$*-saturated*) if, for every d, $d' \in \mathbf{M}_s$, if d belongs to L and $d \sim_s d'$, then d' also belongs to L.

We prove below (Proposition (1.5)) that a subset L of $\mathbf{M}_s$ is $\mathbf{M}$-recognizable iff it is saturated w.r.t. a locally finite congruence on $\mathbf{M}$. This generalizes a well-known characterization of recognizable languages. The notion of syntactic congruence can also be generalized to arbitrary subsets of $\mathbf{M}_s$ and yields another characterization of $\mathbf{M}$-recognizable sets.

Let $L \subseteq \mathbf{M}_u$. We associate with L a congruence $\sim_L = (\sim_{L,s})_{s \in \mathscr{S}}$ on $\mathbf{M}$ as follows:

for $d, d' \in \mathbf{M}_s$:
$d \sim_{L,s} d'$ iff
for all n, for all linear term t in $\mathbf{M}(F, \{x_1, ..., x_n\})_u$ such that $\sigma(x_1) = s$,
for all $d_2, ..., d_n$ in $\mathbf{M}_{\sigma(x_2)}, ..., \mathbf{M}_{\sigma(x_n)}$:

$$t_{\mathbf{M}}(d, d_2, ..., d_n) \in L \Leftrightarrow t_{\mathbf{M}}(d', d_2, ..., d_n) \in L.$$

The congruence $\sim_L$ is called the *syntactic congruence* of L. In the special case where F generates $\mathbf{M}$, the elements $d_2, ..., d_n$ are defined by terms, hence, they can be "merged in t." In other words

$$d \sim_{L,s} d' \qquad \text{iff, for all } t \in \mathbf{Ctxt}(F)_{s,u}\text{: } t_{\mathbf{M}}(d) \in L \Leftrightarrow t_{\mathbf{M}}(d') \in L.$$

(1.4) Definition. **Inductive sets of predicates.** By a *predicate* on a set A, we mean a mapping $A \to \{\mathbf{true}, \mathbf{false}\}$. If $\mathbf{M}$ is a many-sorted F-magma with set of sorts $\mathscr{S}$, a *family of predicates* on $\mathbf{M}$ is an indexed set $\{\hat{p}/p \in P\}$, such that each p in P has a sort $\sigma(p)$ in $\mathscr{S}$, and each $\hat{p}$ is a predicate on $\mathbf{M}_{\sigma(p)}$. Such a family will also be denoted by P. For $p \in P$, we let $L_p = \{d \in \mathbf{M}_{\sigma(p)}/\hat{p}(d) = \mathbf{true}\}$.

The family P is *locally finite* if, for each $s \in \mathscr{S}$, the set $\{p \in P/\sigma(p) = s\}$ is finite.

It is F'*-inductive*, where $F' \subseteq F$, if for every f in F' of profile $s_1 \times s_2 \times \cdots \times s_n \to s$, for every $p \in P$ of sort s, there exist $m_1, ..., m_n$ in $\mathbb{N}$, there exists an $(m_1 + \cdots + m_n)$-place Boolean expression B, and a sequence of $(m_1 + \cdots + m_n)$ elements of P, $(p_{1,1}, ..., p_{1,m_1}, p_{2,1}, ..., p_{2,m_2}, ..., p_{n,m_n})$, such that:

(1) $\sigma(p_{i,j}) = s_i$ for all $j = 1, ..., m_i$

(2) for all $d_1 \in \mathbf{M}_{s_1}, ..., d_n \in \mathbf{M}_{s_n}$:

$$\hat{p}(f_{\mathbf{M}}(d_1, ..., d_n)) = B[\hat{p}_{1,1}(d_1), ..., \hat{p}_{1,m_1}(d_1), \hat{p}_{2,m_2}(d_2), ..., \hat{p}_{n,m_n}(d_n)].$$

The sequence $(B, p_{1,1}, ..., p_{2,1}, ..., p_{n,m_n})$ is called a *decomposition of p w.r.t. f*.

In words, the existence of such a decomposition means that the validity of p for any object of the form $f_{\mathbf{M}}(d_1, ..., d_n)$ can be determined from the truth values for $d_1, ..., d_n$ of finitely many predicates of P, in a way that depends only on p and f.

(1.5) PROPOSITION. *Let* $\mathbf{M}$ *be an F-magma. For every* $s \in \mathscr{S}$, *for every subset* L *of* $\mathbf{M}_s$, *the following conditions are equivalent*:

(i) *L is $\mathbf{M}$-recognizable,*

(ii) *L is saturated w.r.t. a locally finite congruence on* $\mathbf{M}$,

(iii) *the syntactic congruence of L is locally finite,*

(iv) *$L = L_p$ for some predicate p belonging to a locally finite, F-inductive family of predicates on* $\mathbf{M}$.

Proof. (i) $\Rightarrow$ (iv) Let $L = h^{-1}(C) \subseteq \mathbf{M}_s$ for some automaton $(h, \mathbf{A}, C)$. We can assume that the domains of $\mathbf{A}$ are pairwise disjoint. We let then $P = \bigcup\{\mathbf{A}_t / t \in \mathscr{S}\} \cup \{p\}$.

Each element a of $\mathbf{A}_t$ is of sort t (considered as a member of P), and p is of sort s. For $d \in \mathbf{M}_t$, and $a \in \mathbf{A}_t$, we let:

$$\begin{aligned} \hat{a}(d) &= \mathbf{true} \qquad \text{if} \quad h(d) = a, \\ &= \mathbf{false} \qquad \text{otherwise.} \end{aligned}$$

For $d \in \mathbf{M}_s$, we let

$$\begin{aligned} \hat{p}(d) &= \mathbf{true} \qquad \text{if} \quad h(d) \in C, \\ &= \mathbf{false} \qquad \text{otherwise.} \end{aligned}$$

It is clear that P is locally finite. It is not hard to prove that it is F-inductive, and, clearly, $L = L_p$.

(iv) $\Rightarrow$ (ii) Let P be a locally finite F-inductive family of predicates. The relations such that

$$d \sim_s d' :\Leftrightarrow d, d' \in \mathbf{M}_s, \hat{p}(d) = \hat{p}(d') \qquad \text{for all } p \in P \text{ of sort } s$$

are equivalence relations on the sets $\mathbf{M}_s$. Each of them has finitely many classes since P is locally finite. The family $\sim = (\sim_s)_{s \in \mathscr{S}}$ is a congruence since P is F-inductive (the verification is straightforward), and, for every p in P, the set L_p is saturated w.r.t. $\sim$.

(ii) $\Rightarrow$ (i) If L is saturated w.r.t. a locally finite congruence $\sim$ on $\mathbf{M}$, then one takes $(h, \mathbf{M}/\sim, h(L))$ as an automaton defining L, where h is the canonical surjective homomorphism: $\mathbf{M} \to \mathbf{M}/\sim$.

(iii) $\Rightarrow$ (ii) Holds trivially.

(ii) $\Rightarrow$ (iii) If L is $\sim$-saturated, then $\sim \subseteq \sim_L$. Hence $\sim_L$ is locally finite if L is. ∎

A locally finite and F-inductive family of predicates P on an F-magma $\mathbf{M}$ is *effectively locally finite and F-inductive* if the following conditions hold:

(1) $\mathbf{M}$ and P are effectively given,

(2) the mappings σ and σ^{-1} (σ^{-1} is such that $\sigma^{-1}(s) = \{p \in P/\sigma(p) = s\}$) are computable,

(3) the mapping: $P \times \bigcup\{\mathbf{M}_s/s \in \mathscr{S}\} \to \{\mathbf{true}, \mathbf{false}\}$ associating $\hat{p}(d)$ with $p \in P$ and $d \in \mathbf{M}_{\sigma(p)}$ is computable.

(4) there exists an algorithm producing a decomposition of p w.r.t. f, for every f in F and p in P.

(1.6) PROPOSITION. *Let* $\mathbf{M}$ *be an effectively given F-magma. An* $\mathbf{M}$*-recognizable subset* L *of* $\mathbf{M}_s$ *is effectively* $\mathbf{M}$*-recognizable iff* $L = L_p$, *for some predicate* p *of sort* s *belonging to an effectively locally finite and F-inductive family of predicates on* $\mathbf{M}$.

Proof. Only if. By (i) $\Rightarrow$ (iv) of the proof of Proposition (1.5).

If. Let P be an effectively locally finite and F-inductive family of predicates on $\mathbf{M}$.

For every $s \in \mathscr{S}$, we let P_s be the finite set $\sigma^{-1}(s)$, we let Θ_s be the set of all functions: $P_s \to \{\mathbf{true}, \mathbf{false}\}$, and we let $\mathbf{tv}$ be the mapping $\mathbf{M}_s \to \Theta_s$ such that $\mathbf{tv}(m)$ is the mapping $p \mapsto \hat{p}(m)$, for all $m \in \mathbf{M}_s$, $p \in P_s$.

From the hypothesis that P is effectively F-inductive, it follows that one can determine for every $f \in F$, a mapping f_Θ such that:

$$\mathbf{tv}(f_{\mathbf{M}}(m_1, ..., m_k)) = f_\Theta(\mathbf{tv}(m_1), ..., \mathbf{tv}(m_k)) \qquad \text{for all} \quad (m_1, ..., m_k) \in \mathbf{M}_{\alpha(f)}.$$

Hence $\Theta = \langle (\Theta_s)_{s \in \mathscr{S}}, (f_\Theta)_{f \in F} \rangle$ is an F-magma and $\mathbf{tv}$ is a homomorphism $\mathbf{M} \to \Theta$. Hence $(\mathbf{tv}, \Theta)$ is a semi-automaton, since Θ is locally finite. We have $L_p = \mathbf{tv}^{-1}(\Theta')$, where $\Theta' = \{\theta \in \Theta/\theta(p) = \mathbf{true}\}$. Hence L_p is effectively $\mathbf{M}$-recognizable. ∎

(1.7) PROPOSITION. *Let* $\mathbf{M}$ *be generated by F. A subset* L *of* $\mathbf{M}_t$ *is* $\mathbf{M}$*-recognizable iff* $h_{\mathbf{M}}^{-1}(L)$ *is* $\mathbf{M}(F)$*-recognizable. Furthermore, if* $\mathbf{M}$ *is effectively given, and if L is effectively* $\mathbf{M}$*-recognizable then* $h_{\mathbf{M}}^{-1}(L)$ *is effectively* $\mathbf{M}(F)$*-recognizable. The converse holds if F is finite.*

Proof. We first prove the "only if" directions. If $L = h^{-1}(C)$ for some homomorphism $h: \mathbf{M} \to \mathbf{A}$, where $\mathbf{A}$ is locally finite, then $h_{\mathbf{M}}^{-1}(L) = (h \circ h_{\mathbf{M}})^{-1}(C)$, and, since $h \circ h_{\mathbf{M}}$ is a homomorphism: $\mathbf{M}(F) \to \mathbf{A}$, the set

$h_{\mathbf{M}}^{-1}(L)$ is $\mathbf{M}(F)$-recognizable. If h is computable, then so are $h_{\mathbf{M}}$ as observed in Definition (1.1), and $\mathrm{hoh}_{\mathbf{M}}$. Hence $h_{\mathbf{M}}^{-1}(L)$ is effectively given in L is.

Let conversely $L \subseteq \mathbf{M}_t$ be such that $T = h_{\mathbf{M}}^{-1}(L)$ is $\mathbf{M}(F)$-recognizable. We have $T = h_{\mathbf{A}}^{-1}(C)$, where $\mathbf{A}$ is the locally finite F-magma $\mathbf{M}(F)/\sim_T$, and C is some subset of $\mathbf{A}_t$.

Let a, $a' \in \mathbf{M}_s$. Then $a \sim_{L,s} a'$ iff for all $c \in \mathbf{Ctxt}(F)_{s,t}$:

$$c_{\mathbf{M}}(a) \in L \Leftrightarrow c_{\mathbf{M}}(a') \in L.$$

But, for every $t \in \mathbf{M}_s$ such that $t_{\mathbf{M}} = a$,

$$c_{\mathbf{M}}(a) \in L \Leftrightarrow c[t] \in h_{\mathbf{M}}^{-1}(L) = T.$$

Hence for any two terms t and t' such that $t_{\mathbf{M}} = a$ and $t'_{\mathbf{M}} = a'$,

$$a \sim_{L,s} a' \qquad \text{iff} \quad t \sim_{T,s} t'.$$

This proves that $\sim_{L,s}$ and $\sim_{T,s}$ have the same number of classes. Hence L is recognizable, and furthermore $\mathbf{M}(F)/\sim_T$ is isomorphic to $\mathbf{M}/\sim_L$.

If, furthermore, F is finite, then $\mathbf{M}(F)/\sim_T$ is computable and defines an automaton recognizing L. ▮

(1.8) PROPOSITION. *The emptiness of an effectively given* $\mathbf{M}$*-recognizable set is not decidable in general. It is decidable under the additional conditions that the signature F is finite and generates* $\mathbf{M}$.

Proof. We first establish the decidability result. Let $\mathbf{M}$ be effectively given and generated by a finite signature F. If $L \in \mathbf{Rec}(\mathbf{M})_s$, then $h_{\mathbf{M}}^{-1}(L)$ is an effectively given recognizable subset of $\mathbf{M}(F)$. Its emptiness can be decided by a classical algorithm on tree-automata (see, for instance, Gecseg and Steinby [20]), and this also decides the emptiness of L.

We now consider the undecidability. We give two examples showing that none of the two hypotheses can be omitted. We consider the infinite one-sort signature F consisting of a constant, a, and of monadic functions f_n, for all $n \in \mathbb{N}$. Let g be a total recursive mapping $\mathbb{N} \to \{0, 1\}$.

Let $\mathbf{A}$ be the finite F-magma be associated with g as

$$\mathbf{A} = \{0, 1\}, \qquad a_{\mathbf{A}} = 0, \qquad f_{n\mathbf{A}}(1) = 1, \qquad f_{n\mathbf{A}}(0) = g(n).$$

Let $B = h_{\mathbf{A}}^{-1}(\{1\}) \subseteq \mathbf{M}(F)$. It is effectively $\mathbf{M}(F)$-recognizable. It is clear that $B \neq \varnothing$ iff $g(n) = 1$ for some $n \in \mathbb{N}$, and this not decidable.

Here is the second example. We let F' be reduced to the constant a. Let $\mathbf{M} = \langle \mathbb{N}, a_{\mathbf{M}} \rangle$ with $a_{\mathbf{M}} := 0$. (It is not generated by F'). Let $\mathbf{A}$ and g

be as above. The mapping h such that $h(0)=0$, $h(i)=g(i)$ if $i \geqslant 1$, is a homomorphism: $\mathbf{M} \to \mathbf{A}$. Hence $h^{-1}(\{1\})$ is an effectively given $\mathbf{M}$-recognizable set. It is nonempty iff $g(i)=1$ for some $i \geqslant 1$. And this is not decidable. ∎

In the next two propositions, $\mathbf{M}$ is an arbitrary F-magma, and s is one of its sorts.

(1.9) PROPOSITION. *The family of sets* $\mathbf{Rec}(\mathbf{M})_s$ *contains* $\varnothing$, $\mathbf{M}_s$, *and is closed under union, intersection, and difference.*

Proof (Sketch). If L_i is recognized by $(h_i, \mathbf{A}_i, C_i)$, $i=1, 2$, then, L_1 and L_2 are both recognized by the semi-automaton $(h_1 \times h_2, \mathbf{A}_1 \times \mathbf{A}_2)$, with respective sets of "final states" $C_1 \times A_2$ and $A_1 \times C_2$. The closure under union, intersection, and difference follows immediately. The other assertions are easy to verify. ∎

(1.10) PROPOSITION. *If* $K \in \mathbf{Rec}(\mathbf{M})_s$ *and* $L \in \mathbf{Equat}(\mathbf{M})_s$ *then* $L \cap K \in \mathbf{Equat}(\mathbf{M})_s$.

Proof. It follows from Mezei and Wright [25] or Courcelle [5, Section 14] that we can assume that $L = L((S, \mathbf{M}), U')$, where S is a *uniform* polynomial system over F with set of unknowns U, and $U' \subseteq U$. (A polynomial system is *uniform* if its equations are of the form $u = t_1 + t_2 + \cdots + t_m$, where each t_i is of the form $f(u_1, u_2, ..., u_k)$ for some $f \in F$, some $u_1, ..., u_k \in U$).

Let $F' \subseteq F$ be the finite set of symbols occurring in S, and let $\mathscr{S}' \subseteq \mathscr{S}$ be the finite set of sorts of the symbols occurring in S. Hence F' is an $\mathscr{S}'$-signature. Let $h\colon \mathbf{M} \to \mathbf{A}$ be a homomorphism (with $\mathbf{A}$ locally finite), such that $K = h^{-1}(C)$ for some $C \subseteq \mathbf{A}_s$.

For every $u \in U$, we let $L_u := L((S, \mathbf{M}), u)$. Let W be the new set of unknowns $\{[u, a]/u \in U, a \in \mathbf{A}_{\sigma(u)}\}$. It is finite. We shall define a system S', with set of unknowns W, such that

$$L((S', \mathbf{M}), [u, a]) = L_u \cap h^{-1}(a)$$

for all $[u, a] \in W$.

Let $u \in U$ and $a \in \mathbf{A}_{\sigma(u)}$. Let us assume that the defining equation of u in S is of the form $u = t_1 + \cdots + t_k$.

Consider one of the monomials, say t_i. Let us assume that it is of the form $f(u_1, ..., u_n)$.

For every $a_1 \in \mathbf{A}_{\sigma(u_1)}, ..., a_n \in \mathbf{A}_{\sigma(u_n)}$ such that $f_{\mathbf{A}}(a_1, ..., a_n) = a$, we form the monomial $f([u_1, a_1], ..., [u_n, a_n])$, and we let $\hat{t}_i$ denote the sum of these monomials. If no such n-tuple $(a_1, ..., a_n)$ exists, then $\hat{t}_i$ is defined as Ω.

The defining equation of $[u, a]$ in S' is taken as

$$[u, a] = \hat{t}_1 + \hat{t}_2 + \cdots \hat{t}_k.$$

It is clear from this construction that the W-indexed family of sets $(L_u \cap h^{-1}(a))_{[u,a] \in W}$ is a solution of S' in $\mathscr{P}(\mathbf{M})$. Hence $L_u \cap h^{-1}(a) \supseteq L_{u,a}$, where $(L_{u,a})_{[u,a] \in W}$ denotes the least solution of S' in $\mathscr{P}(\mathbf{M})$.

In order to establish the opposite inclusion, we define from $L_{u,a}$ the sets $L'_u = \bigcup \{L_{u,a}/a \in \mathbf{A}_{\sigma(u)}\}$ for $u \in U$. Then $(L'_u)_{u \in U}$ is a solution of S in $\mathbf{M}$ (this is easy to verify). Hence $L_u \subseteq L'_u$ for all u.

For all $a \in \mathbf{A}_{\sigma(u)}$, we have

$$L_u \cap h^{-1}(a) \subseteq L'_u \cap h^{-1}(a) = (\bigcup \{L_{u,a}/a \in \mathbf{A}\}) \cap h^{-1}(a).$$

The latter set is equal to $L_{u,a} \cap h^{-1}(a)$, since $L_{u,a'} \subseteq L_u \cap h^{-1}(a')$ and, $h^{-1}(a) \cap h^{-1}(a') = \varnothing$ for all a, a' with $a \neq a'$. Hence $L_u \cap h^{-1}(a) \subseteq L_{u,a}$. By the first part of the proof, we have an equality, and $(L_u \cap h^{-1}(a))_{[u,a] \in W}$ is the least solution of S' in $\mathscr{P}(\mathbf{M})$. Finally, we have

$$\begin{aligned} L \cap K &= (\bigcup \{L_u/u \in U'\}) \cap h^{-1}(C) \\ &= \bigcup \{L((S', \mathbf{M}), [u, a])/u \in U', a \in C\}. \end{aligned}$$

Hence $L \cap K \in \mathbf{Equat}(\mathbf{M})_s$. ∎

The above construction is effective if K is effectively given, and L is defined by a given system. Hence since the emptiness of an equational set (defined by a system of equations) is decidable, we have the following corollary that can be contrasted with the undecidability result of Proposition (1.8).

(1.11) COROLLARY. *If K is an effectively given $\mathbf{M}$-recognizable set, and if L is an $\mathbf{M}$-equational set, one can test whether $L \cap K = \varnothing$, or whether $L \subseteq K$.*

The following result is due to Mezei and Wright [25].

(1.12) PROPOSITION. *A subset L of $\mathbf{M}_s$ is $\mathbf{M}$-equational iff $L = h_{\mathbf{M}}(T)$ for some $T \in \mathbf{Rec}(\mathbf{M}(F'))_s$, and some finite subset F' of F.*

In the following corollary, $\mathbf{Rec}(\mathbf{M}) \subseteq \mathbf{Equat}(\mathbf{M})$ means: $\mathbf{Rec}(\mathbf{M})_s \subseteq \mathbf{Equat}(\mathbf{M})_s$ for all s in $\mathscr{S}$.

(1.13) COROLLARY. *Let $\mathbf{M}$ be generated by F. Then $\mathbf{Rec}(\mathbf{M}) \subseteq \mathbf{Equat}(\mathbf{M})$ iff for every $s \in \mathscr{S}$, there exists a finite subset F' of F such that $h_M(\mathbf{M}(F')_s) = \mathbf{M}_s$.*

Proof. If. Let $L \in \mathbf{Rec}(\mathbf{M})$, let F' be such that $h_{\mathbf{M}}(\mathbf{M}(F')_s) = \mathbf{M}_s$. Then $T = h_{\mathbf{M}}^{-1}(L) \cap \mathbf{M}(F')_s \in \mathbf{Rec}(\mathbf{M}(F'))_s$ (since $h_{\mathbf{M}}^{-1}(L) \in \mathbf{Rec}(\mathbf{M}(F))_s$, and by Proposition (1.9)). Hence $L = h_{\mathbf{M}}(T)$, and is $\mathbf{M}$-equational.

Only if. Let $\mathbf{Rec}(\mathbf{M}) \subseteq \mathbf{Equat}(\mathbf{M})$. Then $\mathbf{M}_s \in \mathbf{Equat}(\mathbf{M})$ and $\mathbf{M}_s = h_{\mathbf{M}}(T')$ for some $T' \in \mathbf{Rec}(\mathbf{M}(F'))_s$ with F' finite, $F' \subseteq F$. Hence $\mathbf{M}_s = h_{\mathbf{M}}(\mathbf{M}(F'))_s$. ∎

(1.14) COROLLARY. $\mathbf{Rec}(\mathbf{M}(F)) = \mathbf{Equat}(\mathbf{M}(F))$ *if F is finite.*

In the following proposition we assume that F and F' are two signatures over a same set of sorts $\mathscr{S}$, that $F' \subseteq F$, that $\mathbf{M}$ is an F-magma, and that $\mathbf{M}'$ is a sub-F'-magma of $\mathbf{M}$ (we write this $\mathbf{M}' \subseteq \mathbf{M}$). If G is a new $\mathscr{S}$-signature disjoint from F, and $\mathbf{P}$ be a G-magma with the same family of domains as $\mathbf{M}$, such that $g_{\mathbf{P}}$ is a derived operation of $\mathbf{M}$, we say that $\mathbf{P}$ is a *derived magma of* $\mathbf{M}$.

