Structural Properties of Context-Free Sets of Graphs Generated by Vertex Replacement*

BRUNO COURCELLE

Université Bordeaux I, LABRI (CNRS), 351 cours de la Libération, 33405 Talence Cedex, France
E-mail: courcelle@labri.u-bordeaux.fr

We establish that a Vertex Replacement set of graphs, i.e., a set of graphs generated by a C-edNCE or, equivalently, by a separated handle rewriting graph grammar is Hyperedge Replacement, i.e., is generated by a hyperedge replacement graph grammar, if its graphs do not contain arbitrary large complete bipartite graphs $K_{n,n}$ as subgraphs. Another equivalent condition is that its graphs have a number of edges that is linearly bounded in terms of the number of vertices. These properties are decidable by means of an appropriate extension of the theorem by Parikh that characterizes the commutative images of context-free languages. We extend these results to hypergraphs. © 1995

Academic Press, Inc.

INTRODUCTION

Among the many notions of graph grammars which have been defined, some of them can be called context-free because their derivation sequences can be adequately described by derivation trees, and because the sets they generate form the least solution of systems of equations associated with them. These ideas are developed in [4].

There are two main types of context-free graph grammars, the hyperedge replacement grammars (HR for short) that generate graphs or hypergraphs by using hypergraphs as righthand sides of production rules and sentential forms [1, 23, 24]), and the vertex replacement grammars (VR for short) based on NLC-rewriting [21, 25]. The boundary NLC grammars [28, 30], the confluent NLC grammars [4], the confluent edNCE grammars [16], and the separated handle hypergraph grammars [14], which all generate sets of simple graphs, are VR grammars.

The generative powers of these two types of grammars have been compared in several papers [2, 13, 14, 18, 20, 29]. We improve some results of these papers. Let us recall some facts:

Fact 1 [14, 20]. Every HR set of simple graphs is VR.

The following fact gives a characterization of the sets of graphs that are VR but not HR.

Fact 2 [13]. A VR set of graphs is HR if and only if it is a subset of some HR set iff it does not contain some clique $K_n$ as a minor, iff it does not contain some complete bipartite graph $K_{n,n}$ as a minor.

Let us recall that $G$ is a minor of $H$ if it is obtained from a subgraph of $H$ by edge contractions; saying that a set of graphs $L$ does not contain $G$ as a minor means that no $H$ in $L$ contains $G$ as a minor.

The present article establishes the following results. (A set of graphs is sparse if its members have a number of edges that is linearly bounded in terms of the number of vertices.)

Main Theorem. A VR set of graphs is HR if and only if it does not contain some complete bipartite graph $K_{n,n}$ as a subgraph, iff it is sparse.

This result extends a result by Welzl [30] concerning certain boundary NLC sets of graphs that we shall discuss in Section 4.

Corollary. One can decide whether a VR set is HR.

We now present the three tools used for the proofs. The first one is a way to “factorize” graphs. The inverse of the factorization is called the “expansion” and is akin to the removal of $e$-transitions in finite state automata. We prove that a set of simple graphs is VR iff it is the set of expansions of the graphs of some HR set. This result shows some deep relationships between HR and VR sets of graphs.

Our second tool is an extension of Parikh’s Theorem to context-free sets of graphs that uses monadic second-order logic: instead of just counting vertex or edge labels as in [23, Chap. IV], we count the cardinalities of sets defined by monadic second-order formulas. Let us recall briefly the role of logic. A graph $G$ can be described by (logical) structures of two types denoted by $|G|_1$ and $|G|_2$, respectively. The first structure has a domain consisting of the set of ver-
tices of $G$, and relations expressing the existence of edges between vertices. The structure $\{G\}_i$ has a domain consisting of vertices and edges put together; its relations express the incidences. For every set of graphs $L$, we let $|L|_i := \{|G|_i | G \in L\}$, for $i = 1, 2$. Roughly speaking, the structures $|G\}_i$ are adequate for describing VR sets of graphs by formulas of monadic second-order logic (MS formulas for short), i.e., by first-order formulas augmented with quantifications on sets, whereas the structures $|G\}_2$ are adequate for describing HR sets by such formulas.

As third tool, we shall use transformations from trees to structures representing graphs, that are specified by MS formulas. Let $i$ be a tree, represented by the structure $|i|_1$, and let $S$ be a structure. We write $S = f(i)$ if the domain of $S$ can be specified as a subset of that of $|i|_1$ by an MS formula, and if the relations of $S$ can also be specified by MS formulas (an $n$-ary relation is specified by an MS formula with $n$ free individual variables). Hence, such a mapping $f$ is specified by a finite tuple of MS formulas. We say that it is a definable transduction from trees to graphs. (This is only an approximation of the formal definition; see Section 1.) We shall handle VR and HR sets of graphs via the characterization of $[13, 17]$ stating that a set $L$ is VR iff $|L|_1 = \{ f(t) | t \in K \}$ for some recognizable set $K$ of trees and some definable transduction $f$, and that a set $L$ is HR iff $|L|_2 = \{ f(t) | t \in K \}$, where $K$ and $f$ are as above.

At the end of the paper (Section 5), we extend the main result to VR and HR sets of hypergraphs. These sets are defined and handled in terms of definable transductions from trees to appropriate structures, exactly as are the corresponding sets of graphs. The proof uses a result stating that the set of pairs $|H|_1, |H|_2$ for all simple hypergraphs $H$ of fixed tree-width is a definable transduction. This result has been proved in [9] for unlabelled directed graphs, and then extended to labelled directed graphs in [13, Lemma (3.6)], by some kind of boot-strapping, making it possible not to repeat the complicated constructions of [9]. Theorem (5.2) is the extension to hypergraphs. In order to obtain the new result as a consequence of that of [13], we introduce a new coding of hypergraphs by graphs, of independent interest. The idea of this coding is to replace a hyperedge $e$ having $n$ vertices by $n - 1$ "new" vertices (representing initial subsuccesses of the sequence of vertices of $e$) and $2n - 2$ edges, linking appropriately these new vertices to one another and to the ones of $e$. If two hyperedges share an initial subsequence of vertices, the "new" vertices encoding this common initial sequence are also shared. This gives some kind of factorization of the hyperedges of the given hypergraph.

The paper is organized as follows. Section 0 reviews the notation, Section 1 defines graphs, graph operations, VR and HR sets of graphs, MS logic, definable transductions and reviews basic results. Section 2 introduces graph expansion and establishes the relations between VR and HR sets of graphs via graph expansion. Section 3 gives the new "Parikh's Theorems" for VR and HR sets, Section 4 gives the proof of the main result, and Section 5 extends it to VR and HR sets of hypergraphs.

0. NOTATION

We use := for "equal by definition," i.e., to introduce a new notation, and ::=, similarly, to introduce a logical condition. We denote by $\mathbb{N}$ the set of nonnegative integers, and by $\mathbb{N}_+$ the set of positive ones. We denote by $[n]$ the interval $\{1, 2, 3, ..., n\}$ for $n \geq 0$ (with $[0] = \emptyset$). For sets $A$ and $B$, we denote by $A \cdot B$ the set $\{a \in A \mid a \notin B\}$. The cardinality of a set is denoted by $\text{Card}(A)$, and its powerset by $\nu(A)$.

A binary relation $R \subseteq A \times B$ is also called a transduction. We write $arB$ for $(a, b) \in R$, and we consider every $b$ such that $arB$ as an image of $a$ under $R$. The domain of $R$ is $\text{Dom}(R) := \{a \in A \mid \text{arB for some } b \in B\}$ and the image of $R$ is $\text{Im}(R) := \{b \in B \mid \text{arB for some } a \in A\}$. For $L \subseteq A$, the image of $L$ under $R$ is $R(L) := \{b \in B \mid \text{arB for some } a \in L\}$. The transduction $R^{-1} := \{(b, a) \mid (a, b) \in R\}$ is called the inverse of $R$. If $L$ is a subset of $B$, then $R^{-1}(L)$ is the inverse image of $L$ under $R$. We say that $R$ is functional if $R(\{a\})$ is a singleton or is empty for every $a \in A$. We identify functional relations $R \subseteq A \times B$ with partial functions $R : A \to B$, and we write $(a, b) \in R$ instead of $arB$. The composition of two relations $R \subseteq A \times B$ and $S \subseteq B \times C$ is the relation $R \circ S \subseteq A \times C$, defined as $\{(a, c) \mid (a, b) \in R \text{ and } (b, c) \in S\}$ for some $b \in B$.

The restriction of a partial function $f : A \to B$ to a subset $A'$ of $A$ is denoted by $f \upharpoonright A'$. If two partial functions $f : A \to B$ and $g : A' \to B$ coincide on $\text{Dom}(f) \cap \text{Dom}(g)$ we denote by $f \cup g$ their common extension into a partial function: $A \cup A' \to B$, with domain $\text{Dom}(f) \cup \text{Dom}(g)$. By a mapping, we shall mean a total function.

Trees

By a tree, we shall mean an element of the set $\mathbb{T}(F)$ of finite well-formed terms written with the symbols of a ranked alphabet $F$. The rank of $f$ in $F$ is a nonnegative integer denoted by $\rho(f)$. These trees will also be called terms in order to emphasize the fact that they have a value (they will denote graphs along the lines of [1, 14]). They will also be considered as directed labelled graphs in the usual way.

Certain sets of trees, said to be recognizable, can be defined in terms of (finite-state) tree automata. In decidability results concerning recognizable sets of trees, these sets will be (implicitly) given by tree automata. If we want to emphasize this, we shall say that they are effectively given. Since we shall not use these automata in proofs, we need not recall the precise definition. We refer the reader to the book by Geese and Steinby [22] for a thorough study of finite-state tree-automata.
We shall also use certain tree transformations. A linear
1. VR AND HR SETS OF GRAPHS

VR REDISTRIBUTION. For $G$ and $G' \in \mathbf{GR}_r(A, P)$, let $G \oplus G'$ be their disjoint union. Assuming that $V_G \cap V_{G'} = \emptyset$ and $E_G \cap E_{G'} = \emptyset$ (we replace $G'$ by an isomorphic copy if necessary), we let $G \oplus G' := H$ where $V_H = V_G \cup V_{G'}$, $E_H = E_G \cup E_{G'}$, $\mathbf{lab}_H = \mathbf{lab}_G \cup \mathbf{lab}_{G'}$, $\mathbf{vert}_H = \mathbf{vert}_G \cup \mathbf{vert}_{G'}$, $\mathbf{port}_H = \mathbf{port}_G \cup \mathbf{port}_{G'}$. It is clear that $\tau(G \oplus G') = \tau(G) \cup \tau(G')$.

Port Redefinition. For $z \in P \times P$ we let $\pi_z$ be the unary operation such that $\pi_z(G) = H$ where $H$ is as $G$ except that $\mathbf{port}_H = \mathbf{port}_{G'} \cup z$. We have $\tau(\pi_z(G)) = z \tau(G)$.

Elementary Graphs. The elementary graphs are the

met. Let $A$ and $P$ be two finite sets of labels called respectively the set of edge labels and the set of port labels.

We denote by $\mathbf{GR}_r(A, P)$ the set of finite labelled graphs formally defined as tuples of the form $G = (V_G, E_G, \mathbf{vert}_G, \mathbf{lab}_G, \mathbf{port}_G)$, where $V_G$ is the finite set of vertices, $E_G$ is the finite set of edges, with $V_G \cap E_G = \emptyset$, $\mathbf{lab}_G$ is a mapping: $E_G \to A$, $\mathbf{vert}_G$ is a mapping: $E_G \to V_G \times V_G$, $\mathbf{port}_G$ is a subset of $V_G \times P$. These graphs are finite, directed; we say that an edge links a vertex $x$ to a vertex $y$ if it is directed from $x$ to $y$; we say that it links $x$ and $y$ if the direction is irrelevant. Each edge has a label $a$, and an edge linking a vertex to itself is a loop. If a pair $(t, p)$ belongs to $\mathbf{port}_G$, we say that $v$ is a $p$-port of $G$. A vertex is a port if it is a $p$-port for some $p$. (The use of ports will appear soon.) We shall denote by $\tau(G)$ the set of labels $p$ such that $G$ has a $p$-port, and call it the type of $G$. Any two isomorphisms (for the obvious notion of isomorphism) are considered as equal. A graph $G \in \mathbf{GR}_r(A, P)$ is simple if it has no multiple edges, i.e., if for every $e, e' \in E_G$, $\mathbf{vert}_G(e) = \mathbf{vert}_G(e')$ and $\mathbf{lab}_G(e) = \mathbf{lab}_G(e')$ imply to $e = e'$. We shall denote by $\mathbf{GR}_s(A, P)$ the set of simple graphs in $\mathbf{GR}_r(A, P)$. We define some operations on $\mathbf{GR}_s(A, P)$.

Edge Creation. For $a \in A$ and $p, q \in P$, we let $\eta_{a\cdot p\cdot q}$ be the unary operation such that $\eta_{a\cdot p\cdot q}(G) = G'$ iff $G'$ is $G$ augmented with edges from any $p$-port $x$ to any $q$-port $y$ such that $y \neq x$. These new edges are labeled by $a$. Since we wish $G'$ to be simple, we do not create a new edge from $x$ to $y$ with label $a$ when one exists already.
We shall not use graph grammars in the present paper. All our proofs will rest on algebraic and logical characterizations of VR and HR sets like the one above and others to be recalled below.

