On Families of Graphs Having a Decidable First Order Theory with Reachability

Thomas Colcombet
Thomas.Colcombet@irisa.fr

Irisa, Campus de Beaulieu, 35042, Rennes, France

Abstract. We consider a new class of infinite graphs defined as the smallest solution of equational systems with vertex replacement operators and unsynchronised product. We show that those graphs have an equivalent internal representation as graphs of recognizable ground term rewriting systems. Furthermore, we show that, when restricted to bounded tree-width, those graphs are isomorphic to hyperedge replacement equational graphs. Finally, we prove that on a wider family of graphs — interpretations of trees having a decidable monadic theory — the first order theory with reachability is decidable.

1 Introduction

Automatic verification of properties on programs is one of the challenging problems tackled by modern theoretical computer-science. An approach to this kind of problems is to translate the program in a graph, the property in a logic formula and to use a generic algorithm which automatically solves the satisfaction of the formula over the graph. The use of potentially unbounded data structures such as integers, or stacks in programs leads to infinite graphs. Thus, algorithms dealing with infinite graphs are needed. Many families of infinite graphs have been recently described. For the simplest one, it is possible to verify automatically powerful formulas. For the most complex families, nearly nothing can be said. We are interested here into three logics. The less expressive is the first order logic. The first order logic with reachability extends it with a reachability predicate. The most expressive is the monadic (second order) logic.

The reader may find a survey on infinite graphs in [16]. The study of infinite graphs started with pushdown graphs [11]. Vertices are words and edges correspond to the application of a finite set of prefix rewriting rules. The first extension is the family of HR-equational graphs [5] defined as the smallest solutions of equational systems with hyperedge replacement (HR) operators. The more general family of prefix recognizable graphs [3] is defined internally as systems of prefix rewriting by recognizable sets of rules (instead of finite sets for pushdown graphs). VR-equational graphs are defined as the smallest solutions of equational systems with vertex replacement (VR) operators. VR-equational graphs are isomorphic to prefix recognizable graphs [1]. All those families of graphs share a decidable monadic theory (those results are, in some sense, extensions of the famous decidability result of Rabin [12]). Some more general families have also been introduced. Automatic graphs are defined by synchronized transducers on words [14, 2]. Only the first order theory remains decidable and the reachability problem cannot be handled anymore. The class of rational graphs is defined by general transductions [10]. The first order theory is not decidable anymore. The common point of all those families is that they are (explicitly or not) defined as rewriting systems of words. This is not true anymore with the ground term rewriting systems [9]. Vertices are now terms and transitions are described by a finite set of ground rewriting rules. A practical interest of this family is that it has a decidable first order theory with reachability (but not a decidable monadic theory) [7, 4]. Studies have also been pursued on external properties of graphs. VR-equational graphs of bounded tree-width (this notion is known for long in the theory of finite graphs, see [13] for a survey) are HR-equational graphs [1]. Pushdown graphs are VR-equational graphs of finite degree [3]. Ground term rewriting systems of bounded tree-width are also pushdown graphs [9].

In this paper, we define a new family of infinite graphs, namely, the VRP-equational graphs. It is a natural extension of the VR-equational graphs (solution of equational systems with vertex replacement operators) with an unsynchronised product operator (VRP stands for vertex replacement with product). VRP-equational graphs are formally defined as interpretations of regular infinite trees. The first result of this paper gives an equivalent internal representation to VRP-equational graphs — the recognizable ground term rewriting systems. Secondly, we study the VRP-equational graphs of bounded tree-width and prove that those graphs are isomorphic to the HR-equational graphs. Finally, we show the decidability of the first order theory with reachability on a more general family of graphs, the graphs obtained by VRP-interpretation of infinite trees having a decidable monadic theory. Recent results tend to prove that important families of trees have a decidable monadic theory (algebraic trees [6] and higher order trees with a safety constraint [8]).

The remaining of this article is organized as follows. Section 2 gives the basic definitions. Section 3 describes VRP-equational graphs. The following three sections are independent. Section 4 introduces recognizable ground term rewriting systems and states the isomorphism with VRP-equational graphs. In Section 5 we study the VRP-equational graphs of bounded tree-width. In Section 6 we study the first order logic with reachability of VRP-interpretation of trees.

2 Definitions

We design by $\mathbb{N}$ the set of integers. The notation $[n]$ stands for $\{0, \ldots, n-1\}$. Let S be a set of symbols, S^* is the set of words with letters in S. The empty word is written ε. The length of a word w is written $|w|$.