(1.15) PROPOSITION. *Let $F' \subseteq F$ and $\mathbf{M}' \subseteq \mathbf{M}$. For every $s \in \mathscr{S}$:*

(1) $L \cap \mathbf{M}'_s \in \mathbf{Rec}(\mathbf{M}')_s$ *for all* $L \in \mathbf{Rec}(\mathbf{M})_s$.

If $\mathbf{P}$ is a derived magma of $\mathbf{M}$, then for every $s \in \mathscr{S}$:

(2) $\mathbf{Rec}(\mathbf{M})_s \subseteq \mathbf{Rec}(\mathbf{P})_s$.

We omit the proof which is a straightforward verification from the definitions. The inclusions are strict in general, and $\mathbf{M}'_s$ is not necessarily in $\mathbf{Rec}(\mathbf{M})_s$. Note also that, if $\mathbf{M}'_s = \mathbf{M}_s$ in (1), then

$$\mathbf{Rec}(\mathbf{M})_s \subseteq \mathbf{Rec}(\mathbf{M}')_s.$$

2. GRAPHS, GRAPH OPERATIONS, AND GRAPH EXPRESSIONS

As in [2, 7–9, 11, 13–15], we deal with labeled, directed hypergraphs, equipped with a sequence of distinguished vertices called the sequence of sources.

The labels are chosen in a *ranked alphabet*, i.e., in a finite set A, each element of which has an associated nonnegative integer, that we call its *type*. The type is defined by a mapping $\tau: A \to \mathbb{N}$. The type of the label of an hyperedge must be equal to the length of its sequence of vertices. This type may be 0. In order to shorten the statements, we shall simply call *graphs* these hypergraphs, and *edges* their hyperedges.

(2.1) DEFINITION. **Graphs**. Let A and τ as above, let $n \in \mathbb{N}$. A *concrete n-graph* is a quintuple

$$G = \langle \mathbf{V}_G, \mathbf{E}_G, \mathbf{lab}_G, \mathbf{vert}_G, \mathbf{src}_G \rangle,$$

where

— $\mathbf{V}_G$ is a set whose elements are the vertices of the graph;

— $\mathbf{E}_G$ is a set whose elements are the edges;

— $\mathbf{lab}_G : \mathbf{E}_G \to A$ defines the label of an edge;

— $\mathbf{vert}_G : \mathbf{E}_G \to \mathbf{V}_G^*$ associates with every edge e of G, the sequence of its vertices, a sequence of length $\tau(e) := \tau(\mathbf{lab}_G(e))$; its ith element is denoted by $\mathbf{vert}_G(e, i)$;

— $\mathbf{src}_G$ is a sequence of length n in $\mathbf{V}_G^*$, or equivalently, a mapping: $[n] \to \mathbf{V}_G$. Hence, $\mathbf{src}_G(i)$ denotes the ith element of the sequence $\mathbf{src}_G$. It is called a *source*. If $n = 0$, then G has no source. "Source" is just an easy sounding word for "distinguished vertex." There is no notion of flow involved. The integer n is the *type* of G.

Whenever we need to specify the alphabet A, we say that G is a *concrete n-graph over A*. A *concrete graph* is a concrete n-graph for some $n \geqslant 0$.

A vertex v *belongs* to an edge e if $v = \mathbf{vert}_G(e, i)$ for some i. A vertex is *isolated* if it belongs to no edge. An edge e is *binary* if it is of type 2. If this is the case then $\mathbf{vert}_G(e, i)$ is called the *origin* of e, and $\mathbf{vert}_G(e, 2)$ is called its *target*. An *internal* vertex of G is a vertex that does not appear in the sequence $\mathbf{src}_G$.

A concrete n-graph G and a concrete n'-graph H (both over A) are *isomorphic* if $n' = n$, and if there exist bijective mappings $h_{\mathbf{V}}$ and $h_{\mathbf{E}}$,

$$h_{\mathbf{V}} : \mathbf{V}_G \to \mathbf{V}_H$$

$$h_{\mathbf{E}} : \mathbf{E}_G \to \mathbf{E}_H,$$

such that

$$\mathbf{lab}_H \circ h_{\mathbf{E}} = \mathbf{lab}_G,$$
$$h_{\mathbf{V}}(\mathbf{vert}_G(e, i)) = \mathbf{vert}_H(h_{\mathbf{E}}(e), i) \text{ for all } i \in [\tau(e)], \text{ all } e \text{ in } \mathbf{E}_G,$$
$$h_{\mathbf{V}}(\mathbf{src}_G(i)) = \mathbf{src}_H(i) \text{ for all } i \in [n].$$

A *graph* is the isomorphism class of a concrete graph. A graph G is *finite* if $\mathbf{V}_G$ and $\mathbf{E}_G$ are finite. By a graph, we shall mean a finite graph in the present paper. Infinite countable graphs are considered in Courcelle [11, 13–15].

We denote by $\mathbf{FCG}(A)_n$ (resp. by $\mathbf{FCG}(A)$) (resp. by $\mathbf{FG}(A)_n$) (resp. by $\mathbf{FG}(A)$), the sets of concrete n-graphs (resp. of concrete graphs) (resp. of n-graphs) (resp of graphs) over A.

(2.2) EXAMPLES. The following very simple graphs will be useful to build nontrivial graphs by means of graph expressions:

(1) The *discrete graph* $\mathbf{n}$, for $n \geqslant 0$, is the graph G such that $\mathbf{V}_G = [n]$, $\mathbf{E}_G = \varnothing$, $\mathbf{lab}_G = \varnothing$, $\mathbf{vert}_G = \varnothing$, $\mathbf{src}_G$ is the sequence $(1, 2, ..., n)$. In particular we have the *empty* graph $\mathbf{0}$ which is (necessarily) of type 0.

(2) If b is an element of A type n, then b also denotes the graph G with a single edge e labeled by b, and such that $\mathbf{V}_G = [n]$, $\mathbf{E}_G = \{e\}$, $\mathbf{lab}_G(e) = b$, $\mathbf{vert}_G(e) = \mathbf{src}_G = (1, 2, ..., n)$. The graph b is reduced to an edge with no vertex in the special case where $n = 0$.

(2.3) DEFINITION. **Subgraphs.** Let G be a concrete graph. A concrete graph H such that $\mathbf{V}_H \subseteq \mathbf{V}_G$, $\mathbf{E}_H \subseteq \mathbf{E}_G$, $\mathbf{lab}_H = \mathbf{lab}_G \upharpoonright \mathbf{E}_H$, $\mathbf{vert}_H = \mathbf{vert}_G \upharpoonright \mathbf{E}_H$, and $\mathbf{src}_H$ is obtained from $\mathbf{src}_G$ by the deletion of the vertices not in H, is called a *subgraph of* G. We crite this $H \subseteq G$.

(2.4) DEFINITION. **Quotient graphs.** Let G be a concrete graph, let $\simeq$ be an equivalence relation on $\mathbf{V}_G$. We denote by $[v]$ the equivalence class w.r.t. $\simeq$ of a vertex v. Then, we denote by $G/\simeq$ the concrete graph H such that $\mathbf{V}_H = \mathbf{V}_G/\simeq$, $\mathbf{E}_H = \mathbf{E}_G$, $\mathbf{lab}_G = \mathbf{lab}_H$, $\mathbf{vert}_H(e, i) = [\mathbf{vert}_G(e, i)]$ for all $e \in \mathbf{E}_H$ $(= \mathbf{E}_G)$ and all $i \in [\tau(e)]$, $\mathbf{src}_H(i) = [\mathbf{src}_G(i)]$ for all $i \in [\tau(G)]$. We call $G/\simeq$ *the quotient graph of* G *by* $\simeq$. If G is a graph, then $G/\simeq$ is the isomorphism class of $\bar{G}/\simeq$, where $\bar{G}$ is any concrete graph in the class G.

(2.5) DEFINITION. **Graph operations.** We recall from [2, 7] the definitions of three operations on graphs (or rather of three families of operations) making the set of graphs into a many-sorted magma.

The first operation is the *disjoint sum*. Let G and H be two graphs of respective types n' and n''. We can assume that they are the isomorphism classes of two concrete graphs also denoted by G and H, such that $\mathbf{V}_G \cap \mathbf{V}_H = \varnothing$, $\mathbf{E}_G \cap \mathbf{E}_H = \varnothing$. Then $G \oplus H$ is the isomorphism class of the concrete $(n' + n'')$-graph K such that:

$$\mathbf{V}_K = \mathbf{V}_G \cup \mathbf{V}_H,$$

$$\mathbf{E}_K = \mathbf{E}_G \cup \mathbf{E}_H,$$

$$\mathbf{lab}_K = \mathbf{lab}_G \cup \mathbf{lab}_H,$$

$$\mathbf{vert}_K = \mathbf{vert}_G \cup \mathbf{vert}_H,$$

$$\mathbf{src}_K = (\mathbf{src}_G(1), ..., \mathbf{src}_G(n'), \mathbf{src}_H(1), ..., \mathbf{src}_H(n'')).$$

Here is the second operation. With a map α from $[p]$ to $[n]$, we associate the *source redefinition* map $\sigma_\alpha : \mathbf{FG}(A)_n \to \mathbf{FG}(A)_p$ defined as follows. We let $\sigma_\alpha(G) := \langle \mathbf{V}_G, \mathbf{E}_G, \mathbf{lab}_G, \mathbf{vert}_G, \mathbf{src}_G \circ \alpha \rangle$. If $p = 0$, then α is necessarily the empty map (always denoted by $\varnothing$) and $\sigma_\varnothing(G)$, is the

0-graph obtained from G by "forgetting" its sources. We call it *the 0-graph associated with G.*

When p is small it is convenient to write $\sigma_{i_1, i_2, \ldots, i_p}(G)$ instead of $\sigma_\alpha(G)$, by letting $i_j := \alpha(j)$ for $j = 1, \ldots, p$.

The third operation is the *source fusion.* For every equivalence relation δ on $[n]$, we define a mapping $\theta_\delta : \mathbf{FG}(A)_n \to \mathbf{FG}(A)_n$ as follows. We let $\theta_\delta(G)$ be the quotient graph $G/\simeq$, where $\simeq$ is the equivalence relation on $\mathbf{V}_G$ such that

$$v \simeq v' \Leftrightarrow v = v' \text{ or } \{v = \mathbf{src}_G(i),\ v' = \mathbf{src}_G(j), \text{ and } (i, j) \in \delta\}.$$

If δ is the equivalence relation on $[n]$ generated by the single pair (i, j), then we denote θ_δ by $\theta_{i,j}$. It is clear that if δ is the equivalence relation generated by a set of pairs $\{(i_1, j_1), \ldots, (i_k, j_k)\}$ then

$$\theta_\delta = \theta_{i_1, j_1} \circ \cdots \circ \theta_{i_k, j_k}.$$

(2.6) DEFINITION. **The many-sorted magma FG(A).** Let $\mathbb{N}$ be considered as a set of sorts. We define an $\mathbb{N}$-signature $\mathbf{H}_A$ consisting of the following symbols:

$\bigoplus_{n,m}$, of profile $n \times m \to n + m$ for all $n, m \in \mathbb{N}$

$\theta_{\delta,n}$, of profile: $n \to n$ for all $n \in \mathbb{N}$, all equivalence relations δ on $[n]$.

$\sigma_{\alpha,p,n}$, of profile: $n \to p$ for all n, $p \in \mathbb{N}$, all mappings $\alpha : [p] \to [n]$.

In addition, we put in $\mathbf{H}_A$ the following symbols:

a, a constant of sort $\tau(a)$, for all a in A,

$\mathbf{0}$, a constant of sort 0,

$\mathbf{1}$, a constant of sort 1.

We obtain an $\mathbf{H}_A$-magma $\mathbf{FG}(A)$. Its domain of sort n is $\mathbf{FG}(A)_n$, the set of graphs of type n. The functions associated with the symbols $\bigoplus_{n,m}$, $\theta_{\delta,n}$ and $\sigma_{\alpha,p,n}$ are defined in Definition (2.5). The graphs associated with the constants a, $\mathbf{0}$, and $\mathbf{1}$ are defined in Examples (2.2). It is clear that $\mathbf{FG}(A)$ is effectively given.

(2.7) DEFINITION. **Graph expressions.** An element of $\mathbf{FE}(A) := \mathbf{M}(\mathbf{H}_A)$ is called a *graph expression.* Every graph expression t defines a unique finite graph $t_{\mathbf{FG}(A)}$, also denoted by $\mathbf{val}(t)$ and called its *value.* The following proposition says that $\mathbf{H}_A$ generates $\mathbf{FG}(A)$.

(2.8) PROPOSITION ([2]). *Every graph in* $\mathbf{FG}(A)$ *is the value of a graph expression.*

When writing expressions we shall omit the subscripts n, m, in the symbols $\oplus_{n,m}$, $\sigma_{\alpha,n,m}$, $\theta_{\delta,n}$. Provided the sorts of the variables appearing in an expression are known, its sort can be computed and its well-formedness can be checked.

Since $\oplus$ is associative (more precisely $G \oplus_{m,n+p} (G' \oplus_{n,p} G'') = ((G \oplus_{m,n} G')_{m+n,p} G'')$ for all graphs G, G', G'' of respective types m, n, p), we denote it as an infix operator and we omit parentheses.

(2.9) DEFINITION. **The width of a graph.** For every k, we let $\mathbf{H}_A^{[k]}$ be the $[0, k]$-signature consisting of the symbols of $\mathbf{H}_A$ having their sort in $[0, k]$, and their arity in $[0, k]^*$.

We denote by $\mathbf{FE}(A)_n^{[k]}$ the set $\mathbf{M}(\mathbf{H}_A^{[k]})_n$ and call it the set of *graph expressions over A, of type n, and of width at most k* (this is meaningful if $k \geqslant n$). Hence $\mathbf{FE}(A)_n$ is equal to $\bigcup\{\mathbf{FE}(A)_n^{[k]}/k \geqslant n\}$.

Whereas $\mathbf{H}_A$ generates $\mathbf{FG}(A)$, the set $\mathbf{FG}(A)_n^{[k]}$ of values of expressions in $\mathbf{FE}(A)_n^{[k]}$ is a proper subset of $\mathbf{FG}(A)_n$. We denote by $\mathbf{FG}(A)^{[k]}$ the $\mathbf{H}_A^{[k]}$-magma with domains $\mathbf{FG}(A)_n^{[k]}$, $n \leqslant k$. This magma has finitely many sorts and operations.

The *width* of a finite graph G is defined as the minimal k such that $G \in \mathbf{FG}(A)_n^{[k]}$ for some k. It is denoted by $\mathbf{wd}(G)$.

(2.10) DEFINITION. **Equational and recognizable sets of graphs.** The $\mathbf{FG}(A)$-equational and the $\mathbf{FG}(A)$-recognizable sets are called the *equational* and the *recognizable* sets of graphs.

The equational sets of graphs are also the *context-free sets*, i.e., the sets of graphs generated by the context-free graph grammars of Bauderon and Courcelle [2, 8]. We recall the definition.

A *context-free* graph-grammar is a 3-tuple $\Gamma = \langle A, U, P\rangle$, where A is a finite ranked set (the *terminal* alphabet), $U = \{u_1, ..., u_n\}$ is a finite ranked set (the *nonterminal* alphabet), P is a finite set of production rules. A *production rule p* is a pair (u, e) with u in U and e in $\mathbf{FE}(A \cup U)_{\tau(u)}$. We write $p: u \to e$, and we use p as a *name*, identifying the production rule in a unique way. We also denote by P the set of names of the production rules in P. If $p: u \to e$, if $h, h' \in \mathbf{FE}(A \cup U)$, we write $h \to_p h'$ if h' is obtained from h by the substitution of e for u, at one of its occurrences. We write $h \to_P h'$ if $h \to_p h'$ for some p in P. (Hence, we consider P as ground term rewriting system on $\mathbf{FE}(A \cup U)$.)

The *set of graphs generated by Γ from u_i is* $L(\Gamma, u_i) := \{\mathbf{val}(h)/h \in \mathbf{FE}(A),\ u_i \xrightarrow{*}_P h\}$, and we let $L(\Gamma) := L(\Gamma, u_1)$.

A system of equations over $\mathscr{P}(\mathbf{FG}(A))$ can be associated with Γ as follows:

$$S_\Gamma := \langle u_1 = t_1, ..., u_n = t_n\rangle,$$

where t_i is the polynomial $e_1 + \cdots + e_k$ and $\{e_1, ..., e_k\}$ is the set of right-hand sides of the production rules of Γ, the left-hand side of which is u_i.

It has been proved in [2, Theorem (4.9)] that the least solution of S_Γ in $\mathscr{P}(\mathbf{FG}(A))$ is the n-tuple $(L(\Gamma, u_1), ..., L(\Gamma, u_n))$.

Conversely, a context-free graph-grammar can be associated with a polynomial system on $\mathscr{P}(\mathbf{FG}(A))$, and we have the following result [2, Propositions (4.11) and (4.17)]:

(2.11) PROPOSITION. *A set of finite graphs is equational iff it is context-free. The graphs of an equational set are of bounded width.*

This proposition is effective: a system of equations can be constructed from a grammar and conversely. An upper bound on the widths of the graphs of an equational set can be computed. Since $\mathbf{H}_A^{[k]}$ is a finite signature, the sets $\mathbf{FG}(A)_n^{[k]}$, $n \leqslant k$, are all equational.

The following proposition characterizes the recognizable sets in terms of graph substitutions, rather than in terms of the graph operations of Definition (2.5).

(2.12) PROPOSITION. *A subset L of $\mathbf{FG}(A)_n$ is recognizable iff for every k, the equivalence relation on $\mathbf{FG}(A)_k$ defined by*

> $G \equiv_k G'$ *iff, for every H in $\mathbf{FG}(A)_n$, for every edge e of H of type k, $H[G/e] \in L$ iff $H[G'/e] \in L$*

is finite.

(We denote by $H[G/e)$ the result of the substitution in H of G for the edge e of H.)

Proof. For every graph H of type n, for every edge e of H of type k, there exists t in $\mathbf{Ctxt}(\mathbf{H}_A)_{k,n}$ such that

$$H[G/e] = t_{\mathbf{FG}(A)}(G)$$

for every graph G in $\mathbf{FG}(A)_k$. Conversely, for every context t, there exists H satisfying this for every graph G. (This follows from [2, Lemma (4.15)].) Hence $(\equiv_k)_{k \in \mathbb{N}}$ is the syntactic congruence of L. The result follows. ∎

Lengauer and Wanke have introduced in [24] the notion of a finite graph property. Restating their definition in our terminology, we have that a property of 0-graphs is *finite* if it is decidable and the equivalences $\equiv_k$ associated with the set of 0-graphs satisfying it as in the statement of Proposition (2.12) are finite. Hence, up to a few minor details, the notion of a finite graph property is equivalent to that of an effectively given recognizable set of graphs.

We now compare the families of equational and recognizable sets of graphs. It is well known that the family of recognizable languages is included in the family of context-free ones, and that the inclusion is strict if the alphabet contains at least two symbols. In short, $\mathbf{Rec}(X^*)$ (the class of recognizable languages) is strictly included in $\mathbf{Equat}(X^*)$ (the class of context-free languages), provided $\mathbf{Card}(X) \geqslant 2$.

An analogous result holds for $\mathbf{FG}(A)^{[k]}$, for all k, but it does not hold for $\mathbf{FG}(A)$: the families $\mathbf{Rec}(\mathbf{FG}(A))$ and $\mathbf{Equat}(\mathbf{FG}(A))$ are incomparable.

(2.13) PROPOSITION. (1) *For every* $k \geqslant 0$, *and* $n \leqslant k$, *the following inclusion holds*:

$$\mathbf{Rec}(\mathbf{FG}(A)^{[k]})_n \subseteq \mathbf{Equat}(\mathbf{FG}(A)^{[k]})_n.$$

(2) *If* A *contains at least one symbol of type* p *strictly larger than* 1, *and if* $k \geqslant \mathbf{Max}\{n, p+2\}$, *the above inclusion is strict.*

(3) *If* A *is as in* (2), *then, the families* $\mathbf{Rec}(\mathbf{FG}(A))_n$ *and* $\mathbf{Equat}(\mathbf{FG}(A))_n$ *are incomparable.*

Proof. (1) Let $K \in \mathbf{Rec}(\mathbf{FG}(A)^{[k]})_n$. We have $K = K \cap \mathbf{FG}(A)_n^{[k]}$, hence K is equational by Proposition (1.10), since $\mathbf{FG}(A)_n^{[k]}$ is.

(3) Let us first assume that A contains one symbol a of type 2, and two symbols b and c of type 1. Let L be the set of 0-graphs of the form shown on Fig. 1, with as many b's and c's.

They correspond in an obvious way to the words of the language $L' = \{b^n c^n / n \geqslant 1\}$. It is easy to construct a context-free graph-grammar generating L. If L would be recognizable, so would be the language L'. (From an automaton defining L, it is not hard to construct an automaton defining the language L'). But L' is known to be not recognizable. This proves that $\mathbf{Equat}(\mathbf{FG}(A))_0$ is not included in $\mathbf{Rec}(\mathbf{FG}(A))_0$.

If A contains one symbol d of type $p \geqslant 2$, then one considers

$$\bar{L} = \{G[K_a/a, K_b/b, K_c/c]/G \in L\}$$

instead of L, where $K_a = \sigma_{1,2}(d)$, $K_b = \sigma_1(d)$, $K_c = \sigma_p(d)$. (By $G[K_a/a, K_b/b, K_c/c]$, we denote the result of the simultaneous substitution of K_a for all edges of G labeled by a, and similarily for b and c.)

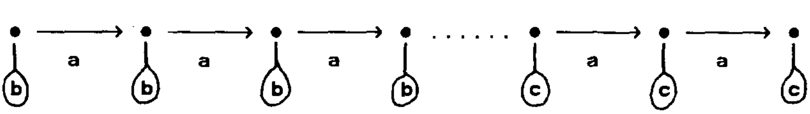

FIGURE 1

The result of this substitution when $p=3$, and G corresponds to the word b^2c^2 and is shown on Fig. 2.

It is easy to construct a context-free graph-grammar generating $\bar{L}$, and, again, from an automaton recognizing $\bar{L}$, one could obtain an automaton recognizing L'. By equipping the graphs of L (and of $\bar{L}$) with sources, one could establish similarily that $\mathbf{Equat}(\mathbf{FG}(A))_n$ is not included in $\mathbf{Rec}(\mathbf{FG}(A))_n$ for any $n \geqslant 0$.