We first establish a technical result saying that \( \mathcal{V}(A, P) \) can be limited to a strict subset that is as powerful for the generation of VR sets of graphs as the full set \( \mathcal{V}(A, P) \). We let \( \mathbf{GR}_{sp}(A, P) \) be the set of graphs in \( \mathbf{GR}_s(A, P) \) with simple ports, i.e., such that no vertex is both a p-port and a q-port for \( p \neq q \). We let \( \mathcal{V}_{sp}(A, P) \) be the subset of \( \mathcal{V}(A, P) \) consisting of the following operations:

- all nullary symbols denoting elementary graphs belonging to \( \mathbf{GR}_{sp}(A, P) \),
- \( \oplus \), and all operations of the form \( \eta_{a, p, q} \), and finally,
- all operations of the form \( \pi_z \), such that \( z \) is functional, i.e., such that, if \( (p, q) \) and \( (p, q') \in z \), then \( q = q' \).

It is clear that any graph formed from graphs in \( \mathbf{GR}_{sp}(A, P) \) by the operations \( \oplus, \eta_{a, p, q} \), and \( \pi_z \) (where \( z \) is functional) is also in \( \mathbf{GR}_{sp}(A, P) \). Hence all terms in \( \mathcal{T}(\mathcal{V}_{sp}(A, P)) \) denote graphs in \( \mathbf{GR}_{sp}(A, P) \). The following lemma expresses formally that, in a certain sense, the operations of \( \mathcal{V}_{sp}(A, P) \) are as powerful as those of \( \mathcal{V}(A, P) \). (Although it follows from Lemma 1 of [19], we prefer to prove it for sake of completeness.)

**Lemma.** If \( L \) is a VR set of graphs included in \( \mathbf{GR}_{sp}(A, P) \), then \( L = \text{val}(M) \) for some recognizable subset \( M \) of \( \mathcal{T}(\mathcal{V}_{sp}(A, P')) \) and some finite set \( P' \supset P \).

**Proof.** We shall construct \( P' \) and a linear tree homomorphism \( h: \mathcal{T}(\mathcal{V}(A, P)) \rightarrow \mathcal{T}(\mathcal{V}_{sp}(A, P')) \) such that, for all \( t \in \mathcal{T}(\mathcal{V}(A, P)) \), if \( \text{val}(t) \in \mathbf{GR}_{sp}(A, P') \), then \( \text{val}(t) = \text{val}(h(t)) \). The result will follow because, if \( L \subseteq \mathbf{GR}_{sp}(A, P) \) is VR, then \( L = \text{val}(M') \) for some recognizable set \( M' \subseteq \mathcal{T}(\mathcal{V}(A, P)) \) and then, \( L = \text{val}(h(M')) \). But \( h(M') \) is recognizable by Lemma (0.1), hence can be taken as the desired set \( M \).

We use a power set construction on port labels. Let \( P = \wp(P) - \emptyset \); we identify \( \{p\} \) with \( p \) so that \( P \) can be considered as a subset of \( P' \). For every \( G \in \mathbf{GR}_{sp}(A, P) \), we let \( \overline{G} \in \mathbf{GR}_{sp}(A, P') \) be obtained as follows: whenever \( v \) is a port of \( G \), we make it into an \( x \)-port of \( \overline{G} \) where \( x = \{v\} \); everything else in \( \overline{G} \) is as in \( G \). Hence, \( \overline{G} = G \) if \( G \in \mathbf{GR}_{sp}(A, P) \) since \( P \) is identified with the set of singletons of \( P' \).

We construct a linear tree homomorphism \( h: \mathcal{T}(\mathcal{V}(A, P)) \rightarrow \mathcal{T}(\mathcal{V}_{sp}(A, P')) \) such that for all \( t \in \mathcal{T}(\mathcal{V}(A, P)) \)

\[
\text{val}(h(t)) = \text{val}(t) \tag{1}
\]

**Case 1.** If \( t \) is a nullary symbol denoting an elementary graph \( G \), then we let \( h(t) \) be the nullary symbol denoting \( \overline{G} \).

**Case 2.** If \( t = t_1 \oplus t_2 \), then \( h(t) := h(t_1) \oplus h(t_2) \).

**Case 3.** If \( t = \eta_{a, p, q}(t_1) \), then we take

\[
h(t) := \eta_{a, x_1, p}(\eta_{a, x_2, p}(\cdots \eta_{a, x_n, p}(h(t_1)) \cdots ))
\]

where \( (x_1, \beta_1), \ldots, (x_n, \beta_n) \) is an enumeration of the set of pairs \( (x, \beta) \) such that \( x, \beta \subseteq P \) and \( p \in x, q \in \beta \).

**Case 4.** If \( t = \pi_z(t_1) \), then we take

\[
h(t) := \pi_z(h(t_1))
\]

where \( y = \{(x, \beta) \mid x \neq \emptyset, \beta \neq \emptyset, \beta = z(x)\} \).

In order to establish (1) it suffices to verify that the following equalities hold for all \( G \) and \( G' \) in \( \mathbf{GR}_s(A, P) \):

\[
\overline{G \oplus G'} = \overline{G} \oplus \overline{G'}
\]

(2)

\[
\overline{\eta_{a, p, q}(G)} = \eta_{a, x_1, p}(\cdots \eta_{a, x_n, p}(\overline{G})) \tag{3}
\]

\[
\overline{\pi_z(G)} = \pi_z(\overline{G}) \tag{4}
\]

where \( x_1, \beta_1, \ldots, x_n, \beta_n \), \( y \) are as in the corresponding cases of the definition of \( h \). We only verify equality (4), the other ones being easier. Let \( x \) be a vertex in \( G \) and \( \alpha = \{p \in P \mid x \text{ a p-port}\} \). If \( \alpha = \emptyset \), then \( x \) is not a port of either \( G \), \( \pi_x(G) \) or \( \overline{\pi_x(G)} \). If \( \alpha \neq \emptyset \), then \( x \) is a port of \( G \) and of \( \overline{G} \) and there are two cases. If \( \beta := z(x) = \emptyset \), then \( x \) is neither a port of \( \pi_x(G) \) nor of \( \overline{\pi_x(G)} \), nor of \( \pi_x(\overline{G}) \). If \( \beta \neq \emptyset \), then \( x \) is a q-port of \( \pi_x(G) \) iff \( q \in \beta \), and a \( \beta \)-port of both \( \pi_x(G) \) and \( \pi_x(\overline{G}) \). Hence the graphs \( \pi_x(\overline{G}) \) and \( \overline{\pi_x(G)} \) have the same ports with the same port labels, and are equal.

The class of HR sets is defined similarly in terms of other graph operations. We let \( \mathbf{GR}_r(A, P) \) be the set of graphs \( G \) in \( \mathbf{GR}_s(A, P) \) such that \( \text{port}_G \subseteq \mathcal{V}_G \times P \) is a partial one-to-one function. This means that these graphs have simple ports and, furthermore, that they have at most one p-port for each \( p \). These ports will be called sources (as in [1, 12]). We let \( \mathcal{H}(A, P) \) be the set consisting of the following operations.

**Parallel Composition.** We let \( G \| G' \) be obtained from the disjoint union of \( G \) and \( G' \) by the fusion of the p-source of \( G \) and the p-source of \( G' \) for every \( p \) in \( \tau(G) \cap \tau(G') \). The type of \( G \| G' \) is thus \( \tau(G) \cup \tau(G') \).

**Source Redefinition.** For every \( z \subseteq P \times P \) that is functional and one-to-one, the operation \( \pi_z \) defined above maps \( \mathbf{GR}_s(A, P) \) into itself. We call it a source redefinition. The type of \( \pi_z(G) \) is thus \( z(\tau(G)) \), as already observed.

In addition, we put in \( \mathcal{H}(A, P) \) nullary symbols denoting the following elementary graphs:
a single vertex that is the $p$-source,

a loop with label $a$, the vertex of which is the $p$-source,

an edge with label $a$ directed from the $p$-source to the $q$-source, where $p \neq q$.

Every term $t$ in $T(H(A, P))$ evaluates to a graph in $GR(A, P)$ that we denote by $\text{val}(t)$. An $HR$ set of graphs is a subset of $GR(A, P)$ of the form $\text{val}(M)$ for some recognizable subset $M$ of $T(H(A, P))$. This is not the classical definition [1, 2, 4, 15], but it is equivalent (see [10]) and is appropriate for the present purposes. It is easy to see that if $K$ is a set of nullary symbols denoting fixed elements of $GR(A, P)$, then, for every recognizable subset $M$ of $T(H(A, P) \cup K)$, the set $\text{val}(M)$ is also $HR$. Here is a lemma following from [13, 14, 20].

(1.2) Lemma. Every $HR$ set of simple graphs is $VR$.

The remaining part of this preliminary section reviews definitions, notations and results from [10, 11, 12]. The reader knowing any of these papers may skip it.

Monadic Second-Order Logic

Let $R$ be a finite ranked set of symbols such that each element $r$ of $R$ has a rank $\rho(r)$ in $N$. A symbol $r$ in $R$ is considered as a $\rho(r)$-ary relation symbol. An $R$-(relational) structure is a tuple $S = \langle D_s, (r)_s, r \in R \rangle$ where $D_s$ is a finite (possibly empty) set, called the domain of $S$, and $r_s$ is a subset of $D_s^{\rho(r)}$ for each $r$ in $R$. We denote by $P(R)$ the class of $R$-structures. The formulas of monadic-second order logic (called MS-formulas for short), intended to describe properties of structures $S$ as above, are written with variables of two types, namely lowercase symbols $x, x', y, \ldots$, called object variables, denoting elements of $D_s$, and uppercase case symbols $X, Y, Y', \ldots$, called set variables, denoting subsets of $D_s$. The atomic formulas are of the forms $x = y$, $r(x_1, \ldots, x_n)$ (where $r$ is in $R$ and $n = \rho(r)$), and $x \in X$, and formulas are formed with propositional connectives and quantifications over the two kinds of variables. For a finite set $W$ of object and set variables, we denote by $L(R, W)$ the set of all formulas that have their free variables in $W$; we also let $L(R) := L(R, \emptyset)$ denote the set of closed formulas.

Let $S$ be an $R$-structure, let $\varphi \in L(R, W)$, and let $\gamma$ be a $W$-assignment in $S$ (i.e., $\gamma(X)$ is a set of $D_s$ for every set variable $X$ in $W$, and $\gamma(x) \in D_s$ for every object variable $x$ in $W$; we write this $\gamma; W \rightarrow S$ to be short). We write $(S, \gamma) \models \varphi$ in the case where $\varphi$ has no free variable. A set of $R$-structures $L$ is (monadic second-order) definable if there is a formula $\gamma$ in $L(R)$ such that $L$ is the set of all $R$-structures $S$ such that $(S, \gamma) \models \varphi$.

A graph $G$ in $GR(A, P)$ can be represented by an $R_{A, P}$-structure, where $R_{A, P} := \{ \text{edg}_a/a \in A \} \cup \{ \text{pt}_p/p \in P \}$ with $\rho(\text{edg}_a) = 3$ and $\rho(\text{pt}_p) = 1$. The structure representing $G$ is $|G|_2 := \langle V, (\text{edg}_a)_{a \in A}, (\text{pt}_p)_{p \in P} \rangle$, where $V := V_G \cup E_G$, (let us recall that $V_G \cap E_G = \emptyset$),

$\text{edg}_a(x, y, z) := \exists x \in E_G, \text{lab}_G(x) = a$, and $\text{vert}_G(x) = (y, z)$, and

$\text{pt}_p(x) := x$ is a $p$-port.

Clearly $|G|_2$ is isomorphic to $|G'|_2$ as a relational structure iff $G \cong G'$ (i.e., $G$ is isomorphic to $G'$; we consider two isomorphic graphs as equal).

Another structure can be associated with $G$. We let $|G|_1 := \langle V, (\text{edg}_a')_{a \in A}, (\text{pt}_p')_{p \in P} \rangle$, where

$\text{edg}_a'(x, y, z) := \exists e \in E_G$, we have $\text{lab}_G(e) = a$, and $\text{vert}_G(e) = (y, z)$, and

$\text{pt}_p'(x) := x$ is a $p$-port. Note that the new relation symbol $\text{edg}_a'$ is binary. For simple graphs $G$ and $G'$, $|G|_1$ is isomorphic to $|G'|_1$ iff $G \cong G'$.

For a set $L$ of graphs, we let $|L|_2 := \{ |G|_2 | G \in L \}$, and similarly for $|L|_1$. By a MS$_r$ formula, we shall mean a monadic second-order formula that describes graph properties via this representation of a graph $G$ by the structure $|G|_1$.

Definable Graph Transductions

We recall the notion of a definable transduction of relational structures already used in [8–11]. Let $R$ and $R'$ be two ranked sets of relation symbols. Let $W$ be a finite set of set variables, called here the set of parameters. (It is not a loss of generality to assume that all parameters are set variables.) An $(R', R)$-definition scheme is a tuple of formulas of the form

$A = (\varphi, \psi_1, \ldots, \psi_k, (\theta_i)_{i \in R \times R'})$, where

$k > 0, R'^*k := \{ (r, j) | r \in R', j \in [k]^{\rho(r')} \}$,

$\varphi \in L(R, W)$,

$\psi_i \in L(R, W \cup \{ x_i \})$ for $i = 1, \ldots, k$,

$\theta_{r, j} \in L(R, W \cup \{ x_1, \ldots, x_{\rho(r')} \})$, for $r \in R', j \in [k]^{\rho(r')}$.