Let Θ be a finite set of base types. A *typed alphabet* $\mathcal{F}$ (over Θ) is a family $(\mathcal{F}_\tau)_{\tau \in \Theta^* \times \Theta}$ where for all τ, $\mathcal{F}_\tau$ is the set of symbols of type τ. We assume that alphabets are finite. We will use notations such as $\mathcal{F} = \{f : \tau \mid f \in \mathcal{F}_\tau\}$ instead of describing the $\mathcal{F}_\tau$ sets separately. The types of the form (ε, θ) are simply written θ. For any symbol f in $\mathcal{F}_{(\theta_0\ldots\theta_{i-1},\theta)}$, the base type θ is the *codomain* of f, the integer i is the *arity* and the base type θ_k is the type of the $k+1$-th argument of f.

A *tree* t (over the typed alphabet $\mathcal{F}$) is a function from $\mathbb{N}^*$ into $\mathcal{F}$ with a non-empty prefix closed domain D_t. The elements of D_t are called *nodes*, and the node ε is the *root* of the tree. A tree t is *well typed* if for all node v and all integer k, vk is a node if and only if k is smaller than the arity of $t(v)$, and, in this case, the codomain of $t(vk)$ is the type of the $k+1$-th argument of $t(v)$. The *type* of a well typed tree is the codomain of its root symbol. Let t be a well typed tree and v one of its nodes, the *subtree* of t rooted at v is the well typed tree t^v defined for all node vu by $t^v(u) = t(vu)$. The set of subtrees of t is $sub(t)$. A tree t is *regular* if $sub(t)$ is finite.

We call *terms* the trees of finite domain. We denote $\mathcal{T}^\infty_\theta(\mathcal{F})$ the set of well typed trees over $\mathcal{F}$ of type θ, and $\mathcal{T}_\theta(\mathcal{F})$ the set of well typed terms over $\mathcal{F}$ of type θ. $\mathcal{T}^\infty(\mathcal{F})$ is the set of all well typed trees of any type (resp. $\mathcal{T}(\mathcal{F})$ for terms).

Here, we only consider infinite trees over ranked alphabets. A *ranked alphabet* is an alphabet typed over only one base type. Types are uniquely identified by their arity. We write n instead of (θ^n, θ) (where θ is the only base type). We want to describe such infinite trees as limits of sequences of 'growing' terms. We slightly extend the alphabet and equip the corresponding trees with a structure of *complete partial order* (cpo) in which sequences of growing terms are chains, and limits are least upper bounds. Formaly, let us define the new typed alphabet $\mathcal{F}_\perp = \mathcal{F} \cup \{\perp : 0\}$ where $\perp$ is a new symbol. We define a binary relation $\sqsubseteq$ over $\mathcal{T}^\infty(\mathcal{F}_\perp)$. Let t_1 and t_2 be trees, then $t_1 \sqsubseteq t_2$ states if $D_{t_1} \subseteq D_{t_2}$ and for all nodes v of t_1, $t_1(v) = t_2(v)$ or $t_1(v) = \perp$. This relation is a cpo — it has a smallest element, the tree with symbol $\perp$ at root, and the least upper bound of a chain of trees $(t_i)_{i\in\mathbb{N}}$ is $\sqcup t$ with $(\sqcup t)(v) = f$ where f is the symbol such that $v \in D_{t_j}$ and $t_j(v) = f$ for all $j \geq k$ for some k. Let t be a tree of $\mathcal{T}^\infty(\mathcal{F}_\perp)$, the *cut at depth* n of t, written $t\downarrow_n$, is the term defined by $t\downarrow_n(v) = t(v)$ for all $v \in D_t$ such that $|v| < n$, and $t\downarrow_n(v) = \perp$ for all $v \in D_t$ such that $|v| = n$. The sequence $(t\downarrow_n)_{n\in\mathbb{N}}$ is a chain of terms of least upper bound t.

3 VRP-graphs

In this section, we describe the cpo of colored graphs (or simply graphs) and the VRP operators working on them. The interpretation of those operators over infinite regular trees defines the VRP-equational graphs.

From now on, we fix a finite set A of *labels. Colored graphs* (or simply graphs) are triple (V, E, η), where V is a countable set of *vertices*, $E \subseteq V \times A \times V$ is the set of edges and η is a mapping from V into a finite set (of colors). If G is a graph, we write V_G its set of vertices, E_G its set of edges and η_G its color mapping. For simplicity, we will assume that there exists an integer N such that the range of η is $[N]$. The set of graphs with coloring functions ranging in $[N]$ is $\mathcal{G}_N$.

We define the relation $\subseteq$ over graphs of $\mathcal{G}_N$ by $G \subseteq G'$ if $V_G \subseteq V_{G'}$, $E_G \subseteq E_{G'}$ and $\eta_G = \eta_{G'}|_{V_G}$ (η_G is the restriction of $\eta_{G'}$ over V_G). This relation is a cpo: the smallest element is the empty graph and the least upper bound is written $\cup$ (it corresponds to the union of vertices, the union of edges and the 'union' of coloring functions).

