An Algebraic Presentation of Term Graphs, via GS-Monoidal Categories *

Andrea Corradini

University of Pisa, Dipartimento di Informatica, Corso Italia 40, I-56124 Pisa, Italy. email: andrea@di.unipi.it

Fabio Gadducci

Technical University of Berlin, Fachbereich 13 Informatik, Franklinstraße 28/29, D-10587 Berlin, Germany. email: gfabio@cs.tu-berlin.de

Abstract. We present a categorical characterisation of term graphs (i.e., finite, directed acyclic graphs labeled over a signature) that parallels the well-known characterisation of terms as arrows of the algebraic theory of a given signature (i.e., the free Cartesian category generated by it). In particular, we show that term graphs over a signature Σ are one-to-one with the arrows of the free gs-monoidal category generated by Σ . Such a category satisfies all the axioms for Cartesian categories but for the naturality of two transformations (the discharger! and the duplicator ∇), providing in this way an abstract and clear relationship between terms and term graphs. In particular, the absence of the naturality of ∇ and! has a precise interpretation in terms of explicit sharing and of loss of implicit garbage collection, respectively.

Keywords: algebraic theories, directed acyclic graphs, gs-monoidal categories, symmetric monoidal categories, term graphs.

Mathematical Subject Classifications (1991): 68R10, 18C10, 18D10, 05C75, 68Q42

1. Introduction

The classical theory of term graph rewriting studies the issue of representing terms as directed graphs, and of modeling term rewriting via graph rewriting (we refer for a survey to [58] and to the references therein). With respect to the standard representation of terms as trees, the main operational appeal of using graphs is that the sharing of common sub-terms can be represented explicitly. Intuitively, the rewrite process is speeded up, because rewriting steps do not have to be repeated for each copy of an identical, shared sub-term. For these reasons,

^{*} Research partly supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems) through the Technical University of Berlin and the University of Pisa; and by the EC Fixed Contribution Contract No. EBRFMBICT960840. Research carried out in part while the first author was on leave at CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands.



© 1999 Kluwer Academic Publishers. Printed in the Netherlands.

term graph rewriting is often used in the implementation of functional programming languages [52].

In our opinion, however, there is an unsatisfactory gap between the achievements of the theories of term and of term graph rewriting, respectively. In particular, a solid ground for the theory of term rewriting is provided by the existence of three different but equivalent presentations, namely the operational, classical one (described in terms of redexes and substitutions [39]), the *logical* one (we think of Meseguer's rewriting logic [48]), and the categorical one, based on algebraic (Cartesian) 2-theories [53, 57]. While the operational presentation is the best suited for implementation purposes, both the logical and the categorical ones provide an *inductive* definition of the rewrite relation over terms, that lays the ground for the development of proof and analysis techniques based on structural induction. Moreover, the categorical account has the advantage of being independent of representation details, and of stressing the intrinsic algebraic structure of terms and their rewriting. In fact, one can safely claim that the essential structure of the collection of terms over a signature is 'Cartesianity'.

Much less satisfactory are the achievements of the theory of term graph rewriting, which has been studied by many authors, but only in operational style. Furthermore, there is no common agreement on the way term graphs should be defined. In fact, they have been represented as directed graphs satisfying a number of constraints [4], as suitable labeled hyper-graphs called *jungles* [32], or as sets of recursive equations [1], among others.

This paper proposes a first contribution towards a categorical theory of term graph rewriting, which parallels the categorical description of term rewriting. In fact, we present a characterisation of term graphs over a given signature as arrows of the *gs-monoidal* category freely generated by it,¹ and we relate this presentation to a categorical characterisation of terms. Indeed, we focus in this paper on the representation of terms and term graphs only, leaving to other works the description of their rewriting using suitable 2-categories (see the discussion in Section 6.4).

Before explaining concisely the main contribution of the paper, it is worth recalling that the terms over a given signature Σ can be regarded as the arrows of a *Cartesian category* (called the *algebraic theory* of Σ) freely generated (in a suitable way) by Σ (see, e.g., [40, 43]). Such a category has (underlined) natural numbers as objects, and its generators are arrows like $g: \underline{n} \to \underline{1}$, where g is an operator of rank n in Σ . It

¹ Here 'gs' stands for 'graph substitution', an acronym whose explanation we defer to the end of Section 5.



Figure 1. Two term graphs that correspond to the same tuple of terms.

can be shown that the arrows from \underline{n} to \underline{m} are in one-to-one correspondence with the m-tuples of terms over n variables; furthermore, arrow composition faithfully corresponds to term substitution.

Cartesian categories can be defined as symmetric monoidal categories equipped with two natural (symmetric monoidal) transformations denoted by ∇ (the duplicator) and ! (the discharger), respectively. The definition of Cartesian categories can be weakened slightly by dropping the requirement of naturality for ∇ and !: We call the resulting categories gs-monoidal.

The main result of the paper shows that the term graphs over a given signature Σ are in one-to-one correspondence with the arrows of the free gs-monoidal category generated by Σ . More precisely, the arrows from \underline{n} to \underline{m} in that category are in one-to-one correspondence with the term graphs with m roots and n variables. This categorical characterisation of term graphs makes evident that the only difference with terms is that the naturality of the duplicator and of the discharger does not hold anymore. And this fact has a clear interpretation in terms of 'sharing of sub-terms' and of 'garbage collection', in the following sense.

Suppose $g: \underline{n} \to \underline{1}$ be the arrow corresponding to a n-ary operator $g(x_1,\ldots,x_n)$. Then, using ';' for composition of arrows in diagrammatic order, arrow $a_2=g$; $\nabla_{\underline{1}}:\underline{n}\to\underline{2}$ represents, intuitively, a structure having two pointers to $g(x_1,\ldots,x_n)$, while arrow $a_1=\nabla_{\underline{n}}$; $g\oplus g:\underline{n}\to\underline{2}$ represents a structure where two copies of g share the same variables (Figure 1 shows the term graphs corresponding to a_1 and a_2 , using conventions to be introduced in Section 2; in particular the numbers to the left represent the variables, and those to the right the roots).

Now, regarded as term graphs the two structures must be distinct, because they exhibit a different degree of sharing for some substructure: There are two 'pointers' to g in G_2 , but only one to each copy of g in G_1 . Indeed, arrows a_1 and a_2 are distinct in the free gs-monoidal category concerned. On the contrary, in the free Cartesian category, the naturality of ∇ implies that g; $\nabla_{\underline{1}} = \nabla_{\underline{n}}$; $g \oplus g$, which means that the two arrows are provably the same. Therefore structures with different

degree of sharing are identified, and, by convention, one can take as representative of an equivalence class of such arrows the structure with least sharing, that is, a tuple of trees (in our example, arrow a_1 (= a_2) in the concerned algebraic theory represents the tuple of terms $\langle g(x_1, \ldots, x_n), g(x_1, \ldots, x_n) \rangle$).

By similar arguments, it can be shown that the naturality of the discharger! can be interpreted as an 'implicit garbage collection'. Let us call 'garbage' a substructure that is not accessible by any root. Then arrow $g; !_1 : \underline{n} \to \underline{0}$ can be regarded as the structure $g(x_1, \ldots, x_n)$ where g is garbage, because, intuitively, its only root is deleted by the discharger $\underline{1}$. On the other hand, arrow $\underline{1}_{\underline{n}} : \underline{n} \to \underline{0}$ represents an empty structure. Now the naturality of! implies that g;! $_1 = !_n$, i.e., that any structure with some garbage is equivalent to the same structure with the garbage removed, and we may call this 'implicit garbage collection'. Such a property certainly holds for terms, but not necessarily for term graphs; actually, in the term graph rewriting literature both implicit and explicit garbage collection have been considered [4]. In our definition of term graphs implicit garbage collection is not allowed for, and this fits well with the non-naturality of! in gs-monoidal categories (a variant of the main result of the paper could show that term graphs with implicit garbage collection are one-to-one with arrows of a free gsmonoidal category where naturality of! is imposed: This is conceptually easy, but would require a major reworking of all definitions).

The paper is organised as follows. In Section 2 we introduce ranked $term\ graphs$, i.e., equivalence classes of directed acyclic graphs labeled over a signature Σ , with distinguished lists of variable and root nodes. Such nodes are used for defining an operation of 'composition', which is the counterpart of term substitution, and that provides term graphs with a natural categorical structure. The main result of the section shows that every ranked term graph can be obtained from a small set of atomic term graphs by using the operations of composition and 'union', i.e., disjoint union with concatenation of the lists of roots and variables.