Proposition (2.14) below says that $\mathbf{Rec}(\mathbf{FG}(A))_0$ is uncountable. On the other hand, $\mathbf{Equat}(\mathbf{FG}(A))_0$ is countable since there are countably many systems of equations (or grammars). Hence one cannot have $\mathbf{Rec}(\mathbf{FG}(A))_0 \subseteq \mathbf{Equat}(\mathbf{FG}(A))_0$ and the families $\mathbf{Rec}(\mathbf{FG}(A))$ and $\mathbf{Equat}(\mathbf{FG}(A))$ are incomparable.

(2) In order to finish the comparison of $\mathbf{Rec}(\mathbf{FG}(A))^{[k]})_n$ and $\mathbf{Equat}(\mathbf{FG}(A)^{[k]})_n$, it suffices to observe that a system of equations defining L (or $\bar{L}$) can be constructed with symbols from $\mathbf{H}_A^{[h]}$ where $h = \mathbf{Max}\{n, p+2\}$. Hence L (or $\bar{L}$) belongs to $\mathbf{Equat}(\mathbf{FG}(A)^{[k]})_n$ for all $k \geqslant \mathbf{Max}\{n, p+2\}$. We omit the details. ▮

(2.14) PROPOSITION. *If A contains at least one symbol of type strictly larger than* 1, *then* $\mathbf{Rec}(\mathbf{FC}(A))_0$ *is uncountable.*

The proof of this proposition needs several definitions and lemmas. We let A consist of one symbol, a, of type 2.

(2.15) DEFINITION. **Grids**. We denote by G_n the *$n \times n$-grid*, a graph belonging to $\mathbf{FG}(A)$. Rather than giving a formal definition, we show the grid G_3 on Fig. 3. All its edges are labeled by a, and these labels are omitted on the drawing.

We let $L_G = \{G_n / n \geqslant 2\}$. Our purpose is to establish that every subset L of L_G is recognizable. To do so, we shall prove that the syntactic congruence $\sim_L$ of every such set is locally finite.

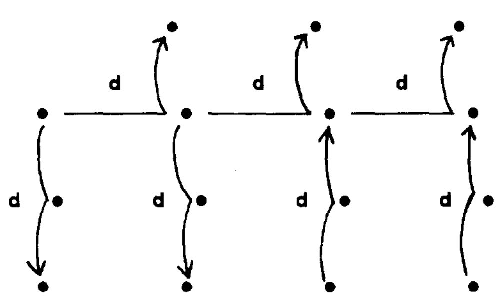

FIGURE 2

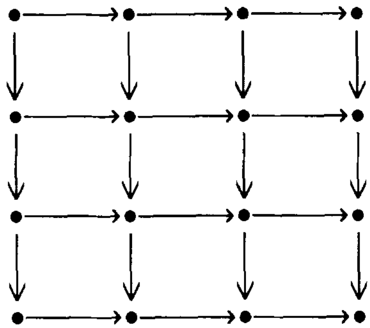

FIGURE 3

We denote by $\delta(p, n)$ the equivalence relation on $[p+2n]$ generated by $\{(p+1, p+n+1), ..., (p+n, p+2n)\}$. In the following lemma, we let B be an arbitrary finite ranked alphabet.

(2.16) LEMMA. *Let $M \subseteq \mathbf{FG}(B)_p$, $p \geqslant 0$. Let $G, G' \in \mathbf{FG}(B)_n$, $n \geqslant 0$. Then $G \sim_{M,n} G'$ iff, for all $K \in \mathbf{FG}(B)_{n+p}$:*

$$\sigma_{1,2,...,p}(\theta_{\delta(p,n)}(K \oplus G)) \in M \Leftrightarrow$$
$$\sigma_{1,2,...,p}(\theta_{\delta(p,n)}(K \oplus G')) \in M$$

Proof. By Proposition (1.10), $G \sim_{M,n} G'$ iff $c_{\mathbf{FG}(B)}(G) \in M \Leftrightarrow c_{\mathbf{FG}(B)}(G') \in M$ for all $c \in \mathbf{Ctxt}(\mathbf{H}_s)_{n,p}$. It follows from [2, Remark p. 117] that, for every c in $\mathbf{Ctxt}(\mathbf{H}_B)_{n,p}$, there exists a graph $K \in \mathbf{FG}(B)_{p+n}$ such that, for all $G \in \mathbf{FG}(B)_n$:

$$c_{\mathbf{FG}(B)}(G) = \sigma_{1,2,...,p}(\theta_{\delta(p,n)}(K \oplus G)).$$

Conversely, with every graph K, a context c can be associated, such that this equality holds, for all G in $\mathbf{FG}(B)_n$. The desired characterization of $\sim_{L,n}$ follows immediately. ∎

We shall use this lemma for $p = 0$. Hence, we introduce a derived operation $\square_m$: $\mathbf{FG}(B)_m \times \mathbf{FG}(B)_m \to \mathbf{FG}(B)_0$, defined by

$$G \square_m G' = \sigma_\phi(\theta_{\delta(0,m)}(G \oplus_{m,m} G')).$$

This operation on graphs can be described as follows. In order to construct $G \square_m G'$, one glues G and G' by fusing $\mathbf{src}_G(i)$ and $\mathbf{src}_{G'}(i)$ for all $i = 1, ..., m$, and the resulting graph has no source. This operation is commutative. If $m = 0$, then $\square_m$ is the disjoint sum.

We write $\square$ instead of $\square_m$, when m is known from the context.

(2.17) LEMMA. *Let $n \geqslant 2m+3 \geqslant 5$. Let $G, G' \in \mathbf{FG}(A)_m$ be such that $G \square G' = G_n$. Then one of G, G', say G, has less than $m+m^2$ vertices, the other has more than $3m^2$ vertices, and for every G'' in $\mathbf{FG}(A)_m$, if $G'' \square G' \in L_G$ then $G'' \square G' = G_n$.*

Proof. Let G and G' be two concrete disjoint m-graphs such that $G \square G'$ is isomorphic to G_n.

We let H be the *restriction of* G to its set of internal vertices. More precisely:

$\mathbf{V}_H =$ the set of internal vertices of G

$\mathbf{E}_H =$ the set of edges of G having all their vertices in $\mathbf{V}_H$

$\mathbf{vert}_H = \mathbf{vert}_G \upharpoonright \mathbf{E}_H$

$\mathbf{lab}_H = \mathbf{lab}_G \upharpoonright \mathbf{E}_H$

$\mathbf{src}_H = (\ )$.

Similarily we let H' be the restriction of G' to its set of internal vertices.

By the isomorphism $j\colon G \square G' \to G_n$, the subgraphs H and H' of G and G' are isomorphic to disjoint subgraphs $\bar{H}$ and $\bar{H}'$ of G_n. In order to simplify the notations, we denote $\bar{H}$ and $\bar{H}'$ by H and H', respectively.

Hence H and H' are two subgraphs of G_n. Note that G_n has no edge linking a vertex of H to a vertex of H', and that an edge of G_n linking two vertices of H (or of H') is in H (or in H').

Let $S = \mathbf{V}_{G_n} - (\mathbf{V}_H \cup \mathbf{V}_{H'})$. Each vertex of S corresponds by j to at least one source of G and at least one source of G'. Hence $\mathbf{Card}(S) \leqslant m$.

Figure 4 below shows an example of such a situation with $n = 4$ (and a large m). The vertices of H are indicated by o, the vertices of H' are indicated by •, the vertices of S are indicated by ∘.

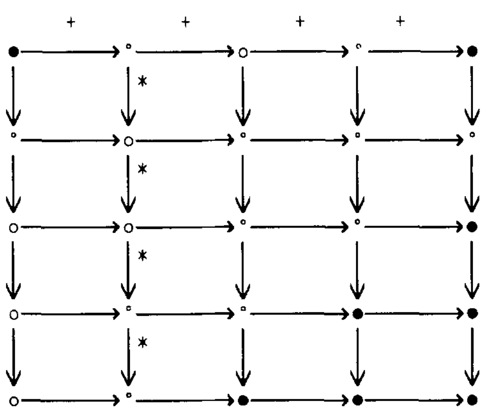

FIGURE 4

A path in G_n like the one marked with $+$'s on Fig. 4 is called a *complete horizontal path*. A path in G_n like the one marked with *'s is called a *complete vertical path*. These paths have $n+1$ vertices.

We are now ready to start the proof; we assume that $n \geqslant 2m+3$.

If H and H' both contain a complete vertical path of G_n, then all complete horizontal paths of G_n contain vertices from H and from H'. Hence they all contain vertices from S, and $\mathbf{Card}(S) \geqslant n+1$. But we have proved what $\mathbf{Card}(S) \leqslant m$, and we have assumed that $n \geqslant 2m+3$. This gives a contradiction.

Hence one of H and H', say H, does not contain any complete vertical path. Let K be the set of complete vertical paths of G_n that are not contained in H'. They all have vertices in S. Hence $\mathbf{Card}(K) \leqslant \mathbf{Card}(S)$. Since $\mathbf{Card}(S) \leqslant m$, H' contains at least $n+1-m$ complete vertical paths. The graph H is contained in the union of the paths of K.

Since H and H' are disjoint, H cannot contain any complete horizontal path. As above for vertical paths, H' contains at least $n+1-m$ complete horizontal paths, and H is contained in the union of a set K' of at most m complete horizontal paths.

Hence $\mathbf{Card}(\mathbf{V}_H) \leqslant m^2$. Since $\mathbf{Card}(S) \leqslant m$, we have $\mathbf{Card}(\mathbf{V}_{H'}) \geqslant (n+1)^2 - m - m^2 \geqslant 3m^2$ (since $n \geqslant 2m+3$). It follows that $\mathbf{Card}(\mathbf{V}_G) \leqslant m + m^2$ and that $\mathbf{Card}(\mathbf{V}_{G'}) \geqslant \mathbf{Card}(\mathbf{V}_{H'}) \geqslant 3m^2$.

Now let $G'' \in \mathbf{FG}(A)_m$ be such that $G'' \square G'$ is isomorphic to $G_{n'}$ for some $n' \geqslant 2$. We wish to establish that $n = n'$.

Let p be an integer $\geqslant 2$. Two complete horizontal paths of G_p are *neighbours* if they are distinct and if there is an edge of G_p linking one vertex of one path the one vertex of the other. A *border path* is a complete horizontal path having only one neighbour path. A *nonborder* path is one having two neighbour paths. Similar definitions can be given for complete vertical paths. Let Q_p be the $(2p+2)$-graph shown on Fig. 5.

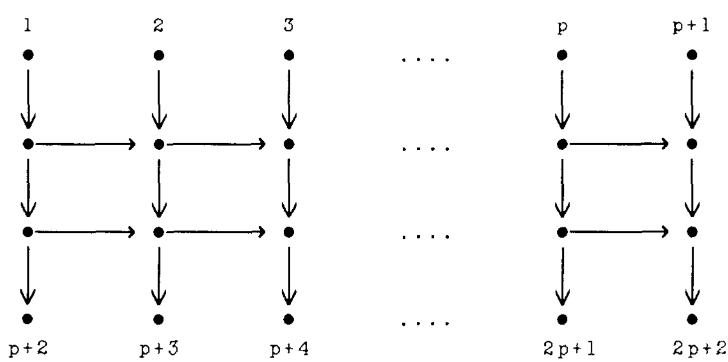

FIGURE 5

CLAIM. *If* $G_n = C[Q_p]$ *for some* $C \in \mathbf{Ctxt}(A)_{2p+2,0}$, *then* $n = p$.

Proof. If $G_n = C[Q_p]$, then G_n has either two nonborder neighbour complete horizontal or vertical paths with p edges. Hence $n = p$. ▮

Let us now go back to H' and G', as in the first part of the proof.

We have established that H' has at least $n+1-m$ complete horizontal paths. Hence, it has at least $m+4$ such paths, since $n \geqslant 2m+3$.

At least $m+2$ of them are nonborder paths. If two of these paths are not neighbour, there is between them, either a complete horizontal path of G_n, totally in H', or at least one vertex of S. Since $\mathbf{Card}(S) \leqslant m$, there are in H' at least two nonborder neighbour paths.

It follows that $G' = C'[Q_n]$ for some C' in $\mathbf{Ctxt}(A)_{2n+2,0}$. Since $G'' \square G'$ is isomorphic to $G_{n'}$, there exists a context C in $\mathbf{Ctxt}(A)_{2n+2,0}$ such that $G_{n'} = C[Q_n]$. It follows from the claim that $n' = n$, and this completes the proof of Lemma (2.17). ▮

The proof of Proposition (2.14) will use another lemma.

(2.18) LEMMA. *Let* E *and* B *be sets, let* f *be a commutative mapping*: $E \times E \to B$. *With* $L \subseteq B$ *we associate an equivalence relation on* E *defined by*: $a \approx a'$ *iff for all* $d \in E$, $f(a, d) \in L \Leftrightarrow f(a', d) \in L$. *Then,* $\approx$ *is finite if there exist* $E_0 \subseteq E$, *and* $C \subseteq B$ *satisfying the following conditions*:

(1) $L - C$ *and* E_0 *are finite,*

(2) *for every* $a, a' \in E$ *such that* $f(a, a') \in C$: *either* $a \in E_0$, $a' \in E - E_0$, *and for all* d *in* E, *if* $f(d, a') \in C$ *then* $f(d, a') = f(a, a')$, *or* $a' \in E_0$, $a \in E - E_0$, *and for all* d *in* E, *if* $f(a, d) \in C$, *then* $f(a, d) = f(a, a')$.

(3) *for every* $b \in B$, *there exist finitely many pairs* (a, a') *in* $E \times E$ *such that* $f(a, a') = b$.

Proof. From condition (2) the condition

$$a \in E - E_0, \text{ and there exists } d \in E_0 \text{ such that } f(a, d) = c \in C \tag{4}$$

defines c in a unique way from a. Let us write $c = g(a)$, where g is the partial mapping: $E \to C$ defined by (4).

We now prove that $\approx$ is finite.

We let $E' := \{a \in E / f(a, d) \notin L \cup C \text{ for all } d \in E\}$. The elements of E' are pairwise equivalent w.r.t. $\approx$; hence they define a single class.

We now let $E'' := E_0 \cup \{a \in E / f(a, d) \in L - C \text{ for some } d \in E\}$. By conditions (1) and (3), the set E'' is finite; hence its elements define finitely many classes.

Let finally $E''' := \{a \in E - E_0 / f(a, d) \in C \text{ for some } d \in E_0\}$. For every $a \in E'''$, let us define $K(a) = \{a' \in E_0 / f(a, a') = g(a)\}$. Note that $K(a)$ is not empty. We now claim that for a, b in E''', if $K(a)=K(b)$, and if $g(a)\in L \Leftrightarrow g(b)\in L$, then $a\approx b$.

Let $d\in E$. Assume that $f(a,d)\in L$. By condition (2), $d\in E_0$ (since $a\in E-E_0$). Furthermore, $g(a)\in L$ and $d\in K(a)$. Since we assume that $g(b)\in L$ and $K(b)=K(a)$, we also have $d\in K(b)$; hence $f(b,d)=g(b)$ and $f(b,d)\in L$. This proves that $a\approx b$.

Since E_0 is finite, there are finitely many sets $K(a)$; hence the elements of E''' define finitely many classes.

Since E is the union of E, E', and E''', we have proved that $\approx$ is finite. ∎

Proof of Proposition (2.14). We first assume that A consists of one symbol a, of type 2. Let $L\subseteq L_G$ and $m\geqslant 1$. By Lemma (2.16), the syntactic equivalence relation $\sim_{L,m}$ is characterized by

$$G\sim_{L,m} G' \text{ iff for every } K\in \mathbf{FG}(A)_m,\ G\square K\in L \Leftrightarrow G'\square K\in L.$$

We shall prove that this equivalence relation is finite.

We apply Lemma (2.18) by letting $E=\mathbf{FG}(A)_m$, $B=\mathbf{FG}(A)_0$, $f=\square_m$, $C=\{G_n/n\geqslant 2m+3\}$, $\approx\ =\ \sim_{L,m}$, $E_0=\{G\in\mathbf{FG}(A)_m/G\square G'\in L_G$ for some $G'\in\mathbf{FG}(A)_m$, and $\mathbf{Card}(\mathbf{V}_G)<m+m^2\}$.

Condition (1) of Lemma (2.18) clearly holds. Condition (2) is proved in Lemma (2.17), and Condition (3) is easy to establish. Hence Lemma (2.18) shows that $\sim_{L,m}$ is finite for $m\geqslant 1$.

Consider finally the special case where $m=0$. Then $G\square G'=G\oplus G'$. Since the grids are connected, if $G\square G'\in L_G$, then one and only one of G and G' is the empty graph $\mathbf{0}$. This means that $\sim_{L,0}$ has exactly two classes: L and $\mathbf{FG}(A)_0-L$. Hence L is recognizable.

If A does not contain any symbol of type 2, but one symbol, say d, of type >2, then for every subset L of L_G, the set $L':=\{G[K_a/a]/G\in L\}$, where K_a is as in the proof of Proposition (2.13), is also recognizable; the above proof can be adapted. ∎

3. Writing Graph Properties in Monadic Second-Order Logic

A graph can be considered as a logical structure with two domains, the set of vertices and the set of edges. Hence logical formulas can express properties of graphs. First-order formulas can express local properties of graphs, as proved by Gaifmann [19]. Monadic second-order formulas written with quantifications over sets of edges and sets of vertices are much more powerful.

We establish that every monadic second-order definable set of graphs is recognizable and that the monadic second-order theory of a context-free set of graphs is decidable. These results do not hold if quantifications over binary relations are also used.

In addition to the usual features of monadic second-order logic, we introduce atomic formulas testing whether the cardinality of a set is equal to n modulo p, where n and p are integers such that $0 \leqslant n < p$ and $p \geqslant 2$. This extension of the usual language is called the *counting monadic second-order logic*. It yields an extension of the result of Doner [16] saying that a subset of $\mathbf{M}(F)$ (considered as a set of trees) is recognizable iff it is definable in monadic second-order logic, to the class of unordered finite trees with no bound on the degrees of nodes. This result is established in Section 5.

(3.1) DEFINITION. **Graphs as logical structures.** In order to express properties of graphs in $\mathbf{FG}(A)_k$ we define the symbols:

$\mathbf{v}$: the *vertex* sort,

$\mathbf{e}$: the *edge* sort,

$\mathbf{s}_i$, a constant of sort $\mathbf{v}$, for each i, $1 \leqslant i \leqslant k$,

$\mathbf{edg}_a$, a predicate symbol of arity $\mathbf{evv} \cdots \mathbf{v}$ (with $\tau(a)$ occurrences of $\mathbf{v}$), for each a, $a \in A$.

With $G \in \mathbf{FG}(A)_k$ we associate the logical structure $|G| = \langle \mathbf{V}_G, \mathbf{E}_G, (\mathbf{s}_{iG})_{i \in [k]}, (\mathbf{edg}_{aG})_{a \in A} \rangle$, where $\mathbf{V}_G$ is the domain of sort $\mathbf{v}$, $\mathbf{E}_G$ is the domain of sort $\mathbf{e}$, $\mathbf{s}_{iG}$ is the ith source of G, and $\mathbf{edg}_{aG}(e, v_1, ..., v_n) = \mathbf{true}$ iff $\mathbf{lab}_G(e) = a$ and $\mathbf{vert}_G(e) = (v_1, ..., v_n)$.

(3.2) DEFINITION. **Counting monadic second-order logic**. We shall build formulas by using *object variables* u, x, y, z, u', ... of sort $\mathbf{v}$ or $\mathbf{e}$, denoting respectively vertices or edges, and *set variables* U, X, Y, Z, U' of sort $\mathbf{v}$ or $\mathbf{e}$, denoting respectively sets of vertices or sets of edges. Since the graphs we consider are finite, the set variables always represent finite sets.

Let $\mathscr{W}$ be a sorted set of variables $\{u, u', ..., U, U', ...\}$ each of them having a sort $\sigma(u)$, $\sigma(u')$, ... $\sigma(U)$, $\sigma(U')$, ... in $\{\mathbf{v}, \mathbf{e}\}$. We denote by $\mathscr{W}_s$ the set $\mathscr{W} \cup \{\mathbf{s}_1, ..., \mathbf{s}_k\}$. (Uppercase letters denote set variables and lowercase letters denote the remaining elements of $\mathscr{W}_s$, i.e., object variables or constants).

The set $\mathscr{A}_{A,k,q}(\mathscr{W})$ of *atomic formulas* consists of:

$$u = u' \quad \text{with} \quad u, u' \in \mathscr{W}_s,\ \sigma(u) = \sigma(u'),$$

$$u \in U \quad \text{with} \quad u, U \in \mathscr{W}_s,\ \sigma(u) = \sigma(U),$$

$$\mathbf{edg}_a(u, u'_1, ..., u'_n) \quad \text{with} \quad u, u'_1, ..., u'_n \in \mathscr{W}_s,$$
$$\sigma(u) = \mathbf{e},\ \sigma(u'_1) = \cdots = \sigma(u'_n) = \mathbf{v},$$

$$\mathbf{card}_{n,p}(U) \quad \text{with} \quad U \in \mathscr{W},\ 0 \leqslant n < p,\ 2 \leqslant p \leqslant q.$$

If U denotes a set X, then

$$\mathbf{card}_{n,p}(U) = \mathbf{true} \quad \text{iff} \quad \mathbf{Card}(X) = n \bmod p.$$

The meaning of the other atomic formulas is clear or has been already defined.

The language of *counting monadic second-order logic* is the set of logical formulas formed with the above atomic formulas together with the Boolean connectives $\wedge$, $\vee$, $\neg$, the object quantifications $\forall u$, $\exists u$ (over vertices or edges), and the set quantifications $\forall U$, $\exists U$ (over sets of vertices or sets of edges).

The language of monadic second-order logic is the set of such formulas that do not use the atomic formulas $\mathbf{card}_{n,p}(U)$.

We denote by $\mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W})$ the set of formulas inductively defined as follows:

$\varphi \in \mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W})$ if $\varphi \in \mathscr{A}_{A,k,q}(\mathscr{W})$,
$\varphi_1 \wedge \varphi_2, \varphi_1 \vee \varphi_2, \neg\varphi_1 \in \mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W})$ if $\varphi_1, \varphi_2 \in \mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W})$,
$\exists u\varphi, \forall u\varphi \in \mathscr{CL}^{(h+1)}_{A,k,q}(\mathscr{W})$ if $\varphi \in \mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W} \cup \{u\})$, $u \notin \mathscr{W}$,
$\exists U\varphi, \forall U\varphi \in \mathscr{CL}^{(h+1)}_{A,k,q}(\mathscr{W})$ if $\varphi \in \mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W} \cup \{U\})$, $U \notin \mathscr{W}$.