Given a graph G and given a one-to-one function Φ with domain containing V_G, $\Phi(G) = (V', E', \eta')$ is defined by:

$$\begin{aligned} V' &= \Phi(V_G) \\ E' &= \{(\Phi(v), e, \Phi(v')) \mid (v, e, v') \in E_G\} \\ \eta'(\Phi(v)) &= \eta_G(v) \ . \end{aligned}$$

If G and G' are two graphs such that there is an injective mapping Φ verifying $\Phi(G) = G'$ then G and G' are said *isomorphic*, written $G \sim G'$.

We define now the five basic operations on graphs used in VRP-equational graphs. The four first are the classical VR operations. The fifth is new.

Single vertex constant: for $n \in [N]$, $\dot{n} = (\{0\}, \emptyset, \{0 \mapsto n\})$.

Recoloring: for ϕ mapping from $[N]$ into $[N]$, $[\phi](V, E, \eta) = (V, E, \phi \circ \eta)$.

Edges adding: for $n, n' \in [N]$, $e \in A$, $[n \overset{e}{\bowtie} n']G = (V_G, E', \eta_G)$ with $E' = E_G \cup \{(v, e, v') \mid \eta_G(v) = n,\ \eta_G(v') = n'\}$.

Disjoint union: $(V_0, E_0, \eta_0) \oplus (V_1, E_1, \eta_1) = (V, E, \eta)$

$$\text{with } \begin{cases} V = \{0\} \times V_0 \ \cup \ \{1\} \times V_1 \\ E = \{((\alpha, v), e, (\alpha, v')) \mid \alpha \in [2], (v, e, v') \in E_\alpha\} \\ \eta(\alpha, v) = \eta_\alpha(v) \qquad \text{for } \alpha \in [2] \ . \end{cases}$$

Unsynchronised product: $(V_0, E_0, \eta_0) \otimes (V_1, E_1, \eta_1) = (V', E', \eta')$

$$\text{with } \begin{cases} V' = V_0 \times V_1 \\ E' = \{((v_0, v_1), e, (v_0', v_1)) \mid (v_0, e, v_0') \in E_0,\ v_1 \in V_1\} \\ \qquad \cup\ \{((v_0, v_1), e, (v_0, v_1')) \mid v_0 \in V_0,\ (v_1, e, v_1') \in E_1\} \\ \eta'(v_0, v_1) = \eta_0(v_0) + \eta_1(v_1) \bmod N \ . \end{cases}$$

Let us remark that — up to isomorphism — $\oplus$ and $\otimes$ are commutative and associative and $\otimes$ is distributive over $\oplus$. The empty graph is an absorbing element for $\otimes$ and the neutral element for $\oplus$ and the graph $\dot{0}$ is neutral for $\otimes$.

Let us now define the ranked alphabets $\mathcal{F}_N^{\mathrm{VR}}$ and $\mathcal{F}_N^{\mathrm{VRP}}$:

$$\begin{aligned} \mathcal{F}_N^{\mathrm{VR}} &= \{\dot{n}_N : 0 \mid n \in [N]\} \\ &\cup \{[\phi]_N : 1 \mid \phi \text{ mapping from } [N] \text{ to } [N]\} \\ &\cup \{[n \overset{e}{\bowtie} n']_N : 1 \mid n, n' \in [N],\ e \in A\} \\ &\cup \{\oplus_N : 2\} \\[1em] \mathcal{F}_N^{\mathrm{VRP}} &= \mathcal{F}_N^{\mathrm{VR}} \ \cup \ \{\otimes_N : 2\} \end{aligned}$$

The trees of $\mathcal{T}^\infty(\mathcal{F}_N^{\mathrm{VR}})$ are called *VR-trees*, and the trees of $\mathcal{T}^\infty(\mathcal{F}_N^{\mathrm{VRP}})$ are *VRP-trees*. We define the *VRP-interpretation* $[\![]\!]_N$ as the natural interpretation

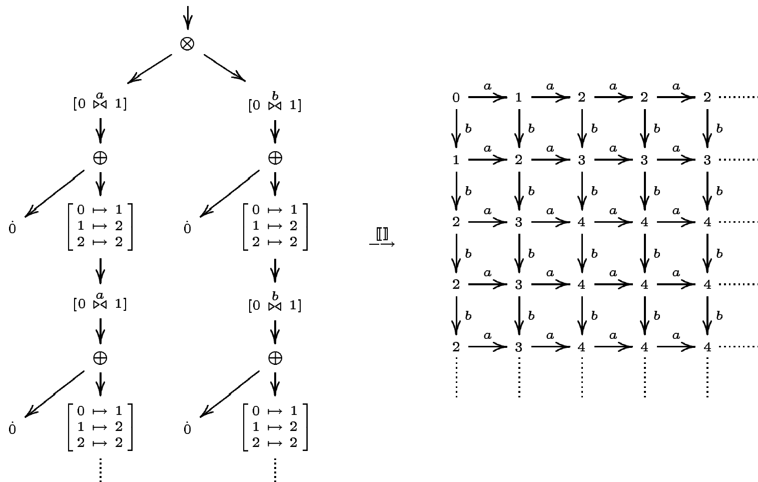

Fig. 1. A VRP-tree producing the infinite grid — assume $N = 5$

of terms of $\mathcal{T}(\mathcal{F}_N^{\mathrm{VRP}})$ over $\mathcal{G}_N$.