Section 3 recalls the basic definitions about symmetric monoidal categories, and introduces gs-monoidal categories: These are symmetric monoidal categories equipped with two transformations, satisfying a few properties. Next the gs-monoidal theory of a signature is defined as the free gs-monoidal category generated by the signature, and a decomposition lemma for the arrows in such theories is presented.

The main result of the paper, stating that the category of term graphs over a signature is isomorphic to its gs-monoidal theory, is presented in Section 4. In Section 5 we discuss the relationship between classical algebraic theories and gs-monoidal theories, and, as a

consequence, between terms and term graphs over the same signature. Many extensions of the characterisation result of this paper are possible, including the treatment of more general signatures, of cyclic term graphs, and of rewriting: These are discussed in the concluding section, Section 6, where the related literature is briefly described as well.

2. Term Graphs

This section introduces (ranked) term graphs as isomorphism classes of (ranked) directed acyclic graphs. This presentation departs slightly from the standard definition (see for example [4]), because our main concern is the algebraic structure of term graphs.

DEFINITION 1. (Directed acyclic graphs, dag's). Let Σ be a signature, i.e., a ranked set of operator symbols, and let arity be the function returning the arity of an operator symbol, i.e., $\operatorname{arity}(f) = n$ iff $f \in \Sigma_n$. A labeled graph d (over Σ) is a triple $d = \langle N, l, s \rangle$, where N is a set of nodes, $l: N \to \Sigma$ is a partial function called the labeling function, $s: N \to N^*$ is a partial function called the successor function, and such that the following conditions are satisfied:

- -dom(l) = dom(s), i.e., labeling and successor functions are defined on the same subset of N; a node $n \in N$ is called empty if $n \notin dom(l)$.
- for each node $n \in dom(l)$, arity(l(n)) = length(s(n)), i.e., each nonempty node has as many successor nodes as the arity of its label.

If $s(n) = \langle n_1, \ldots, n_k \rangle$, we say that n_i is the i-th successor of n and denote it by $s(n)_i$. A labeled graph is discrete if all its nodes are empty. A path in d is a sequence $\langle n_0, i_0, n_1, \ldots, i_{m-1}, n_m \rangle$, where $m \geq 0$, $n_0, \ldots, n_m \in N$, $i_0, \ldots, i_{m-1} \in \mathbb{N}$ (the natural numbers), and n_k is the i_{k-1} -th successor of n_{k-1} for $k \in \{1, \ldots, m\}$. The length of this path is m; if m = 0, the path is empty. A cycle is a path like above where $n_0 = n_m$.

A directed acyclic graph (over Σ), shortly dag, is a labeled graph having no nonempty cycles. For a dag d we shall often denote its components by N(d), l_d and s_d , respectively.

DEFINITION 2. (dag morphisms, category \mathbf{Dag}_{Σ}). Let d and d' be two dag's. A (dag) morphism $f: d \to d'$ is a function $f: N(d) \to N(d')$ that preserves labeling and successors, i.e., such that for each nonempty

node $n \in N(d)$, $l_{d'}(f(n)) = l_d(n)$, and $s_{d'}(f(n))_i = f(s_d(n)_i)$ for each $i \in \{1, \ldots, arity(l_d(n))\}.$

Dag's over Σ and dag morphisms clearly form a category that will be denoted \mathbf{Dag}_{Σ} .

In the following, for each $i \in \mathbb{N}$ we shall denote by \underline{i} the set $\underline{i} = \{1, \ldots, i\}$ (thus $\underline{0} = \emptyset$).

DEFINITION 3. (Ranked dag's and term graphs). An (i,j)-ranked dag (also, a dag of rank (i,j)) is a triple $g = \langle r,d,v \rangle$, where d is a dag with exactly j empty nodes, $r:\underline{i} \to N(d)$ is a function called the root mapping, and $v:\underline{j} \to N(d)$ is a bijection between \underline{j} and the empty nodes of d, called the variable mapping. Node r(k) is called the k-th root of d, and v(k) is called the k-th variable of d, for each admissible k.

Two (i, j)-ranked dag's $g = \langle r, d, v \rangle$ and $g' = \langle r', d', v' \rangle$ are isomorphic if there exists a ranked dag isomorphism $\phi : g \to g'$, i.e., a dag isomorphism $\phi : d \to d'$ such that $\phi \circ r = r'$ and $\phi \circ v = v'$. A (i, j)-ranked term graph G (or with rank (i, j)) is an isomorphism class of (i, j)-ranked dag's. We shall often write G_j^i to recall that G has rank (i, j).

Term graphs are defined in the seminal paper [4] as labeled graphs (as in Definition 1) with a distinguished node, the *root*. Thus there are three differences with the above definition. Firstly, we restrict to the acyclic case; a categorical characterisation of cyclic term graphs is possible, but it goes beyond the goals of this paper (see Section 6).

Secondly, our notion of term graph looks 'more abstract', because it is defined as an isomorphism class of dag's. This allows us to disregard the concrete identity of nodes when manipulating term graphs, provided that the operations we introduce are well defined on isomorphism classes. Admittedly, in most of the literature about term graph rewriting, isomorphic dag's are informally considered as equivalent (for example, the identity of nodes is not depicted in the usual graphical representation of term graphs), thus our definition just formalises a common way of reasoning 'up to isomorphism'.

Thirdly, our term graphs are ranked. The idea of equipping graphs with lists of distinguished nodes in order to define composition operations on them is not new (see for example [5, 25]), but for the first time, to our knowledge, it is applied here to the class of term graphs. In [4] this technique is not used simply because it is not needed for the goals of the paper, and the single root used there has a different role, being used to relate a term graph with a term unfolded from it. Summarising,

apart from the restriction to acyclic graphs, we can safely see our term graphs as a minor variation of those in [4].

We introduce now two operations on ranked term graphs. The *composition* of two ranked term graphs is obtained by gluing the variables of the first one with the roots of the second one, and it is defined only if their number is equal. This operation allows us to define a category having ranked term graphs as arrow. Next the *union* of term graphs is introduced: It is always defined, and it is a sort of disjoint union where roots and variables are suitably renumbered. This second operation provides the category of term graphs with a monoidal structure that will be made explicit in Section 4.

DEFINITION 4. (Composition of ranked term graphs). Let $G_k^{'j} = [\langle r', d', v' \rangle]$ and $G_j^i = [\langle r, d, v \rangle]$ be two ranked term graphs. Their composition is the ranked term graph $H_k^i = G_k^{'j}$; G_j^i defined as $H_k^i = [\langle in_d \circ r, d'', in_{d'} \circ v' \rangle]$, where $d'', in_d : d \to d''$ and $in_{d'} : d' \to d''$ are obtained as follows. Assuming that $d = \langle N(d), l_d, s_d \rangle$ and $d' = \langle N(d'), l_{d'}, s_{d'} \rangle$, we have $d'' = \langle (N(d) \uplus N(d'))/_{\approx}, l'', s'' \rangle$, where \uplus denotes disjoint union, \approx is the least equivalence relation such that $v(i) \approx r'(i)$ for $i \in \underline{j}$, and l'' and s'' are determined by l_d and s_d , respectively, for all \approx -equivalence classes containing only nodes of d, and by $l_{d'}$ and $s_{d'}$, respectively, for all other classes. Furthermore, the injections $in_d : N(d) \to N(d'')$ and $in_{d'} : N(d') \to N(d'')$ map each node to its \approx -equivalence class.

It is worth stressing that the composed dag can be characterised elegantly as a pushout, in the sense that $\langle d'', in_d, in_{d'} \rangle$ is a pushout of $\langle v : \underline{j} \to d, r' : \underline{j} \to d' \rangle$ in \mathbf{Dag}_{Σ} (set \underline{j} is regarded as a discrete dag). Thus the well-definedness of composition of term graphs easily follows from the uniqueness of pushouts up to isomorphism.²

DEFINITION 5. (The category of term graphs). For a given signature Σ , \mathbf{TG}_{Σ} denotes the category having as objects underlined natural numbers, and as arrows from \underline{j} to \underline{i} all (i,j)-ranked term graphs. Arrow composition is defined as in \overline{D} efinition 4, and the identity on \underline{i} is the discrete term graph $G^{\underline{i}}_{id}$ of rank (i,i) having i nodes, and where the k-th root is also the k-th variable, for all $k \in \underline{i}$. Then it is easy to check that \mathbf{TG}_{Σ} is a well-defined category, because composition is associative, and the identity laws hold.