The least h such that $\varphi \in \mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W})$ is called the *height* of φ (this integer is the maximal depth of nested quantifications in φ). We let

$$\mathscr{CL}_{A,k,q}(\mathscr{W}) := \bigcup \{\mathscr{CL}^{(h)}_{A,k,q}(\mathscr{W}) / h \geqslant 0\},$$

and

$$\mathscr{CL}_{A,k}(\mathscr{W}) := \bigcup \{\mathscr{CL}_{A,k,q}(\mathscr{W}) / q \geqslant 2\}.$$

In many cases the subscripts A, k, and q can be omitted.

Similar sets of formulas, where the atomic formulas $\mathbf{card}_{n,p}(U)$ are not used, are denoted by $\mathscr{L}_{A,k}(\mathscr{W})$, $\mathscr{L}^{(h)}_{A,k}(\mathscr{W})$, etc. (the parameter q is irrelevant).

(3.3) DEFINITION. **Definability of graph properties.** Let $\mathscr{W}$ be a finite set of variables. Let G be a graph in $\mathbf{FG}(A)_k$. A *$\mathscr{W}$-assignment in G* is a mapping ν associating with every variable in $\mathscr{W}$ a vertex, or an edge, or a set of vertices, or a set of edges of G, depending on its sort and case (lower or upper).

If $\varphi \in \mathscr{CL}_{A,k,q}(\mathscr{W})$, then for each G and ν as above, φ is either true or false in $|G|$ for ν. The classical notation in the former case is $(|G|, \nu) \models \varphi$ and we say that *φ holds in G for ν*. We shall also use $\varphi/_G(\nu)$ as a Boolean value, that is equal to **true** if φ holds in G for ν, and equal to **false** otherwise.

If φ is *closed*, then v disappears, and φ_G is either equal to **true** or to **false**.

A property of graphs in $\mathbf{FG}\ (A)_k$ is *$\mathscr{L}$-definable* (resp. *definable*) if there exists a closed formula φ in $\mathscr{L}_{A,k}$ (resp. in $\mathscr{CL}_{A,k}$) such that G satisfies this property iff φ holds in G. A set $L \subseteq \mathbf{FG}(A)_k$ is *$\mathscr{L}$-definable* (resp. is *definable*) if the membership in L is so. The set of graphs *defined by* φ is the set of graphs G where φ holds, and it is denoted by L_φ.

More generally, a property P of a graph G taking as parameters vertices, edges, sets of vertices, sets of edges, denoted by variables from a finite set $\mathscr{W}$, is *$\mathscr{L}$-definable* (or *definable*) iff there is a formula φ in $\mathscr{L}(\mathscr{W})$ (or in $\mathscr{CL}(\mathscr{W})$) such that, for every $\mathscr{W}$-assignment v in G, φ holds in G for v iff P holds in G for the values $v(x)$, $x \in \mathscr{W}$, of the parameters. For example, we shall see below that the property reading: "there is a simple path from x to y, the set of edges of which is U," where x and y are vertices and U is a set of edges, is $\mathscr{L}$-definable.

In Section 6, we shall prove that $\mathscr{CL}$ is more powerful than $\mathscr{L}$, i.e., that certain graph properties are definable without being $\mathscr{L}$-definable.

We now give a few examples of definable graph properties.

(3.4) EXAMPLE. **Colorability**. Let A consist of symbols of type 2.

The existence of a coloring of the vertices of a graph G in $\mathbf{FG}(A)$, using at most m colors, can be expressed as follows:

> There exist sets of vertices $X_1, ..., X_m$ such that $X_1 \cup \cdots \cup X_m = \mathbf{V}_G$, $X_i \cap X_j = \varnothing$ for $i \neq j$, and the two vertices of any edge do not belong both to X_i for any i.

From this formulation a formula φ in $\mathscr{L}_{A,0}$ can be constructed such that φ holds in G iff G is m-colorable. Hence, the m-colorability of a graph is $\mathscr{L}$-definable.

(3.5) EXAMPLE. **Flows**. Let A be as in Example (3.4), and $G \in \mathbf{FG}(A)_0$. Let $\mathbf{M} = \langle M, +, -, \mathbf{0} \rangle$ be an abelian group.

An $\mathbf{M}$-*flow* on G is a mapping $\theta: \mathbf{E}_G \to M$ such that for every vertex $v \in \mathbf{V}_G$:

$$\Sigma\{\theta(e)/e \in \mathbf{in}(v)\} = \Sigma\{\theta(e)/e \in \mathbf{out}(v)\},$$

where $\mathbf{in}(v) := \{e/\mathbf{vert}_G(e, 2) = v\}$ and $\mathbf{out}(v) := \{e/\mathbf{vert}_G(e, 1) = v\}$.

A flow θ is *nowhere-zero* if $\theta(e) \neq \mathbf{0}$ for all $e \in \mathbf{E}_G$. A *k-flow* is a $\mathbb{Z}$-flow θ such that $-k < \theta(e) < k$ for all $e \in \mathbf{E}_G$.

There exists a formula $\varphi_{n,k}$ in $\mathscr{L}_{A,0}$ such that, for every $G \in \mathbf{FG}(A)_0$ such that $\mathbf{Max}\{\mathbf{Card}\ (\mathbf{in}(v)) + \mathbf{Card}(\mathbf{out}(v))/v \in \mathbf{V}_G\} \leqslant n$:

$$G \models \varphi_{n,k} \quad \text{iff } G \text{ has a nowhere-zero } k\text{-flow}.$$

The limitation to graphs of degree at most n is due to the impossibility to "count in $\mathscr{L}$ beyond fixed integers." In $\mathscr{C}\mathscr{L}$, one can "count modulo p." It follows that the existence of a nowhere-zero $\mathbb{Z}/p\mathbb{Z}$-flow can be expressed in $\mathscr{C}\mathscr{L}_{A,0,p}$ without any limitation on the degree of the considered graphs.

(3.6) DEFINITIONS. **Paths and simple paths.** Let G be a graph. Let v, v' be vertices. A *path from v to v'* is a nonempty sequence of binary (i.e., type 2) edges $e_1, ..., e_n$ such that $\mathbf{vert}_G(e_1, 1) = v$, $\mathbf{vert}_G(e_i, 2) = \mathbf{vert}_G(e_{i+1}, 1)$ for all $i \in [n-1]$, and $\mathbf{vert}_G(e_n, 2) = v'$. (One may have $v = v'$.) Such a path is *simple* if $\mathbf{vert}_G(e_i, 1) \neq \mathbf{vert}_G(e_j, 1)$ for $i \neq j$. (A more general notion of path, that concerns graphs with edges of type larger than 2, can be found in Courcelle [9, 15].)

(3.7) LEMMA. *The transitive closure of an $\mathscr{L}$-definable binary relation is $\mathscr{L}$-definable.*

Proof. (Sketch). Let R be a binary relation on a set D. A subset X of D is *R-closed* if, for every x in X and every pair (x, y) in R, the element y belongs to X. A pair (x, y) belongs to R^+ iff it belongs to the smallest R-closed subset of D containing x. ("Smallest" is taken w.r.t. set inclusion). From this observation it is easy to construct a monadic second-order formula defining R^+ from one defining R. ∎.

(3.8) PROPOSITION. *The following properties of a graph G are $\mathscr{L}$-definable*:

(1) *A given set of edges is the set of edges of a simple path linking two given vertices.*

(2) *G is connected,*

(3) *G has k connected components (for some fixed k),*

(4) *G is strongly connected,*

(5) *G has a Hamiltonian circuit.*

Proof. Let G be a graph. Let U be a set variable of sort $\mathbf{e}$. Let x, y be object variables of sort $\mathbf{v}$. Let φ express that there is in U an edge e such that $\mathbf{vert}_G(e) = (x, y)$. By using Lemma (3.7), one can construct a formula θ in $\mathscr{L}(\{x, y, U\})$ saying that there exists a path from x to y, all edges of which are in U. Then, the formula μ defined as

$$\theta \wedge \forall W[\text{“}W \subseteq U\text{”} \wedge \theta[W/U] \Rightarrow \text{“}W = U\text{”}]$$

says that U is the set of edges of a simple path from x to y. (Formulas can easily be constructed to express what is written inside quotes. We denote by $\theta[W/U]$ the result of the substitution in θ of W for U, after some possibly necessary renamings of bound variables.) This proves (1). The other assertions follow more or less easily.

Consider, for instance, the existence of a Hamiltonian circuit. This property can be written

$\exists U, x, y, e[\mu \wedge x \neq y \wedge$ "e is an edge from y to x" $\wedge$
"every vertex belongs to some edge in U"].

"Forbidden configurations" can be expressed in monadic second-order logic. Some properties of sets of graphs defined by forbidden configurations are investigated in Courcelle [11, 12, 14]. ∎

(3.9) PROPOSITION. *Let A contain at least two symbols, one of which is of type 2. The following properties of a graph G over A are not definable*:

(1) *G has a nontrivial automorphism.*

(2) *G has as many edges labeled by a as by b, where $a, b \in A$.*

The proof uses results to be established below. It will be given at the end of Section 5. (Note that it is easy to express these two properties in second-order logic, by formulas using quantifications on binary relations.)

In order to obtain a relatively short proof for the result of the next section, we define a syntactical variant of the language $\mathscr{CL}$, that we shall denote by $\mathfrak{CL}$. This new language has a simpler syntax than $\mathscr{CL}$, but the formulas are not easily readable.

(3.10) DEFINITION. **The language** $\mathfrak{CL}$. The language $\mathfrak{CL}$ is a variant of $\mathscr{CL}$ using set variables only (still denoted by uppercase letters), of the two possible sorts $\mathbf{v}$ and $\mathbf{e}$.

Let $\mathscr{W}$ be a $\{\mathbf{v}, \mathbf{e}\}$-sorted set of set variables, U, U', V, W, ... Let $k \in \mathbb{N}$. A *term of sort* $\mathbf{e}$ is either a variable U, of sort $\mathbf{e}$, or the constant $\boldsymbol{\phi}$. A *term of sort* $\mathbf{v}$ is an expression of the two possible forms $\mathbf{S}_I(\boldsymbol{\phi})$ and $\mathbf{S}_I(U)$, where U is a variable of sort $\mathbf{v}$, and I is a subset of $[k]$. The set of these terms is denoted by $\mathfrak{Z}_k(\mathscr{W})$.

For every $\mathscr{W}$-assignment v in a graph $G = \langle \mathbf{V}_G, \mathbf{E}_G, \mathbf{lab}_G, \mathbf{vert}_G, \mathbf{src}_G \rangle$ of type k, we state that:

a term of the form $\boldsymbol{\phi}$ denotes $\varnothing$,

a term of the form U denotes $v(U)$,

a term of the form $\mathbf{S}_I(\boldsymbol{\phi})$ denotes $\{\mathbf{src}_G(i)/i \in I\}$,

a term of the form $\mathbf{S}_I(U)$ denotes $v(U) \cup \{\mathbf{src}_G(i)/i \in I\}$.

We let $v(X)$ be the set denoted by a term X. We shall use $\boldsymbol{\phi}$ and U as shorthands for $\mathbf{S}_\phi(\boldsymbol{\phi})$ and $\mathbf{S}_\phi(U)$, respectively. Hence $\boldsymbol{\phi}$ is a constant of both sorts $\mathbf{v}$ and $\mathbf{e}$, but this will not create any difficulty.

The set $\mathfrak{U}_{A,k,q}(\mathscr{W})$ consists of the following atomic formulas, where by a term, we mean an element of $\mathfrak{Z}_k(\mathscr{W})$:

(1) $X \subseteq Y$ for terms X, Y of the same sort,

(2) $\mathbf{sgl}(U)$ for a variable U of sort $\mathbf{e}$,

(3) $\mathbf{edg}_a(U, X_1, ..., X_n)$ for a variable U in $\mathscr{W}$ of sort $\mathbf{e}$, and terms $X_1, ..., X_n$ of sort $\mathbf{v}$, where $a \in A$ and $n = \tau(a)$,

(4) $\mathbf{card}_{p,r}(X)$ for $0 \leqslant p < r \leqslant q$, $r \geqslant 2$, and a term X.

For every $\mathscr{W}$-assignment v in a graph G, these formulas hold true iff, one has, respectively,

(1) $v(X) \subseteq v(Y)$,

(2) $v(U)$ is a singleton,

(3) $v(U)$ is a singleton $\{e\}$, $\mathbf{lab}_G(e) = a$ and $\mathbf{vert}_G(e) \in v(X_1) \times \cdots \times v(X_n)$

(4) $\mathbf{Card}(v(X)) = p \ \mathbf{mod}\ r$.

Finally, we denote by $\mathfrak{CL}_{A,k,q}(\mathscr{W}')$ the set of formulas formed from $\mathfrak{U}_{A,k,q}(\mathscr{W})$ by Boolean combinations and existential quantifications (over set variables), having their free variables in $\mathscr{W}'$.

The simplified notations $\mathfrak{CL}(\mathscr{W})$, $\mathfrak{CL}_A$, etc... will be used similarily, as for $\mathscr{CL}$. The set $\mathfrak{CL}^{(h)}_{A,k,q}(\mathscr{W})$ of formulas with at most h levels of nested quantifications is defined as for $\mathscr{CL}$.

The two languages $\mathscr{CL}$ and $\mathfrak{CL}$ have the same expressive power as shown by the following lemma. In its statement, we use the following notations.

If $\mathscr{W}$ a set of object and set variables $\{u, v, w, ... U, V, W, ...\}$, we denote by $\bar{\mathscr{W}}$ the set of set variables $\{\bar{u}, \bar{v}, \bar{w}, ..., U, V, W, ...\}$ (where $\bar{u}, \bar{v}, \bar{w}, ...$ are new set variables associated with $u, v, w, ...$).

If v is a $\mathscr{W}$-assignment in a graph G, then we denote by $\bar{v}$ the $\bar{\mathscr{W}}$-assignment such that $\bar{v}(U) = v(U)$, $\bar{v}(\bar{u}) = \{v(u)\}$ for U, u in $\mathscr{W}$.

(3.11) Lemma. (1) *Let $\varphi \in \mathscr{CL}(\mathscr{W})$. One can construct a formula $\bar{\varphi}$ in $\mathfrak{CL}(\bar{\mathscr{W}})$ such that, for every graph G, and every $\mathscr{W}$-assignment v in G*:

$$(G, \bar{v}) \models \bar{\varphi} \quad \text{iff} \quad (G, v) \models \varphi.$$

(2) *Conversely, if $\mathscr{W}$ consists of set variables, and $\psi \in \mathfrak{CL}(\mathscr{W})$, one can construct ψ' in $\mathscr{CL}(\mathscr{W})$ such that, for every $\mathscr{W}$-assignment v in G,*

$$(G, v) \models \psi' \quad \text{iff} \quad (G, v) \models \psi.$$

Proof (Sketch). (2) Each formula in $\mathfrak{U}_{A,k,q}(\mathscr{W})$ can be easily translated into a formula in $\mathscr{CL}_{A,k,q}(\mathscr{W})$. The result follows immediately.

(1) Every object variable $u, v, \ldots$ of φ can be represented by the set variable $\bar{u}, \bar{v}, \ldots$, subject to the additional condition that $\bar{u}$ denotes a singleton.

Here are the main steps of the translation of φ into $\bar{\varphi}$:

$$\overline{\exists u\psi} \text{ is } \exists\bar{u}[\mathbf{sgl}(\bar{u}) \wedge \bar{\psi}].$$

If u is of sort $\mathbf{v}$ then $\mathbf{sgl}(\bar{u})$ is not an atomic formula, but stands for the following formula (expressing that $\bar{u}$ denotes a singleton):

$$\exists U_1[U_1 \subseteq \bar{u} \wedge \neg(\bar{u} \subseteq U_1)] \wedge \neg[\exists U_1, \exists U_2(U_1 \subseteq U_2 \wedge U_2 \subseteq \bar{u} \wedge \neg(U_2 \subseteq U_1) \wedge \neg(\bar{u} \subseteq U_2)].$$

Then

$$\overline{\forall u\psi} \text{ is } \neg\exists\bar{u}\,[\mathbf{sgl}(\bar{u}) \wedge \neg\bar{\psi}].$$

The translations of the atomic formulas are

$$\overline{u=v} \quad \text{is} \quad \bar{u} \subseteq \bar{v} \wedge \bar{v} \subseteq \bar{u},$$

$$\overline{u \in U} \quad \text{is} \quad \bar{u} \subseteq U,$$

$$\overline{\mathbf{edg}_a(w, v_1, \ldots, v_n)} \quad \text{is} \quad \mathbf{edg}_a(\bar{w}, \bar{v}_1, \ldots, \bar{v}_n),$$

$$\overline{\mathbf{card}_{p,q}(U)} \quad \text{is} \quad \mathbf{card}_{p,q}(U),$$

where u, w, $v_1, \ldots, v_n$ are object variables of the appropriate sorts. If any of these variables, say u, is the constant $\mathbf{s}_i$, then $\bar{u}$ is the term $\mathbf{S}_{\{i\}}(\boldsymbol{\phi})$. We omit the remaining definitions and verifications. ∎

4. The Main Theorem

We establish that every definable set of graphs is recognizable. By Lemma (3.11), every definable set is defined by a closed formula in $\mathfrak{CL}$. In our proof, we shall use this syntactical variant of $\mathscr{CL}$.

(4.1) Definition. **Tautological equivalence**. Two formulas φ and φ' of $\mathfrak{CL}(\mathscr{W})$ are *tautologically equivalent* if φ can be transformed into φ' by finitely many renamings of bound variables, and applications of the Boolean laws on $\vee$, $\wedge$, $\neg$, **true, false** like $\varphi \vee \varphi \equiv \varphi$ and $\neg\neg\varphi \equiv \varphi$.

Hence in particular, if Φ is finite, there are finitely many tautologically inequivalent Boolean combinations of formulas of Φ.

It is clear that for every two tautologically equivalent formulas φ and φ' in $\mathfrak{CL}_{A,k,p}(\mathscr{W})$, for every graph $G \in \mathbf{FG}(A)_k$, for every $\mathscr{W}$-assignment v in G:

$$(G, v) \models \varphi \Leftrightarrow (G, v) \models \varphi'.$$

For every subset Φ of $\mathfrak{CL}$, we denote by $\bar{\Phi}$ the quotient set of Φ w.r.t. tautological equivalence.

In the following lemma we fix a finite ranked alphabet A, an integer q, and we denote by $\mathfrak{CL}_k^{(h)}(\mathscr{W})$ the set $\mathfrak{CL}_{A,k,q}^{(h)}(\mathscr{W})$.

(4.2) LEMMA. *For every k and h in $\mathbb{N}$, for every finite set of variables $\mathscr{W}$, the set $\overline{\mathfrak{CL}_k^{(h)}(\mathscr{W})}$ is finite.*

Proof. By induction on h. Let $h = 0$. It is clear that $\mathfrak{A}_{A,k,q}(\mathscr{W})$ is finite. Since $\mathfrak{CL}_k^{(0)}(\mathscr{W})$ is the set of Boolean combinations of formulas in $\mathfrak{A}_{A,k,q}(\mathscr{W})$, it is finite, up to tautological equivalence.

Let $h = h' + 1$. Since $\mathfrak{CL}_k^{(h')}(\mathscr{W} \cup \{U\})$ is finite up to tautological equivalence, so is the set of formulas in $\mathfrak{CL}_k^{(h)}(\mathscr{W})$ that are of the forms $\exists U\varphi$, and so is $\mathfrak{CL}_k^{(h)}(\mathscr{W})$ that is the set of Boolean combinations of formulas of this latter form. ∎

Since one can decide whether two formulas are tautologicaly equivalent, the finite set $\overline{\mathfrak{CL}_k^{(h)}(\mathscr{W})}$ can be effectively constructed.

We now make $(\overline{\mathfrak{CL}}_k)_{k \in N}$ into a family of predicates. For every φ in $\overline{\mathfrak{CL}}_k$, we let k be the sort of φ, and $\hat{\varphi}$ be the predicate on $\mathbf{FG}(A)_k$ defined by:

$$\hat{\varphi}(G) = \mathbf{true} :\Leftrightarrow G \models \varphi.$$

We shall establish the following result:

(4.3) PROPOSITION. *For every $h \geqslant 0$, the family of predicates $\Phi^{(h)} := \{\hat{\varphi} / \varphi \in \overline{\mathfrak{CL}_k^{(h)}}, k \geqslant 0\}$ is effectively locally-finite and $\mathbf{H}_A$-inductive.*

The main result of this paper is an immediate consequence of this proposition. We state it immediately.

(4.4.) THEOREM. *Every definable subset of $\mathbf{FG}(A)_k$ is an effectively given recognizable set of graphs.*

Proof. Let $L \subseteq \mathbf{FG}(A)_k$ be defined by a formula φ in $\mathfrak{CL}_{A,k,q}$. There exist $h \geqslant 0$ and q such that $\varphi \in \mathfrak{CL}_{A,k,q}^{(h)}$. The set A and the integer q being fixed, we can apply Propositions (4.3), and (1.5). They yield that $L = L_\varphi$ is $\mathbf{FG}(A)$-recognizable.

Since the family of predicates $\Phi^{(h)}$ is effectively locally-finite and $\mathbf{H}_A$-inductive, the set L is effectively recognizable by Proposition (1.6). ∎

The proof of Proposition (4.3) is based on three lemmas, stating that the family of predicates $\Phi^{(h)}$ is inductive w.r.t. the sets of operations $\{\oplus_{n,m}/n, m \geqslant 0\}$, $\{\theta_{i,j,n}/1 \leqslant i, j \leqslant n, \ n \geqslant 1\}$, and $\{\sigma_{\alpha,n,p}/\alpha\colon [p] \to [n], n, p \geqslant 0\}$, respectively.

These lemmas will be proved by induction on formulas with free variables, in order to handle quantifications. Hence, we need a few more technical notations.

Let $\mathscr{W}$ be a finite set of set variables. If v' is a $\mathscr{W}$-assignment in $G' \in \mathbf{FG}(A)_{k'}$, if v'' is a $\mathscr{W}$-assignment in $G'' \in \mathbf{FG}(A)_{k''}$, then, we denote by $v := v' \cup v''$ the $\mathscr{W}$-assignment in $G' \oplus G''$ defined by $v(U) = v'(U) \cup v''(U)$ for all U in $\mathscr{W}$.