Let $S \in L(R); r$ be a $W$-assignment in $S$. An $R'$-structure $S'$ with domain $D_s \subseteq D_s \times [k]$ is defined by $A$ in $(S, r)$ if

$(S, r) \models \varphi$, $D_s = \{ (d, i) | d \in D_s, i \in [k], (S, r, d) \models \psi_i \}$. 

and, for each \( r \) in \( R \),

\[
\mathcal{r}_r = \left\{ (d_1, i_1, \ldots, d_i, i) \mid (S, \gamma, i, d_1, \ldots, d_i) \models \theta_{(r, r)} \right\},
\]

where \( j = (i_1, \ldots, i_i) \) and \( t = p(r) \).

(By \( (S, \gamma, d_1, \ldots, d_i) \models \theta_{(r, r)} \), we mean \( (S, \gamma') \models \theta_{(r, r)} \), where \( \gamma' \) is the assignment extending \( \gamma \), such that \( \gamma'(x) = d_i \) for all \( i = 1, \ldots, t \); a similar convention is used for \( (S, \gamma, d) \models \psi_r \). Note that \( S' \) is associated in a unique way with \( S, \gamma, \) and \( \mathcal{d} \) if it is defined, i.e., if \( (S, \gamma) \models \psi \); hence we use the functional notation \( \text{def}_d(S, \gamma) \).)

The transduction defined by \( \mathcal{d} \) of the relation \( \text{def}_d(S, \gamma) \) for some \( W \)-assignment \( \gamma \) in \( S \) is defined. The transduction \( \mathcal{f} \) is definable if it is equal to \( \text{def}_d \) for some definition scheme \( \mathcal{d} \) of appropriate type.

(1.3) Proposition [10, 11, 17]. The composition of two definable transductions is definable.

These definitions apply to graphs via their representations by relational structures as defined above. Since we have two representations of graphs by structures, we shall be more precise. We say that a graph transduction \( \mathcal{f} \) is \((i, j)\)-definable, where \( i, j \) are 0 or 1, iff the transduction \( \{((G, \gamma), (G', \gamma')) \mid (G, G') \in \mathcal{f}\} \) is definable. We shall also use transductions from trees to graphs. Since a tree \( t \) is a graph, it can be represented by either \( |t|_1 \) or \( |t|_2 \). However, it follows from the results of [9] that both structures are equally powerful for expressing in monadic second-order logic properties of trees and definable transductions from trees to graphs. A tree \( t \) will be represented by the structure \( |t|_1 \), since it is the smallest of the two. The results of the following theorem are proved in [13, 17] respectively.

(1.4) Theorem. A subset of \( \text{GR}_\mathcal{f}(A, P) \) is \( \text{HR} \) iff it is the image of a recognizable set of trees under a \((1,2)\)-definable transduction. A subset of \( \text{GR}_\mathcal{f}(A, P) \) is \( \text{VR} \) iff it is the image of a recognizable set of trees under a \((1,1)\)-definable transduction.

Since by Proposition (1.3) definable transductions are closed under composition we also get immediately

(1.5) Theorem. The class of HR sets (resp. of VR sets) of graphs is closed under \((2,2)\)-definable (resp. under \((1,1)\)-definable) graph transductions.

2. GRAPH EXPANSION

The expansion of a graph, some edges of which are labelled by the special label \( \varepsilon \), eliminates these edges more or less as one usually removes \( \varepsilon \)-transitions from a finite-state automaton. The result of the expansion of \( G \) is denoted by \( \exp(G) \). One can consider \( G \) as a succinct (or factorized) representation of the graph \( \exp(G) \), although in some particular cases \( \exp(G) \) may be smaller than \( G \).

(2.1) Definition. \( \varepsilon \)-graphs and their expansions. We let \( A \) and \( P \) be fixed; we let \( \varepsilon \) be a new edge label. An \( \varepsilon \)-graph is a graph \( G \) in \( \text{GR}(A \cup \{\varepsilon\}, P) \) such that the subgraph \( G_\varepsilon \) of \( G \) formed of all \( \varepsilon \)-edges of \( G \) (i.e., of all the edges of \( G \) with label \( \varepsilon \)) and, of course, of all vertices of these edges, is acyclic. In particular, \( \varepsilon \)-graphs have no loop labelled by \( \varepsilon \). We denote by \( \text{GR}_\varepsilon \) (or by \( \text{GR}_\varepsilon(A, P) \) if we need to specify \( A \) and \( P \)) the set of \( \varepsilon \)-graphs.

The expansion of a graph \( G \in \text{GR}_\varepsilon \) is the simple graph \( H = \exp(G) \in \text{GR}_\varepsilon(A, P) \) defined as follows:

\[
(1) \quad V_H = W_G, \quad \text{where we denote by } W_G \text{ the set of vertices of } G \text{ that are not the target of any } \varepsilon\text{-edge}.
\]

\[
(2) \quad \text{If } x \text{ and } y \text{ are distinct vertices of } H, \text{ there is in } H \text{ an edge with label } a, \text{ directed from } x \text{ to } y, \text{ if there is in } G \text{ an edge with label } a \text{ from } x' \text{ to } y', \text{ where } x' \text{ and } y' \text{ are such that there exist } \varepsilon\text{-paths (i.e., possibly empty directed paths consisting of } \varepsilon\text{-edges}) \text{ from } x \text{ to } x' \text{ and } y \text{ to } y'. \text{ We say that the edge of } H \text{ linking } x \text{ to } y \text{ is produced by the edge of } G \text{ linking } x' \text{ to } y'. \text{ We may have } x' = y', \text{ which means loops in } G \text{ may produce non-loop edges of } H.
\]

\[
(3) \quad \text{A vertex } x \text{ of } H \text{ is incident with a loop with label } a \text{ in } H \text{ iff it is incident with a loop with label } a \text{ in } G.
\]

\[
(4) \quad \text{A vertex } x \text{ is a } p\text{-port of } H \text{ iff there is in } G \text{ an } \varepsilon\text{-path from } x \text{ to some } p\text{-port of } G.
\]

Remark that loops are not created by \( \varepsilon \)-edges. They exist in the expansion if and only if they exist in the original graph. This is coherent with the definition we took of the edge creating operation that does not create loops.

(2.2) Example. Figure 1 shows an \( \varepsilon \)-graph \( G \) and the graph \( H = \exp(G) \). These graphs have no port and \( A = \{a, b\} \). The graph \( G \) has one \( a \)-labelled loop producing six \( a \)-labelled edges, and two \( b \)-labelled loops that remain \( b \)-labelled loops in \( H \).

(2.3) Proposition. If \( L \) is a VR or a HR set of \( \varepsilon \)-graphs then \( \exp(L) \) is a VR set.

Proof: It follows from the definition that the transduction \( \exp: \text{GR}_\varepsilon \rightarrow \text{GR}(A, P) \) is \((1,1)\)-definable and so \((2,1)\)-definable. The result then follows from Theorem (1.5).

The following result is a converse.

(2.4) Proposition. If a set of simple graphs is VR, then it is equal to \( \exp(M) \) for some HR set \( M \subseteq \text{GR}_\varepsilon \).

Proof: We shall construct a linear tree transduction \( h \) from \( \mathcal{T}(V(A, P)) \) into \( \mathcal{T}(\psi(A, P')) \), where \( P' \) is a finite set containing \( P \), such that, for every term \( l \) in \( \mathcal{T}(V(A, P)) \), \( \forall l(h(t)) \) belongs to \( \text{GR}_\varepsilon(A, P') \) and \( \forall l(t) = \exp(\forall l(h(t))) \).

Assuming this, the proof is as follows. If \( L \) is VR, then \( L = \forall l(R) \) for some recognizable subset \( R \) of \( \mathcal{T}(V(A, P)) \).
The set $h(R)$ is a recognizable subset of $T(H(A, P'))$ by Lemma (0.1), and we have

$$L = \{\text{val}(t) \mid t \in h(R)\}$$

$$= \{\exp(\text{val}(h(t))) \mid t \in R\}$$

$$= \exp(M),$$

where $M$ is the set $\{\text{val}(t) / t \in h(R)\}$, which is HR by the definition of HR sets.

We now explain how $h$ can be defined. We shall need an auxiliary set of labels $\bar{P} := \{\bar{p} / p \in P\}$. We let $\text{GR}_{ch} := \text{GR}_{ch}(A \cup \{\varepsilon\}, P \cup \bar{P}) \cap \text{GR}_{ch}(A \cup \{\varepsilon\}, P \cup \bar{P})$ be the subset of $\text{GR}_{ch}$ consisting of graphs having only simple ports and at most one $p$-port for each $p \in P \cup \bar{P}$. Such ports are called $p$-sources, as explained in Section 1. The expressions to be constructed by $h$ will be written with the operations of $H(A, P \cup \bar{P})$ together with some nullary symbols denoting finitely many fixed graphs in $\text{GR}_{ch}$.

The definition of $h(t)$ for $t \in T(\forall(A, P'))$ is by induction on the structure of $t$, using the auxiliary information $\tau(t) := \tau(\text{val}(t))$. (Let us recall that the type $\tau(K)$ of a graph $K$ is the set of labels of its ports.) Together with the definition of $h(t)$, we verify inductively that $\text{val}(h(t))$ belongs to $\text{GR}_{ch}$ and that $\text{val}(t) = \exp(\text{val}(h(t)))$.

1. If $t$ is a nullary symbol denoting an elementary graph $G$ with one vertex $x$ and port labels $p_1, \ldots, p_n$, we let $h(t)$ be a nullary symbol denoting the $\varepsilon$-graph $G'$ consisting of $x$ and of $\varepsilon$-edges linking $x$ to $n$ vertices $y_1, \ldots, y_n$. We let $y_i$ be the $p_i$-port for each $i$. This graph is shown in Fig. 2 (its vertex to the left is $x$) in the case $n = 3$.

If $G$ has loops with labels $a_1, \ldots, a_m$, then the vertex $x$ of $G'$ has the same loops. If $x$ is not a port of $G$, then we let $h(t) = t$. By these definitions, $\text{val}(h(t))$ belongs to $\text{GR}_{ch}$ and $\text{val}(t) = \exp(\text{val}(h(t)))$.

2. Let $t = \eta_{a, p \cdot q}(t_1)$. If at least one of $p$ or $q$ is not in $\tau(t_1)$, we let $h(t) := h(t_1)$. We have $\text{val}(t) = \exp(\text{val}(h(t_1)))$, since $\text{val}(t) = \text{val}(t_1)$, and by the induction hypothesis, $\text{val}(t_1) = \exp(\text{val}(h(t_1)))$.

Otherwise, we let $h(t)$ be $h(t_1) / [p \cdot \varepsilon \rightarrow q]$ if $p \neq q$ and $h(t_1) / [p \cdot \circ q]$ if $p = q$. (We let $[p \cdot \varepsilon \rightarrow q]$ be a nullary symbol denoting the graph $p \cdot \varepsilon \rightarrow q$ and similarly for $[p \cdot \circ q]$.) From the induction hypothesis that $\text{val}(h(t_1))$ belongs to $\text{GR}_{ch}$ and $\text{val}(t_1) = \exp(\text{val}(h(t_1)))$, we obtain that $\text{val}(h(t))$ belongs to $\text{GR}_{ch}$ and $\text{val}(t) = \exp(\text{val}(h(t)))$.

The hypothesis that $p$ and $q$ belong to $\tau(t_1)$ ensures that $\text{val}(h(t_1)) / [p \cdot \varepsilon \rightarrow q]$ has the same vertices as $h(t_1)$, and that $\exp(\text{val}(h(t_1)) / [p \cdot \varepsilon \rightarrow q])$ is obtained from $\exp(\text{val}(h(t_1)))$ by the addition of edges. (Otherwise, if for example $\text{val}(t_1)$ has no $q$-port, then the graph $\text{val}(t)$ has no $q$-port and is not equal to the graph $\exp(\text{val}(h(t_1)) / [p \cdot \varepsilon \rightarrow q])$ that has one $q$-port.)

3. If $t = \pi_1(t_1)$, then we let $h(t) := \pi_1(h(t_1) / H)$, where $x$ is the functional one-to-one relation $\{(\bar{p}, p, ) / p \in x(t_1)\}$ and $H$ is the graph in $\text{GR}_{ch}$ such that:

- its set of vertices is $\text{Dom}(z) \cap \tau(t_1) \times \{1\} \cup (\text{Im}(z) \cap z(\tau(t_1))) \times \{2\}$.
• its edges are ε-edges linking \((p, 1)\) to \((q, 2)\) for every pair \((p, q)\) in \(z \cap (\tau(t_1) \times \tau(t_1))\);

• the vertex \((p, 1)\) is the \(p\)-source and the vertex \((q, 2)\) is the \(q\)-source for every \(p \in \text{Dom}(z) \cap \tau(t_1)\) and \(q \in \text{Im}(z) \cap \tau(t_1)\).

We give an example in Fig. 3. We let \(P = \{p, q, r, s, u\}\). If \(t_1\) denotes a graph \(G_1\), with sources labeled by \(p, q, r, s\), if \(z = \{(p, p), (p, q), (r, q), (q, u), (r, u)\}\), then \(H\) and \(G = \pi_p(G_1)/H\) are shown in Fig. 3 (where all edges shown in the drawings are ε-edges).

By induction, \(\text{val}(h(t_1))\) belongs to \(\text{GR}_{\text{eh}}\) and \(\text{val}(t_1) = \text{exp}(\text{val}(h(t_1)))\), from which we get that \(\text{val}(h(t))\) belongs to \(\text{GR}_{\text{eh}}\) and \(\text{val}(t) = \text{exp}(\text{val}(h(t)))\).