² Actually, the pushout of two arrows in \mathbf{Dag}_{Σ} does not always exist: It does exist however in the case we are interested in, since morphism $v:\underline{j}\to d$ is injective and has only empty nodes in the codomain. See [16] for necessary and sufficient conditions for the existence of pushouts in the equivalent category of *jungles*.

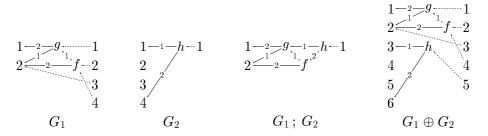


Figure 2. Two term graphs, their composition and their union.

DEFINITION 6. (Union of ranked term graphs). Let $G_j^i = [\langle r, d, v \rangle]$ and $G_l^{'k} = [\langle r', d', v' \rangle]$ be two ranked term graphs. Their union or parallel composition is the term graph of rank (i + k, j + l) $G_j^i \oplus G_l^{'k} = [\langle r'', d \uplus d', v'' \rangle]$, where $r'' : \underline{i + k} \to d \uplus d'$ and $v'' : \underline{j + l} \to d \uplus d'$ are defined as

$$-r''(x) = \begin{cases} r(x) & \text{if } x \in \underline{i}, \\ r'(x-i) & \text{if } x \in \{i+1,\dots,i+k\}. \end{cases}$$
$$-v''(x) = \begin{cases} v(x) & \text{if } x \in \underline{j}, \\ v'(x-j) & \text{if } x \in \overline{\{j+1,\dots,j+l\}}. \end{cases}$$

EXAMPLE 7. (Term graphs, composition and union). Figure 2 shows four term graphs. Empty nodes are represented by the natural numbers corresponding to their position in the list of variables, and are depicted as a vertical sequence on the left; nonempty nodes are represented by their labels, from where the edges pointing to the successors leave; the list of numbers on the right represent pointers to the roots: A dashed arrow from j to a node indicates that it is the j-th root. For example, the first term graph G_1 has rank (4,2), four nodes (two empty, 1 and 2, and two nonempty, f and g), the successors of g are the variables 2 and 1 (in this order), the successors of f are g and 2, and the four roots are g, f, 2, and f.

These graphical conventions make easy the operation of composition, that can be performed by matching the roots of the first graph with the variables of the second one, and then by eliminating them. For example, term graph G_1 ; G_2 is the composition of G_1 and of G_2 of rank (1,4). The last term graph is $G_1 \oplus G_2$, the union of G_1 and G_2 , of rank (5,6).

We are now ready to show the main result of this section, that is, that every term graph can be constructed, using composition and union, from a small set of atomic term graphs.

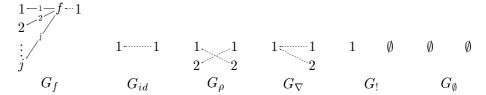


Figure 3. Atomic term graphs.

DEFINITION 8. (Atomic term graphs). An atomic term graph is a term graph in $\{G_f \mid f \in \Sigma\} \cup \{G_{id}, G_\rho, G_\nabla, G_!, G_\emptyset\}$: These graphs are depicted in Figure 3.

Since every node of an atomic term graph is a root or a variable, such term graphs can be formally defined as follows (using, a bit improperly, r, v, s, and l for the root, variable, successor, and labeling functions, respectively):

- For each $f \in \Sigma$ with arity(f) = j, G_f has rank (1, j), with l(r(1)) = f, and $s(r(1))_x = v(x)$ for each $x \in \underline{j}$.
- G_{id} has rank (1, 1), with r(1) = v(1).
- G_{ρ} has rank (2,2), with r(1) = v(2) and r(2) = v(1).
- $-G_{\nabla} \ has \ rank \ (2,1), \ with \ r(1) = r(2) = v(1).$
- $-G_1$ has rank (0,1), one empty node, and no roots.
- G_{\emptyset} is the empty graph having rank (0,0).

THEOREM 9. (Decomposition of term graphs). Every term graph can be obtained as the value of an expression containing only atomic term graphs as constants and composition and union as operators.

Proof. We first need to define some auxiliary term graphs using atomic ones, composition and union. They are shown in Figure 4, and are defined formally as follows:

[*Identities*] Term graph $G_{id}^{\underline{j}}$ is defined as $G_{id}^{\underline{j}} = G_{id} \oplus \ldots \oplus G_{id}$ (j times).

[**Permutations**] For each permutation Π over \underline{j} (i.e., a bijective mapping $\Pi: \underline{j} \to \underline{j}$), G_{Π} is the discrete term graph of rank (j, j) such that for all $x \in \underline{j}$, $v(x) = r(\Pi(x))$. Since every permutation can be obtained by a finite sequence of exchanges of pairs of adjacent elements, every such term graph can be obtained as the composition

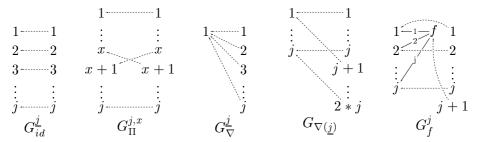


Figure 4. Auxiliary term graphs.

of some elementary graphs like $G_{\Pi}^{j,x}$ of Figure 4. In turn, we have $G_{\Pi}^{j,x}=G_{id}^{x-1}\oplus G_{\rho}\oplus G_{id}^{j-x-1}$.

For each permutation Π , we shall say that G_{Π} implements Π .

- [*Multiplicators*] For each $j \in \mathbb{N}$, term graph $G^{\underline{j}}_{\nabla}$ has rank (j,1). It is defined recursively as $G^{\underline{0}}_{\nabla} = G_!, G^{\underline{1}}_{\nabla} = G_{id}, G^{\underline{2}}_{\nabla} = G_{\nabla}$, and $G^{\underline{j}}_{\nabla} = G_{\nabla}$; $(G_{id} \oplus G^{\underline{j-1}}_{\nabla})$ if j > 2.
- [**Duplicators**] Term graph $G_{\nabla(\underline{j})}$ for $j \in \mathbb{N}^+$ has rank (2*j,j) and it is defined recursively as $G_{\nabla(\underline{1})} = G_{\nabla}$, and $G_{\nabla(\underline{j})} = (G_{\nabla} \oplus G_{\nabla(\underline{j}-1)})$; $(G_{id} \oplus G_{\Pi'} \oplus G_{id}^{\underline{j}-1})$ if j > 1, where Π' is the permutation on \underline{j} which maps 1 to j and all other numbers to their predecessors.
- [Multi-rooted atoms] Term graph G_f^j (for $f \in \Sigma$ and arity(f) = j) is similar to the atomic graph G_f , but all its nodes are roots. It has rank (j+1,j), and is defined as $G_f^j = G_{\nabla(\underline{j})}$; $(G_{id}^{\underline{j}} \oplus G_f)$.
- [Elementary term graphs] A term graph is elementary if it has rank (j+1,j) for some $j \in \mathbb{N}$, has only one nonempty node which is the j+1-th root, and for each $x \in \underline{j}$, r(x) = v(x). Every elementary term graph can be obtained as the composition of six term graphs which are easily expressed in terms of the above defined auxiliary graphs. Figure 5 shows an elementary graph G and its decomposition, that can be described, informally, as follows. G_3 has the form $G_{id}^{\underline{k}} \oplus G_f^j$; G_2 implements a permutation that puts close to each other the successor nodes of f which are shared in G; G_1 is a union of multiplicators and identities and glues together such nodes; G_4 is made of $G_!$'s and identities and 'forgets' nodes that are glued together with other nodes by G_1 ; and G_0 and G_5 are suitable permutation graphs.

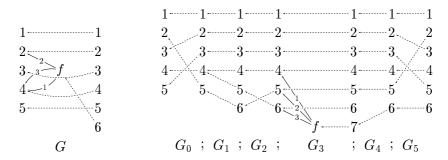


Figure 5. Example of decomposition of an elementary term graph.