Letting $k = k' + k''$, we have

(4.5) Lemma. *Given φ in $\mathfrak{CL}_k^{(h)}(\mathscr{W})$, one can construct a finite sequence of formulas $\varphi'_1, \dots, \varphi'_n$ in $\mathfrak{CL}_{k'}^{(h)}(\mathscr{W})$, a finite sequence of formulas $\varphi''_1, \dots, \varphi''_m$ in $\mathfrak{CL}_{k''}^{(h)}(\mathscr{W})$, and an $(n+m)$-place Boolean expression B such that, for every k'-graph G', for every k''-graph G'', for every $\mathscr{W}$-assignment v' in G', for every $\mathscr{W}$-assignment v'' in G'':*

$$\varphi_{G' \oplus G''}(v' \cup v'') = B[\varphi'_{1G'}(v'), \dots, \varphi'_{nG'}(v'), \varphi''_{1G''}(v''), \dots, \varphi''_{mG''}(v'')]. \quad (*)$$

Proof. The proof is by induction on the structure of φ.

First Case. φ is atomic. The various possibilities are as follows: (In each case, we write the equality corresponding to $(*)$ of the statement.)

(1) If φ is $X \subseteq Y$ for terms X and Y of sort $\mathbf{e}$, then

$$\varphi_{G' \oplus G''}(v' \cup v'') = \varphi_{G'}(v') \wedge \varphi_{G''}(v'').$$

(1′) If φ is $\mathbf{S}_I(U) \subseteq \mathbf{S}_J(U')$ with U, U' in $\mathscr{W} \cup \{\boldsymbol{\phi}\}$ of sort $\mathbf{v}$, then

$$\varphi_{G' \oplus G''}(v' \cup v'') = \varphi'_{G'}(v') \wedge \varphi''_{G''}(v''),$$

where φ' is $\mathbf{S}_{I'}(U) \subseteq \mathbf{S}_{J'}(U')$,
with $I' := I \cap [k']$ and $J' := J \cap [k']$,
and φ'' is: $\mathbf{S}_{I''}(U) \subseteq \mathbf{S}_{J''}(U')$,
with $I'' := \{i \in [k'']/i + k' \in I\}$ and $J'' := \{i \in [k'']/i + k' \in J\}$.

(2) If φ is $\mathbf{edg}_a(U, \mathbf{S}_I(X), S_J(Y))$, then

$$\begin{aligned}\varphi_{G' \oplus G''}(v' \cup v'') = {} & [\mathbf{edg}_a(U, \mathbf{S}_{I'}(X), \mathbf{S}_{J'}(Y))_{G'}(v') \wedge (U \subseteq \boldsymbol{\phi})_{G''}(v'')] \\ & \vee [(U \subseteq \boldsymbol{\phi})_{G'}(v') \wedge \mathbf{edg}_a(U, \mathbf{S}_{I''}(X), \mathbf{S}_{J''}(Y))_{G''}(v'')],\end{aligned}$$

where I', I'', J', J'' are as in (1′).

(In order to simplify the writing, we have assumed that the symbol a is of type 2; the general case is similar.)

(3) If φ is $\mathbf{sgl}(U)$, where U is a variable of sort $\mathbf{e}$, then

$$\varphi_{G'\oplus G''}(v'\cup v'') = [\mathbf{sgl}(U)_{G'}(v') \wedge (U\subseteq \boldsymbol{\phi})_{G''}(v'')] \vee [(U\subseteq \boldsymbol{\phi})_{G'}(v') \wedge \mathbf{sgl}(U)_{G''}(v'')].$$

(4) If φ is $\mathbf{card}_{p,q}(U)$, then

$$\varphi_{G'\oplus G''}(v'\cup v'') = \bigvee\!\!\bigvee \{\psi_{r,q,G'}(v') \wedge \psi_{s,q\,G''}(v'')/0\leqslant r<q,\ 0\leqslant s<q,\ r+s=p \ \mathbf{mod}\ q\},$$

where $\psi_{x,y}$ is the atomic formula $\mathbf{card}_{x,y}(U)$.

In order to understand case (4), one should remember that $v'(U)\cap v''(U)=\varnothing$. The validity of the stated equality follows, since for disjoint sets X and Y, $\mathbf{Card}(X\cup Y)=\mathbf{Card}(X)+\mathbf{Card}(Y)$. The verifications of the other cases are easy from the definitions.

Second Case. φ is $\neg\psi$, or $\psi_1\wedge\psi_2$, or $\psi_1\vee\psi_2$. We only consider the case where φ is $\psi_1\wedge\psi_2$. The other ones are similar. We can assume that we have constructed $B_i[\psi'_{i,1},...,\psi''_{i,1},...]$, such that $\psi'_{i,1},...\in\mathfrak{CL}^{(h)}_{k'}(\mathscr{W})$, $\psi''_{i,1},...\in\mathfrak{CL}^{(h)}_{k''}(\mathscr{W})$ and

$$\psi_{iG'\oplus G''}(v'\cup v'') = B_i[\psi'_{i,1\,G'}(v'),...,\psi''_{i,1\,G''}(v''),...]$$

for $i=1, 2$. Then, clearly,

$$\varphi_{G'\oplus G''}(v'\cup v'') = B_1[\psi'_{1,1\,G'}(v'),...] \wedge B_2[\psi''_{2,1\,G''}(v'),...].$$

This gives the desired decomposition of φ.

Third Case. φ is $\exists U\psi$, with ψ in $\mathfrak{CL}^{(h-1)}_{k}(\mathscr{W}\cup\{U\})$, and U not in $\mathscr{W}$. Without loss of generality, we can assume that U is of sort $\mathbf{v}$.

If $X'\subseteq\mathbf{V}_G$ and v' is a $\mathscr{W}$-assignment in G', we denote by $v'_{X'}$ its extension into the $(\mathscr{W}\cup\{U\})$-assignment in G', defined by taking X' as value of U, and similarly for $v''_{X''}$, if $X''\subseteq\mathbf{V}_{G''}$. Hence

$$(\exists U\psi)_{G'\oplus G''}(v'\cup v'') = \mathbf{true}$$

iff

$$\psi_{G'\oplus G''}(v'_{X'}\cup v''_{X''}) = \mathbf{true}$$

for some subsets X' and X'', of $\mathbf{V}_{G'}$ and $\mathbf{V}_{G''}$.

By the induction hypothesis, one can assume that one has defined $B[\psi'_1, ..., \psi''_1, ...]$ such that $\psi'_1, ... \in \mathfrak{CL}_{k'}^{(h-1)}(\mathscr{W} \cup \{U\})$, $\psi''_1, ... \in \mathfrak{CL}_{k''}^{(h-1)}(\mathscr{W} \cup \{U\})$, and

$$\psi_{G' \oplus G''}(v'_{X'} \cup v''_{X''}) = B[\psi'_{1G'}(v'_{X'}), ..., \psi''_{1G''}(v''_{X''}), ...].$$

We can write the right-hand side of this equality as a disjunction $C_1[\cdots] \vee \cdots \vee C_n[\cdots]$ of formulas $C_i[\cdots]$ of the form

$$\psi'_{i,1G'}(v'_{X'}) \wedge \psi'_{i,2G'}(v'_{X'}) \wedge \cdots \wedge \psi''_{i,1G''}(v''_{X''}) \wedge \psi''_{i,2G''}(v''_{X''}) \wedge \cdots,$$

where each $\psi'_{i,j}$ is either a formula, or the negation of a formula in $\{\psi'_1, ...\}$, and similarly for $\psi''_{i,j}$. Hence

$$\varphi_{G' \oplus G''}(v) = \mathbf{true}$$

iff there exist $X' \subseteq \mathbf{V}_{G'}$, and $X'' \subseteq \mathbf{V}_{G''}$ such that

$$C_1[\cdots] \vee \cdots \vee C_n[\cdots] = \mathbf{true}.$$

The ith element of this disjunction is equivalent to

$$\exists X' \subseteq \mathbf{V}_{G'}[\psi'_{i,1G'}(v'_{X'}) \wedge \cdots] \wedge \exists X'' \subseteq \mathbf{V}_{G''}[\psi''_{i,1G''}(v''_{X''}) \wedge \cdots],$$

i.e., to

$$(\exists U \theta'_i)_{G'}(v') \wedge (\exists U\, \theta''_i)_{G''}(v''),$$

where θ'_i is the formula $\psi'_{i,1} \wedge \psi'_{i,2} \wedge \cdots$ (in $\mathfrak{CL}_{k'}^{(h-1)}(\mathscr{W} \cup \{U\})$ and θ''_i is defined similarly as $\psi''_{i,1} \wedge \psi''_{i,2} \wedge \cdots$ (in $\mathfrak{CL}_{k''}^{(h-1)})(\mathscr{W} \cup \{U\})$.

Hence $\varphi_{G' \oplus G''}(v)$ is equivalent to a Boolean combination of formulas in $\mathfrak{CL}_{k'}^{(h)}(\mathscr{W}) \cup \mathfrak{CL}_{k''}^{(h)}(\mathscr{W})$, expressing properties of v' in G', and of v'' in G''. ∎

The next lemma expresses the validity in $\theta_{i,j}(H)$ of a formula φ, in terms of the validity in H of a formula φ' constructed from φ. Let us recall that the graph $G = \theta_{i,j}(H)$ is the result of the fusion of the two vertices $\mathbf{src}_H(i)$ and $\mathbf{src}_H(j)$. Formally, it is defined by a surjective mapping $f: \mathbf{V}_H \to \mathbf{V}_G$, where $\mathbf{V}_G = \mathbf{V}_H/\sim$, and $\sim$ is the equivalence relation on $\mathbf{V}_H$ generated by the pair $(\mathbf{src}_H(i), \mathbf{src}_H(j))$.

For every $\mathscr{W}$-assignment v' in H, we define the assignment $v = \theta_{i,j}(v')$ in G by letting

$$v(U) := v'(U) \qquad \text{for } U \text{ of sort } \mathbf{e},$$

$$v(U) := f(v'(U)) = \{f(v)/v \in v'(U)\} \qquad \text{for } U \text{ of sort } \mathbf{v}.$$

As in the proof of the last lemma, we use v_X to denote the extension of a $\mathscr{W}$-assignment v into a $\mathscr{W} \cup \{U\}$-assignment, in such a way that $v_X(U) = X$ (where U is not in $\mathscr{W}$).

(4.6) LEMMA. *Given $\varphi \in \mathfrak{CL}_k^{(h)}(\mathscr{W})$ and $i, j \in [k]$, one can construct a formula $\varphi' \in \mathfrak{CL}_k^{(h)}(\mathscr{W})$ such that, for every $H \in \mathbf{FG}(A)_k$, for every $\mathscr{W}$-assignment v' in H, if $G = \theta_{i,j}(H)$ and $v = \theta_{i,j}(v')$, then*

$$\varphi_G(v) = \varphi'_H(v').$$

Proof. By induction on the structure of φ.

First Case. φ is atomic.

(1) If φ is $X \subseteq Y$, or is $\mathbf{card}_{p,q}(U)$ for X, Y, U of sort $\mathbf{e}$, then, we let φ' be φ, and we have

$$\varphi_G(v) = \varphi'_H(v').$$

(2) If φ is $\mathbf{S}_I(X) \subseteq \mathbf{S}_J(Y)$, then, we let φ' be

$$\mathbf{S}_I(X) \subseteq \mathbf{S}_{J'}(Y) \vee (\mathbf{S}_I(X) \subseteq \mathbf{S}_{J''}(Y) \wedge \rho_{i,j}(Y)),$$

where

$$J'' := J \cup \{i, j\},$$
$$J' := \textbf{if } i \text{ or } j \text{ is in } J \textbf{ then } J'' \textbf{ else } J,$$

and $\rho_{i,j}(Y)$ is the formula: $\mathbf{S}_{\{i\}}(Y) \subseteq Y \vee \mathbf{S}_{\{j\}}(Y) \subseteq Y$ expressing that Y contains at least one of the two sources of H that are being fused.

(3) If φ is $\mathbf{edg}_a(U, \mathbf{S}_I(X), \mathbf{S}_J(Y))$, then φ' is the disjunction of the following four formulas:

$$\mathbf{edg}_a(U, \mathbf{S}_{I'}(X), \mathbf{S}_{J'}(Y))$$
$$\mathbf{edg}_a(U, \mathbf{S}_{I'}(X), \mathbf{S}_{J''}(Y)) \wedge \rho_{i,j}(Y)$$
$$\mathbf{edg}_a(U, \mathbf{S}_{I''}(X), \mathbf{S}_{J'}(Y)) \wedge \rho_{i,j}(X)$$
$$\mathbf{edg}_a(U, \mathbf{S}_{I''}(X), \mathbf{S}_{J''}(Y)) \wedge \rho_{i,j}(X) \wedge \rho_{i,j}(Y),$$

(We have only considered the case of a symbol a of type 2; for a symbol of type n, φ' is a disjunction of 2^n formulas, that are straightforward to write; J' and J'' are as in (2); I' and I'' are similar.)

(4) If φ is $\mathbf{card}_{p,q}(X)$ for a term X of sort $\mathbf{v}$, then φ' is

$$(\mathbf{card}_{p+1,q}(X) \wedge \psi) \vee (\mathbf{card}_{p,q}(X) \wedge \neg \psi),$$

where ψ is the formula

$$\mathbf{S}_{\{i\}}(\boldsymbol{\phi}) \subseteq X \wedge \mathbf{S}_{\{j\}}(\boldsymbol{\phi}) \subseteq X \wedge \neg [\mathbf{S}_{\{i\}}(\boldsymbol{\phi}) \subseteq \mathbf{S}_{\{j\}}(\boldsymbol{\phi})],$$

expressing that the ith and jth sources are two distinct vertices of X.

Second Case. φ is $\varphi_1 \wedge \varphi_2$ or $\varphi_1 \vee \varphi_2$ or $\neg \varphi_1$. Assuming that φ'_1, φ'_2 have already been constructed, then one takes for φ' respectively $\varphi'_1 \wedge \varphi'_2$ or $\varphi'_1 \vee \varphi'_2$, or $\neg \varphi'_1$.

Third Case. φ is $\exists U \psi$, where $\psi \in \mathfrak{CL}_k^{(h-1)}(\mathscr{W} \cup \{U\})$, U not in $\mathscr{W}$. We assume that U is of sort $\mathbf{v}$. Let ψ' be obtained from ψ. Then

$$\begin{aligned} \varphi_G(v) = \mathbf{true} \quad & \text{iff} \quad \psi_G(v_X) = \mathbf{true} \text{ for some } X \subseteq \mathbf{V}_G, \\ & \text{iff} \quad \psi'_H(v'_Y) = \mathbf{true} \text{ for some } Y \subseteq \mathbf{V}_H, \\ & \text{iff} \quad (\exists U \psi')_H(v') = \mathbf{true}, \end{aligned}$$

since for all $Y \subseteq \mathbf{V}_H$, $\theta_{i,j}(v'_Y) = v_{f(Y)}$ and f is surjective. Hence φ' is the formula $\exists U \psi'$.

This completes the proof of Lemma (4.6). ∎

The next lemma deals similarily with the source redefinition map $\sigma_\alpha : \mathbf{FG}(A)_n \to \mathbf{FG}(A)_k$.

(4.7) LEMMA. *Given $\varphi \in \mathfrak{CL}_k^{(h)}(\mathscr{W})$, and α: $[k] \to [n]$, one can construct a formula $\varphi' \in \mathfrak{CL}_n^{(h)}(\mathscr{W})$ such that, for every $H \in \mathbf{FG}(A)_n$, for every $\mathscr{W}$-assignment v in H, if $G = \sigma_\alpha(H)$ then $\varphi_G(v) = \varphi'_H(v)$.*

Proof. For every formula φ, we let φ' be the result of the simultaneous substitution in φ of $\mathbf{S}_{\alpha(I)}(X)$, for every occurrence of $\mathbf{S}_I(X)$ (where X is either a variable or $\boldsymbol{\phi}$), for all $I \subseteq [k]$. It is easy to see that

$$\varphi_G(v) = \varphi'_H(v).$$

It is clear that $\varphi' \in \mathfrak{CL}_n^{(h)}(\mathscr{W})$. ∎

Proof of Proposition (4.3).

For $\mathscr{W} = \varnothing$, Lemmas (4.5) and (4.7) yield that $\Phi^{(h)}$ is inductive with respect to $\{\oplus_{m,n}/m, n \in \mathbb{N}\} \cup \{\sigma_{\alpha,n,p}/n, p \in \mathbb{N}, \alpha:[p] \to [n]\}$. Lemma (4.6) yields similarly that $\Phi^{(h)}$ is inductive w.r.t. $\{\theta_{i,j,n}/n \geqslant 0, i, j \in [n]\}$. But every operation $\theta_{\delta,n}$ can be written as a composition of at most n operations of the form $\theta_{i,j,n}$, with $1 \leqslant i, j, \leqslant n$. The appropriate extension of Lemma (4.6) holds and yields the desired result.

Finally, $\mathbf{H}_A$ also contains constants $\mathbf{0}$, $\mathbf{1}$, and a for all $a \in A$. These constants define finite graphs. For every closed formula φ and every finite graph G, one can decide whether φ holds in G. This means that one can determine whether $\varphi_G = \mathbf{true}$, or $\varphi_G = \mathbf{false}$. This gives (trivial) decompositions for formula φ w.r.t. the constants $\mathbf{0}$, $\mathbf{1}$, and a, $a \in A$. ∎

Our main theorem has the following consequences.

(4.8) COROLLARY. *Let $L \subseteq \mathbf{FG}(A)_k$ be a context-free set of graphs, and let $\varphi \in \mathscr{CL}_{A,k}$.*

(1) *The set $L \cap L_\varphi$ is context-free and a context-free graph-grammar can be constructed to generate it.*

(2) *The following properties are decidable:*

(2.1) *φ holds in all graphs G in L* (i.e., $L \subseteq L_\varphi$),

(2.2) *φ holds in some graph G in L* (i.e., $L \cap L_\varphi \neq \varnothing$).

Proof. By Proposition (2.11), context-free graph-grammars and systems of equations define the same sets of graphs. In this proof, it is convenient to describe context-free sets of graphs by systems of equations. The set L_φ is an effectively given recognizable set of graphs. Hence, by Proposition (1.8), one can construct a system of equations defining $L \cap L_\varphi$ (hence also a context-free graph-grammar). One can test whether $L \cap L_\varphi = \varnothing$. One can also test whether $L \cap L_{\neg\varphi} = \varnothing$, i.e., whether $L \subseteq L_\varphi$. ∎

As an application, we get that the set of planar (or connected, or Hamiltonian) graphs belonging to a given context-free set, is context-free, and that a grammar can be constructed to generate it. One can also decide whether a context-free set of graphs contains a planar (or a connected, or a Hamiltonian) graph.

(4.9) *Remarks.* The algorithms doing these things, that one derives from Corollary (4.8), are "uniform" in terms of the graph properties. This uniformity is a source of inefficiency: the grammar generating $L \cap L_\varphi$, that one can contruct in this way has approximately $m.\,\mathbf{exp}^{h+2}(b.h^n)$ nonterminals, where m is the number of nonterminals of the grammar generating L, and the constants b and n depend polynomialy on $\mathbf{Card}(A)$, $\mathbf{Max}\{\tau(a)/a \in A\}$, k, and q, where $\varphi \in \mathfrak{CL}_{A,k,q}$, and h is the height of φ. (We denote 2^x by $\mathbf{exp}(x)$ for $x \in \mathbb{N}$.)

But Corollary (4.8) provides us with an easily testable decidability criterion. Furthermore, the notion of an inductive set of predicates yields a methodology for finding efficient algorithms. If a context-free set L as in Corollary (4.8), is given by a system of m equations over $\mathbf{H}_A^{[k]}$, if φ belongs to $\mathfrak{CL}_{A,n,q}^{(h)}$, then, in order to contruct a system of equations (or a grammar) defining $L \cap L_\varphi$, it suffices to find a finite $\mathbf{H}_A^{[k]}$-inductive family P of predicates containing $\hat{\varphi}$. The number of nonterminals of the context-free grammar obtained in this way is then at most $m.\mathbf{exp}(p)$, where $p = \mathbf{card}(P)$. This number p can be much smaller than $\mathbf{Card}(\bigcup\{\mathfrak{CL}_{A,i,q}^{(h)}/i \leqslant k\})$, that is precisely the cardinality of the family of predicates used in the proofs of Theorem (4.4) and of Corollary (4.8). This idea is exploited by Lengauer and Wanke in [24]. ∎

We now review a few applications to the logic of graphs and to the complexity of certain graph decision problems.

Sets of Graphs Having a Decidable Monadic Theory

Let $M \subseteq \mathbf{FG}(A)_k$ be a set of graphs. The *monadic (second-order) theory of M* is the set of formulas $\mathbf{th}(M) := \{\varphi \in \mathscr{CL}_{A,k}/G \models \varphi$ for all G in $M\}$.

(4.10) COROLLARY. *The following sets of graphs have a decidable monadic theory*:

(1) *The set of k-graphs of width at most m, for every k and $m \geqslant k$,*

(2) *Every context-free set of graphs.*

Proof. The set of k-graphs of width at most m is context-free by Proposition (2.11). The two results follow immediately from Corollary (4.8), assertion (2.1). ∎

One cannot hope to break the limitation to sets of graphs of bounded width, because of the following results:

(4.11) PROPOSITION. (1) *The first-order theory of the set of all finite graphs is undecidable.*

(2) *The monadic theory of a set of graphs L of unbounded width is undecidable.*

Proof. Result (1) is known from Trahtenbrot [30]. It follows in particular that the monadic theory of the set of all finite graphs is undecidable. Result (2) is essentially due to Seese [27, 28]. Technical details can be found in Courcelle [14]. ∎

On the other hand, decidability results can be obtained for noncontext-free sets of graphs of bounded width, defined by certain controlled context-free graph-grammars.

Controlled Grammars

Let Γ be a context-free graph grammar, let $L(\Gamma, C)$ be the set of graphs in $L(\Gamma)$ having a derivation tree in C, where C is a given set of trees. We call C a *control set*, and we say that (Γ, C) is a *controlled* (*context-free*) *graph-grammar*. We shall prove that certain controlled graph-grammars generate sets of graphs having a decidable monadic theory.

If Γ is a context-free (word) grammar, and if C is the set of trees having all their branches of the same length, then $L(\Gamma, C)$ is an EOL language. (Rozenberg and Salomaa [26]). Every EOL language can be considered as a subset of a context-free language, defined by such a control set.

We now define the derivation trees of a context-free graph grammar, $\Gamma = \langle A, U, P\rangle$. We first turn P into a signature. Let p in P name rule $u \to e$. Let $(u_{i_1}, ..., u_{i_k})$ be the sequence of nonterminal symbols occurring in e, in this order (a same symbol may occur several times in this list). We let $\sigma(p) := \tau(u)$ and $\alpha(p) := (\tau(u_{i_1}), ..., \tau(u_{i_k}))$.

We let also $\hat{p}$ be the monomial $p(u_{i_1}, ..., u_{i_k})$. Hence P is an N-signature, where N is $\{\tau(u)/u \in U\}$.