$$\begin{aligned} [\![\dot{c}_N]\!]_N &= \dot{c} \\ [\![[\phi]_N(t)]\!]_N &= [\phi][\![t]\!]_N \\ [\![[c \overset{e}{\bowtie} c']_N(t)]\!]_N &= [c \overset{e}{\bowtie} c'][\![t]\!]_N \end{aligned} \qquad \begin{aligned} [\![\oplus_N(s,t)]\!]_N &= [\![s]\!]_N \oplus [\![t]\!]_N \\ [\![\otimes_N(s,t)]\!]_N &= [\![s]\!]_N \otimes [\![t]\!]_N \end{aligned}$$

VRP-interpretation is extended by continuity to any tree — let $[\![\perp]\!]_N$ be the empty graph, the interpretation of a tree $t \in \mathcal{T}^\infty(\mathcal{F}_N^{\mathrm{VRP}})$ is $[\![t]\!]_N = \cup_k [\![t \downarrow_k]\!]_N$. This definition makes sense since all operators are continuous with respect to the $\subseteq$ order.

Definition 1. *The VR-equational graphs (resp. VRP-equational graphs) are the VRP-interpretations of the regular VR-trees (resp. VRP-trees).*

Figure 1 gives the example of the infinite grid described by a VRP-tree.

Remark 1. Given two VRP-equational graphs, we cannot yet combine them easily because the number of colors used may be different in the two underlying VRP-trees, and VRP-interpretation depends of this value. The problem comes from the color mapping used in the definition of $\otimes$ — it depends of N. In order to get rid of this drawback, we use VRP-trees without overflows: a VRP-tree t has *no overflow* if for all node u such that $t^u = \otimes_N(s, s')$, the sum of maximum colors appearing in $[\![s]\!]_N$ and $[\![s']\!]_N$ is strictly smaller than N. Under this constraint, the color mapping $\eta'(v_0, v_1) = \eta_0(v_0) + \eta_1(v_1) \bmod N$ is a simple sum and does not depend of N anymore. Increasing the value of N everywhere in a

VRP-tree without overflow (and accordingly extending the recoloring operators) do not change the VRP-interpretation of the tree.

Furthermore if a VRP-tree has overflows, it is easy to obtain an equivalent VRP-tree without overflow by doubling the value of N and replacing everywhere $\otimes_N(s, s')$ by $[\mathrm{mod}_N]_{2N}(\otimes_{2N}(s, s'))$.

For now and on, we make the assumption that all VRP-trees have no overflow and we omit everywhere the N indices. We also allow ourself to increase the value of N whenever it is needed.

4 Internal representation

In this part, we give to VRP-equational graphs an equivalent internal representation (Theorem 1). Instead of describing those graphs as the interpretation of regular trees, we describe them by explicitly giving the set of vertices and explicitly defining the edge relation.

We use here deterministic bottom-up tree automata without final states. Let $\mathcal{F}$ be a typed alphabet over Θ. A *deterministic tree automaton* is a tuple $(\mathcal{F}, Q, \delta)$, where $(Q_\theta)_{\theta \in \Theta}$ is a finite alphabet of states (of arity 0), and δ is a function which maps for all type $\tau = (\theta_0 \dots \theta_{n-1}, \theta)$ tuples of $\mathcal{F}_\tau \times Q_{\theta_0} \times \dots \times Q_{\theta_{n-1}}$ into Q_θ. The function δ is naturally extended by induction into a function from $\mathcal{T}_\theta(\mathcal{F} \cup Q)$ into Q_θ (for any type θ).

A set $T \subseteq \mathcal{T}(\mathcal{F})$ is *recognizable* if there exists a deterministic automaton $\mathcal{A} = (\mathcal{F}, Q, \delta)$ and a set $F \subseteq Q$ (of final states) verifying that $t \in T$ iff $\delta(t) \in F$. Similarly, a relation $R \subseteq \mathcal{T}(\mathcal{F}) \times A \times \mathcal{T}(\mathcal{F})$ is *recognizable* if there exists a deterministic tree automaton $\mathcal{A} = (\mathcal{F}, Q, \delta)$ and a set $F \subseteq Q \times A \times Q$ verifying that $(t, e, t') \in R$ iff $(\delta(t), e, \delta(t')) \in F$. Such a relation R *preserves* types if for all $(t, e, t') \in R$, t and t' have the same type.