We prove the statement by induction on the number of nonempty nodes of the term graph. Suppose first that G is a discrete term graph of rank (i,j), and let $g_j^i = \langle r,d,v \rangle$ be a ranked dag in G. Clearly d has exactly j nodes. For each $x \in \underline{j}$, let $a_x = \#\{y \in \underline{i} \mid r(y) = v(x)\}$ be the number of times the x-th variable of g appears in the list of roots. Then we have that $G = (G_{\overline{\nabla}}^{\underline{a_1}} \oplus \ldots \oplus G_{\overline{\nabla}}^{\underline{a_j}})$; G_{Π} for a suitable permutation Π on \underline{i} .

Now, let G be a term graph of rank (i, j) with at least one nonempty node, and let \hat{n} be a nonempty node such that all its successors are empty: Such a node must exist by acyclicity. Let G' be the term graph of rank (i, j + 1) obtained from G by making the node \hat{n} empty, and adding it as the j+1-th variable. Furthermore, let G'' be the elementary term graph of rank (j+1,j) which is the sub-graph of G containing all its variables and node \hat{n} . It is easy to see that G = G''; G'. Then since elementary term graphs can be constructed from atomic ones, the statement follows by induction hypothesis on G', since it has one nonempty node less than G.

3. GS-Monoidal Theories

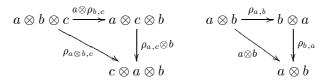
In this section we introduce graph substitution (shortly, gs-) monoidal categories: Roughly, symmetric monoidal categories equipped with two transformations (i.e., families of arrows indexed over the objects of the category). We then present the notion of gs-monoidal theory for a signature, a generalisation of the algebraic theory due to Lawvere [43], as the free gs-monoidal category built over that signature.

Since the focus of the paper is the representation result for term graphs (see Section 4), our presentation is tailored over the need of our

main theorem.³ For a full account on (symmetric) monoidal categories we refer the reader to [44]; gs-monoidal theories have been introduced in [21], under the name of *s-monoidal* theories.

DEFINITION 10. (Symmetric monoidal categories). A monoidal category \mathbf{C} is a triple $\mathbf{C} = \langle \mathbf{C}_0, \otimes, e \rangle$ where \mathbf{C}_0 is a category, $e \in \mathbf{C}_0$ is a distinguished object and $\otimes : \mathbf{C}_0 \times \mathbf{C}_0 \to \mathbf{C}_0$ is a functor, satisfying the axioms $(t \otimes t_1) \otimes t_2 = t \otimes (t_1 \otimes t_2)$ and $t \otimes e = e \otimes t = t$ for all arrows $t, t_1, t_2 \in \mathbf{C}_0$.

A symmetric monoidal category is a four-tuple $\langle \mathbf{C}_0, \otimes, e, \rho \rangle$ where $\langle \mathbf{C}_0, \otimes, e \rangle$ is a monoidal category, and $\rho : \otimes \Rightarrow \otimes \circ \mathbf{X} : \mathbf{C}_0 \times \mathbf{C}_0 \to \mathbf{C}_0$ is a natural transformation⁵ (where X is the functor that swaps its two arguments) such that $\rho_{e,e} = e$, and satisfying the axioms:



A monoidal functor $F: \mathbf{C} \to \mathbf{C}'$ is a functor $F: \mathbf{C}_0 \to \mathbf{C}'_0$ such that F(e) = e' and $F(a \otimes b) = F(a) \otimes' F(b)$. It is symmetric if moreover $F(\rho_{a,b}) = \rho'_{F(a),F(b)}$.

The category of symmetric monoidal categories and symmetric monoidal functors is denoted by SM-Cat.

EXAMPLE 11. (Free symmetric monoidal category generated by a singleton). The forgetful functor |.|: SM-Cat \rightarrow Set mapping a symmetric monoidal category to its set of objects has a left adjoint SM that maps a set S to the free symmetric monoidal category generated by S. An explicit description of category SM(S) can be given via inference rules and equations. We present them for our specific case of interest, namely SM(1) (where 1 is a singleton set).

Category $SM(1) = \langle \underline{\mathbb{N}}, \otimes, \underline{0}, \rho \rangle$ has as objects underlined natural numbers $\underline{n} \in \underline{\mathbb{N}}$, on which functor \otimes is defined as $\underline{n} \otimes \underline{m} = \underline{n+m}$ (in fact, objects must form a monoid generated by a singleton set; the unit

³ In fact, the reader will note that our definition of monoidal categories actually characterises what are called *strict* monoidal categories in the literature. Since no confusion can arise, we omitted the prefix 'strict' in order to ease the notation. Also, we only consider monoidal functors that preserve monoidal product and unit 'on the nose'.

⁴ We often denote the identity of an object by the object itself.

⁵ Given functors $F, G: \mathbf{A} \to \mathbf{B}$, a transformation $\tau: F \Rightarrow G: \mathbf{A} \to \mathbf{B}$ is a family of arrows of \mathbf{B} indexed by objects of \mathbf{A} , $\tau = \{\tau_a: F(a) \to G(a) \mid a \in |\mathbf{A}|\}$. Transformation τ is natural if for every arrow $f: a \to a'$ in \mathbf{A} , $\tau_a; G(f) = F(f); \tau_{a'}$.

is clearly $\underline{0}$). Arrows are equivalence classes of terms generated by the following rules

$$\frac{n \in \mathbb{N}}{id_{\underline{n}} : \underline{n} \to \underline{n}}, \qquad \frac{n, m \in \mathbb{N}}{\rho_{\underline{n},\underline{m}} : \underline{n} \otimes \underline{m} \to \underline{m} \otimes \underline{n}},$$

$$\frac{t:\underline{n}\to\underline{m},t':\underline{m}\to\underline{k}}{t;t':\underline{n}\to\underline{k}},\qquad \frac{t:\underline{n}\to\underline{m},t':\underline{n}'\to\underline{m}'}{t\otimes t':\underline{n}\otimes\underline{n}'\to\underline{m}\otimes\underline{m}'}$$

with respect to the following equations: For all $\underline{n}, \underline{m}, \underline{k}, \underline{n}', \underline{m}' \in \underline{\mathbb{N}}$, and for all arrows $t, t_1, t_2, t_3 \in SM(\mathbf{1})$

(categories)

$$id_{\underline{n}}; t = t = t; id_{\underline{m}}$$
 for $t : \underline{n} \to \underline{m}$
 $(t; t_1); t_2 = t; (t_1; t_2)$

(functoriality)

$$id_{\underline{n}\otimes \underline{m}} = id_{\underline{n}} \otimes id_{\underline{m}}$$

 $(t;t_1)\otimes (t_2;t_3) = (t\otimes t_2); (t_1\otimes t_3)$ whenever both sides are defined

(monoidality)

$$t \otimes id_{\underline{0}} = t = id_{\underline{0}} \otimes t (t \otimes t_1) \otimes t_2 = t \otimes (t_1 \otimes t_2)$$

(naturality)

$$\rho_{n',n}$$
; $(t \otimes t_1) = (t_1 \otimes t)$; $\rho_{m',m}$ for $t : \underline{n} \to \underline{m}$, $t_1 : \underline{n'} \to \underline{m'}$

(symmetricity)

$$\begin{array}{l} \rho_{\underline{0},\underline{0}} = id_{\underline{0}} \\ \rho_{\underline{n},\underline{m}}; \rho_{\underline{m},\underline{n}} = id_{\underline{n}\otimes\underline{m}}, \ and \\ \rho_{\underline{k}\otimes\underline{n},\underline{m}} = (id_{\underline{k}}\otimes\rho_{\underline{n},\underline{m}}); (\rho_{\underline{k},\underline{m}}\otimes id_{\underline{n}}). \end{array}$$

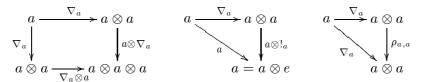
Note also that for all the arrows of SM(1) their source and target actually coincide (that is, for $\underline{n} \neq \underline{m}$, the hom-set $SM(1)[\underline{n},\underline{m}]$ is empty). For each $\underline{n} \in \mathbb{N}$, we call an arrow in $SM(1)[\underline{n},\underline{n}]$ a symmetry over \underline{n} .

Intuitively, symmetric monoidal categories formalise in categorical terms the basic notions of *pairing* and *permutation*, as shown by the many coherence results for these categories (originating from [45]). The following well-known fact provides a simple characterisation of symmetries that will be of use in the next section.