Let us consider the polynomial system

$$\bar{S}_\Gamma = \langle u_1 = \bar{t}_1, ..., u_n = \bar{t}_n\rangle,$$

where $\bar{t}_i$ is the polynomial $\hat{p}_1 + \cdots + \hat{p}_k$, and $\{p_1, ..., p_k\}$ is the set of production rules with left-hand side u_i. The least solution of $\bar{S}_\Gamma$ in $\mathscr{P}(\mathbf{M}(P))$ is an n-tuple of sets of trees. The first component of this tuple is the set of derivation trees of Γ. It is denoted by $\mathbf{Der}(\Gamma)$. (Let us recall from (2.10) that $L(\Gamma) = L(\Gamma, u_1)$). It is $\mathbf{M}(P)$-recognizable. Every tree t in $\mathbf{Der}(\Gamma)$ defines a graph in $L(\Gamma)$, denoted by $\mathbf{yield}(t)$. We characterize the mapping **yield** algebraically as the unique homomorphism $\mathbf{M}(P) \to \mathbf{FG}_\Gamma$, where $\mathbf{FG}_\Gamma$ is a derived magma of $\mathbf{FG}(A)$ that we now define.

We let $(\mathbf{FG}_\Gamma)_n := \mathbf{FG}(A)_n$ for $n \in N$.

We now define the operation $p_{\mathbf{FG}_\Gamma}$ for every p in P. Let p name $u \to e$, let $(u_{i_1}, ..., u_{i_k})$ be the sequence of nonterminals of e, let $x_j, j = 1, ..., k$ be a variable of sort $\tau(u_{i_j})$, let $\bar{e}$ be the expression in $\mathbf{FE}(A, X_k)$ obtained by replacing in e the jth nonterminal symbol by x_j. (It follows that $\bar{e}$ is linear in X_k.) We let $p_{\mathbf{FG}_\Gamma}$ be the derived operation $\bar{e}_{\mathbf{FG}(A)}$. Hence there is a unique homomorphism **yield**: $\mathbf{M}(P) \to \mathbf{FG}_\Gamma$, and it easy to verify that $\mathbf{yield}(\mathbf{Der}(\Gamma)) = L(\Gamma)$. More details on derivation trees can be found in Courcelle [6].

The following result, is a generalization of a result by Lengauer and Wanke [24].

(4.12) PROPOSITION. *Let (Γ, C) be a controlled context-free graph grammar defining a subset of $\mathbf{FG}(A)_s$. Let us assume that it can be decided whether $K \cap C = \varnothing$ for every effectively given recognizable set of trees K. The following properties of a formula φ in $\mathscr{CL}_{A,s}$ can be decided:*

(1) *$G \models \varphi$ for some graph G in $L(\Gamma, C)$,*

(2) *$G \models \varphi$ for all graphs G in $L(\Gamma, C)$.*

Proof. The set $K_\varphi := \{G \in \mathbf{FG}(A)_s / G \models \varphi\}$ is effectively $\mathbf{FG}(A)$-recognizable. Since $\mathbf{FG}_\Gamma$ is a derived magma of $\mathbf{FG}(A)$, it is also $\mathbf{FG}_\Gamma$-recognizable (Proposition (1.15)). Hence $\mathbf{yield}^{-1}(K_\varphi)$ is effectively $\mathbf{M}(P)$-recognizable by Proposition (1.7). It follows that

$$L(\Gamma, C) \cap K_\varphi = \mathbf{yield}(\mathbf{Der}(\Gamma) \cap \mathbf{yield}^{-1}(K_\varphi) \cap C).$$

Since $\mathbf{Der}(\Gamma) \cap \mathbf{yield}^{-1}(K_\varphi)$ is effectively recognizable, the emptiness of this set can be tested. Property (1) holds iff it is nonempty. Property (2) holds iff the set constructed similarily from $\neg\varphi$ is empty. ∎

(4.13) EXAMPLE. Let C_i, $i \geqslant 1$, be the set of trees all branches of which are of length i. Let $C = \bigcup\{C_i / i \geqslant 1\}$.

Let us establish that for every recognizable subset K of $\mathbf{M}(P)$, one can decide whether $C \cap K = \varnothing$. Without loss of generality, we assume that N is reduced to only one sort. (The general case is no more difficult.)

Let $K = h^{-1}(Q')$, where h is a homomorphism: $\mathbf{M}(P) \to \mathbf{Q}$, $\mathbf{Q}$ is a finite P-magma, and $Q' \subseteq \mathbf{Q}$. For every n, let $Q_n := h(C_n)$. Then

$$\begin{aligned} Q_1 &= \{h(p)/p \in P, \rho(p) = 0\}, \\ Q_{n+1} &= \{h(t)/t \in C_{n+1}\} \\ &= \{h(p(t_1, ..., t_k))/t_1, ..., t_k \in C_n, \rho(p) = k\} \\ &= \{p_Q(q_1, ..., q_k)/q_1, ..., q_k \in Q_n, \rho(p) = k\}. \end{aligned}$$

It follows that the sequence $Q_1, Q_2, ..., Q_n, ...$ is computable. Since the sets Q_n are subsets of a finite set, there exists q such that $Q_q = Q_m$ for some $m < q$. Hence $K \cap C \neq \varnothing$ iff $Q' \cap \bigcup\{Q_i / i < q\} \neq \varnothing$, and this is decidable.

Hence, for every context-free graph-grammar Γ, the set of graphs $L(\Gamma, C)$, that is not necessarily context-free, has a decidable monadic theory. ∎

Complexity Issues

We present a few applications to the complexity of graph algorithms. Other results can be found in Courcelle [12, 14].

(4.14) PROPOSITION. *Let φ be a formula in $\mathscr{CL}_{A,k,q}$.*

(1) *Let $m \geqslant k$. One can decide in time $\mathbf{O}(\mathbf{size}(e))$ whether φ holds in the graph $\mathbf{val}(e)$ defined by a given expression e in $\mathbf{FE}(A)_k^{[m]}$.*

(2) *Let Γ be a context-free graph-grammar generating a subset L of*

$\mathbf{FG}(A)_k$. *One can decide in time* $\mathbf{O}(\mathbf{length}(d))$ *whether a graph* G *in* L *given by a derivation sequence* d *of* Γ *satisfies* φ.

Proof. (1) From the proofs of Propositions (4.3) and (1.6), one can contruct a deterministic bottom-up tree-automaton recognizing the set of graph expressions e of width at most m and of type k, such that $\mathbf{val}(e) \models \varphi$. (This set can be considered as a set of trees $L \subseteq \mathbf{M}(\mathbf{H}_A^{[m]})_k$. This tree automaton is of large but fixed size, depending on φ, k, and m. It makes it possible to decide in time $\mathbf{O}(\mathbf{size}(e))$ whether e belongs to L.

(2) Let Γ be a context-free graph-grammar. Let $G \in \mathbf{FG}(A)_k$ be generated by Γ, by means of a derivation sequence d. By the definition we gave in Section 2 of context-free graph-grammars, this derivation sequences produces an expression e that defines G. This expression is of size $\mathbf{O}(\mathbf{length}(d))$ and of width at most m, where m is the maximum sort of a symbol occurring in Γ. It can be constructed in linear time from d.

Hence, by the first part of the lemma, one can decide whether $\mathbf{val}(e) \models \varphi$ in time $\mathbf{O}(\mathbf{size}(e))$, hence, one can decide in time $\mathbf{O}(\mathbf{length}(d))$ whether $G \models \varphi$. ∎

(4.15) *Remarks.* From the above result, it follows that, if a context-free set of graphs L has a polynomial parsing algorithm, then one can decide in polynomial time whether a graph G belongs to L, and, if this is the case, if it satisfies a given monadic second-order formula. Lauteman gives conditions on context-free graph-grammars ensuring the existence of polynomial parsing algorithms [23].

Monadic second-order formulas can express NP-complete problems. (The existence of a Hamiltonian circuit in a graph is an example of such a problem). This gives examples of NP-complete problems, becoming polynomial when restricted to special classes of graphs. Johnson [22] discusses several such situations.

Arnborg *et al.* [1] introduce a more powerful calculus, called the *extended monadic second-order logic*, for which Proposition (4.14) holds. This logical calculus makes possible a few numerical computations and comparisons. In particular, one can express that a graph has as many edges labeled by a and by b. But the set of graphs satisfying this property is not recognizable. (Otherwise, the set K used below in the proof that the converse to (4) in Theorem (5.3) would be recognizable, and we shall prove that it is not.) Hence, Theorem (4.4) does not hold for the extended monadic second-order logic.

Families of Sets of Graphs: A Comparison

We have established that every definable set of graphs is recognizable. In the case of words, a theorem by Büchi [4] (also Theorem 3.2 of Thomas

[29]) states that a language is recognizable iff it is definable. In the case of graphs, since there are countably many definable sets of graphs and uncountably many recognizable ones, some recognizable sets of graphs are not definable.

Here is an example of such a set. Let $K \subseteq \mathbb{N}$ be a recursively enumerable nonrecursive set. Let $L = \{G_n / n \in K,\ n \geqslant 2\}$, where G_n is the $(n \times n)$-grid defined in Definition (2.15). Given a graph H and a closed formula φ in $\mathscr{C}\mathscr{L}$, one can decide whether $H \models \varphi$ (because H is finite). If L would be equal to L_φ for some formula φ, one could decide whether $G_n \in L$, i.e., one could decide whether $n \in K$. This contradicts the choice of K.

We conclude this section by giving a diagram, comparing the various families of sets of graphs we have discussed. (On this diagram, shown on Fig. 6, the scope of a family name is the largest rectangle, at the upper left corner of which it is written.)

The following families of sets of graphs are compared:

REC, the family of recognizable sets of graphs,

CMSOL, the family of definable sets of graphs,

MSOL, the family of $\mathscr{L}$-definable sets of graphs,

CF, the family of context-free sets of graphs,

B, the family of width-bounded sets of graphs.

Provided the reference alphabet contains at least one symbol of type at least 2, the families **REC** and **B** are uncountable. The other ones are countable. The inclusions shown on the diagram, are strict, except possibly the inclusion:

$$\mathbf{CF} \cap \mathbf{CMSOL} \subseteq \mathbf{CF} \cap \mathbf{REC}.$$

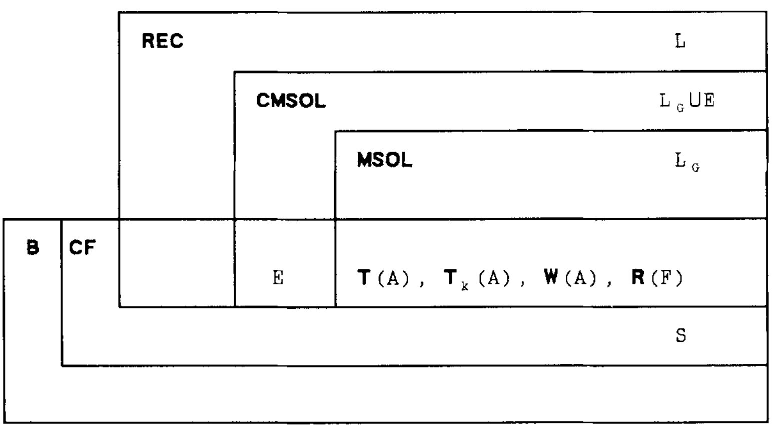

FIGURE 6

Whether it is strict raises an open problem, that can be restated as follows.

Open Problem. Does there exist k and a recognizable set $L \subseteq \mathbf{FG}(A)_0^k$ that is not definable (in counting monadic second-order logic)?

Such a set exists iff the shaded area of the diagram is not empty. The diagram also locates several sets of graphs:

L_G, the set of square grids,

L, the set of all grids G_n, where n is an element of some nonrecursive subset of $\mathbb{N}$,

E, the set of discrete graphs (all vertices of which are isolated), having an even number of vertices.

S, the set of graphs corresponding to the language $\{a^n b^n / n > 0\}$ (see Proposition (6.9)).

The sets of graphs $\mathbf{T}(A)$, $\mathbf{T}_k(A)$, $\mathbf{W}(A)$, and $\mathbf{R}(F)$ are introduced in Sections 5 and 6 below. They correspond to certain representations of trees and words by graphs.

It follows from Proposition (6.2) (and the proof of its Corollary (6.6)) that E belongs to **CMSOL-MSOL**.

5. Recognizable Sets of Trees

Büchi has proved in [4] that a set of words is recognizable iff it is $\mathscr{L}$-definable. A similar result has been proved for sets of ordered ranked trees (i.e., for subsets of $\mathbf{M}(F)$, where F is a finite signature) by Doner in [16]. (This latter result is essentially contained in Theorems (3.7) and (3.9) of [16]. See also Thomas [29, Theorem (11.1)] for a formulation closer to ours than that of Doner.)

In this section, we extend the result of Doner to sets of unordered unranked trees. This extension makes an essential use of *counting* monadic second-order logic. It does not work with the "ordinary" one, as we shall see in Section 6.

In this section, A is a finite alphabet consisting of symbols of type 1 or 2, and A_i is the set of symbols of A of type i.

(5.1) Definition. **Trees.** A *tree* is (here) a graph G in $\mathbf{FG}(A)_1$ satisfying the following conditions:

(1) for each vertex v, there is a path from $\mathbf{src}_G(1)$ to v; the vertex $\mathbf{src}_G(1)$ is called the *root* of G;

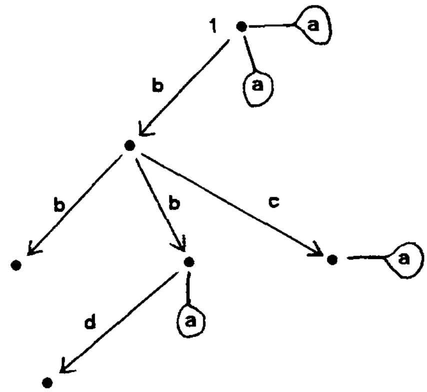

FIGURE 7

(2) every vertex different from $\mathbf{src}_G(1)$ is the target of one and only one binary edge;

(3) $\mathbf{src}_G(1)$ is the target of no edge.

An example of such a tree is shown on Fig. 7. There is no ordering on the set of edges originating from a same vertex. The vertices may belong to one, or several, or no unary edge ("unary" means "of type 1"). The graph **1** is also a tree.

It is clear that the set $\mathbf{T}(A)$ of all trees (over A) is $\mathscr{L}$-definable. Hence it is also recognizable.

(5.2) DEFINITION. **An algebraic structure on the set of trees.** We define a few derived operations on $\mathbf{FG}(A)_1$. If $G, G' \in \mathbf{FG}(A)_1$, we let

$$G \parallel G' := \sigma_1(\theta_{1,2}(G \oplus G')).$$

If $G \in \mathbf{FG}(A)_1$, and $b \in A_2$, then we let

$$\hat{b}(G) := \sigma_1(\theta_{2,3}(b \oplus G)).$$

Figure 8 shows the graphs $G \parallel G'$ and $\hat{b}(G)$, respectively, to the left and to the right.

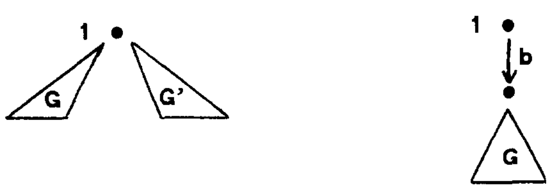

FIGURE 8

It is clear that the operation $\|$ is associative and commutative. We shall denote it as an infixed operation, without parentheses. $G \| G'$ and $\hat{b}(G)$ are trees if G and G' are.

Let $\mathbf{K}_A$ be the finite one-sort signature $\{\|, \mathbf{1}\} \cup A$, where $\|$ is of rank 2, $\mathbf{1}$ is of rank 0, a is of rank 0 if $a \in A_1$, and $\hat{a}$ is of rank 1 if $a \in A_2$. By the above definitions, $\mathbf{T}(A)$ is a $\mathbf{K}_A$-magma.

It is clear that $\mathbf{K}_A$ generates $\mathbf{T}(A)$. Since the operations $\|$ and $\hat{a}$ are defined on $\mathbf{FG}(A)_1$, the set $\mathbf{T}(A)$ is $\mathbf{FG}(A)$-equational. It is the least solution in $\mathscr{P}(\mathbf{FG}(A)_1)$ of the equation E:

$$L = L \| L + \sum \{\hat{b}(L)/b \in A_2\} + \sum \{a/a \in A_1\} + \mathbf{1}.$$

(As in Courcelle [5], $+$ and $\sum$ refer to set unions.)

A set of trees can be recognizable or equational, either w.r.t. $\mathbf{FG}(A)$, or w.r.t. $\mathbf{T}(A)$. We shall compare the two notions in the following proposition. Since $\mathbf{T}(A)$ is L-definable as a subset of $\mathbf{FG}(A)_1$, a subset L of $\mathbf{T}(A)$ is definable (resp. $\mathscr{L}$-definable) iff it is definable (resp. $\mathscr{L}$-definable) as a set of trees, i.e., iff there exists a closed formula φ in $\mathscr{CL}$ (resp. in $\mathscr{L}$) such that $L = \{G \in \mathbf{T}(A)/G \models \varphi\}$. (We shall prove in Section 6 that certain definable sets are not $\mathscr{CL}$-definable.)

(5.3) THEOREM. *Let $L \subseteq \mathbf{T}(A)$. The following conditions are equivalent*:

(1) $L \in \mathbf{Rec}(\mathbf{FG}(A))_1$

(2) $L \in \mathbf{Rec}(\mathbf{T}(A))$

(3) *L is definable.*

The following implications hold, and the converse implications do not:

(4) $L \in \mathbf{Rec}(\mathbf{T}(A)) \Rightarrow L \in \mathbf{Equat}(\mathbf{T}(A))$

(5) $L \in \mathbf{Equat}(\mathbf{T}(A)) \Rightarrow L \in \mathbf{Equat}(\mathbf{FG}(A))_1$.

Proof. (3) $\Rightarrow$ (1) by Theorem (4.4).

(1) $\Rightarrow$ (2) by Proposition (1.15).

(2) $\Rightarrow$ (3) by Proposition (5.4) established below.

(4) is an immediate consequence of Proposition (1.13), since the finite signature $\mathbf{K}_A$ generates $\mathbf{T}(A)$.

(5) holds because it is easy to construct a system of equations defining L in $\mathbf{FG}(A)$ from a system of equations defining L in $\mathbf{T}(A)$.

We now consider examples showing that the converses to (4) and (5) do not hold. Let K be the set of trees of the form:

$$(\hat{b}(\mathbf{1}) \| \hat{c}(\mathbf{1}) \| \cdots \| (\hat{b}(\mathbf{1}) \| \hat{c}(\mathbf{1}))$$

with as many b's as c's. It is easy to construct an equation defining it in $\mathbf{T}(A)$. Hence $K \in \mathbf{Equat}(\mathbf{FG}(A))_1$. Let us assume that K is recognizable in $\mathbf{T}(A)$. Then so is $h^{-1}(K)$ in $\mathbf{M}(\mathbf{K}_A)$, where h is the unique homomorphism: $\mathbf{M}(\mathbf{K}_A) \to \mathbf{T}(A)$. Let us denote by b^n the term

$$\hat{b}(\mathbf{1}) \| \cdots \| \hat{b}(\mathbf{1}),$$

belonging to $\mathbf{M}(\mathbf{K}_A)$, written with n occurrences of $\hat{b}$. Let us denote by c^n the similar term with $\hat{c}$ instead of $\hat{b}$. The set of terms $M := \{b^n \| c^m / n, m \geqslant 1\}$ is recognizable in $\mathbf{M}(\mathbf{K}_A)$. Hence $K' := M \cap h^{-1}(K)$ is recognizable too. But $K' = \{b^n \| c^n / n \geqslant 1\}$, and it is easy to establish that it is not recognizable. Hence K is not recognizable in $\mathbf{T}(A)$, and the converse to (4) does not hold.

Let N be the set of trees in $\mathbf{T}(A)$ of the form $\hat{b}^n(\hat{c}^n(\mathbf{1}))$, $n \geqslant 0$. It is easy to find a context-free graph grammar generating N. Hence $N \in \mathbf{Equat}(\mathbf{FG}(A))_1$. If $N \in \mathbf{Equat}(\mathbf{T}(A))$, then $N = h(N')$ for some recognizable subset N' of $\mathbf{M}(\mathbf{K}_A)$, by Proposition (1.12). From a top-down tree-automaton that would recognize N', it would not be difficult to construct an automaton recognizing the set of prefixes of $\{b^n c^n / n \geqslant 0\}$. Hence no such recognizable N' can exist, and N is not in $\mathbf{Equat}(\mathbf{T}(A))$. Hence, the converse to (5) does not hold. ∎

(5.4) PROPOSITION. *Every* $\mathbf{T}(A)$*-recognizable set of trees is definable.*

Proof. Let $L = h^{-1}(C)$, where h is a homomorphism: $\mathbf{T}(A) \to \mathbf{Q}$, $\mathbf{Q}$ is a finite $\mathbf{K}_A$-magma, and C is a subset of the domain Q of $\mathbf{Q}$.

The subset $Q' = h(\mathbf{T}(A))$ of Q can be computed (as the least solution in $\mathscr{P}(\mathbf{Q})$ of the equation E introduced in Definition (5.2); an explicit computation is possible since $\mathbf{Q}$ is finite).

For every $b \in A_2$, the function $\hat{b}_Q$ maps Q' into Q', and the function $\|_Q$ maps $Q' \times Q'$ into Q'. Furthermore, for all q, q', q'' in Q':

$$q \|_Q \mathbf{1}_Q = q$$
$$q \|_Q q' = q' \|_Q q$$
$$q \|_Q (q' \|_Q q'') = (q \|_Q q') \|_Q q''.$$

Hence $\|_Q$ can be extended to finite multisets as follows:

$$\|_Q Z := q_1 \|_Q q_2 \|_Q \cdots \|_Q q_k,$$

where $\{q_1, ..., q_k\}$ is any enumeration of a finite multiset Z of elements of Q' (and $\|_Q \varnothing = \mathbf{1}_Q$). We also denote by $n \cdot q$, for q in Q', n in $\mathbb{N}_+$, the object $q \|_Q q \|_Q \cdots \|_Q q$ (with n times q). We let $0 \cdot q = \mathbf{1}_Q$ for every $q \in Q'$. We have $(n \cdot q) \|_Q (n' \cdot q) = (n + n') \cdot q$.

Let $\bar{q}_0, \bar{q}_1, \ldots, \bar{q}_m$ be an enumeration of Q' without repetitions, such that $\bar{q}_0 = \mathbf{1}_Q$. For every $q \in Q'$, we let W_q be the set

$$\{(n_1, \ldots, n_m) \in \mathbb{N}^m / n_1 \cdot \bar{q}_1 \|_Q \cdots \|_Q n_m \cdot \bar{q}_m = q\}.$$

For every sequence $w = (\alpha_1, \ldots, \alpha_m, \beta_1, \ldots \beta_m)$ in $\mathbb{N}^{2m}$, we let $L(w) = \{(\alpha_1 + \lambda_1 \beta_1, \ldots, \alpha_m + \lambda_m \beta_m) / \lambda_1, \ldots, \lambda_m \in \mathbb{N}\} \subseteq \mathbb{N}^m$.