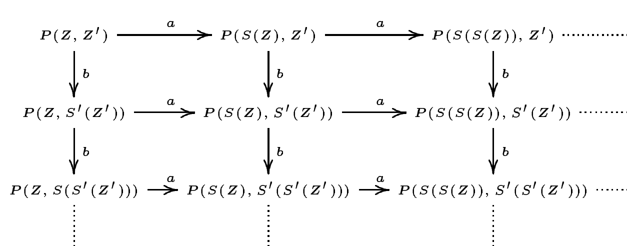

Fig. 2. The infinite grid as a recognizable ground term rewriting system

Let $\mathcal{F}$ be a typed alphabet over the set of types Θ, $\theta_r \in \Theta$ be a (root) type, and R a recognizable subset of $\mathcal{T}(\mathcal{F}) \times A \times \mathcal{T}(\mathcal{F})$ preserving types. The *(recognizable) ground term rewriting system* (RGTRS) are not colored graphs defined by a recognizable relation R. The vertices are terms of $\mathcal{T}_{\theta_r}(\mathcal{F})$, and there is an edge labelled by e between two terms t_1 and t_2 if one can obtain t_2 by replacing a subterm t'_1 by t'_2 in t_1, with $(t'_1, e, t'_2) \in R$. Formally, the graph of ground term rewriting by R is the graph:

$$GTR_{\theta_r}(R) = (\mathcal{T}_{\theta_r}(\mathcal{F}), \{(t_1, e, t_2) \mid \exists u \in D_{t_1} \cap D_{t_2}, \\ (t_1^u, e, t_2^u) \in R, \ \ t_1|_{(D_{t_1} - u\mathbb{N}^*)} = t_2|_{(D_{t_2} - u\mathbb{N}^*)}\})$$

Figure 2 gives an example of the infinite grid as a RGTRS. The types are $\Theta = \{\theta_r, \theta, \theta'\}$ where θ_r is the root type. The set of symbols is $\mathcal{F} = \{P : (\theta\theta', \theta_r),\ Z : \theta,\ Z' : \theta',\ S : (\theta, \theta),\ S' : (\theta', \theta')\}$. The recognizable relation used is $R = \{(Z, a, S(Z)),\ (Z', b, S'(Z'))\}$.

Theorem 1. *The VRP-equational graphs are exactly the RGTRS up to isomorphism and color removal.*

The idea of the translation is almost straightforward. One identifies subtrees of regular VRP-trees with types, and colors with states of the automaton. The same construction can also be applied for non regular VRP-trees. It leads to an infinite set of types.

Remark 2. In the RGTRS, the typing policy is used as a technique for restricting the set of vertices. The prefix recognizable graphs are defined as prefix rewriting systems of words (or equivalently linear terms), but the vertices are restricted by a general recognizable (rational) language. Restricting vertices of RGTRS with a recognizable relation increase the power of the system and leads to graphs which do not have a decidable reachability problem (whereas RGTRS have, see Theorem 3). However the RGTRS restricted to linear terms gives exactly the prefix recognizable graphs up to isomorphism[1].

Remark 3. Another technique for restricting vertices is to use a root vertex and keep only vertices reachable from the root. This approach is used by Löding [9].

Löding studies those graphs when restricted to bounded tree-width and shows that the resulting graphs are pushdown graphs. In the next section we perform the same study for VRP-equational graphs.

5 Tree-width of VRP-equational graphs

In this section we study the VRP-equational graphs of bounded tree-width. We show that those graphs are exactly the HR-equational graphs (Theorem 2).

Definition 2. *A tree decomposition of a graph G is an undirected tree (a connected undirected graph without cycle) $T = (V_T, E_T)$ with subsets of V_G as vertices such that:*

[1] In fact, one can restrict RGTRS by a top-down deterministic tree automaton without changing the power of the system. In general, top-down deterministic tree automaton are less expressive than recognizability. On linear terms, both are equivalent.

1. *for any edge $(v, e, v') \in E_G$, there is a vertex $N \in V_T$ such that $v, v' \in N$.*
2. *for any vertex $v \in V_G$, the set of vertices containing v, $\{N \in V_t \mid v \in N\}$ is a connected subpart of T.*

The tree-width of a graph G is the smallest $k \in \mathbb{N} \cup \{\infty\}$ such that G admits a tree decomposition with vertices of cardinality smaller than or equal to $k+1$. If k is finite, the graph is of bounded tree-width.