4. If \(t = t_1 \oplus t_2\), we let \(h(t) := \pi_s(h(t_1))/H_1 \cup \pi_p(h(t_2))/H_2\), where \(H_1\) and \(H_2\) are constructed as \(H\) in case (3) from \(z = \{(p, p)/p \in P\}\) and \(\tau(t_i)\) for \(i = 1, 2, x = \{(p, p)/p \in \tau(t_1)\}\) and \(\beta = \{(p, p)/p \in \tau(t_2)\}\). This case is illustrated in Fig. 4 (where again all the edges represented are ε-edges), with \(G = \text{val}(h(t_1)), K = \text{val}(h(t_2)), \tau(G) = \{p, q, r, s\}, \tau(K) = \{q, r, s, u\}\). The third graph in Fig. 4 is \(\text{val}(h(t))\), the expansion of which equals \(\text{exp}(G) \cup \text{exp}(K)\).

By induction, \(\text{val}(h(t_1))\) and \(\text{val}(h(t_2))\) belong to \(\text{GR}_{\text{eh}}\), and \(\text{val}(t_i) = \text{exp}(\text{val}(h(t_i)))\) for \(i = 1, 2\), from which we get that \(\text{val}(t_1 \oplus t_2) = \text{exp}(\text{val}(\pi_s(h(t_1))/H_1 \cup \pi_p(h(t_2))/H_2))\), hence \(\text{val}(t) = \text{exp}(\text{val}(h(t)))\) and \(\text{val}(h(t))\) belongs to \(\text{GR}_{\text{eh}}\).

It follows from these four cases that \(h\) is a linear tree homomorphism where the substitution rules are determined by the auxiliary information \(\tau(t_i)\). Hence, \(h\) is a linear (deterministic bottom-up) tree transduction. (See Section 1 and [22, 27].)

(2.5) Definition. Normal ε-graphs. Let \(G = \text{val}(h(t))\), where \(t \in \begin{array}{c}
T(\forall(A, P))
\end{array}\) and \(h\) as is in the proof of Proposition (2.4). By looking at the definition of \(h\) one gets the following facts.

1. There do not exist in \(G\) four vertices \(x, y, z, u\) such that there are non-empty ε-paths from \(x\) to \(y\) and from \(z\) to \(u\), and non-ε-edges between \(x\) and \(u\) on one hand and between \(y\) and \(z\) on the other.

This is a consequence of the way ε- and non-ε-edges are put into the graphs \(\text{val}(h(t'))\) for all subterms \(t'\) of \(t\). In this statement, we may have \(x = z\) and \(y = u\), which excludes the possibility of a non-ε-edge between \(x\) and \(y\) when there is a non-empty ε-path from \(x\) to \(y\).

If furthermore \(t \in \begin{array}{c}
T(\forall(A, P))
\end{array}\), then \(\text{val}(h(t))\) is an ε-graph \(G\) satisfying the following property:

2. The subgraph \(G_{\varepsilon}\) of \(G\) consisting of its ε-edges has no vertex of outdegree more than 1.
Hence, $G_r$ is a union of directed trees (with edges directed toward the root) because, in the construction of Proposition (2.4), the basic graphs of case (1) and the graphs $H_r$ of cases (3) and (4) are of this form (by the special form of the operations of $\nu_{sp}(A, P)$) and because in case (2), $e$-edges are not modified.

An $e$-graph satisfying conditions (1) and (2) will be called a normal $e$-graph.

### 3. AN EXTENSION OF PARikh'S THEOREM

We shall give an extension of Parikh's theorem where one counts the cardinalities of sets of vertices defined by $MS_1$ formulas on the graphs generated by VR grammars, and of sets of vertices and edges defined by $MS_2$ formulas in the case of HR grammars.

A subset $Z$ of $\mathbb{N}^k$ is linear if it is of the form

$$Z = \{(f_1(x), ..., f_k(x)) / x \in \mathbb{N}^l\},$$

where each $f_i$ is an affine function, i.e., a function of the form

$$f_i(x) = a_{i,1}x_1 + a_{i,2}x_2 + \cdots + a_{i,l}x_l + \beta_i,$$

where $x = (x_1, ..., x_l)$ and $a_{i,1}, ..., a_{i,l}, \beta_i$ belong to $\mathbb{N}$. We write more briefly $Z = \{f(x) / x \in \mathbb{N}^l\}$, where $f$ denotes $(f_1, ..., f_k)$. We say that a set is semi-linear if it is a finite union of linear sets. We say that it is effectively semi-linear if one can compute the coefficients $a_{i,j}$ and $\beta_i$ in the tuples $f$ describing its linear components.

We say that $Z' \subseteq \mathbb{N}^k$ is the image of $Z \subseteq \mathbb{N}^k$ under an affine transformation if $Z' = \{g(y) / f \in Z\}$, where $g$ is a $k'$-tuple of affine functions. It is clear that, if $Z'$ is in this way from $Z$ and if $Z$ is semi-linear, then $Z'$ is semi-linear too. If $Z$ is effectively semi-linear, then $Z'$ is effectively semi-linear too.

If $K$ is a set of relational structures and $\varphi$ is a monadic second-order formula with free variables $X_1, ..., X_k$, we define $Z(K, \varphi) \subseteq \mathbb{N}^k$ as the set

$$Z(K, \varphi) := \{\text{Card}(X_1), ..., \text{Card}(X_k) \mid S \in K, (S, X_1, X_2, ..., X_k) \models \varphi\}.$$

Let $K$ be an effectively given recognizable set of trees and $\varphi$ be a $MS_1$ formula written with the relation symbols appropriate for trees in $K$. The set $Z(K, \varphi)$ is effectively semi-linear.

Proof: Let $K \subseteq \mathcal{T}(F)$. For every tree $t \in \mathcal{T}(F)$, for every $k$-tuple $(X_1, ..., X_k)$ of sets of nodes of $t$, we let $t(X_1, ..., X_k)$ be the tree in $\mathcal{T}(F \times \{0, 1\}^k)$ obtained by changing the label $f$ of every node $u$ of $t$ into $(f, i_1, ..., i_k)$, where $i_j = 0$ if $u \notin X_j$ and $i_j = 1$ if $u \in X_j$. The set

$$K_\varphi := \{t(X_1, ..., X_k) \mid t \in K, (t, X_1, ..., X_k) \models \varphi\}$$

is recognizable by the theorem of Doner stating that a set of trees is recognizable iff it is MS definable [15]. Hence, the set $L$ of Polish prefix representations of all trees in $K_\varphi$ is a context-free language. Let $k' = \text{Card}(F \times \{0, 1\}^k)$ and let $h$ be a bijection $[k'] \rightarrow F \times \{0, 1\}^k$. For every tree $s$ in $\mathcal{T}(F \times \{0, 1\}^k)$, we let $\#(s) := (\#(s, 1), ..., \#(s, k'))$, where $\#(s, i)$ is the number of occurrences of the symbol $h(i)$ in $s$. Hence, $\{\#(t') / t' \in K_\varphi\}$ is semi-linear by Parikh's theorem applied to the context-free language $L$. Since for each $t(X_1, ..., X_k) \in K_\varphi$.

$$\text{Card}(X_i) = \sum \{\#(t(X_1, ..., X_k), j) \mid \text{the } i\text{th bit of the second component of } h(j) = 1\},$$

the set $Z(K, \varphi)$ is the image of the semi-linear set $\{\#(t') / t' \in K_\varphi\}$ under an affine transformation. Hence $Z(K, \varphi)$ is semi-linear. Since all constructions used in this proof are effective, one can construct the expression of $Z(K, \varphi)$ as a semi-linear set from $\varphi$ and a tree automaton describing $K$.

(3.2) Theorem. Let $L$ be a set of structures given as $\theta(K)$, where $K$ is a recognizable set of trees and $\theta$ is a definable transduction. Let $\varphi$ be an $MS_1$ formula with free variables $X_1, ..., X_k$. The set $Z(L, \varphi) \subseteq \mathbb{N}^k$ is effectively semi-linear.

Proof. Let $\theta$ be defined by a definition scheme $A$ with parameters $Y_1, ..., Y_m$ and let $n$ be such that $D_n \subseteq D_1 \times \{0, 1\}^n$ whenever $S \in \theta(t)$. We shall use new set variables $X_{i,j}$ for $1 \leq i \leq k$, $1 \leq j \leq n$. By Proposition (5.2) of [18], one can construct a formula $\psi$ with free variables $Y_1, ..., Y_m, X_{1,1}, ..., X_{k,n}$ such that, for every tree $t$, for all sets of nodes $Y_1, ..., Y_m, X_{1,1}, ..., X_{k,n}$ of $t$, if we let $X_i = \bigcup \{X_{i,j} \times \{j\} / 1 \leq j \leq n\}$, then

$$((t, Y_1, ..., Y_m, X_{1,1}, ..., X_{k,n}) \models \psi)$$

iff

$$t \in K, \text{def}_A((t, Y_1, ..., Y_m)) \text{ is a well-defined structure } S, X_{1,1}, ..., X_{k,n} \subseteq D_S \text{ (where } X_i = \bigcup \{X_{i,j} \times \{j\} / 1 \leq j \leq n\}),$$

and $((S, X_1, ..., X_k) \models \varphi).$

In each case where these conditions hold, we have $\text{Card}(X_i) = \text{Card}(X_{i,1}) + \cdots + \text{Card}(X_{i,n})$. It follows that $Z(L, \varphi)$ is the image of $Z(K, \psi)$ under an affine transformation. Hence $Z(L, \varphi)$ is semi-linear, since $Z(K, \psi)$ is, by Lemma (3.1). All constructions involved in this proof are effective.

(3.3) Corollary. Let $L$ be a VR (resp. a HR) set of graphs and $\varphi$ be a $MS_1$-formula (resp. a $MS_2$-formula). The set $Z([L_1, \varphi] \text{ (resp. } Z([L_2, \varphi]))$ is effectively semi-linear.
This corollary yields extensions of Parikh’s theorem where one does not simply count occurrences of vertex or edge labels (as in [23, Chap. IV] for HR sets), but cardinalities of sets satisfying arbitrary MS formulas.

Here are examples of such sets: one can express by a MS₂ formula that a set \( X \) is a minimal set of edges, the deletion of which makes planar the considered graph \( G \) (minimality is w.r.t. set inclusion); one can express similarly that a set of vertices \( Y \) is the set of occurrences in \( G \) of a fixed graph \( H \), where \( H \) is a graph with one source, and we say that \( v \) is an occurrence of \( H \) in \( G \) if some induced subgraph of \( G \) is isomorphic to \( H \), in such a way that \( v \) corresponds to the source of \( H \) in the isomorphism.

The cardinalities of these sets can be compared with the cardinalities of the set of vertices or of edges: by looking at the coefficients occurring in effective descriptions of semilinear sets, one can decide whether, on all graphs of a given HR set, some quantity such as \( \text{Card}(X) \) or \( \text{Card}(Y) \), where \( X \) and \( Y \) are as above, is uniformly bounded or linearly bounded in terms of another one. These techniques will be used in Section 4.

4. THE MAIN THEOREM

This section is devoted to the proof of the main theorem. We say that a set of graphs \( L \) is sparse if there exists a constant \( k \) such that \( \text{Card}(E_G) \leq k \cdot \text{Card}(V_G) \) for every graph \( G \) in \( L \). For every graph \( G \), we denote by \( \text{und}(G) \) the undirected graph obtained from \( G \) by forgetting orientations and by deleting edge labels, source- and port-labels, and loops and by fusing multiple edges. Hence, \( \text{und}(G) \) is undirected, simple, loop-free, and unlabelled and has no port or source. If \( L \) is a set of graphs, we denote by \( \text{und}(L) \) the set of graphs \( \text{und}(G) \) for \( G \) in \( L \).

(4.1) Theorem. Let \( L \) be a VR set of graphs. The following conditions are equivalent:

1. \( L \) is HR,
2. there exists an integer \( n \) such that \( K_{n,n} \) is not a subgraph of \( \text{und}(G) \) for any graph \( G \) in \( L \),
3. \( L \) is sparse.

One can decide whether these conditions hold.

Before starting the proof of this theorem, we recall that Welzl has established in [30] that for every r-BNLC ("relabelling Boundary Node Label Controlled") set of graphs \( L \), the following are equivalent:

1. the set of all subgraphs of the graphs in \( L \) is r-BNLC,
2. there exists an integer \( n \) such that \( K_{n,n} \) is not a subgraph of \( \text{und}(G) \) for any graph \( G \) in \( L \),
3. \( L \) is sparse.

The graphs generated by an r-BNLC grammar are undirected; their vertices have labels but their edges do not. They form a VR set in the extended sense sketched in Section 1, before Lemma (1.1). This can be established by the results of [10, 17] but our purpose here is not to prove such a minor technical point.

Condition (1) in the theorem of Welzl implies that \( L \) is HR by the result proved in [13] that a VR set of graphs is HR if the set of all its subgraphs is VR. Hence, Theorem (4.1) together with the result of [13] extends to the class of VR sets the result of [30].