CLAIM 1. *For every $q \in Q'$, one can find a finite subset $\bar{w}_q$ of $\mathbb{N}^{2m}$ such that $W_q = \bigcup \{L(w) / w \in \bar{w}_q\}$.*

Proof of Claim 1. Since Q' is finite, the infinite sequence $0 \cdot q$, $1 \cdot q$, $2 \cdot q, \ldots, n \cdot q, \ldots$ is ultimately periodic. One can determine its period (we let β_q be the length of the periodic factor) and its nonperiodic initial part. Hence, for every q, $q' \in Q'$, the set of integers $\{n \in \mathbb{N} / n \cdot q = q'\}$ can be written as the union of a finite set, and finitely many sets of the form $\{\alpha + \lambda \beta_q / \lambda \in \mathbb{N}\}$, with $\alpha \in \mathbb{N}$. The result follows then easily. ∎

CLAIM 2. *For every sequence $w \in \mathbb{N}^{2m}$, one can find a formula in $\mathscr{CL}(\{X_1, \ldots, X_m\})$ expressing that $(\mathbf{Card}(X_1), \ldots, \mathbf{Card}(X_m))$ belongs to $L(w)$.*

Proof. It suffices to construct φ_i in $\mathscr{CL}(\{X_i\})$ expressing that $\mathbf{card}(X_i) = \alpha_i + \lambda \beta_i$ for some $\lambda \in \mathbb{N}$.

If $\beta_i = 0$ or 1, then a formula φ_i in $\mathscr{L}(\{X_i\})$ can be constructed. If $\beta_i \geqslant 2$, then one takes for φ_i the following formula:

$$\exists Y, Y'[\text{“}X_i = Y \cup Y'\text{”} \wedge \text{“}Y \cap Y' = \varnothing\text{”} \wedge \text{“}\mathbf{Card}(Y) = \alpha_i\text{”} \wedge \mathbf{card}_{\alpha_i, \beta_i}(Y')].$$

The desired formula is then: $\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_m$. ∎

We now go back to the proof of Proposition (5.4). Let $G \in \mathbf{T}(A)$ and $v \in \mathbf{V}_G$. We denote by $\mathbf{D}(G, v)$ the tree H in $\mathbf{T}(A)$ such that

$\mathbf{V}_H = \{v' \in \mathbf{V}_G / v' = v$ or there is a path in G from v to $v'\}$,

$\mathbf{E}_H = \{e \in \mathbf{E}_G /$ all vertices of e are in $\mathbf{V}_H\}$,

$\mathbf{lab}_H = \mathbf{lab}_G \restriction \mathbf{E}_H$,

$\mathbf{vert}_H = \mathbf{vert}_G \restriction \mathbf{E}_H$,

$\mathbf{src}_H = (v)$.

If $e \in \mathbf{E}_G$ we denote by $\mathbf{D}(G, e)$:

the tree a if $\mathbf{lab}_G(e) = a \in A_1$,

the tree $\hat{b}(\mathbf{D}(G, v))$ if $\mathbf{lab}_G(e) = b \in A_2$ and $v = \mathbf{vert}_G(e, 2)$.

Let ∂ and $\bar{\partial}$ be the mappings $\partial: \mathbf{V}_G \to Q'$ and $\bar{\partial}: \mathbf{E}_G \to Q'$ such that:

(1) $\partial(v) = h(\mathbf{D}(G, v))$ for all $v \in \mathbf{V}_G$,

(2) $\bar{\partial}(e) = h(\mathbf{D}(G, e))$ for all $e \in \mathbf{E}_G$.

They satisfy the properties:

(3) $\bar{\partial}(e) = a_Q$ if $\mathbf{lab}_G(e) = a \in A_1$,

(4) $\bar{\partial}(e) = \hat{b}_Q(\partial(\mathbf{vert}_G(e, 2)))$ if $\mathbf{lab}_G(e) = b \in A_2$,

(5) $\partial(v) = \mathbf{1}_Q = \bar{q}_0$ if $\mathbf{D}(G, v) = \mathbf{1}$,

(6) $\partial(v) = \|_Q \{\bar{\partial}(e)/e \in \mathbf{E}_G,\ \mathbf{vert}_G(e, 1) = v\}$ if $\mathbf{D}(G, v)$ is not $\mathbf{1}$.

By Claim 1, Eq. (6) can be written as

(6′) $(n_1, ..., n_m) \in W_{\partial(v)}$

where $n_i = \mathbf{card}(\{e \in \mathbf{E}_G/\mathbf{vert}(e, 1) = v,\ \bar{\partial}(e) = \bar{q}_i\})$ for $i \in [m]$.

If follows then that

(7) $h(G) = h(\mathbf{D}(G, \mathbf{src}_G(1))) = \partial(\mathbf{src}_G(1))$.

It is not hard to see that Eqs. (3) to (6) define a unique pair of mappings ∂, $\bar{\partial}$, and this pair of mappings satisfies (1) and (2). Hence, for every tree G in $\mathbf{T}(A)$,

(8) $G \in L$ iff there exists a pair of mappings ∂, $\bar{\partial}$ satisfying (3)–(5) and (6′), and such that $\partial(\mathbf{src}_G(1)) \in C$.

The required mappings ∂ and $\bar{\partial}$ take their values in the finite set $\{\bar{q}_0, ..., \bar{q}_m\}$. Hence they can be represented by $(m+1)$-tuples of sets, $X_0, ..., X_m \subseteq \mathbf{V}_G$, and $Y_0, ..., Y_m \subseteq \mathbf{E}_G$, such that $X_i = \{v \in \mathbf{V}_G/\partial(v) = \bar{q}_i\}$ and $Y_i = \{e \in \mathbf{E}_G/\bar{\partial}(e) = \bar{q}_i\}$. With this coding, and by Claim 2 conditions (3)–(5) and (6′) can be written in counting monadic second-order logic. Note that the atomic formulas with $\mathbf{card}_{n,p}$ are used to express condition (6′).

Hence condition (8) can be rewritten as

$$G \in L \quad \text{iff} \quad \exists X_0, ..., X_m, Y_0, ..., Y_m[\Psi],$$

where Ψ is a formula in $\mathscr{CL}_{A,1,p}(\{X_0, ..., X_m, Y_0, ..., Y_m\})$ and $p = \mathbf{Max}\{\beta_{\bar{q}_1}, ..., \beta_{\bar{q}_m}\}$. Hence L is definable. ∎

Proof of Proposition (3.9). We now complete the proof of Proposition (3.9) by proving that the set of graphs having a nontrivial automorphism is not definable.

Let $a \in A$ be of type 2. Let L be the set of 0-graphs of the form:

$$\bullet \xrightarrow{a} \bullet \xrightarrow{a} \cdots \bullet \xrightarrow{a} \bullet \xleftarrow{a} \bullet \cdots \bullet \xleftarrow{a} \bullet$$

The set L is $\mathscr{L}$-definable. Let L' be the set of graphs in L having as many edges pointing to the left and to the right.

Let K be the set of graphs in $\mathbf{FG}(A)_0$ having at least one nontrivial automorphism. More precisely, K is the set of isomorphism classes of concrete graphs G such that there exists an isomorphism $G \to G$ that is not the identity.

Let us assume that K is definable. Then $L \cap K$ is also definable. Hence $L \cap K$ is recognizable. But $L \cap K = L'$. Hence the set of graph expressions defining graphs in L' is recognizable (as a set of ranked trees).

By using a proof technique similar to the one used in Theorem (5.3), to establish that the converse to (4) does not hold, one obtains a contradiction. Hence the existence of a nontrivial automorphism in a graph is not definable.

If the set of graphs having as many edges labeled by a and by b would be definable then, the set K used in the proof of Theorem (5.3) would be definable, hence recognizable. We know that this is not the case. This proves the second part of Proposition (3.9). ∎

6. The Expressive Power of Counting Monadic Second-Order Logic

We establish that the *counting* monadic second-order logic is strictly more powerful than the "ordinary" one. We also prove that, if, in a many-sorted structure $\mathbf{M}$ linear orders on the domains are $\mathscr{L}$-definable, then every formula φ of $\mathscr{CL}$ can be translated into a formula $\hat{\varphi}$ of $\mathscr{L}$, equivalent to φ in $\mathbf{M}$. Hence for words and ranked trees (in which linear orders are definable), the atomic formulas of the form $\mathbf{card}_{p,q}(U)$ do not add expressive power to $\mathscr{L}$.

(6.1) Definition. **Monadic second-order logic dealing with sets**. We shall consider a one-sorted language without constants (like $\mathbf{s}_1, \mathbf{s}_2, \ldots$), or basic relations (like $\mathbf{edg}_a$). We denote by $\mathscr{L}(\mathscr{V})$ the set of monadic second-order formulas with free variables in $\mathscr{V}$. (The atomic formulas are $x = y$ and $x \in X$, for object variables x, y and set variable X). Hence $\mathscr{L}(\mathscr{V})$ is the subset of $\mathscr{L}_{\varnothing,0}(\mathscr{V})$, that one would use to express properties of graphs in $\mathbf{FG}(\varnothing)_0$, i.e., of graphs consisting of finite sets of isolated vertices. It is clear that the formulas in $\mathscr{L}(=\mathscr{L}(\varnothing))$ can only express conditions on the cardinalities of the sets in which they are interpreted.

(6.2) Proposition. *There is no formula φ in $\mathscr{L}(\{X\})$ such that for every finite set V, for every subset X of V, $\mathbf{Card}(X)$ is even iff $(V, X) \models \varphi$.*

Since the formulas $\mathbf{card}_{0,2}(X)$ have no equivalent in $\mathscr{L}$, the counting monadic second-order logic $\mathscr{CL}$ is strictly more powerful than the "ordinary" one $\mathscr{L}$.

This proof will use several technical definitions and lemmas.

(6.3) Definitions. For every positive integer n, we enrich $\mathscr{L}$ into $\mathscr{L}'_n$, by allowing terms defining subsets of V (the domain of the logical structure where the formulas are evaluated), formed with set variables and Boolean operations $\bigcup$, $\bigcap$, $-$, and $\bar{\ }$ (the last one denotes the complementation w.r.t. V). These terms are called *set terms*. The atomic formulas are as in $\mathscr{L}$, together with $\mathbf{card}_i(t)$, and $\mathbf{card}_{>i}(t)$ for set terms t and $i \in [0, n]$. The meanings of these formulas are respectively "the set defined by t has exactly i elements" and "the set defined by t has more than i elements."

It is clear that $\mathscr{L}$ and $\mathscr{L}'_n$ have the same power since for set terms t and t', $\mathbf{Card}(t) = i$ and $\mathbf{Card}(t') > i$ are definable in $\mathscr{L}$. The formulas $x \in t$ and $t = t'$ are also definable in $\mathscr{L}$.

We eliminate object variables from $\mathscr{L}'_n$:

— for every object variable x, we let Z_x be a new set variable
— $\exists x \cdots$ is replaced by $\exists Z_x, \mathbf{card}_1(Z_x) \wedge \cdots$
— $x \in X$ is replaced by $\mathbf{card}_0(Z_x - X)$
— $x = y$ is replaced by $\mathbf{card}_0(Z_x - Z_y) \wedge \mathbf{card}_0(Z_y - Z_x)$.

Let $\mathscr{L}''_n$ be the set of formulas of $\mathscr{L}'_n$ without object variables. It follows from the above remarks that for every formula φ in $\mathscr{L}(\{X_1, ..., X_k\})$ one can find an equivalent formula φ' in $\mathscr{L}''_1(\{X_1, ..., X_k\})$. Our next aim is to eliminate quantifiers in the formulas of $\mathscr{L}''_n$.

We let $\mathscr{Q}_{k,n}$ be the set of quantifier-free formulas in $\mathscr{L}''_n(\{X_1, ..., X_k\})$. We also introduce sets $\mathscr{N}_{k,n} \subseteq \mathscr{Q}_{k,n}$ of formulas said to be in *normal form*.

For this purpose, we let $\mathscr{T}_k$ be the set of set terms of the form $Y_1 \cap \cdots \cap Y_k$, where each Y_i is either X_i or $\bar{X}_i$. We let $\mathscr{A}_{k,n}$ be the set of atomic formulas of the forms $\mathbf{card}_i(t)$ or $\mathbf{card}_{>n}(t)$ for $i \in [0, n]$ and $t \in \mathscr{T}_k$. We let $\mathscr{B}_{k,n}$ be the set of *basic formulas*, i.e., of formulas of the form:

$$\bigwedge\!\!\!\bigwedge \{\varphi_t / t \in \mathscr{T}_k\},$$

where for each $t \in \mathscr{T}_k$, φ_t is a formula in $\mathscr{A}_{k,n}$ of the form $\mathbf{card}_x(t)$ for some $x \in \{0, 1, ..., n, >n\}$.

Finally, we let $\mathscr{N}_{k,n}$ be the set of finite disjunctions of formulas in $\mathscr{B}_{k,n}$.

(6.4) Lemma. *Let $m \geqslant n$ be positive integers. For every formula φ in $\mathscr{Q}_{k,n}$ one can construct an equivalent formula $\bar{\varphi}$ in $\mathscr{N}_{k,m}$.*

Proof. The formula φ is transformed into $\bar{\varphi}$ by several steps.

Step 1. One replaces every atomic formula occurring in φ that is of the form $\mathbf{card}_{>i}(t)$ with $i<m$ by

$$\mathbf{card}_{i+1}(t) \vee \mathbf{card}_{i+2}(t) \vee \cdots \vee \mathbf{card}_m(t) \vee \mathbf{card}_{>m}(t).$$

This gives φ_1, that is equivalent to φ.

Step 2. Consider an atomic formula in φ_1 of the form $\mathbf{card}_x(t)$, $x \in \{0, 1, ..., m, >m\}$ such that $t \notin \mathscr{T}_k$. The term t can be rewritten into an equivalent term of the form $t_1 \cup \cdots \cup t_s$ where $t_1, ..., t_s$ are pairwise distinct terms in $\mathscr{T}_k$.

Then, for $i \in \{0, 1, ..., m\}$, the atomic formula $\mathbf{card}_i(t)$ is replaced by

$$\bigvee\!\!\!\bigvee \{\mathbf{card}_{i_1}(t_1) \wedge \cdots \wedge \mathbf{card}_{i_s}(t_s)/i_1, ..., i_s \in \mathbb{N}, i_1 + i_2 + \cdots + i_s = i\}.$$

The atomic formula $\mathbf{card}_{>m}(t)$ is replaced by

$$\bigvee\!\!\!\bigvee \{\mathbf{card}_{x_1}(t_1) \wedge \cdots \wedge \mathbf{card}_{x_s}(t_s)/x_1, ..., \\ x_s \in \{0, 1, ..., m, >m\}, \sigma(x_1, x_2, ..., x_s) > m\}.$$

(In this formula, $\sigma(x_1, x_2, ..., x_s)$ denotes the integer $x_1' + x_2' + \cdots x_s'$, where $x_i' = x_i$ if $x_i \in \{0, ..., m\}$, and $x_i' = m+1$ if x_i is "$>m$".)

We do this for all atomic formulas occurring in φ_1, and this gives a formula φ_2, equivalent to φ_1. This formula is a Boolean combination of atomic formulas in $\mathscr{A}_{k,m}$.

Step 3. We now eliminate the negation. It is sufficient to do this for atomic formulas. Note that $\neg(\mathbf{card}_{>m}(t))$ is equivalent to

$$\mathbf{card}_0(t) \vee \cdots \vee \mathbf{card}_m(t)$$

and that $\neg(\mathbf{card}_i(t))$ is equivalent to

$$\mathbf{card}_0(t) \vee \cdots \vee \mathbf{card}_{i-1}(t) \vee \mathbf{card}_{i+1}(t) \vee \cdots \vee \mathbf{card}_m(t) \vee \mathbf{card}_{>m}(t),$$

if $i \in [0, m]$. We can transform φ_2 into an equivalent formula φ_3, that is a disjunction of conjunctions of atomic formulas in $\mathscr{A}_{k,m}$.

Step 4. Let the obtained formula φ_3 be of the form $\bigvee\!\!\!\bigvee\{\psi_i/1 \leqslant i \leqslant r\}$. Each of its composing conjunctions can be simplified as follows:

If ψ_i is of the form

$$\cdots \wedge \mathbf{card}_x(t) \wedge \cdots \wedge \mathbf{card}_y(t) \wedge \cdots$$

with $x, y \in \{0, 1, ..., m, >m\}$, and $x \neq y$ then it can be replaced by **false**. If $x = y$, then one of these two atomic formulas can be deleted.

After finitely many steps, ψ_i is transformed into an equivalent formula ψ'_i, no two atomic formulas of which concern a same set term in $\mathscr{T}_k$.

If every set term t in $\mathscr{T}_k$ occurs in ψ'_i, then $\psi'_i \in \mathscr{B}_{k,m}$. Otherwise, if some t does not occur in ψ'_i, then ψ'_i can be replaced by

$$(\psi'_i \wedge \mathbf{card}_0(t)) \vee (\psi'_i \wedge \mathbf{card}_1(t)) \vee \cdots \vee (\psi'_i \wedge \mathbf{card}_{>m}(t)).$$

Hence, after finitely many such replacements, φ_3 is transformed into an equivalent formula φ_4 in $\mathscr{N}_{k,m}$. ∎

(6.5) LEMMA. *For every formula φ in $\mathscr{L}''_n(\{X_1, ..., X_k\})$, one can find $m \in \mathbb{N}$ and a formula $\bar{\varphi}$ in $\mathscr{N}_{k,m}$ that is equivalent to φ.*

Proof. By induction on the structure of φ.

(1) If φ is atomic (or even quantifier-free) then the existence of m and $\bar{\varphi}$ follows from Lemma (6.4).

(2) Let φ be of the form $\exists X_{k+1}\, \varphi'(X_1, ..., X_k, X_{k+1})$. Let $\bar{\varphi}' \in \mathscr{N}_{k+1,m}$, be associated with φ', by way of induction.

Then $\bar{\varphi}'$ is $\bigvee\!\!\bigvee\{\theta_i/1 \leqslant i \leqslant r\}$, where $\theta_i \in \mathscr{B}_{k+1,m'}$. Hence φ is equivalent to $\bigvee\!\!\bigvee\{\exists X_{k+1}.\ \theta_i/1 \leqslant i \leqslant r\}$. Consider θ_i. It is of the form

$$\bigwedge\!\!\bigwedge\{\mu_t/t \in \mathscr{T}_k\},$$

where μ_t is of the form $\mathbf{card}_{x_t}(t \cap X_{k+1}) \wedge \mathbf{card}_{x'_t}(t \cap \bar{X}_{k+1})$ for some $x_t, x'_t \in \{0, 1, ..., m', >m'\}$.

Since the sets defined by the various terms t in $\mathscr{T}_k$ are pairwise disjoint, $\exists X_{k+1}\, \theta_i$ is equivalent to

$$\bigwedge\!\!\bigwedge\{\exists X_{k+1} \cdot \mu_t/t \in \mathscr{T}_k\}.$$

Now consider now the formula

$$\exists X_{k+1}(\mathbf{card}_{x_t}(t \cap X_{k+1}) \wedge \mathbf{card}_{x'_t}(t \cap \bar{X}_{k+1}));$$

it is equivalent to:

$\mathbf{card}_{x_t + x'_t}(t)$ if $x_t, x'_t \in \{0, ..., m'\}$

$\mathbf{card}_{>(2m'+1)}(t)$ if x_t and x'_t are both "$>m'$"

$\mathbf{card}_{>(x_t+m')}(t)$ if $x_t \in \{0, ..., m'\}$ and $x_{t'}$ is "$>m$"

$\mathbf{card}_{>(m+x'_t)}(t)$ if $x'_t \in \{0, ..., m'\}$, x_t is "$>m$".

Hence φ is equivalent to a formula in $\mathscr{Q}_{k,m}$ with $m = 2m' + 1$. By Lemma (6.4) this formula can be transformed into an equivalent formula in $\mathscr{N}_{k,m}$ as wanted.

(3) If φ is a Boolean combination of formulas $\varphi_1, ..., \varphi_r$ such that the associated formulas $\bar{\varphi}_1, ..., \bar{\varphi}_r$ have been found in $\mathcal{N}_{k,m_1}, ..., \mathcal{N}_{k,m_r}$, respectively, then, the existence of $\bar{\varphi}$ in $\mathcal{N}_{k,m}$ with $m = \mathbf{Max}\{m_1, ..., m_r\}$ also follows from Lemma (6.4).

(4) If φ is $\forall X_{k+1} \varphi'(X_1, ..., X_k, X_{k+1})$, then it can be written $\neg \exists X_{k+1}[\neg \varphi'(X_1, ..., X_{k+1})]$ and the above cases (2) and (3), together with Lemma (6.4) yield the desired result. ∎

Proof of Proposition (6.2). If $\varphi \in \mathscr{L}(\{X_1\})$ defines $\mathbf{card}_{0,2}(X_1)$ then is equivalent to a formula in $\mathcal{N}_{1,m}$ for some m by Lemma (6.3), i.e., to a disjunction of formulas of the forms:

$\mathbf{card}_i(X_1)$ or
$\mathbf{card}_{>m}(X_1)$ or
$\mathbf{card}_j(\bar{X}_1)$ or
$\mathbf{card}_{>m}(\bar{X}_1)$.

The atomic formulas of the last three types allow X_1 to have an odd number of elements. Hence they cannot appear. Hence φ is equivalent to

$$\mathbf{card}_{i_1}(X_1) \vee \cdots \vee \mathbf{card}_{i_n}(X_1).$$

Hence φ does not allow sets X_1 with an even number of elements larger than $\mathbf{Max}\{i_1, ..., i_n\}$. This contradicts the initial assumption. ∎

(6.6) COROLLARY. *There exists a definable set of trees that is not $\mathscr{L}$-definable.*

Proof. Let A consist of one symbol, a, of type 1. Let $L \subseteq \mathbf{T}(A)$ be the set of trees of the form $a \| a \| \cdots \| a$ with an even positive number of a's. This set is definable. Let us assume that it is $\mathscr{L}$-definable. There exists a formula φ in $\mathscr{L}_{A,1}$ such that, for all $G \in \mathbf{T}(A)$:

$$G \models \varphi \Leftrightarrow G \in L.$$

Let L' be the set of trees of the form $a \| a \| a \| \cdots \| a$, with an arbitrary positive number of a's. If $G \in L'$, then the structure $|G|$ is of the form $\langle \mathbf{V}_G, \mathbf{E}_G, \mathbf{edg}_{aG}, \mathbf{s}_{1G} \rangle$ with $\mathbf{V}_G = \{\mathbf{s}_{1G}\}$, $\mathbf{E}_G \neq \varnothing$, $\mathbf{edg}_a(e, \mathbf{s}_{1G}) = \mathbf{true}$ for all $e \in \mathbf{E}_G$. Furthermore, $G \in L$ iff $G \models \varphi$ iff $\mathbf{Card}(\mathbf{E}_G)$ is even.