Intuitively, a graph is of tree-width k if it can be obtained by 'pumping up at width k' a tree. An equivalent definition is that a graph is of tree-width k if it can be obtained by a (non-deterministic) graph grammar with at most $k+1$ vertices per rule. This notion does not rely on orientation nor on labelling of edges. It is defined up to isomorphism and is continuous. Let $\leq$ be the binary relation defined by $G \leq G'$ if G is isomorphic to a subgraph of G' up to relabelling and edge reversal. It has the important property that if $G \leq G'$ then the tree-width of G is smaller or equal to the tree-width of G'.

Given a regular VRP-tree t, we aim at eliminating the $\otimes$ operators of the tree under the constraint of bounded tree-width. We then obtain a regular VR-tree of isomorphic interpretation. To this purpose, we assume that t is *normalized*: none of its subtrees has the empty graph as interpretation (normalization is possible if $[\![t]\!]$ is not empty). Under this constraint, for any subtree s of t, we have $[\![s]\!] \leq [\![t]\!]$ (this would not be true anymore if t was not normalized because the empty graph is an absorbing element of $\otimes$). The first consequence of normalization is that all subtrees of t have an interpretation of bounded tree-width.

First step: The first step of the transformation eliminates subtrees of the form $s = \otimes(s_0, s_1)$ appearing infinitely in a branch of the tree. Using normalization, we obtain that for all n, either $[\![s_0]\!]^n \leq [\![s]\!]$, either $[\![s_1]\!]^n \leq [\![s]\!]$ (with $G^0 = \dot{0}$ and $G^{n+1} = G \otimes G^n$). Lemma 1 handles such cases.

Lemma 1. *Let G be a graph such that for all n, $G^n \leq G'$ and G' is of bounded tree-width, then all edges in G are loops*[2] *(of the form (x, e, x)).*

In fact, the unsynchronised product of a graph G with a graph with only loops does not behaves like a real product. It leads to the graph G duplicated a finite or an infinite number of times, with colors changed and possibly loops added. The same result can be obtained without product, and furthermore, this transformation can be performed simultaneously everywhere in a tree. The first step of the transformation amounts to apply this technique until there is no branch of the tree with an infinite number of product operators.

Second step: After the first step of the transformation, among the subtrees with $\otimes$ at root, at least one does not contain any other $\otimes$ operator. Let $\otimes(s_0, s_1)$ be this subtree. Its interpretation is of bounded tree-width, and s_0 and s_1 are VR-trees. Lemmas 2 and 3 then hold.

[2] If G has a non-looping edge, then G^n has the hypercube of dimension n as subgraph. The hypercubes do not have a bounded tree-width, thus G' is not of bounded tree-width.

Lemma 2. *If the unsynchronised product of two graphs is of bounded tree-width then one of the graphs has all its connected components of bounded size.*

Lemma 3. *VR-equational graphs are closed by unsynchronised product with graphs having all their connected components of bounded size*[3]*.*

It follows that $[\![\otimes(s_0, s_1)]\!]$ is a VR-equational graph. It is then sufficient to replace in the tree all occurrences of $\otimes(s_0, s_1)$ by a VR-tree of isomorphic interpretation. After iteration of this process, there is no more product operator left, and thus the original graph is VR-equational up to isomorphism.

The HR-equational graphs are originally described as the smallest solution of equational systems with hyperedge replacement operators. Barthelmann has proved that those graphs were exactly the VR-equational graphs of bounded tree-width (see [1]). Theorem 2 concludes.

Theorem 2. *The VRP-equational graphs of bounded tree-width are exactly the HR-equational graphs up to isomorphism.*

It is interesting as the HR-equational graphs have a decidable monadic theory whereas VRP-equational graphs do not (the infinite grid is the classical counter-example). In Section 6 we extend the known result that the first order theory with reachability remains decidable for VRP-equational graphs.

6 Decidability of logic

In this section, we prove that for all trees t which have a decidable monadic theory, $[\![t]\!]$ has a decidable first order theory with reachability predicate (Theorem 3).

6.1 Monadic second order logic

During this part, t will be a (not necessarily regular) VRP-tree. We use here the monadic second order logic on VRP-trees. We do not define precisely what this logic is (the reader may find a description of it in [15]). Informally, we are allowed in monadic formula to use the classical logic connectives (e.g $\vee, \wedge, \neg$), existential ($\exists$) and universal ($\forall$) quantifiers over variables of first and monadic second order. Variables of first order are interpreted as nodes of the tree t and are written in small letters. Monadic second order variables are interpreted as sets of nodes and are written in capital letters. If x is a first order variable then $x0$ represents its left child, and $x1$ the right one (this notation is compatible with the definition of nodes as words). The predicates allowed are equality of first order variables (e.g $x = y0$) and membership (e.g $x \in X$). The first order constant ε is the root of the tree. The symbols of t are described by a finite set of second order constants of same name. For instance, expressing that the symbol at node x is $\otimes$ is written $x \in \otimes$.