Proof of Theorem (4.1), Easy Directions. Let \( L \) be a HR set of graphs. The tree-width of its members is bounded by some integer \( k \) that one can determine from the grammar or the system of equations defining \( L \); see, e.g., [7, 12] (we shall recall in Section 5 the definition of tree-width; we denote by \( \text{twd}(G) \) the tree-width of a graph \( G \). Then Condition (2) holds because if \( K \) is a subgraph of \( \text{und}(G) \) we have

\[
\text{twd}(K) \leq \text{twd}(\text{und}(G)) = \text{twd}(G)
\]

by classical properties of tree-width. Since \( \text{twd}(K_{n,n}) = n \) for every \( n \), \( K_{n+1,n+1} \) is not a subgraph of \( \text{und}(G) \) for any graph \( G \) in \( L \), and Condition (2) holds.

We now consider Condition (3). A simple undirected loop-free graph of tree-width at most \( k \) is a partial \( k \)-tree. A partial \( k \)-tree \( H \) with at least \( k \) vertices results from a \( k \)-tree \( H \) by deletions of edges (see [26] on \( k \)-trees). Since for every \( k \)-tree \( H \) we have \( \text{Card}(E_H) \leq k \cdot \text{Card}(V_H) \), the same inequality holds for \( G \) in \( \text{und}(L) \). Since the graphs in \( L \) are simple, we have for every graph \( G \) in \( L \) the inequality

\[
\text{Card}(E_G) \leq b \cdot \text{Card}(V_{\text{und}(G)}) + 2b \cdot \text{Card}(E_{\text{und}(G)}) \leq (b+2bk) \cdot \text{Card}(V_G),
\]

where \( b = \text{Card}(A) \), because \( G \) has at most \( b \cdot \text{Card}(V_{\text{und}(G)}) \) loops (at most one for each pair of an edge label and a vertex) and at most \( 2b \cdot \text{Card}(E_{\text{und}(G)}) \) non-loop edges, since each edge of \( \text{und}(G) \) comes from at most \( 2b \) non-loop edges of \( G \) (because there are two orientations and \( b \) labels). Hence, \( L \) is sparse and Condition (3) holds. (We shall give another proof of this result for the more general case of HR sets of hypergraphs in Theorem (5.1).)

For the other directions, we need some preliminary definitions and results.

(4.2) Definition. \( k \)-limited \( \varepsilon \)-graphs. Let \( H \) be an \( \varepsilon \)-graph. We recall that we denote by \( W_H \), the set of vertices of \( H \) that are not the target of any \( \varepsilon \)-edge, so that \( V_{\text{exp}(H)} = W_H \). For every \( v \in W_H \), we let \( C(v) := \{ w \in W_H \mid \varepsilon \to v \} \) (where \( \varepsilon \to v \) means that there is an \( \varepsilon \)-path in \( H \) from \( w \) to \( v \)). We let \( N(v) \) be the set of non-\( \varepsilon \)-edges \( e \) of \( E_H \), one end of which is a vertex \( w \) such that \( w \varepsilon \to v \). Finally, we let \( D(v) \) be \( \{ v' \mid C(v') \text{ some edge } e \in N(v) \text{ links } w \text{ and } v' \text{, where } v \varepsilon \to v' \} \). See Fig. 5 for a picture. We say that an \( \varepsilon \)-graph \( H \) is \( k \)-limited iff for every \( v \in V_H \),

\[
\text{Min}\{ \text{Card}(C(v)), \text{Card}(D(v)) \} < k.
\]
A set of $\varepsilon$-graphs is $k$-limited if each graph in this set is $k$-limited.

4.3 Example. Figure 5 shows an $\varepsilon$-graph: the $\varepsilon$-edges are those having no label. The labelled edges except the loop form the set $N(v)$. We have $C'(v') = \{v'\}$ and $C(v') = \{v''\}$. The sets $N(v')$ and $N(v'')$ are both equal to the set $N(v)$ augmented with the loop. The edges of $N(v)$ produce a set of edges forming a subgraph $K$ of $\exp(H)$ such that $\operatorname{und}(K)$ is a complete bipartite graph with edges between every vertex in $C(v) \cup \{v', v''\}$ and every vertex in $D(v)$.

4.4 Lemma. If a normal $\varepsilon$-graph $H$ is not $k$-limited, then $K_{k,k}$ or $K_k$ is a subgraph of $\operatorname{und}(\exp(H))$.

Proof. Let $v$ be a vertex in $V_H$ such that $\operatorname{Card}(C(v))$ and $\operatorname{Card}(D(v))$ are both $\geq k$. If there are vertices $w$, $w'$ with $v \xrightarrow{e} w$ and $v \xrightarrow{e} w'$, and if there is a non-$\varepsilon$-edge $e$ linking $w$ to $w'$, then $e$ is a loop and $w = w'$, since $H$ is normal. Thus $e$ produces in $\exp(H)$ edges linking any two distinct vertices in $C(v)$. (Note that $D(v) \supseteq C(v)$ in this case.) Hence, $K_k$ is a subgraph of $\operatorname{und}(\exp(H))$. Otherwise, $C(v) \cap D(v) = \emptyset$ and the edges in $N(v)$ produce edges linking any vertex of $C(v)$ with any vertex of $D(v)$, and $K_{k,k}$ is a subgraph of $\operatorname{und}(\exp(H))$.  

The following proposition is the crux of the proof of Theorem (4.1).

4.5 Proposition. For every $k$ the restriction of the mapping $\exp$ to normal $k$-limited $\varepsilon$-graphs is $(2, 2)$-definable.

Proof. Let us first observe that, for each $k$, the set of normal $k$-limited $\varepsilon$-graphs (over fixed alphabets) is $\mathit{MS}_1$-definable. (Normal $\varepsilon$-graphs are defined in Definition (2.5)). We say that an $\varepsilon$-graph $H$ is irredundant if it is simple and if every edge of $\exp(H)$ is produced by at most one (hence by exactly one) edge of $H$. In a normal $\varepsilon$-graph $H$, if two distinct edges $e$ and $e'$ produce the same edge of $\exp(H)$, then they have the same label and one of them, say $e$, must be below the other, say $e'$, which means that $e$ links $x$ to $y$, that $e'$ links $x'$ to $y'$, and that there exist possibly empty $\varepsilon$-paths from $x$ to $x'$ and from $y$ to $y'$. (We may have $x = x'$ and $y = y'$ in the case where $e$ and $e'$ are multiple edges.) In such a case, $e$ can be removed and $\exp(H - e) = \exp(H)$, because every edge produced by $e$ is also produced by $e'$. (One removes only the edge and not its vertices).

Let $H$ be a normal $\varepsilon$-graph. Since it is normal, it has no two $\varepsilon$-edges with the same pair of vertices. By removing non-$\varepsilon$-edges if necessary, one can transform it into a normal irredundant $\varepsilon$-graph $\mathit{irr}(H)$ such that $\exp(\mathit{irr}(H)) = \exp(H)$, because, as observed above, one can remove an edge $e$ if it is below an edge $e'$ having the same label. Similarly, if $e$ and $e'$ are two loops with the same label and incident with the same vertex of $W_H$, then they yield a unique loop in $\exp(H)$, and one of them can be deleted. Note that $\mathit{irr}(H)$ is still $k$-limited if $H$ is, because the sets $C(v)$ remain the same, whereas the sets $D(v)$ can only decrease. The transformation of $H$ into $\mathit{irr}(H)$ is a $(2, 2)$-definable transduction. Since the composition of two $(2, 2)$-definable transductions is $(2, 2)$-definable by Proposition (1.3), we need only establish that $\exp$ is $(2, 2)$-definable on normal irredundant $k$-limited $\varepsilon$-graphs in order to get the desired result.

Let $H$ be such a graph and $G = \exp(H)$ be its expansion. We know that $\exp$ is $(1, 1)$-definable. In order to obtain the result it suffices to encode $E_G$ as a subset of $D_H \times \{2, ... , n\}$ for some large enough integer $n$ depending only on $k$ and $\operatorname{Card}(A)$. (Let us recall that for every graph $K$, $D_K = V_K \cup E_K$ and is the domain of the structure $|K|_2$ representing $K$.) This encoding must be done in such a way that the two end vertices, the label, and the direction of an edge in $G$ can be defined in $H$ by MS-formulas from the pair $(v, i)$ in $D_H \times \{n\}$ that encodes this edge.

We first consider the encoding of loops. A loop with label $a$, incident with a vertex $v$ in $W_H = W_{\varepsilon}$, can be encoded by the pair $(v, m_a)$, where $m_a$ is a fixed integer in $\{2, ..., \operatorname{Card}(A) + 1\}$, and $m_a \neq m_b$ for $a \neq b$.

We now consider the encoding of edges of $G$ which are not loops. A vertex $v$ of $H$ is called a big vertex if $\operatorname{Card}(C(v)) \geq k$. The other vertices of $H$ are called small. (Without loss of generality, we assume that $k \geq 2$, so that every vertex in $W_H$ is small.) Every non-$\varepsilon$-edge of $H$ has at most one end that is a big vertex, since $H$ is $k$-limited. We call it a big edge if it has one, and a small edge otherwise, i.e., if its two ends are small. Note that a small edge of $H$ produces at most $(k - 1)^2$ edges of $G$, and that a big edge cannot be a loop. The set of big edges is $\{N(w) | w \text{ is a big vertex}\}$. A small vertex $v$ such that $v \xrightarrow{e} v'$ for no small vertex $v'$ is said to be maximally small. Since $H$ is normal, there is a unique maximal $\varepsilon$-path starting from any vertex. Note also that if $w$ is big and $w \xrightarrow{e} w'$, then $w'$ is big. It follows that for every small vertex $v$, there is a unique maximally
small vertex \( v' \) such that \( v \xrightarrow{e} v' \). We have thus \( C(v) \subseteq C(v') \) in this case, and two sets \( C(v') \) and \( C(v') \) are disjoint whenever \( v' \) and \( v' \) are maximally small and distinct. Hence, the sets \( C(v) \) where \( v \) is maximally small form a partition of \( W_H \).

A \( k \)-nice coloring of \( H \) is a mapping \( c : V_H \to [m] \), where \( m = (2b + 1) \cdot (k - 1) \), and \( b \) is the cardinality of \( A \), that satisfies the following conditions:

1. \( c(V_H) \subseteq [k - 1] \) and \( c \) is one-to-one on each set \( C(v) \) such that \( v \) is maximally small.

2. \( c(E_H) \subseteq \{k, \ldots, m\} \) and \( c \) is one-to-one on each set \( N(w) \) such that \( w \) is big.

As already noted, if \( v \) is small, there is an \( e \)-path from \( v \) to some maximally small vertex \( v' \), and \( C(v) \subseteq C(v') \). Hence, if (1) holds, then \( c \) is one-to-one on every set \( C(v) \) where \( v \) is small.

**Claim 1.** Every normal, irredundant, \( k \)-limited \( e \)-graph \( H \) has a \( k \)-nice coloring.

**Proof.** The sets \( C(v) \) for \( v \) maximally small are of cardinality at most \( k - 1 \) and are pairwise disjoint; hence, Condition (1) is fulfilled.

In order to fulfill Condition (2), one constructs \( c(N(w)) \) inductively, assuming \( c(N(w')) \) to be known if \( w \xrightarrow{e} w' \). (This condition implies that \( N(w') \subseteq N(w) \).) We make first a few observations.

Let \( w \) be big; let \( v \) be a vertex in \( C(w) \). The edges in \( N(w) \) produce edges linking \( v \) and the vertices of \( D(w) \) and there are at most \( 2b \cdot \text{Card}(D(w)) \) such edges (between \( v \) and vertices of \( D(w) \)), since \( \exp(H) \) is simple and each edge has two possible orientations and \( b \) possible labels. Since \( H \) is irredundant, distinct edges in \( N(w) \) produce distinct edges. Hence \( \text{Card}(N(w)) \leq 2b \cdot \text{Card}(D(w)) \). For a big vertex \( w \), \( \text{Card}(D(w)) \leq k - 1 \) (because \( H \) is \( k \)-limited), hence \( \text{Card}(N(w)) \leq 2b(k - 1) \).

We now construct \( c(N(w)) \), where \( w \) is big. If \( w \) has no neighbour vertex \( w' \) with \( w' \), we take \( c(N(w)) \) any function \( N(w) \to \{k, \ldots, m\} \), which is possible by the above observations and the definition of \( m \). Otherwise \( w \xrightarrow{e} w' \), \( w' \) is unique for a given \( w \), and, by the definition of \( N(w) \), we have \( N(w') \subseteq N(w) \). Since the \( e \)-edges do not form cycles, we can construct \( c \) inductively and assume that \( c(N(w')) \) is already known. We need only to extend \( N(w) \). It is possible to do that with at most \( 2b(k - 1) \) colors since \( \text{Card}(N(w)) \leq 2b(k - 1) \), and we can do this independently for the various big vertices \( w' \) such that \( w \xrightarrow{e} w' \); since for any such \( w' \neq w \), we have \( N(w') \cap N(w) \subseteq N(w) \) (because otherwise, we would have an edge linking \( w \) and \( w' \), which are both big, and this would contradict the assumption that \( H \) is \( k \)-limited).

This completes the proof of Claim 1.

We now explain how a \( k \)-nice coloring \( c \) can help to encode the edges of \( G = \exp(H) \) as triples in \( D_H \times [m] \times \{m\} \), where \( H \) is normal, irredundant, and \( k \)-limited. Let \( e' \) be an edge of \( G \) linking \( x' \) to \( y' \) (let us recall that linking “to” means “directed from \( x \) to \( y \)” as opposed to “linking \( x \) and \( y \)” which does not specify any orientation), where \( x' \neq y' \).

**First Case.** \( e' \) is produced by a small edge \( e \) of \( H \) linking \( x \) to \( y \); then we let \( \gamma(x') = (e, c(x'), c(y')) \).