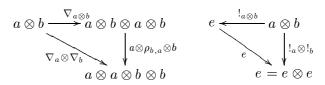
LEMMA 12. (Symmetries are permutations). For any $n \in \mathbb{N}$, the set of symmetries over \underline{n} is isomorphic to the set of permutations over the set $\underline{n} = \{1, \ldots, n\}$.

In other terms, this means that the set of symmetries over \underline{n} is isomorphic to the set of all bijective functions from $\{1, \ldots, n\}$ to itself. An analogous characterisation for the set of *all* possible functions can be obtained, if we enrich the categories at hand.

DEFINITION 13. (gs-monoidal categories). A gs-monoidal category \mathbf{C} is a six-tuple $\langle \mathbf{C}_0, \otimes, e, \rho, \nabla, ! \rangle$, where $\langle \mathbf{C}_0, \otimes, e, \rho \rangle$ is a symmetric monoidal category and $!: Id_{\mathbf{C}_0} \Rightarrow e: \mathbf{C}_0 \to \mathbf{C}_0, \nabla: Id_{\mathbf{C}_0} \Rightarrow \otimes \circ \Delta: \mathbf{C}_0 \to \mathbf{C}_0$ are two transformations (Δ is the diagonal functor), such that $!_e = \nabla_e = e$ and satisfying the coherence axioms:



and the monoidality axioms:



A gs-monoidal functor $F: \mathbf{C} \to \mathbf{C}'$ is a symmetric monoidal functor such that $F(!_a) = !'_{F(a)}$ and $F(\nabla_a) = \nabla'_{F(a)}$. The category of gs-monoidal categories and gs-monoidal functors is denoted by **GSM-Cat**.

Informally, gs-monoidal theories provide an equational description of the notion of duplication of pointers for symmetric monoidal categories. More precisely, if an arrow $t:a\to b$ is considered as a data structure pointed to by object b, then arrow $t;\nabla_b:a\to b\otimes b$ intuitively represents the same data structure but with two pointers to it. The related axioms assure that this duplication of pointers is unique, as shown (for the specific case of the free gs-monoidal category generated by a singleton set) by Lemma 15 and Corollary 16 below. There are many related categorical structures proposed in the literature: We briefly discuss them in Section 6.6.

EXAMPLE 14. (Free gs-monoidal category generated by a singleton). Similarly to Example 11, let $GSM: \mathbf{Set} \to \mathbf{GSM\text{-}Cat}$ be the left adjoint functor mapping each set to its free gs-monoidal category. An explicit description of $GSM(\mathbf{1}) = \langle \underline{\mathbb{N}}, \otimes, \underline{0}, \rho, \nabla, ! \rangle$ is easily obtained by adding to the description of $SM(\mathbf{1})$ in Example 11 the following inference rules for arrows

$$\frac{n \in \mathbb{N}}{!_{\underline{n}} : \underline{n} \to \underline{0}}, \qquad \frac{n \in \mathbb{N}}{\nabla_{\underline{n}} : \underline{n} \to \underline{n} \otimes \underline{n}}$$

and the following axioms:⁶ For all $\underline{n}, \underline{m}, \underline{k} \in \underline{\mathbb{N}}$,

$$\begin{split} &- \ !_{\underline{0}} = \nabla_{\underline{0}} = id_{\underline{0}}, \\ &- \nabla_{\underline{n}}; (id_{\underline{n}} \otimes \nabla_{\underline{n}}) = \nabla_{\underline{n}}; (\nabla_{\underline{n}} \otimes id_{\underline{n}}), \\ &- \nabla_{\underline{n}}; (id_{\underline{n}} \otimes !_{\underline{n}}) = id_{\underline{n}}, \\ &- \nabla_{\underline{n}}; \rho_{\underline{n},\underline{n}} = \nabla_{\underline{n}}, \\ &- \nabla_{\underline{n} \otimes \underline{m}}; (id_{\underline{n}} \otimes \rho_{\underline{m},\underline{n}} \otimes id_{\underline{m}}) = \nabla_{\underline{n}} \otimes \nabla_{\underline{m}}, \\ &- !_{\underline{n} \otimes \underline{m}} = !_{\underline{n}} \otimes !_{\underline{m}}. \end{split}$$

The next result provides an easy characterisation for the arrows in $GSM(\mathbf{1})$.

LEMMA 15. (Arrows of $GSM(\mathbf{1})$ are functions). For any $n, m \in \mathbb{N}$, the set of arrows from \underline{n} to \underline{m} in $GSM(\mathbf{1})$ (i.e., the hom-set $GSM(\mathbf{1})[\underline{n},\underline{m}]$) is isomorphic to the set of all functions from $\{1,\ldots,m\}$ to $\{1,\ldots,n\}$.

Proof outline. A direct proof of the statement is quite cumbersome but not really needed. It follows from the observation that GSM(1) is a Cartesian category, because the equations $t; \nabla_{\underline{m}} = \nabla_{\underline{n}}; (t \otimes t)$ and $t; \underline{!}_{\underline{m}} = \underline{!}_{\underline{n}}$ hold for all $t: \underline{n} \to \underline{m}$ in GSM(1). Hence it is isomorphic to the algebraic theory generated by the empty signature, in which arrows from \underline{n} to \underline{m} are one-to-one with m-tuples of variables taken from the set $\{x_1, \ldots, x_n\}$. See Section 5 for a more detailed discussion.

COROLLARY 16. (Multiplicators). For each $n \in \mathbb{N}$, the hom-set $GSM(\mathbf{1})[\underline{1},\underline{n}]$ is a singleton: We denote the only arrow in it by ∇^n , and call it a multiplicator. We obviously have $\nabla^0 = !_{\underline{1}}$, $\nabla^1 = id_{\underline{1}}$, and $\nabla^2 = \nabla_{\underline{1}}$.

Moreover, if n = m + k + 1, then ∇^n ; $(id_{\underline{m}} \otimes \nabla_{\underline{1}} \otimes id_{\underline{k}}) = \nabla^{n+1}$ and ∇^n ; $(id_{\underline{m}} \otimes !_{\underline{1}} \otimes id_{\underline{k}}) = \nabla^{n-1}$. Finally, for every symmetry $\rho : \underline{n} \to \underline{n}$, we have ∇^n ; $\rho = \nabla^n$.

⁶ Even if these are just instances of the diagrams in the previous definition, we find convenient to summarise them for readers that may prefer an equational presentation for axioms.

Let us explain now how a free gs-monoidal category can be generated from a (one-sorted) signature Σ . The idea is as follows. Every gs-monoidal category M has an underlying generalised signature having a monoid of sorts (the objects of M) and all the arrows of M as operators. This defines a forgetful functor from **GSM-Cat** to **GenSig**, the category of generalised signatures. This functor has an obvious left adjoint (denoted **GS-Th**) that preserves the monoid of sorts of its argument signature, and generates in a free way all the missing structure on arrows. We will obtain the free gs-monoidal category of a signature Σ as the action of **GS-Th** over Σ .

DEFINITION 17. (Generalised signatures). A generalised signature Σ is a four-tuple $\langle M, T, s, t \rangle$, where M is a monoid, T a set of operators, and $s, t : T \to U(M)$ are (the source and target) functions, where U(M) is the underlying set of M. A morphism of generalised signatures $f : \Sigma \to \Sigma'$ is a pair $\langle f_M : M \to M', f_T : T \to T' \rangle$, where f_M is a monoid homomorphism and f_T a function preserving sources and targets. Generalised signatures and their morphisms form a category, denoted GenSig.

Note that a one-sorted signature is a generalised signature where M is the free monoid generated by a singleton (whose elements we denote by underlined natural numbers), and $f: \underline{n} \to \underline{1}$ if arity(f) = n.

DEFINITION 18. (gs-monoidal theories). Let $V: \mathbf{GSM\text{-}Cat} \to \mathbf{GenSig}$ be the forgetful functor mapping a gs-monoidal category to the underlying generalised signature, and let $\mathbf{GS\text{-}Th}: \mathbf{GenSig} \to \mathbf{GSM\text{-}Cat}$ be its left (free) adjoint. Then the gs-monoidal theory of a given (one-sorted) signature Σ is defined as $\mathbf{GS\text{-}Th}(\Sigma)$, i.e., the free gs-monoidal category generated by Σ .

An explicit description of category $\mathbf{GS-Th}(\Sigma)$ for a given signature can be easily provided on the basis of Examples 11 and 14. In fact, such a category is obtained by generating arrows using all the inference rules listed in those examples plus the following rule:

(generators)
$$\frac{f \in \Sigma_n}{f_{\Sigma} : \underline{n} \to \underline{1}}$$

and then by imposing all the axioms listed in the examples.