The formula φ can be transformed into a closed formula $\bar{\varphi}$ belonging to the set $\mathscr{L}$ introduced in Definition (6.1), such that for every G in L':

$$G \models \varphi \quad \text{iff} \quad \mathbf{E}_G \models \bar{\varphi}.$$

Hence, for every $G \in L'$, i.e., for every nonempty set $\mathbf{E}_G$:

$$\begin{aligned} &G \in L \quad \text{iff} \\ &\mathbf{card}(\mathbf{E}_G) \text{ is even} \quad \text{iff} \\ &G \models \varphi \quad \text{iff} \\ &\mathbf{E}_G \models \bar{\varphi}. \end{aligned}$$

That $\mathbf{E}_G \models \bar{\varphi}$ iff $\mathbf{card}(\mathbf{E}_G)$ is even, contradicts Proposition (6.2). Hence L is definable, but not $\mathscr{L}$-definable. ∎

We now prove that the languages $\mathscr{C}\mathscr{L}$ and $\mathscr{L}$ are equally powerful, for expressing properties of finite structures, the domains of which are linearly ordered by $\mathscr{L}$-definable orderings. In order to avoid the introduction of new notations, we state our result for a class of graphs. But its proof can be extended to any class of finite many-sorted logical structures.

We fix A and k, and we let $\mathscr{L}(\mathscr{W})$ denote $\mathscr{L}_{A,k}(\mathscr{W})$, where $\mathscr{W}$ is a finite $\{\mathbf{v}, \mathbf{e}\}$-sorted set of object and set variables.

In the following proposition, we let $\rho \in \mathscr{L}(\mathscr{W} \cup \{x, y\})$, where $\sigma(x) = \sigma(y)$ and $x, y \notin \mathscr{W}$. For every G in $\mathbf{FG}(A)_k$, for every $\mathscr{W}$-assignment v in G, we let $\rho_{G,v}$ be the binary relation on $\mathbf{V}_G$ (or $\mathbf{E}_G$) such that

$$(m, m') \in \rho_{G,v} :\Leftrightarrow (G, v, m, m') \models \rho.$$

(In the right-hand side of this definition, we assume that m is assigned to x, and that m' is assigned to y.)

Finally, we let X be a set variable of sort $\sigma(x)$.

(6.7) PROPOSITION. *For every $p, q \in \mathbb{N}$ such that $0 \leqslant p < q$, and $q \geqslant 2$, one can contruct a formula φ in $\mathscr{L}(\mathscr{W} \cup \{X\})$, such that, for every G in $\mathbf{FG}(A)_k$, for every $\mathscr{W}$-assignment v in G, if $\rho_{G,v}$ is a linear order on $\mathbf{V}_G$ (or on $\mathbf{E}_G$, depending on the sort of X) then for every subset X of $\mathbf{V}_G$ (or of $\mathbf{E}_G$):*

$$\mathbf{Card}(X) = p \;\mathbf{mod}\; q \quad \textit{iff} \quad (G, v, X) \models \varphi.$$

Proof. Without loss of generality, we assume that $\sigma(X) = \mathbf{v}$. Let Y, X', X'' be set variables of sort $\mathbf{v}$, that are not in $\mathscr{W}$.

Let G be such that $\rho_{G,v}$ is a linear order on $\mathbf{V}_G$. We denote this order by $\leqslant$. If $X, Y \subseteq \mathbf{V}_G$, and $y \in Y$, we let:

$$I(X, Y, y) := \{x \in X / y \leqslant x \text{ and for all } y' \in Y, \text{ either } y' \leqslant y \text{ or } x < y'\},$$

i.e., is the set of elements of X greater than or equal to y and strictly less than the successor of y in Y if it exists. It follows that if $y, \bar{y} \in Y$, $\bar{y} \neq y$ then $I(X, Y, y) \cap I(X, Y, \bar{y}) = \varnothing$.

Observe that $\mathbf{Card}(X) \equiv p \ \mathbf{mod}\ q$ iff there exists a partition $X_1 \cup \cdots \cup X_k \cup X''$ of X such that $\mathbf{Card}(X'') = p$, and $\mathbf{Card}(X_i) = q$ for all $i = 1, ..., k$.

Claim. $\mathbf{Card}(X) \equiv p \ \mathbf{mod}\ q$ *iff there exist* X', X'', *and* Y *such that*

$$\begin{aligned}
&X = X' \cup X'', \qquad X' \cap X'' = \varnothing \\
&\mathbf{Card}(X'') = p \\
&X' = \bigcup\{I(X', Y, y)/y \in Y\} \\
&\mathbf{Card}(I(X', Y, y)) = q \qquad \text{for all} \quad y \in Y.
\end{aligned}$$

The "if" direction follows from the fact that $\{I(X', Y, y)/y \in Y\}$ is a partition of X'. For the converse, let X be enumerated in increasing order as $\{x_1, ..., x_{kq+p}\}$ for some $k \geqslant 0$. Then let $X'' = \{x_{kq+1}, ..., x_{kq+p}\}$, $X' = X - X''$, and $Y = \{x_1, x_{q+1}, ..., x_{(k-1)q+1}\}$. Hence $I(X', Y, x_{iq+1}) = \{x_{iq+1}, x_{iq+2}, ..., x_{iq+q}\}$ for $i = 0, ..., k-1$. ▮

The conditions of the claim are expressible by a formula ψ in $\mathscr{L}(\{X, X', X'', Y\})$. Hence the desired formula φ is $\exists X', X'', Y[\psi]$. ▮

If $\mathscr{G}$ is a family of k-graphs such that two formulas ρ in $\mathscr{L}(\{x, y\})$, and ρ' in $\mathscr{L}(\{x', y'\})$ with $\sigma(x) = \sigma(y) = \mathbf{v}$, $\sigma(x') = \sigma(y') = \mathbf{e}$ are such that, for every G in $\mathscr{G}$, $\rho_{|G|}$ is a linear order on $\mathbf{V}_G$, and $\rho'_{|G|}$ is a linear order on $\mathbf{E}_G$, then, every formula ψ in $\mathscr{CL}_{A,k}(\mathscr{W})$ can be translated into a formula θ in $\mathscr{L}_{A,k}(\mathscr{W})$ such that, for every G in $\mathscr{G}$, every $\mathscr{W}$-assignment v in G:

$$(G, v) \models \psi \Leftrightarrow (G, v) \models \theta.$$

It follows, then that, a subset L of $\mathscr{G}$ is definable iff it is $\mathscr{L}$-definable.

We shall apply this result to words, to ranked trees and to k-bounded unordered unranked trees, but we first use it to compare $\mathscr{CL}$ to the full (nonmonadic) second-order logic.

(6.8) *Remark.* Every formula of counting monadic second-order logic can be translated into a equivalent formula of second-order logic, written with existential quantifications over binary relations.

In order to prove this fact, we introduce two variables R_v and R_e, denoting binary relations on the domains of sorts $\mathbf{v}$ and $\mathbf{e}$, respectively. One can construct first-order formulas φ_v and φ_e expressing that R_v and R_e are linear orders.

Hence, a formula ψ of $\mathscr{CL}(\mathscr{W})$ can be translated into the formula

$$\exists R_v, R_e[\varphi_v \wedge \varphi_e \wedge \theta]$$

of second-order logic, where θ is the translation of ψ, done with the technique of Proposition (6.7), in terms of the linear orders R_v and R_e.

Application to words

A word in A^* is considered as a graph in $\mathbf{FG}(A)_2$, where the symbols of A are of type 2. The word abac is identified with the graph:

$$1 \bullet \xrightarrow{a} \bullet \xrightarrow{b} \bullet \xrightarrow{a} \bullet \xrightarrow{c} \bullet\, 2$$

We denote by $\mathbf{W}(A)$ the subset of $\mathbf{FG}(A)_2$, consisting of all graphs corresponding to words in the above sense. Note that $\mathbf{W}(A)$ is $\mathscr{L}$-definable. Linear orders on $\mathbf{V}_G$ and $\mathbf{E}_G$, where $G \in \mathbf{W}(A)$ can be defined as follows.

A vertex x is "smaller" than a vertex y if there exists a simple path from x to y, and this relation is $\mathscr{L}$-definable by Proposition (3.8).

An edge e is "smaller" than e' if $\mathbf{vert}_G(e, 1)$ is "smaller" than $\mathbf{vert}_G(e', 1)$.

Hence Proposition (6.7) can be applied to words. For every $L \subseteq A^*$, we also denote by L the corresponding subset of $\mathbf{W}(A)$.

(6.9) PROPOSITION. *For every language $L \subseteq A^*$, the following conditions are equivalent*:

(1) *L is $\mathscr{L}$-definable,*

(2) *L is definable,*

(3) *L is A^*-recognizable, hence is a regular language,*

(4) *L is $\mathbf{FG}(A)$-recognizable.*

Proof. (1) $\Leftrightarrow$ (3) is known by Büchi [4].

(1) $\Leftrightarrow$ (2) is a consequence of Proposition (6.7).

(3) $\Leftarrow$ (4): as in Theorem (5.3).

(1) $\Rightarrow$ (4) follows from Theorem (4.4). ∎

Application to Ranked Trees

Ranked trees, i.e., terms can be treated in a similar way. Let F be a one-sort signature. Each symbol f of F has a *rank* $\rho(f)$ in $\mathbb{N}$, i.e., a number of arguments. The elements of $\mathbf{M}(F)$, called *terms*, are usually identified with finite ordered trees. These trees are not graphs in our sense, but we can define a one-to-one mapping making any term t in $\mathbf{M}(F)$ into a graph $\mathbf{H}(t)$ belonging to $\mathbf{FG}(F)_1$.

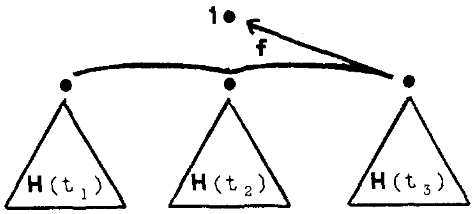

FIGURE 9

Let us define the type $\tau(f)$ of f in F as $\rho(f)+1$. We give the inductive definition of the graph $\mathbf{H}(t)$ associated with an element t of $\mathbf{M}(F)$:

If $t=a$, $\rho(a)=0$, then $\mathbf{H}(t)=a$.

If $t=f(t_1, ..., t_k)$, $k=\rho(f)\geqslant 1$ then $\mathbf{H}(t)$ is obtained by connecting $\mathbf{H}(t_1)$, $\mathbf{H}(t_2)$, ..., $\mathbf{H}(t_k)$ by their sources by means of a new hyperedge labeled by f. The source of $\mathbf{H}(t_i)$ is identified with the ith vertex of this new hyperedge. (The case $k=3$ is shown on Fig. 9).

Formally, this can be written:

$$\mathbf{H}(t)=\sigma_{2k+1}(\theta_\delta(\mathbf{H}(t_1)\oplus \cdots \oplus \mathbf{H}(t_k)\oplus f)),$$

where δ is the equivalence relation on $[2k+1]$ generated by $\{(1, k+1), ..., (k, 2k)\}$.

The graph $\mathbf{H}(f(g(a, a), a))$, where $\rho(f)=\rho(g)=2$, $\rho(a)=0$ is shown on Fig. 10.

We denote by $\mathbf{R}(F)$ the set $\{\mathbf{H}(t)/t \in M(F)\}$. It is not hard to prove that $\mathbf{R}(F)$ is $\mathscr{L}$-definable.

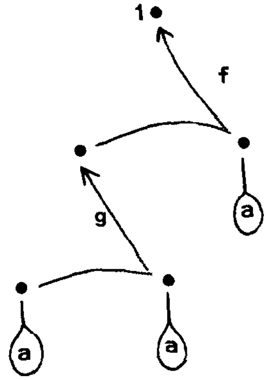

FIGURE 10

The postorder is the linear order on $\mathbf{V}_G$ that we can define as follows, for G in $\mathbf{R}(F)$. If $G = \mathbf{H}(f(t_1, ..., t_k))$, if $v, v' \in \mathbf{V}_G$, then $v \leqslant v'$ iff:

either v' is the source of G or

v and v' belong both to $\mathbf{H}(t_i)$ for some $i \in [k]$, and $v \leqslant v'$ w.r.t. $\mathbf{H}(t_i)$ or

v belongs to $\mathbf{H}(t_i)$, and v' belongs to $\mathbf{H}(t_j)$ for some $i, j \in [k]$, $i < j$.

For edges e and e' of G, one lets

$$e \leqslant e' \text{ iff } \mathbf{vert}_G(e, \tau(e)) \leqslant \mathbf{vert}_G(e', \tau(e')).$$

It is not hard to establish that these relations are linear orders and that they are $\mathscr{L}$-definable. Hence we have the following result, where $\bar{L}$ denotes $\{\mathbf{H}(t)/t \in L\}$ for every subset L of $\mathbf{M}(F)$.

(6.10) PROPOSITION. *For every subset L of $\mathbf{M}(F)$, the following conditions are equivalent*:

(1) *$\bar{L}$ is $\mathscr{L}$-definable,*

(2) *$\bar{L}$ is definable,*

(3) *L is $\mathbf{M}(F)$-recognizable,*

(4) *$\bar{L}$ is $\mathbf{FG}(A)$-recognizable.*

Proof. (1) $\Leftrightarrow$ (3) is known from Doner [16] (see Thomas [29, Theorem (11.1)]). The other equivalences are as in Proposition (6.9). ∎

Application to k-bounded (Unordered) Trees

Let A be as in Section 5. A tree G in $\mathbf{T}(A)$ is *k-bounded*, where $k \in \mathbb{N}_+$ if, for every vertex v of G, the set $\mathbf{out}(v) := \{e \in \mathbf{E}_G / \mathbf{vert}_G(e, 1) = v\}$ is of cardinality at most k. We denote by $\mathbf{T}_k(A)$ the set of k-bounded trees over A. This set is $\mathscr{L}$-definable.

Let $G \in \mathbf{T}_k(A)$. A partition π of $\mathbf{E}_G$ in k classes is *good*, if no two edges of any set $\mathbf{out}(v)$ belong to the same class. From every good partition $\pi = (X_1, ..., X_k)$ of $\mathbf{E}_G$, one can define linear orders on $\mathbf{V}_G$ and $\mathbf{E}_G$ as follows:

$v \leqslant v'$ iff, either there exists a path from v to v' or

$v = v'$ or

there exist two edges e, e' such that $e \in X_i$, $e' \in X_j$ with $i < j$, $\mathbf{vert}_G(e, 1) = \mathbf{vert}_G(e', 1)$, and there exist paths from $\mathbf{vert}_G(e, 2)$ to v, and from $\mathbf{vert}_G(e', 2)$ to v'.

$e \leqslant e'$ iff, either $e = e'$ or

$\mathbf{vert}_G(e, 1) < \mathbf{vert}_G(e', 1)$ or

$\mathbf{vert}_G(e, 1) = \mathbf{vert}_G(e', 1)$ and $e \in X_i$, $e' \in X_j$, $i < j$.

These two linear orders are defined by two formulas, respectively, ρ (belonging to $\mathscr{L}_{A,1}(\{v, v', X_1, ..., X_k\})$) and ρ' (belonging to $\mathscr{L}_{A,1}(\{e, e', X_1, ..., X_k\})$), where v, v' are variables of sort $\mathbf{v}$ and e, e' are variables of sort $\mathbf{e}$.

(6.11) PROPOSITION. *For a subset L of $\mathbf{T}_k(A)$, the following conditions are equivalent:*

(1) *L is $\mathscr{L}$-definable,*

(2) *L is definable,*

(3) *L is $\mathbf{FG}(A)$-recognizable.*

Proof. (2)$\Rightarrow$(1) Let $L \subseteq \mathbf{T}_k(A)$ be definable. Let $\varphi \in \mathscr{C}\mathscr{L}_{A,1,q}$ be a formula defining it. Let $\theta \in \mathscr{L}_{A,1}(\{X_1, ..., X_k\})$ be a formula expressing that $(X_1, ..., X_k)$ is a good partition. Let $\psi \in \mathscr{L}_{A,1}(\{X_1, ..., X_k\})$ be the formula that translates φ, according Proposition (6.7), by means of ρ and ρ'. Then, for every $G \in \mathbf{T}_k(A)$,

$$G \models \varphi \qquad \text{iff} \quad G \models \exists X_1, ..., X_k[\theta \wedge \psi].$$

Hence, L is $\mathscr{L}$-definable.

(1)$\Rightarrow$(3) and (3)$\Rightarrow$(2) are consequences of Theorem (5.3). ∎

ACKNOWLEDGMENTS

I thank J. Engelfriet, F. Jaeger, A. Meyer, G. Rozenberg, D. Seese, W. Thomas, and E. Welzl, for helpful remarks and suggestions.

RECEIVED February 27, 1987; FINAL MANUSCRIPT RECEIVED March 7, 1989

REFERENCES

1. ARNBORG, S., LAGERGREN, J., AND SEESE D. (1988), Problems easy for tree decomposable graphs, *in* "Proceedings, ICALP '88, Tampere," Lecture Notes in Computer Science, Springer-Verlag, Berlin/New York. Vol. 317, pp. 38–51.
2. BAUDERON, M., AND COURCELLE, B. (1987), Graph expressions and graph rewritings, *Math. Systems Theory* **20**, 83–127.
3. BODLAENDER, H. (1988), Dynamic programming on graphs with bounded tree width, *in* "Proceedings, ICALP '88, Tampere," Lect. Notes Comput. Sci., Vol. 317, pp. 105–118.

4. Büchi, J. (1960), Weak 2nd order logic and finite automata, *Z. Math. Logik Gründlag. Math.* **5**, 66–92.
5. Courcelle, B. (1986), Equivalences and transformations of regular systems. Applications to recursive program schemes and grammars, *Theoret. Comput. Sci.* **42**, 1–122.
6. Courcelle, B. (1987), An axiomatic definition of context-free rewriting and its application to NLC graph grammars, *Theoret. Comput. Sci.* **55**, 141–181.
7. Courcelle, B. (1987), A representation of graph by algebraic expressions and its use for graph rewriting systems, *in* "Proceedings, 3rd International Workshop on Graph Grammars," Lect. Notes Comput. Sci., Vol. 291, pp. 112–132, Springer-Verlag, Berlin/New York.
8. Courcelle, B. (1987), On context-free sets of graphs and their monadic second-order theory, Lect. Notes Comput. Sci., Vol. 291, pp. 133–146, Springer-Verlag, Berlin/New York.
9. Courcelle, B. (1988), On using context-free graph grammars for analyzing recursive definitions, *in* "Programming of Future Generation Computers II" (K. Fuchi and L. Kott, Eds.), pp. 83–122, North-Holland-Elsevier, Amsterdam.
10. Courcelle, B. (1989), On recognizable sets and tree automata, *in* "Resolution of Equations in Algebraic Structures" (H. Ait-Kaci and M. Nivat, Eds.), Academic Press, New York.
11. Courcelle, B. (in press), Graph rewriting: an algebraic and logical approach, *in* "Handbook of Theoretical Computer Science" (J. Van Leeuwen, Ed.), Elsevier, Amsterdam.
12. Courcelle, B. (1989), The monadic second-order logic of graphs: Definable sets of finite graphs, *in* "Workshop on Graph Theoretical Concepts in Computer Science, Amsterdam, June 1988," Lect. Notes Comput. Sci., Vol. 344, pp. 30–53, Springer-Verlag, New York/Berlin.
13. Courcelle, B. (1989), The monadic second-order logic of graphs. II. Infinite graphs of bounded width, *Math. Systems Theory* **21**, 187–221.
14. Courcelle, B. (1988), "The Monadic Second-Order Logic of Graphs. III. Tree-Width, Forbidden Minors and Complexity Issues," Report 8852, Bordeaux-1 University, submitted for publication.
15. Courcelle, B. (1988), "The Monadic Second-Order Logic of Graphs. IV. Every Equational Graph Is Definable," Research Report 8830, submitted for publication.
16. Doner, J. (1970), Tree acceptors and some of their applications, *J. Comput. System Sci.* **4**, 406–451.
17. Ehrig, H., and Mahr, B. (1985), "Fundamentals of Algebraic Specification," Springer-Verlag, Berlin.
18. Eilenberg, S., and Wright, J. (1967), Automata in general algebras, *Inform. and Control* **11**, 52–70.
19. Gaifmann, H. (1981), Local and non-local properties, "Logic Colloquium (Herbrand Symposium)" (J. Stern, Ed.), pp. 105–135, North-Holland, Amsterdam.
20. Gecseg, F., and Steinby, M. (1984), "Tree Automata," Akad. Kiado, Budapest.
21. Habel, A., and Kreowski, H. J. (1987), May we introduce to you, Hyperedge replacement, Lect. Notes Comput. Sci., Vol. 291, pp. 15–26, Springer-Verlag, Berlin.
22. Johnson, D. (1985), The NP-completeness column: An on-going guide (16th)," *J. Algorithms* **6**, 434–451.
23. Lautemann, C. (1988), Efficient algorithms on context-free graph grammars, *in* "Proceedings, ICALP '88," Lect. Notes Comput. Sci., Vol. 317, pp. 362–378, Springer-Verlag, Berlin/New York.
24. Lengauer, T., and Wanke, E. (1988), Efficient analysis of graph properties on context-free graph languages, *in* "Proceedings, ICALP '88," Lect. Notes Comput. Sci., Vol. 317, pp. 379–393.

25. MEZEI, J., AND WRIGHT, J. (1967), Algebraic automata and context-free sets, *Inform. and Control* **11**, 3–29.

26. ROZENBERG, G. AND SALOMAA, A. (1980), "The Mathematical Theory of L Systems," Academic Press, New York.

27. SEESE, D. (1975), Ein Unentscheidbarkeitskriterium, *Wiss. Z. Humbold-Univ. Berlin Math.-Natur. Reich.* **24**, 772–780.

28. SEESE, D. (in press), The structure of the models of decidable monadic theories of graphs, *Annals of Pure Appl. Logic.*

29. THOMAS, W. (in press), Automata on infinite objects, *in* "Handbook of Theoretical Computer Science" (J. Van Leeuwen, Ed.), Elsevier, Amsterdam.

30. TRAHTENBROT, B. (1950), Impossibility of an algorithm for the decision problem on finite classes, *Dokl. Nauk. SSSR* **70**, 569–572.

31. WIRSING, M. (in press), Algebraic specification, *in* "Handbook of Theoretical Computer Science" (J. Van Leeuwen, Ed.), Elsevier, Amsterdam.