**Second Case.** \( e' \) is produced by a big edge \( e \) linking a big vertex \( x \) to a small vertex \( y \) or a small vertex \( y \) to a big vertex \( x \) then we let \( \gamma(x') := (x', c(e), c(y')) \) in the former case and \( \gamma(y') := (y', c(x'), c(e)) \) in the latter.

Since \( H \) is assumed to be irredundant, \( \gamma(x') \) is uniquely defined.

**Claim 2.** The mapping \( \gamma \) is one-to-one.

**Proof.** Let some triple \( (u, i, j) = \gamma_e(e') = \gamma_{e''}(e'') \) be given. If \( u \) is an edge, we are in the first case. Hence \( u = e \), \( e \) links \( x \) to \( y \) and \( x \) and \( y \) are small vertices. Let \( e' \) link \( x' \) to \( y' \) and \( e'' \) link \( x'' \) to \( y'' \).

We have \( x', x'' \in C(x) \) and \( y', y'' \in C(y) \), and \( i = c(x') = c(x''), j = c(y') = c(y'') \). Since \( c \) is \( k \)-nice, it is one-to-one on the sets \( C(x) \) and \( C(y) \), and we have \( x' = x'' \) and \( y' = y'' \). Hence \( e' = e'' \).

Let us now consider the case where \( u \) in the triple \( (u, i, j) = \gamma_e(e') = \gamma_{e''}(e'') \) is a vertex \( x' \). Hence \( e' \) and \( e'' \) are produced by big edges, \( e' \) and \( e'' \). There are two cases: either \( j > i \) and \( i \) colors an edge or \( i < j \) and \( j \) colors an edge. Without loss of generality, we consider only the first case. Then \( e' \) and \( e'' \) have both origin \( x' \). Let \( y' \) and \( y'' \) be their respective targets. There exists a unique big vertex \( w \) such that \( x' \xrightarrow{e''} v \xrightarrow{e} w \) and \( v \) is small. Hence, \( e' = e'' \) \( N(w) \). Since \( i = c(x') = c(x'') \) and \( c \) is \( k \)-nice, we have \( e' = e'' \). Let \( y \) be the target of this edge. This vertex is small (since the other edge \( e' \) is big and \( y', y'' \in C(y) \). We have also \( j = c(x') = c(x'') \) hence \( y' = y'' \) since \( c \) is nice and \( e' = e'' \) as desired (since \( G \) is simple).

This completes the proof of Claim 2.
iff \((u, i, j) = y(e)\) for some non-loop edge \(e\) of \(\exp(H)\) with label \(a\) and \(b\)-pair of vertices \((x, y)\).

**Proof.** The construction of \(\varphi_{a, i, j}\) is easy from the definitions and the proof of Claim 2. \qed

The proof or Proposition (4.5) can now be completed: the formulas \(\psi\) and \(\varphi_{a, i, j}\) can be combined in order to form a definition scheme \(\lambda\) defining the restriction of \(\exp\) to normal, irredundant, \(k\)-limited \(\varepsilon\)-graphs. This definition scheme uses parameters \(X_1, \ldots, X_m\), in order to “guess” a nice coloring of the given graph \(H\) and specifies \(E_{\exp(H)}\) as a subset of \(D_H = \{2, \ldots, p\} \cup D_H \times \{p + 1, \ldots, p + m\}\), where \(m = (2\ \text{Card}(A) + 1) \cdot (k - 1)\) and \(p = \text{Card}(A) + 1\). The loops are encoded in the first component of this union, and the other edges in the second (pairs of integers in \(m\)) are coded as integers in \(\{p + 1, \ldots, p + m\}\); hence, finally, it specifies \(D_{\exp(H)}\) as a subset of \(D_H \times \{p + m\}\). \qed

We can now complete the proof of our main theorem.

**Proof of Theorem (4.1), End.** Let \(L\) be a VR set of graphs such that, for some \(n\), the complete bipartite graph \(K_{n,n}\) is not a subgraph of \(\und(G)\) for any \(G\) in \(L\) (Condition (2)). We shall prove that \(L\) is HR.

Since \(L\) is VR, one can construct by Lemma (1.1), Proposition (2.4) and the remarks made in Definition (2.5), a HR set of normal \(\varepsilon\)-graphs \(M\) such that \(L = \exp(M)\). If \(M\) is not 2n-limited, then it follows from Lemma (4.4) that \(K_{2n}\) or \(K_{2n, 2n}\) is a subgraph of some graph in \(\und(\exp(M)) = \und(L)\). Since \(K_{2n}\) is a subgraph of \(K_{2n}\), as well as of \(K_{2n, 2n}\), this contradicts the hypothesis on \(L\). Hence \(M\) is 2n-limited. Since \(M\) is HR, it is the image of a recognizable set of trees \(N\) under a \((1, 2)\)-definable transduction \(\theta\). By Proposition (4.5), \(\exp\) is \((2, 2)\)-definable on a class of graphs containing \(\theta(N)\). Hence the composed transduction \(\theta; \exp\) is \((1, 2)\)-definable by Proposition (1.3) and \(L = \exp(\theta(N))\) is the image of a recognizable set of trees under a \((1, 2)\)-definable transduction. Hence it is HR by Theorem (1.4).

Finally, we prove that \(L\) is HR if it is sparse. Let \(M\) be as in the proof of (2) \(\Rightarrow\) (1). We use the notations of the proof of Proposition (4.5). One can construct a MS_{2} formula \(\varphi\) with free variables \(X, Y, U\) such that, for every graph \(H\) in \(M\)

\[
(\{H\}_{2}, X, Y, U) \models \varphi
\]

iff

\[
X = C(v), \ Y = D(v) \quad \text{and} \quad U = V_{\exp(H)}(X \cup Y)
\]

for some \(v\) in \(V_{H}\).

Then the set of triples of integers

\[
Z(M, \varphi) := \{(\text{Card}(X), \text{Card}(Y), \text{Card}(U)) \mid H \in M, (\{H\}_{2}, X, Y, U) \models \varphi\}
\]

is semi-linear by Corollary (3.3). Hence, \(Z(M, \varphi) = Z_{1} \cup \cdots \cup Z_{m}\) where each set \(Z_{i}\) is linear. If \(M\) is not \(k\)-limited for any \(k\), there exists at least one set \(Z_{k}\), containing triples \((x, y, u)\) where \(\text{Min}\{x, y\} \geq k\), for arbitrarily large \(k\). Let us write this set \(Z_{k} = \{(f_{1}(\tilde{w}), f_{2}(\tilde{w}), f_{3}(\tilde{w})) \mid \tilde{w} \in \mathbb{N}^{2}\}\). None of the functions \(f_{1}\) and \(f_{2}\) is constant. For \(q \in \mathbb{N}\), let

\[
(x_{q}, y_{q}, u_{q}) = (f_{1}(\tilde{q}), f_{2}(\tilde{q}), f_{3}(\tilde{q})) \in Z_{k},
\]

where \(\tilde{q} = (q, q, \ldots, q)\). Then \((x_{q}, y_{q}, u_{q}) = (\text{Card}(X_{q}), \text{Card}(Y_{q}), \text{Card}(U_{q}))\) for some \((\{H_{q}\}_{2}, X_{q}, Y_{q}, U_{q}) \models \varphi, H_{q} \in M\). Consider \(G = \exp(H_{q}) \in L\). It follows from the proof of Lemma (4.4) that:

\[
\text{Card}(E_G) \geq f_{1}(\tilde{q}) \cdot (f_{2}(\tilde{q}) - 1) = aq^{2} + bq + c
\]

we have \(\text{Card}(E_G) \geq f_{1}(\tilde{q}) \cdot f_{2}(\tilde{q})\) if \(X_{q}\) and \(Y_{q}\) are disjoint, and \(\text{Card}(E_G) \geq f_{1}(\tilde{q}) \cdot (f_{2}(\tilde{q}) - 1)\) if \(Y_{q}\) contains \(X_{q}\); see the proof of Lemma (4.4), and from the definition of \(\varphi\) that

\[
\text{Card}(V_G) \leq f_{1}(\tilde{q}) + f_{2}(\tilde{q}) + f_{3}(\tilde{q}) = a'q + b'
\]

for some constants \(a, b, c, a', b'\) where \(a > 0\), because \(f_{1}\) and \(f_{2}\) are not constant. Since \(q\) is arbitrarily large, we do not have any constant \(d\) such that \(\text{Card}(E_G) \leq d\cdot \text{Card}(V_G)\) for all \(G \in L\). Hence \(L\) is not sparse. It follows that \(M\) is \(k\)-limited for some \(k\). We conclude that \(L\) is HR as in the preceding case.

We now prove that these conditions are decidable. The sets \(Z_{1}, \ldots, Z_{m}\) constructed in the proof of (3) \(\Rightarrow\) (1) can be used: \(M\) is \(k\)-limited for some \(k\) if \(\varphi\) for each linear set \(Z_{k}\), at least one of the affine functions defining its first or second component is constant. This can be decided. If \(M\) is \(k\)-limited for some \(k\), then \(L\) is HR (see the implication (2) \(\Rightarrow\) (1)). If it is not, then Lemma (4.4) yields a contradiction with Condition (2). Hence \(L\) is not HR. Hence, one can decide whether a VR set is HR.

Here is another proof of this decidability result involving less complicated constructions, and perhaps closer to be implementable. Let \(\psi\) be the MS formula with free set variables \(X, Y\) such that for every graph \(G\),

\[
(\{G\}_{1}, X, Y) \models \psi
\]

iff

\[
X \cap Y = \emptyset, \text{and for every } x \in X, y \in Y, \text{there is an edge linking } x \text{ and } y.
\]

Since \(L\) is VR and \(\psi\) can be written without edge quantifications, the set \(Z(\{L\}_{1}, \psi)\) is effectively semi-linear (Corollary (3.3)). Then \(L\) satisfies Condition (2) iff each linear set composing \(Z(\{L\}_{1}, \psi)\) is constant on at least one of its coordinates. This is decidable by inspection of the expression of \(Z(\{L\}_{1}, \psi)\). \qed
5. EXTENSION TO HYPERGRAPHS

Our purpose is to extend the main theorem to VR sets of hypergraphs. We review some definitions.

We let \( A \) be a fixed finite ranked alphabet with rank mapping \( \rho : A \to \mathbb{N}_+ \). In the following definitions, we only indicate what differs with the case of graphs. A hypergraph over \( A \) is a tuple \( H = \langle V_H, E_H, \text{lab}_H, \text{vert}_H \rangle \) where \( E_H \) is the set of hyperedges (still called edges for simplicity) and \( \text{vert}_H \) associates with every edge \( e \) in \( E_H \) its sequence of vertices, which is of length \( \rho(\text{lab}_H(e)) \). We say that \( H \) is simple if no two edges have the same label and the same sequence of vertices. It is loop-free if no vertex occurs twice in any sequence \( \text{vert}_H(e) \). We denote by \( \text{HG}(A) \) the set of all hypergraphs over \( A \), by \( \text{HG}_s(A) \) the subset of simple ones, and by \( \text{HG}_s(A) \) the set of simple loop-free ones.

The reader may note that our hypergraphs have neither sources nor ports. Sources and ports are not necessary since we shall define or characterize VR and HR sets of hypergraphs in terms of definable transductions like we did for VR and HR sets of graphs in Theorem (1.4). However, the results we establish extend easily to hypergraphs with sources. A hypergraph \( H \) will be described by the following logical structures

\[
\begin{align*}
|H|_1 & := \langle V_H, (\text{edg}_{a|H})_{a \in A} \rangle \\
|H|_2 & := \langle V_H \cup E_H, (\text{edg}_{a|H})_{a \in A} \rangle,
\end{align*}
\]

where

\[
\begin{align*}
\text{edg}_{a|H}(x, y_1, ..., y_n) & : \equiv x \in E_H, y_1, ..., y_n \in V_H, \\
\text{lab}_H(x) & = a \quad \text{and} \\
\text{vert}_H(x) & = (y_1, ..., y_n), \\
\text{edg}'_{a|H}(y_1, ..., y_n) & := \text{edg}_{a|H}(x, y_1, ..., y_n) \\
& \text{holds for some } x \in E_H.
\end{align*}
\]

It is clear that \( H = H' \) iff \(|H|_2 = |H'|_2\) for any two hypergraphs \( H \) and \( H' \), and that, if \( H \) and \( H' \) are simple, then \( H = H' \) iff \(|H|_1 = |H'|_1\). In order to compare these representations, we let

\[
\text{tr}(\mathcal{F}) := \{ (|H|_1, |H|_2) / H \in \mathcal{F} \},
\]

where \( \mathcal{F} \) is a subset of \( \text{HG}_s(A) \). This transduction is functional and one-to-one. As in [9], we shall consider sets \( \mathcal{F} \) such that \( \text{tr}(\mathcal{F}) \) is definable.

A HR set of hypergraphs is a subset \( L \) of \( \text{HG}(A) \) such that \(|L|_2\) is the image of a recognizable set of trees under a definable transduction. This definition is equivalent to the classical ones by the result of [13], which extends the first assertion of Theorem (1.4) to hypergraphs.

A VR set of hypergraphs is defined as a subset \( L \) of \( \text{HG}_s(A) \) such that \(|L|_1\) is the image of a recognizable set of trees under a definable transduction. An equivalent characterization by means of systems of equations over operations extending those of \( \forall(A, P) \) can be found in [10, Theorem (4.6)]. An equivalent notion of context-free hypergraph grammar remains to be defined.