More explicitly, it will be convenient to regard the arrows of $\mathbf{GS-Th}(\Sigma)$ as equivalence classes of terms as follows. Let $\Psi(\Sigma)$ be the signature having as constants the (countable) set $\{id_n, \rho_{n,m}, \nabla_n, !_n, f_{\Sigma}\}$ for $n, m \in \mathbb{N}$ and $f \in \Sigma$, and as binary operators the set $\{;, \otimes\}$.

Let $\hat{T}_{\Psi(\Sigma)}$ be the set of terms over $\Psi(\Sigma)$ generated by the inference rules of Examples 11 and 14, plus the rule (generators) above. Then, quite obviously, an arrow t of GSM(1) is an equivalence class of terms $t \in \hat{T}_{\Psi(\Sigma)}$ with respect to the equivalence induced by the axioms of Examples 11 and 14.

Note that there is an obvious chain of inclusion functors $SM(1) \hookrightarrow GSM(1) \hookrightarrow GS-Th(\Sigma)$. The following result will be useful in the following.

LEMMA 19. (Decomposition of arrows of gs-monoidal theories). For any signature Σ , for each arrow t of the associated gs-monoidal theory $\mathbf{GS-Th}(\Sigma)$, either $t=id_{\underline{0}}$ or it can be decomposed into the composition $t_0; \ldots; t_k$, with $k \geq 0$, of arrows of the form $t_i = (\bigotimes_{j=1}^{n_i} id_{\underline{1}}) \otimes t_i' \otimes (\bigotimes_{j=1}^{m_i} id_{\underline{1}})$, where $n_i, m_i \geq 0$, and $t_i' \in \{id_{\underline{1}}, \rho_{\underline{1},\underline{1}}, \nabla_{\underline{1}}, !_{\underline{1}}\} \cup \{f_{\Sigma} \mid f \in \Sigma\}$.

Proof. One way of proving this is to show that every term in $\hat{T}_{\Psi(\Sigma)}$ is equivalent to a term of the required shape in $\hat{T}_{\Psi_0(\Sigma)}$, where signature $\Psi_0(\Sigma)$ has the (finite) set of constants $\{id_{\underline{1}}, \rho_{\underline{1},\underline{1}}, \nabla_{\underline{1}}, !_{\underline{1}}\} \cup \{f_{\Sigma} \mid f \in \Sigma\}$, and $\{;, \otimes\}$ as binary operators.

Firstly, note that all constants in $\{id_{\underline{n}}, \rho_{\underline{n},\underline{m}}, \nabla_{\underline{n}}, \underline{!}_{\underline{n}}\}$ are equivalent to terms over $\Psi_0(\Sigma)$. This is obvious for $id_{\underline{n}}$ and $\underline{!}_{\underline{n}}$; for $\rho_{\underline{n},\underline{m}}$ and $\nabla_{\underline{n}}$ it can be shown by repeated applications of the last axiom of Example 11 and of the fifth axiom of Example 14, exploiting also the fact that $\rho_{\underline{n},\underline{m}} = \rho_{\underline{m},\underline{n}}^{-1}$. Since the (binary) operators in $\Psi(\Sigma)$ and $\Psi_0(\Sigma)$ coincide, this is sufficient to shows that every term over $\Psi(\Sigma)$ is equivalent to a term over $\Psi_0(\Sigma)$.

Secondly, any term in $T_{\Psi_0(\Sigma)}$ can be transformed into a sequential composition of parallel components by lifting all semicolons to the outermost level, by repeated applications of the second axiom for functoriality in Example 11; sometimes identities have to be added, as in $t \otimes (t'; t'') = (t; id_{\underline{n}}) \otimes (t'; t'') = (t \otimes t'); ((\bigotimes_{j=1}^n id_{\underline{1}}) \otimes t'')$. Finally the same technique (i.e., adding identities and applying functoriality) can be used to leave in each component of the sequential decomposition only one basic arrow different from id_1 .

4. Term Graphs are Arrows of the GS-Monoidal Theory

In this section we prove the main result of the paper, namely, the correspondence between the gs-monoidal theory of a signature Σ and the category \mathbf{TG}_{Σ} of term graphs over Σ . The first observation is that \mathbf{TG}_{Σ} can be equipped with a gs-monoidal structure.

LEMMA 20. (The category of term graphs is gs-monoidal). Let \mathbf{TG}_{Σ} be the category introduced in Definition 5. Then the structure $\langle \mathbf{TG}_{\Sigma}, \oplus, \underline{0}, G_{\Pi(\underline{\cdot},\underline{\cdot})}, G_{\nabla(\underline{\cdot})}, G_{\overline{\uparrow}} \rangle$ is a gs-monoidal category, where:

- Functor \oplus : $\mathbf{TG}_{\Sigma} \times \mathbf{TG}_{\Sigma} \to \mathbf{TG}_{\Sigma}$ is defined on arrows as in Definition 6, and on objects as $\underline{n} \oplus \underline{m} = \underline{n+m}$;
- For each $n, m, G_{\Pi(\underline{n},\underline{m})}$ is the discrete term graph implementing the permutation on $\underline{n+m}$ such that $\Pi(\underline{n},\underline{m})(x) = x+m$ if $x \leq n$, and $\Pi(\underline{n},\underline{m})(x) = x-n$ otherwise (see the proof of Theorem 9);
- For each n, $G_{\nabla(n)}$ is as defined in the proof of Theorem 9; and
- For each n, $G^{\underline{n}}_{!}$ is the discrete term graph with n variables and with no roots.

Proof. It is easy to show that $\langle \mathbf{TG}_{\Sigma}, \oplus, \underline{0} \rangle$ is a (strict) monoidal category because union of term graphs is associative and functoriality of \oplus holds. Furthermore, by straightforward inspection it can be shown that $G_{\Pi(_,_)}$ is a natural isomorphism, and that the transformations $G_{\Pi(_,_)}, G_{\nabla(_)}$, and $G_{\overline{\cdot}}$ make all the diagrams of Definitions 10 and 13 commute.

Therefore the axioms for gs-monoidality are sound for term graphs, i.e., any equation that can be deduced from the theory of gs-monoidal categories also holds for term graphs (via the translation provided by the functor in the next proposition). The main result of the paper implicitly states that those axioms are also complete. For the time being, we can summarise Lemma 19, Lemma 20 and Theorem 9 with the following proposition.

PROPOSITION 21. (The full functor for gs-monoidal theories). For any signature Σ , a full gs-monoidal functor $F_{\Sigma}: \mathbf{GS-Th}(\Sigma) \to \mathbf{TG}_{\Sigma}$ can be defined inductively.

Proof. By the freeness of $\mathbf{GS-Th}(\Sigma)$ and since \mathbf{TG}_{Σ} is gs-monoidal, a gs-monoidal functor $F_{\Sigma}: \mathbf{GS-Th}(\Sigma) \to \mathbf{TG}_{\Sigma}$ is uniquely determined by imposing $F_{\Sigma}(\underline{1}) = \underline{1}$ on objects, and $F_{\Sigma}(f_{\Sigma}) = G_f$ on generators (i.e., the arrows corresponding to operators of the signature).

More explicitly, functor F_{Σ} will map the basic arrows $id_{\underline{0}}$, $id_{\underline{1}}$, $\rho_{\underline{1},\underline{1}}$, $\nabla_{\underline{1}}$, $!_{\underline{1}}$, and f_{Σ} to the atomic term graphs G_{\emptyset} , G_{id} , G_{ρ} , G_{∇} , $G_{!}$, and G_{f} , respectively, and it will preserve monoidal product and sequential composition. Lemma 19 ensures that in this way F_{Σ} is defined on all arrows of $\mathbf{GS-Th}(\Sigma)$. This explicit definition also shows that all the atomic term graphs of Definition 8 are in the range of F_{Σ} . Thus by Theorem 9 it follows that F_{Σ} is full (i.e., surjective over the hom-sets). \square

Actually, F_{Σ} is also injective on hom-sets: in categorical terms, this is expressed by the following proposition.

LEMMA 22. (Faithfulness). For any signature Σ , the functor F_{Σ} : $\mathbf{GS-Th}(\Sigma) \to \mathbf{TG}(\Sigma)$ is faithful.