[3] Notice that it changes the number of colors used.

We also use for simplicity classical operations on sets (e.g. $\cup$,$\subseteq$, ...) and quantification over known finite sets (colors, labels and functions from colors to colors). We also write $x\mathbb{N}^*$ the set of nodes under x. Those extensions can be encoded into the monadic formulas without any difficulty.

If Ψ is a closed monadic formula, we write $t \vDash \Psi$ to express the satisfaction of Ψ by t.

As an example, we define the predicate finitetree(r, X) which is satisfied if X is a finite connected subset of D_t 'rooted' at r.

$$\begin{aligned}\text{childof}(x, y) \equiv\ & x = y0 \wedge (\forall\ n,\ y \notin \dot{n}) \\ & \vee\ x = y1 \wedge (y \in \oplus\ \vee\ y \in \otimes)\end{aligned}$$

$$\begin{aligned}\text{finitetree}(r, X) \equiv\ & r \in X\ \wedge\ \forall x \in X,\ x = r \text{ xor } \exists y \in X, \text{childof}(x, y) \\ & \wedge \neg(\exists W \subseteq X, W \neq \emptyset \wedge\ \forall x \in W, \exists y \in W, \text{childof}(y, x))\end{aligned}$$

The predicate childof(x, y) means that x appears just under y in the tree t. Then, we define a finite connected subset X 'rooted' at r as a set such that every element, except the root r, has a father in the set and which do not contain an infinite branch W.

6.2 First order theory with reachability of VRP-interpretations

Our first goal is to encode the VRP-interpretation of t into monadic formulas. To obtain this encoding we first remark that each vertex of the resulting graph can be uniquely identified with a special kind of finite subset of D_t. Let t be a VRP-tree and r one of its nodes, it amounts to associate to each vertex of $[\![t^r]\!]$ a non-empty finite connected subpart of D_t rooted at r. This finite rooted subset can be seen as the equivalent term of a RGTRS system.

If $t(r) = \dot{n}$ then $\{r\}$ encodes the only vertex of $[\![t^r]\!]$.

If $t(r) = [\phi]$ or $t(r) = [n \overset{e}{\bowtie} n']$ the vertices of $[\![t^r]\!]$ are exactly the vertices of $[\![t^{r0}]\!]$. Let X be the encoding of a vertex of $[\![t^{r0}]\!]$, then $\{r\} \cup X$ encodes the same vertex in $[\![t^r]\!]$.

If $t(r) = \oplus$, then a vertex of $[\![t^r]\!]$ has its origin either in $[\![t^{r0}]\!]$, either in $[\![t^{r1}]\!]$. Let v be a vertex of $[\![t^{r\alpha}]\!]$ (for some $\alpha \in \{0, 1\}$) and X its encoding, then $\{r\} \cup X$ uniquely encodes the vertex in $[\![t^r]\!]$.

Finally, if $t(r) = \otimes$, then each vertex of $[\![t^r]\!]$ originates from both a vertex of $[\![t^{r0}]\!]$ and a vertex of $[\![t^{r1}]\!]$. Let X_0 be the encoding of the origin vertex in $[\![t^{r0}]\!]$ and X_1 be the encoding of the origin vertex in $[\![t^{r1}]\!]$, then $\{r\} \cup X_0 \cup X_1$ uniquely encodes the vertex in $[\![t^r]\!]$.

We can translate this description into a monadic predicate vertex(r, X) which is satisfied by t if X encodes a vertex of $[\![t^r]\!]$. We can also describe a monadic predicate color$_c(r, X)$ which is satisfied by t if the vertex encoded by X has color c in $[\![t^r]\!]$, and a predicate edge$_e(r, X, X')$ which is satisfied by t if there is an edge of label e between the vertex encoded by X and the vertex encoded by X' in $[\![t^r]\!]$. The validity of the approach is formalized in Lemma 4.

Lemma 4. *For all r, the graph $[\![t^r]\!]$ is isomorphic to $G_r = (V_r, E_r, \eta_r)$ with:*

$$V_r = \{X \subseteq D_t \mid t \vDash \mathrm{vertex}(r, X)\}$$
$$E_r = \{(X, e, X') \mid t \vDash \mathrm{edge}_e(r, X, X')\}$$
$$\eta_r(X) = n \quad \textit{such that} \quad t \vDash \mathrm{color}_n(r, X) \ .$$

We define also the monadic predicate $\mathrm{path}_{A'}(X, X')$ which satisfies the following lemma.