Tree-width is an important complexity measure of graphs and hypergraphs. Let \( H \in \text{HG}(A) \). A tree-decomposition of \( H \) is a pair \((T, g)\), where \( T \) is an undirected tree and \( g \) is a mapping from \( V_T \), the set of nodes (i.e., vertices) of \( T \), into \( \omega(V_H) \) such that

\[
\begin{align*}
(1) & \quad V_H = \bigcup \{ g(v) / v \in V_T \}, \\
(2) & \quad \text{every edge of } H \text{ has all its vertices in } g(v) \text{ for some } v \in V_T, \\
(3) & \quad \text{for every vertex } w \text{ of } H, \{ v \in V_T / w \in g(v) \} \text{ induces a connected subgraph of } T.
\end{align*}
\]

The width of \((T, g)\) is \( \text{wd}(T, g) := \max \{ \text{Card}(g(v)) - 1 / \forall v \in V_T \} \) and the tree-width of \( H \) is the minimum width of a tree-decomposition of \( H \), denoted by \( \text{tw}(H) \). We let \( \mathcal{H}_k \) denote the set of hypergraphs of tree-width at most \( k \).

An important property of HR sets of hypergraphs is that they have bounded tree-width [7, 8, 12].

In order to extend Theorem (4.1) to hypergraphs, we recall from [7] a transformation from hypergraphs to graphs. For every hypergraph \( H \), we let \( K(H) \) denote the graph obtained by the substitution in \( H \) of a complete graph with \( n \) vertices for each hyperedge of \( H \) having \( n \) vertices; if \( L \) is a set of hypergraphs, we let \( K(L) \) denote the set of graphs \( K(H) \) for \( H \) in \( L \). (The orientations and labels of edges of the graphs in \( K(L) \) are actually irrelevant in the present use of \( K \).) If \( L \) is HR, then \( K(L) \) is HR too because the family of HR sets of hypergraphs is closed under substitutions of hypergraphs for hyperedges. Another useful fact is that \( \text{tw}(H) = \text{tw}(K(H)) \) for every hypergraph \( H \) (see [7]).

The following theorem extends both Theorem (4.1) and Theorem (3.9) of [13]. The latter result establishes (1) \( \iff \) (2) for VR sets of graphs.

5.1 Theorem. Let \( L \) be a VR set of hypergraphs. The following conditions are equivalent:

\[
\begin{align*}
(1) & \quad L \text{ is HR}, \\
(2) & \quad L \text{ has bounded tree-width}, \\
(3) & \quad \text{there exists an integer } n \text{ such that } K_{n,n} \text{ is not a subgraph of } \text{und}(G) \text{ for any graph } G \text{ in } K(L), \\
(4) & \quad L \text{ is sparse}.
\end{align*}
\]

One can decide whether these conditions hold.

The proof will use the following result, which extends Theorem (2.2) of [9] and Lemma (3.6) of [13].
(5.2) Theorem. For every k, the transduction \( \text{tr}(\text{HG}_d(A) \cap A) \) is definable.

This result is known from [13, Lemma (3.6)] in the special case of loop-free graphs. We shall reduce the general case to this case by means of a coding of hypergraphs by graphs, more complicated than the mapping \( K \) recalled above.

Let \( \bar{A} := \{ \bar{a} \mid a \in A \} \) be a new set of symbols of rank 2 and we also let \( \mathcal{L} \) be new symbols of rank 2. For \( H \) in \( \text{HG}_d(A) \) we let \( \mathcal{H}(H) \in \mathcal{G}(\bar{A} \cup \{ \mathcal{L}, \mathcal{S}, \mathcal{O} \}) \) be the graph \( G \) defined as follows:

(i) \( \mathcal{V}_G = \mathcal{V}_H \cup \mathcal{E}_H \cup \mathcal{U}_H \), where

\[
\mathcal{U}_H := \{(v_1, \ldots, v_p) \mid p \geq 1, (a, v_1, \ldots, v_p, v_{p+1}, \ldots, v_n) \in \mathcal{E}_H \}
\]
for some \( a \in A \), some \( v_1, \ldots, v_n \in \mathcal{V}_H \), where \( n > p \).

(II) \( \mathcal{E}_G \) consists of the following edges:

\[
(a, v_1, \ldots, v_n, \mathcal{L}) \quad (\text{where } n \geq 1)
\]
\[
(\mathcal{S}, (v_1, \ldots, v_p, \mathcal{L}) \quad (\text{where } p \geq 2)
\]
\[
(\mathcal{L}, (v_1, \ldots, v_p), (v_1, \ldots, v_{p+1})) \quad (\text{where } p \geq 1)
\]
\[
(\mathcal{L}, (v_1, \ldots, v_{n-1}), (a, v_1, \ldots, v_n)) \quad (\text{where } n \geq 2)
\]

for every \( (a, v_1, \ldots, v_n) \in \mathcal{E}_H, (v_1, \ldots, v_p), (v_1, \ldots, v_{p+1}) \in \mathcal{U}_H \).

Figure 6.1 shows a hypergraph \( H \). Its edge labels are \( a, b, c, d \) of respective ranks 4, 3, 2, 1. Its vertices are numbered from 1 to 7. The hyperedges are \((a, 1, 2, 3, 4), (a, 1, 2, 5, 6), (b, 1, 2, 3), (b, 7, 3, 4), (c, 1, 2), \) and \((d, 7)\). The graph \( \mathcal{H}(H) \) is shown in Fig. 6.4. The initial parts of the sequences of vertices of the hyperedges are made into vertices, giving (1, 2), (1, 2, 3), (1, 2, 5), (7, 3). The hyperedges of \( H \) are also made into vertices of \( \mathcal{H}(H) \); see Fig. 6.4. An S-labelled edge links a vertex that is a sequence to the vertex of \( H \) that is the last element of this sequence. An \( \bar{a} \)-labelled edge links a vertex that is a hyperedge with label \( a \) to its last vertex, and similarly for \( b, c, d \).

(5.3) Lemma. The transduction \( \mathcal{H} \) is functional and one-to-one on \( \text{HG}_d(A) \). The transduction \( \mathcal{H}^{-1}: \mathcal{G}(\bar{A} \cup \{ \mathcal{L}, \mathcal{S}, \mathcal{O} \}) \to \text{HG}_d(A) \) is (1, 2)-definable.

Proof. That \( \mathcal{H} \) is one-to-one is clear from the observation that \( H \) can be reconstructed from \( G = \mathcal{H}(H) \) as follows. Let \( x \in \mathcal{V}_G \).

(1) This vertex is a vertex of \( H \) if and only if there is no edge of the form \((\mathcal{L}, x, y)\) or \((\bar{a}, x, y)\) for any \( a \in A \) and any \( y \in \mathcal{V}_G \).

(2) On the other hand, \( x \) is an edge of \( H \) iff there is an edge of the form \((\bar{a}, x, y)\) for some \( y \in \mathcal{V}_G \) and \( a \in \bar{A} \). The label \( \mathcal{L}(x) \) is \( a \). Let \( n \) be its rank. If \( n = 1 \) then the unique vertex of \( x \) (as an edge of \( H \)) is \( y \); if \( n > 1 \), the sequence of vertices \( \mathcal{L}(x) \) is the (unique) sequence \((v_1, v_2, \ldots, v_n)\) of vertices of \( H \) such that for some \( x_2, x_1, \ldots, x_{n-1} \in \mathcal{V}_G \) we have \( v_n = y \) and we have edges of the forms:

\[
(\mathcal{L}, x, x_{n-1}, x), (\mathcal{S}, x_{n-1}, v_{n-1}) \quad \text{for } i = n - 2, n - 3, 2, \]

\[
(\mathcal{L}, v_1, x_2).
\]

This gives the following picture:

![Diagram](attachment:image.png)

Note that \( v_1 \) is the origin of the unique maximal path with end \( x \), all edges of which have label \( \mathcal{L} \).

It is easy to write these conditions in monadic second-order logic (actually in first-order logic since no set variable is necessary). It follows that the transduction \( \{ (\mathcal{H}(H))_1, (\mathcal{H}(H))_2 \mid H \in \text{HG}_d(A) \} \) is definable, i.e., that \( \mathcal{H}^{-1} \) is (1, 2)-definable. \( \square \)

Our objective is to bound the tree-width of \( \mathcal{H}(H) \) in terms of \( H \). To do so, we shall express \( \mathcal{H}(H) \) as \( \theta^{n-1}(H) \), where \( n \) is the maximum rank of a symbol in \( A \) and \( \theta \) is a transformation that squares the tree-width, which we now describe. We shall deal with simple loop-free hypergraphs with edges labelled by symbols in \( A \cup \bar{A} \cup \{ \mathcal{L}, \mathcal{S} \} \). The edges labelled by symbols in \( A \cup \{ \mathcal{L}, \mathcal{S} \} \) will have two vertices. The other ones will have two or more vertices. If \( K \) is such a hypergraph, we let \( K' = \theta(K) \) be as follows:

(i) \( \mathcal{V}_K' = \mathcal{V}_K \cup \{ \langle v_1, v_2 \rangle \mid \langle a, v_1, v_2, \ldots, v_n \rangle \in \mathcal{E}_K \}, \) for some \( a \in A \), some \( n \geq 3 \), some \( v_1, v_2, \ldots, v_n \in \mathcal{V}_K \) \( \cup \{ \langle a, v_1, v_2 \rangle \in \mathcal{E}_K \mid a \in A, v_1, v_2 \in \mathcal{V}_K \} \cup \{ \langle a, v_1 \rangle \in \mathcal{E}_K \mid a \in A, v_1 \in \mathcal{V}_K \} \).

(ii) \( \mathcal{E}_K' \) is constructed as follows:

- for every edge \( \langle a, v_1, v_2, \ldots, v_n \rangle \) of \( \mathcal{E}_K \) with \( a \in A, n \geq 3 \), we put in \( \mathcal{E}_K' \) the edges \( \langle \mathcal{L}, v_1, (v_1, v_2) \rangle \), \( \langle (v_1, v_2), v_3, \ldots, v_n \rangle \);
- for every edge \( \langle a, v_1, v_2 \rangle \) of \( \mathcal{E}_K \) with \( a \in A \), we put in \( \mathcal{E}_K' \) the edges \( \langle \mathcal{L}, (v_1, a, v_2) \rangle \) and \( \langle \bar{a}, (v_1, v_2) \rangle \);
- for every edge \( \langle a, v_1 \rangle \) of \( \mathcal{E}_K \) with \( a \in A \), we put in \( \mathcal{E}_K' \) the edge \( \langle \bar{a}, (v_1, v_2) \rangle \);
- all other edges of \( K \) are of the form \( (b, v, w) \) for \( b \in \bar{A} \cup \{ \mathcal{L}, \mathcal{S} \} \) and are left in \( K' \).
Let us note that the rank of $a$ in $A$ varies as follows. If $a$ is of rank $n \geq 3$ in $K$, it is of rank $n - 1$ in $\theta(K)$. If $a$ is of rank 1 or 2, then it disappears: every edge labeled by such an $a$ is replaced by an edge of rank 2 labelled by $a$. The following figures show a hypergraph $H$ (with $n = 4$), the hypergraph $\theta(H)$, and the graphs $\theta^2(H)$ and $\theta^3(H)$. We claim that $\theta^3(H) = f(H)$.

Figure 6.1 shows $H$. Let us recall that $a, b, c, d$ are of respective ranks 4, 3, 2, 1. Figure 6.2 shows the hypergraph $\theta(H)$. Its "new" vertices, i.e., those that are not in $H$ are represented by circles. The labels $a$ and $b$ are of respective ranks 3 and 2 in $\theta(H)$. The labels $c$ and $d$ have been replaced by $\bar{c}$ and $\bar{d}$, both of rank 2.

Figure 6.3 shows the graph $\theta(\theta(H))$. Its vertices not in $\theta(H)$ are represented by squares. All labels are of rank 2, but there remains one namely $a$, belonging to $A$. Figure 6.4 shows $\theta(\theta(H)) = f(H)$. Its vertices not in $\theta(\theta(H))$ are ovals.

**Lemma.** Let $n = \text{Max}\{p(a) / a \in A\}$ and $m = 2^n - 1$. Then for every $H \in \text{HG}_{n+1}(A)$ of tree-width at least 2, $\text{twd}(f(H)) \leq (\text{twd}(H) + 1)^n$.

**Proof.** Let $H \in \text{HG}_{n+1}(A)$. The graph $G = f(H)$ is the last element of a sequence $H = H_0, H_1, \ldots, H_{n-1} = G$ of hypergraphs such that $H_{i+1} = \theta(H_i)$ for $i = 0, 1, \ldots, n-2$.

Let $(T, g)$ be a tree-decomposition of width $k$ of an arbitrary hypergraph $K$, so that Card$(g(s)) \leq k + 1$ for each node $s$ of $T$. We shall construct a tree-decomposition $(T', g')$ of $K' = \theta(K)$ of width at most $(k + 1)^2 - 1$. We must "enlarge" $(T, g)$ into $(T', g')$ so as to "accommodate" the "new" vertices $x \in V_{K'} - V_K$ which are of three types:

* **Type 1:** $x = (v_1, v_2)$.
* **Type 2:** $x = (a, v_1, v_2)$.
* **Type 3:** $x = (a, v_1)$.