The two propositions imply that the arrows of the hom-set $\mathbf{GS-Th}(\Sigma)[\underline{n},\underline{m}]$ are in a one-to-one correspondence with the term graphs with m roots and n variables. And since F_{Σ} is obviously bijective over objects, our main theorem immediately follows.

THEOREM 23. (Representation theorem for term graphs). For any signature Σ , the gs-monoidal categories $\mathbf{GS-Th}(\Sigma)$ and \mathbf{TG}_{Σ} are isomorphic.

The rest of this section is devoted to the proof of Lemma 22. As a first result, it is easy to show that the functor $F_{\Sigma} : \mathbf{GS-Th}(\Sigma) \to \mathbf{TG}_{\Sigma}$ restricts to an isomorphism between the arrows of GSM(1) and the discrete term graphs over Σ .

PROPOSITION 24. (Faithfulness for the empty signature). For any signature Σ , the functor $F_{\Sigma} : \mathbf{GS-Th}(\Sigma) \to \mathbf{TG}_{\Sigma}$ restricts to a full functor $F_{\emptyset} : GSM(1) \to \mathbf{DTG}_{\Sigma}$, where \mathbf{DTG}_{Σ} is the sub-category of \mathbf{TG}_{Σ} including only discrete term graphs as arrows. Moreover, F_{\emptyset} is faithful.

Proof. An arrow t of $GSM(\mathbf{1}) \subseteq \mathbf{GS-Th}(\Sigma)$ is generated by inference rules and equations of Examples 11 and 14, thus it does not contain any generator. By the definition of F_{Σ} in the proof of Proposition 21, clearly $F_{\Sigma}(t)$ is discrete, and this ensures that $F_{\emptyset}: GSM(\mathbf{1}) \to \mathbf{DTG}_{\Sigma}$ is well defined. Fullness of F_{\emptyset} follows by instantiating Proposition 21 for the empty signature Σ_{\emptyset} , by observing that $GSM(\mathbf{1}) = \mathbf{GS-Th}(\Sigma_{\emptyset})$, $\mathbf{DTG}_{\Sigma} = \mathbf{TG}_{\Sigma_{\emptyset}}$, and $F_{\emptyset} = F_{\Sigma_{\emptyset}}$.

Next we show that discrete term graphs of rank (m, n) are in one-to-one correspondence with functions from \underline{m} to \underline{n} . In fact, let $G = [\langle r, g, v \rangle]$ be discrete, with $r : \underline{m} \to N(g)$ and $v : \underline{n} \to N(g)$. Since all nodes of g are empty, v is an isomorphism and we obtain a function $\phi(G) = v^{-1} \circ r : \underline{m} \to \underline{n}$; it is easy to see that it is well-defined, as it does not depend on the concrete representative chosen in G. Conversely, for a function $h : \underline{m} \to \underline{n}$ let $\psi(h)$ be the discrete term graph $\psi(h) = [\langle h, \underline{n}, id_{\underline{n}} \rangle]$ (where \underline{n} is the discrete dag with n nodes). It is easy to check that ϕ and ψ are inverse to each other.

Finally faithfulness of F_{\emptyset} follows because for each $n, m \in \mathbb{N}$ it restricts to a surjective function (by fullness) from $GSM(1)[\underline{n}, \underline{m}]$ to

 $\mathbf{DTG}_{\Sigma}[\underline{n},\underline{m}]$, which are two isomorphic finite sets, by Lemma 15 and the above discussion.

Now for the general case, recall that arrows of $\mathbf{GS-Th}(\Sigma)$ and \mathbf{TG}_{Σ} are actually equivalence classes. An arrow of $\mathbf{GS-Th}(\Sigma)$ is an equivalence class of terms of a suitable signature, as remarked just before Lemma 19, while a term graph is an isomorphism class of ranked dag's. To prove Lemma 22 we need to move to a more concrete setting, by considering representatives of those equivalence classes.

The next definition shows how to associate with a concrete representative τ of an arrow t, a ranked dag $\delta(\tau)$ that is a representative of the term graph $F_{\Sigma}(t)$, and also a suitable function ξ_{τ} relating generators in τ to nodes of $\delta(\tau)$.

DEFINITION 25. (From representatives of arrows to dag's). Let $\Psi(\Sigma)$ be the signature and $\hat{T}_{\Psi(\Sigma)}$ be the set of terms as defined just before Lemma 19. It is convenient to regard a term $\tau \in \hat{T}_{\Psi(\Sigma)}$ as a partial function $\tau : \{0,1\}^* \to \Psi(\Sigma)$, such that its domain $\mathcal{O}(\tau)$ (also called the set of occurrences of τ) satisfies for $w \in \{0,1\}^*$ and $i \in \{0,1\}$:

1. $\mathcal{O}(\tau)$ is finite, nonempty, and left-closed (i.e., if $wi \in \mathcal{O}(\tau)$ then $w \in \mathcal{O}(\tau)$);

2.wi $\in \mathcal{O}(\tau)$ implies $\tau(w) \in \{;, \otimes\}$, and $\tau(w) \in \{;, \otimes\}$ implies $\{w0, w1\} \subset \mathcal{O}(\tau)$.

For $\tau \in \hat{T}_{\Psi(\Sigma)}$, let $\mathcal{O}_{\Sigma}(\tau) \subseteq \{0,1\}^*$ be the set of all occurrences of generators in τ , i.e., $\mathcal{O}_{\Sigma}(\tau) = \{w \in \mathcal{O}(\tau) \mid \tau(w) = f_{\Sigma} \text{ for some } f \in \Sigma\}$.

Now let t be a fixed arrow in **GS-Th**(Σ). For each term $\tau \in t$ we define a ranked dag $\delta(\tau)$ which is a concrete representative of the term graph $F_{\Sigma}(t)$, and, at the same time, a function $\xi_{\tau}: \mathcal{O}_{\Sigma}(\tau) \to N(\delta(\tau))$, i.e., to the nodes of $\delta(\tau)$. Note that for $\delta(\tau)$ we do not need to describe the entire structure, but it is sufficient to show how a ranked dag in $F_{\Sigma}(t)$ can be determined uniquely. $\delta(\tau)$ and ξ_{τ} are defined by induction on the structure of τ .

1.If τ is in $\{id_n, \rho_{\underline{n},\underline{m}}, \nabla_{\underline{n}}, \underline{!}_{\underline{n}}\}$, then let $\delta(\tau)$ be the (only) discrete ranked dag in $F_{\Sigma}(t)$ having natural numbers as nodes, and such that the variable function is the identity. Function $\xi_{\tau}: \mathcal{O}_{\Sigma}(\tau) \to N(\delta(\tau))$ is necessarily the empty function, because $\mathcal{O}_{\Sigma}(\tau) = \emptyset$.

⁷ Informally, $\mathcal{O}(\tau)$ is the set of nodes of the syntactical tree of τ , represented by their access path.

- 2. If $\tau = f_{\Sigma}$ with $f \in \Sigma$ and arity(f) = n, then let $\delta(\tau)$ be the (only) ranked dag in $F_{\Sigma}(t)$ having $\underline{n+1}$ as set of nodes, such that the variable function is the identity, and where n+1 is the only nonempty node. In this case clearly $\mathcal{O}_{\Sigma}(\tau) = \{\lambda\}$, where λ is the empty string, and we define $\xi_{\tau}(\lambda) = n+1$.
- 3. If $\tau = \sigma_0 \otimes \sigma_1$, then let $\delta(\tau) = \delta(\sigma_0) \underline{\oplus} \delta(\sigma_1)$, where $\underline{\oplus}$ is almost like the union of term graphs (see Definition 6). Only, to ensure that $\delta(\tau)$ is a well defined dag, we have to provide an explicit construction of the disjoint union of ranked dag's. It suffices to provide it for nodes, by stating that $N(d_0 \underline{\oplus} d_1) = \{\langle 0, n \rangle \mid n \in N(d_0)\} \cup \{\langle 1, n \rangle \mid n \in N(d_1)\}$.
 - Next we clearly have $\mathcal{O}_{\Sigma}(\tau) = \{0w \mid w \in \mathcal{O}_{\Sigma}(\sigma_0)\} \cup \{1w \mid w \in \mathcal{O}_{\Sigma}(\sigma_1)\}$, and we define ξ_{τ} as $\xi_{\tau}(0w) = \langle 0, \xi_{\sigma_0}(w) \rangle$, and $\xi_{\tau}(1w) = \langle 1, \xi_{\sigma_1}(w) \rangle$.
- 4. If $\tau = \sigma_0$; σ_1 , then we can follow the same idea as in the previous case to define $\delta(\tau)$ and ξ_{τ} . More precisely, $\delta(\tau)$ can be defined uniquely by making the operation of composition of ranked dag's (see Definition 4) deterministic, for example using an explicit construction of disjoint union as in the previous case; and ξ_{τ} will then be uniquely determined by ξ_{σ_0} and ξ_{σ_1} .