Lemma 5. *Let $A' \subseteq A$ be a set of labels and let X,X' be two vertices of G_ε. There is a path with labels in A' between X and X' in G_ε iff $t \vDash \mathrm{path}_{A'}(X, X')$.*

The proof of this result is technical : it involves the encoding of simpler problems such as "*is there a path between a node of color n and a node of color n' ?*".

Using the two previous lemmas, it is easy to translate a first order formula with reachability predicate over $[\![t]\!]$ into a monadic formula over t. The main theorem is then straightforward.

Theorem 3. *The VRP-interpretation of a tree having a decidable monadic second order theory has a decidable first order theory with reachability predicate.*

7 Conclusion

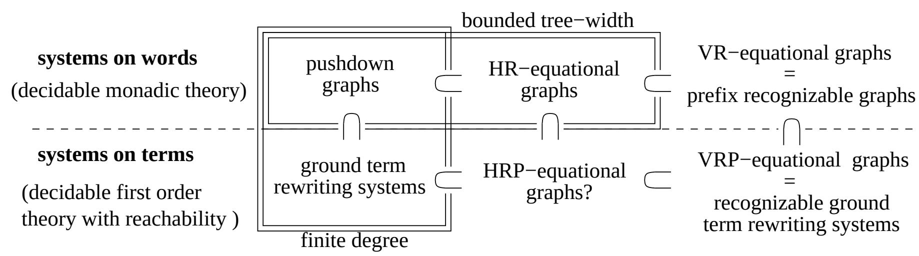

Fig. 3. A partial classification of families of graphs

System on words where already well known. This paper is a step toward a corresponding classification for systems on terms. The hierarchy obtained is depicted in Figure 3. The ground term rewriting systems must be understood recognizable ground term rewriting systems of finite rewriting relations (see Remark 3). Notice that there is probably a natural family of graphs — HRP-equational graphs — between ground term rewriting systems and VRP-equational graphs.

An open question is the nature of the family of graphs obtained by ε-closure of VRP-equational graphs (VR-equational graphs are closed by this operation). It also has a first order theory with reachability predicate decidable (ε-closure preserves it). We assume that this family of graphs strictly contains the family of VRP-equational graphs.

Lastly, we have studied the more general family of graphs obtained by VRP-interpretation of infinite trees with a decidable monadic second order theory. Those graphs have a first order theory with reachability predicate decidable. In fact, it is probable that weak monadic second order theory is sufficient.

Acknowledgments. Many thanks to Didier Caucal who has introduced me to this topic and gave helpful remarks. Thanks to Thierry Cachat, Emmanuelle Garel and Tanguy Urvoy for reading previous versions of this work. Thanks to Marion Fassy for her support.

References

1. K. Barthelmann. When can equational simple graphs be generated by hyperedge replacement? Technical report, University of Mainz, 1998.
2. A. Blumensath and E. Grädel. Automatic Structures. In *Proceedings of 15th IEEE Symposium on Logic in Computer Science LICS 2000*, pages 51–62, 2000.
3. D. Caucal. On infinite transition graphs having a decidable monadic theory. In Icalp 96, volume 1099 of *LNCS*, pages 194–205, 1996.
4. H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on: http://www.grappa.univ-lille3.fr/tata, 1997.
5. B. Courcelle. *Handbook of Theoretical Computer Science*, chapter Graph rewriting: an algebraic and logic approach. Elsevier, 1990.
6. B. Courcelle. The monadic second order logic of graphs ix: Machines and their behaviours. In *Theoretical Computer Science*, volume 151, pages 125–162, 1995.
7. M. Dauchet and S. Tison. The theory of ground rewrite systems is decidable. In *Fifth Annual IEEE Symposium on Logic in Computer Science*, pages 242–248. IEEE Computer Society Press, June 1990.
8. T. Knapik, D. Niwinski, and P. Urzyczyn. Higher-order pushdown trees are easy. In M. Nielsen, editor, *FOSSACS'2002*, 2002.
9. C. Löding. Ground tree rewriting graphs of bounded tree width. In *STACS O2*, 2002.
10. C. Morvan. On rational graphs. In J. Tiuryn, editor, *FOSSACS'2000*, volume 1784 of *LNCS*, pages 252–266, 2000.
11. D. Muller and P. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.
12. M.O. Rabin. Decidability of second-order theories and automata on infinite trees. *Trans. Amer. Math. soc.*, 141:1–35, 1969.
13. D. Seese. The structure of models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53(2):169–195, 1991.
14. G. Sénizergues. Definability in weak monadic second-order logic of some infinite graphs. In *Dagstuhl seminar on Automata theory: Infinite computations, Warden, Germany*, volume 28, page 16, 1992.
15. W. Thomas. Languages, automata, and logic. *Handbook of Formal Language Theory*, 3:389–455, 1997.
16. W. Thomas. A short introduction to infinite automata. In W. Kuich, editor, *DLT'2001*, 2001.