For the vertices of Type 1 we let

$$g'(s) := g(s) \cup \{(v_1, v_2) \mid v_1, v_2 \in g(s), (v_1, v_2) \in V_{K'}\}.$$

For each vertex $x = (a, v_1, v_2)$ of Type 2 (resp. $x = (a, v_1)$ of Type 3) there is some node $s$ of $T$ such that $v_1$ and $v_2 \in g(s)$ (resp. $v_1 \in g(s)$). We add to $T$ a new node $s_x$, we link it to $s$, and we let

$$g'(s_x) := \{x, v_1, v_2\} \quad \text{(resp. } g'(s_x) := \{x, v_1\}).$$
We obtain in this way a pair \((T', g')\) that is a tree-decomposition of \(K'\) (the verification is easy). For every \(s \in V_T\)

\[
\text{Card}(g'(s)) = k + 1 + k(k + 1) = (k + 1)^2
\]

and for every \(s \in V_{T'} - V_T\), we have \(\text{Card}(g'(s)) \leq 3\). Hence,

\[
\text{twd}(\theta(K)) + 1 \leq \max \{ \text{Card}(g'(s)) \mid s \in V_{T'} \}
\]

\[
\leq (k + 1)^2 = (\text{twd}(K) + 1)^2.
\]

Since \(G = \theta^{-1}(H)\), we get \(\text{twd}(G) + 1 \leq (\text{twd}(H) + 1)^m\), which gives the desired result. \(\square\)

\[(5.5)\] \textbf{Lemma.} For every \(k\), the restriction of the transduction \(\theta\) to the set \(\HG_{\text{gl}}(A \cup \overline{A} \cup \{E, S\}) \cap \mathcal{F}_k\) is \((1, 1)\)-definable.

\textbf{Proof.} Let \(K \in \HG_{\text{gl}}(A \cup \overline{A} \cup \{E, S\}) \cap \mathcal{F}_k\) and let \(K'\) be its image under \(\theta\). We want to prove the existence of a definition scheme \(\mathcal{D}\) (independent of \(K\)) such that \(|K'|_1 = \text{def}_\mathcal{D}(|K|_1)|.

We must represent the vertices in \(V_{K'} - V_K\) as pairs \((x, l)\) for \(x \in V_K\) and \(1 \leq l \leq m\), where \(m\) depends only on \(A\) and \(k\). These "new" vertices are actually either the edges of \(K\) of rank 1 or the edges of the graph \(K''\), defined as follows:

- \(V_{K''} = V_K\).
- the edges of \(K''\) are the edges \((a, v_1, v_2)\) of \(K\) with \(a \in A\), and edges of the form \((a, v_1, v_2, ..., v_n)\) for \(a \in A\), where \(n \geq 3\), some \(v_1, v_2, ..., v_n\) and \(a \in A\), where \(a\) is a new symbol.

Since \(K''\) has tree-width at most \(k\) (being some kind of subgraph of \(K\)), one can construct by \([13, \text{Lemma } (3.6)]\) a definition scheme \(\mathcal{D}'\) such that

\[
|K''|_2 = \text{def}_\mathcal{D}'(|K''|_1).
\]

Since \(K''\) is \((1, 1)\)-definable in \(K\), the scheme \(\mathcal{D}'\) can be transformed into \(\mathcal{D}\) such that

\[
|K'|_2 = \text{def}_\mathcal{D}(|K|_1).
\]

The set \(V_{K'} - V_K\) that we are aiming to encode as a subset of \(V_K \times [m]\) (for some fixed \(m\) we shall actually not determine precisely) consists of the edges of \(K\) of rank 1 and of those of \(K''\) (or rather, of a set in bijection with \(E_{K''}\)). An edge \((a, v)\) of \(K\) of rank 1 can be encoded as a pair \((v, i)\) where the integer \(i\) encodes the label \(a\). The encoding of the edges of \(K''\) can be obtained from \(\mathcal{D}'\). Hence, one can construct \(\mathcal{D}\) such that

\[
|K'|_1 = \text{def}_\mathcal{D}(|K|_1),
\]

which completes the proof. \(\square\)

We are now ready to prove Theorem \((5.2)\), which states that the transduction \(\text{tr}(\HG_{\text{gl}}(A) \cap \mathcal{F}_k) = \{|[H], [H'] | H \in \HG_{\text{gl}}(A) \cap \mathcal{F}_k\}\) is definable.

\textbf{Proof of Theorem \((5.2)\).} Let \(H \in \HG_{\text{gl}}(A)\). We first make it into a loop-free hypergraph \(\text{ll}(H) \in \HG_{\text{gl}}(A')\), where \(A'\) is a new finite ranked alphabet containing \(A\). Every (hyper) edge with loops such as \((a, x, y, x, z, y),\) where \(x, y, z\) are pairwise distinct vertices, is replaced by the edge \((a', x, y, z)\), where \(a'\) is the new label \(a\). \(1, 2, 3, 2\). To take another example, a double loop of the form \((b, x, x, x)\) becomes an edge \((b', x)\) of rank 1, where \(b'\) is the new label \((b, 1, 1, 1)\).

It is clear that the transductions

\[
\{|[H], \text{ll}(H) | H \in \HG_{\text{gl}}(A)\}\}
\]

and

\[
\{|\text{ll}(H)|, [H] | H \in \HG_{\text{gl}}(A)\}\}
\]

are definable. It follows that we need only establish the result for \(\text{tr}(\HG_{\text{gl}}(A) \cap \mathcal{F}_k)\), that is, for simple loop-free hypergraphs of tree-width at most \(k\). The transduction \(\text{tr}(\HG_{\text{gl}}(A) \cap \mathcal{F}_k)\) can be expressed as the composition of

\[
\{|[H], \text{ll}(H) | H \in \HG_{\text{gl}}(A) \cap \mathcal{F}_k\}
\]

and

\[
\{|\text{ll}(H)|, [H] | H \in \HG_{\text{gl}}(A) \cap \mathcal{F}_k\}\}
\]

The first one is definable since it is the \((n - 1)\)-fold composition of \(\theta\) on simple loop-free hypergraphs of tree-width at most \((k + 1)^m\) (by the proof of Lemma \((5.4)\)), since the restriction of \(\theta\) to any set \(\mathcal{F}_k\) is \((1, 1)\)-definable by \(\text{Lemma (5.5)}\), and since the composition of two definable transductions is definable \((\text{Proposition (1.3)})\). The second one is definable by \(\text{Lemma (5.3)}\); hence the transduction \(\text{tr}(\HG_{\text{gl}}(A) \cap \mathcal{F}_k)\) is definable, which concludes the proof. \(\square\)

\textbf{Proof of Theorem \((5.1)\).} Let \(L\) be a VR set of simple hypergraphs.

Let us first assume that \(L = \text{HR}(\text{Condition (1)})\). Then \(L\) has bounded tree-width \((\text{Condition (2)})\) and so has \(K(L)\) \(\text{(since } \text{twd}(K(G)) = \text{twd}(G) \text{ for every hypergraph } G)\), hence \(\text{und}(K(L))\) cannot contain large \(K_{n,n}\)'s as subgraphs \((\text{see the argument used in the proof of Theorem (4.1)})\). Hence \(\text{Condition (3)}\) holds. That \(L\) is sparse is proved in \(\text{Lemma (5.6)}\) below; hence \(\text{Condition (4)}\) holds.

Let us now assume that \(\text{twd}(H) \leq k\) for all \(H \in L\) \(\text{(Condition (2)})\). Since \(L\) is VR, we have \(|L|_1 = 0(K)\) for some definable transduction \(0\) and some recognizable set of trees \(K\). Hence \(|L|_2 = \theta'(K)\) where \(\theta' = \theta \circ \text{tr}(\HG_{\text{gl}}(A) \cap \mathcal{F}_k)\). This transduction is definable by \(\text{Theorem (5.2)}\). Hence \(L\) is HR by \(\text{Theorem (1.4)}\).
Let us assume Condition (3). If \( H \) is a hypergraph, we let \( K'(H) \) be obtained from \( K(H) \) by fusing multiple edges. Hence, the mapping \( K' \) is a \((1, 1)\)-definable transduction. It follows from Theorem (1.5) that \( K'(H) \) is VR. Since \( K_n, n \subseteq \text{und}(K(H)) \) if and only if it is a subgraph of \( \text{und}(K'(H)) \), it follows from Theorem (4.1) that \( K'(L) \) is HR, hence has bounded tree-width. It follows that \( L \) has bounded tree-width since a hypergraph \( H \) has the same tree-width as \( K'(H) \). Hence Condition (2) holds.

If \( L \) is sparse then \( K'(L) \) is sparse too (because the edges of the hypergraphs in \( L \) have bounded rank). Hence, \( K'(L) \) is VR and sparse, hence, is HR by Theorem (4.1), and Condition (2) holds by the same argument as above.

Hence we have proved that Conditions (1) to (4) are equivalent. Their decidability follows from the decidability result of Theorem (4.1) applied to \( K'(L) \) which can be effectively described as the image of a recognizable set of trees under a \((1, 1)\)-definable transduction. This concludes the proof.

This proof has used the following lemma.

(5.6) Lemma. Every HR set of simple hypergraphs is sparse.

Proof. We must prove that if \( L \subseteq \text{HG}_1(A) \) is HR, then \( \text{Card}(E_H) \leq k \cdot \text{Card}(V_H) \) for every \( H \in L \), where \( k \) is some constant that does not depend on \( H \). One can construct a monadic second-order formula \( \varphi(X, Y) \) such that, for every \( H \in \text{HG}_1(A) \),

\[
(H_2, X, Y) = \varphi \iff X = V_H \text{ and is not empty, } Y \text{ is a subset of } E_H \text{ such that no two edges in } Y \text{ have the same label and the same sequence of vertices, and every edge in } E_H \text{ has the same sequence of vertices and the same label as some edge in } Y.
\]

Let us establish that there exists a constant \( k \) such that \( y = k \cdot x \) for every \( (x, y) \in Z(\{L_2\}, \varphi) \). This establishes the desired result since, if \( H \) is simple, \((H_2, X, Y) = \varphi \) iff \( X = V_H \) and \( Y = E_H \). By Theorem (3.3), the set \( Z(\{L_2\}, \varphi) \) is a finite union of linear sets \( Z_1, ..., Z_m \). It is enough to establish the desired inequality for each of them. Let \( Z \) be any of the sets \( Z_1, ..., Z_m \). Then

\[
Z = \{(f(\bar{w}), g(\bar{w})) \mid \bar{w} \in \bar{N}' \}
\]

where

\[
f(\bar{w}) = \alpha_1 w_1 + \alpha_2 w_2 + \cdots + \alpha_r w_r + \beta,
\]

\[
g(\bar{w}) = \gamma_1 w_1 + \gamma_2 w_2 + \cdots + \gamma_r w_r + \delta,
\]

for some \( \alpha_1, ..., \alpha_r, \beta, \gamma_1, ..., \gamma_r, \delta \in \mathbb{N} \) and \( \bar{w} \) denotes \((w_1, w_2, ..., w_r) \).

We can assume that \( \alpha_i \) and \( \gamma_i \) are not both 0 for any \( i \) (because otherwise they can be omitted). Furthermore, \( \alpha_i \neq 0 \) for all \( i = 1, ..., r \); otherwise, by taking large values of \( w_i \) and fixed values of \( w_1, ..., w_{r-1}, w_{r+1}, ..., w_r \), one obtains hypergraphs \( H \) with fixed numbers of vertices and arbitrary large sets of edges \( Y \) satisfying \( \varphi(V_H, Y) \), which is impossible, since \( A \) is finite, and by the definition of \( \varphi \). Note also that \( \beta \neq 0 \) since \( \varphi \) says that \( X \) is not empty. Elementary calculation gives that the desired inequality holds with

\[
k = \max\{\delta/|\beta|, \gamma_i / |\alpha_i| \mid 1 \leq i \leq r \}.
\]

We shall conclude this paper with some remarks on the coding \( f \) of simple loop-free hypergraphs by graphs. We have the following result.

(5.7) Proposition. For every set \( L \) of simple loop-free hypergraphs, the following conditions are equivalent:

(1) \( L \) is HR,
(2) \( f(L) \) is HR,
(3) \( f(L) \) is VR.

Proof. Let \( L \) be HR; it is the image of a recognizable set of trees under a \((1, 2)\)-definable transduction by Theorem (1.4). It follows from Lemma (5.5) that the transduction \( f \) is \((1, 1)\)-definable. Hence, it is also \((2, 1)\)-definable and \( f(L) \) is the image of a recognizable set of trees under a \((1, 1)\)-definable transduction (by Proposition (1.3)). Hence, \( f(L) \) is VR by Theorem (1.4). Since by Lemma (5.4) the graphs in \( f(L) \) have bounded tree-width, the set \( f(L) \) is HR by the result of [13] saying that a VR set of graphs of bounded tree-width is HR.

If \( f(L) \) is HR, then it is VR since it consists of simple graphs.

Finally, if \( f(L) \) is VR, then it is the image of a recognizable set of trees under a \((1, 1)\)-definable transduction. Hence, \( L = f^{-1}(f(L)) \) is the image of a recognizable set of trees under a \((1, 2)\)-definable transduction by Lemma (5.3) and Proposition (1.3), hence is HR by Theorem (1.4).

This result shows that the coding \( f \) gives a method to study HR sets of simple loop-free hypergraphs as HR sets of simple loop-free graphs. However, it does not help for VR sets of hypergraphs. We do not know the structure of the sets \( f(L) \) when \( L \) is VR and not HR.

ACKNOWLEDGMENTS

Many thanks to J. Engelfriet and the referees for their numerous suggestions.

Received April 15, 1992; final manuscript received November 25, 1992

REFERENCES