By construction, function ξ_{τ} is easily shown to be an isomorphism between $\mathcal{O}_{\Sigma}(\tau)$ and the set of nonempty nodes of $\delta(\tau)$.

The next lemma shows that for each arrow t of $\mathbf{GS-Th}(\Sigma)$, we can choose a term $\tau \in t$ having a suitable structure that mirrors, to some extent, the structure of the term graph associated with t. For the rest of the section $\rho, \rho', \rho_i, \ldots$ range over symmetries, i.e., arrows of SM(1).

LEMMA 26. (Decomposition lemma). Let Σ be a signature and t: $\underline{n} \to \underline{m}$ be an arrow of the associated gs-monoidal theory **GS-Th**(Σ), different from id_0 .

1. There exists a natural number $q \geq 0$ and a term $\tau \in t$ over $\Psi(\Sigma)$, such that $\tau = \tau_0; \tau_1; \ldots; \tau_q, {}^8$ where $\tau_0 = ((\bigotimes_{j=1}^n \nabla^{k_{0,j}}); \rho_0)$, and for each $i \in \underline{q}$, $\tau_i = (id_{\underline{p_i}} \otimes (\bigotimes_{j=1}^{l_i} (f_{\Sigma}^{i,j}; \nabla^{k_{i,j}})); \rho_i$, for suitable natural numbers $p_i, k_{i,j} \geq 0$ and $l_i \geq 1$, and function symbols $f_{\Sigma}^{i,j}$ in Σ . We

When \otimes and; denote binary operators of $\Psi(\Sigma)$, formally speaking they are not associative. We simplify the notation by assuming that they associate to the right. Thus, for example, a term τ_0 ; τ_1 ; τ_2 over $\Psi(\Sigma)$ should be read as $(\tau_0; (\tau_1; \tau_2))$.

call $\nabla^{k_{i,j}}$ the multiplicator of $f_{\Sigma}^{i,j}$, ρ_i the symmetry of level i, and q the depth of τ .

For each $w \in \mathcal{O}_{\Sigma}(\tau)$ (i.e., an occurrence of an operator of Σ in τ), we say that the level of w in the decomposition is i if and only if w is the occurrence of an operator appearing in τ_i .

2. For a term τ as in the previous point, and $w, v \in \mathcal{O}_{\Sigma}(\tau)$, let $w <_{\tau} v$ iff the level of w is lower than the level of v. Moreover, let $w \sqsubseteq_{\tau} v$ iff there is a nonempty path in $\delta(\tau)$ from node $\xi_{\tau}(v)$ to node $\xi_{\tau}(w)$ (see Definition 25 for the notation).

Then for each $w, v \in \mathcal{O}_{\Sigma}(\tau)$, $w \sqsubseteq_{\tau} v$ implies $w <_{\tau} v$. In other words, the existence of a path between two nodes of a term graph implies that in any term representing it, and having the form described in 1, the (operator corresponding to the) source node appears in a higher level than the target node.

3. Let the graph level of an occurrence $w \in \mathcal{O}_{\Sigma}(\tau)$ be the length of the longest path starting from $\xi_{\tau}(w)$ and ending in a nonempty node, plus one. Moreover, let the graph depth of τ be the maximal graph level for the occurrences in $\mathcal{O}_{\Sigma}(\tau)$. Then for each τ with a structure as in point 1 above, it is possible to build an equivalent term σ having a similar structure, such that (i) the depth of σ is equal to the graph depth of τ , and (ii) for each $w \in \mathcal{O}_{\Sigma}(\sigma)$ the level of w is equal to its graph level.

Proof. Let $t: \underline{n} \to \underline{m}$ be an arrow of **GS-Th**(Σ) different from id_0 .

1. By Lemma 19 t includes a term σ on $\Psi_0(\Sigma)$ of the form $\sigma = \sigma_0; \sigma_1; \ldots; \sigma_k$ with $k \geq 1$, such that for all $i \leq k$, $\sigma_i = (id_{n_i} \otimes \sigma_i' \otimes id_{\underline{m_i}})$, where $n_i, m_i \geq 0$ and $\sigma_i' \in \{id_{\underline{1}}, \rho_{\underline{1},\underline{1}}, \nabla_{\underline{1}}, !_{\underline{1}}\} \cup \{f_{\Sigma} \mid f \in \Sigma\}$. We show by induction on k how σ can be transformed into an equivalent term $\tau = \tau_0; \ldots; \tau_q$ over $\Psi(\Sigma)$ having the required structure.

[Base case] Let k=0, thus $\sigma=\sigma_0=(id_{\underline{n_0}}\otimes\sigma'_0\otimes id_{\underline{m_0}})$. If $\sigma'_0\in\{id_{\underline{1}},\rho_{\underline{1},\underline{1}},\nabla_{\underline{1}},!_{\underline{1}}\}$, then a term τ_0 equivalent to σ and having the desired structure is easily obtained, recalling that $!_{\underline{1}}=\nabla^0$, $id_{\underline{1}}=\nabla^1$, and $\nabla_{\underline{1}}=\nabla^2$. If $\sigma'_0=f_{\Sigma}$ and arity(f)=k, then it is readily checked that σ is equivalent to term $\tau_0;\tau_1$, with $\tau_0=id_{\underline{n_0}}\otimes\rho_{\underline{k},\underline{m_0}}$ and $\tau_1=(id_{\underline{n_0}+\underline{m_0}}\otimes f_{\Sigma});(id_{\underline{n_0}}\otimes\rho_{\underline{m_0},\underline{1}})$, having the required structure.

⁹ More formally, the level of $w \in \mathcal{O}_{\Sigma}(\tau) \subseteq \{0, 1\}^*$ is the greatest $i \in \underline{q}$ such that 1^i is a prefix of w.

[Inductive case] Let $\sigma = \sigma_0; \ldots; \sigma_k$ for k > 0. By induction hypotesis we can assume that $\sigma = \tau_0; \ldots; \tau_q; \sigma_k$, where $\sigma_k = (id_{\underline{n_k}} \otimes \sigma'_k \otimes id_{\underline{m_k}})$, and $\tau_0; \ldots; \tau_q$ has the structure described in the statement.

We proceed by case analysis on σ'_k . If $\sigma'_k \in \{id_{\underline{1}}, \rho_{\underline{1},\underline{1}}\}$, then σ_k is a symmetry and we are done because it can be merged with the symmetry ρ_q of τ_q , providing the required decomposition. If $\sigma'_k = f_{\Sigma}$, then as in the Base case σ_k can be decomposed as the sequential composition of two terms, say τ'_q ; τ_{q+1} . Since τ'_q is actually a symmetry and τ_{q+1} has the desired structure, we get the required decomposition of σ as $\tau_0; \ldots; (\tau_q; \tau'_q); \tau_{q+1}$, where we consider τ'_q as merged with symmetry ρ_q .

Finally, suppose that $\sigma'_k \in \{\nabla_{\underline{1}}, !_{\underline{1}}\}$. By naturality of symmetries and identities, σ_k can be shifted towards left along the decomposition $\tau_0; \ldots; \tau_q$ till when σ'_k happens to be composed with a multiplicator, by which it will be 'absorbed' as described in Corollary 16.

- 2. It is easy to check that in the composition G; G' of two term graphs (see Definition 4) there cannot be any path from a nonempty node of G to a nonempty node of G', because of acyclicity. The statement follows because the inductive construction of $\delta(\tau)$ interprets the operator ';' on terms as (a concrete version of) composition on ranked dag's.
- 3. Let τ be a term with the structure as in point 1 above. From point 2 we deduce that the depth of τ is greater than or equal to the graph depth of τ , and that for each $w \in \mathcal{O}_{\Sigma}(\tau)$ the level of w is greater than or equal to its graph level. Clearly, if for each $w \in \mathcal{O}_{\Sigma}(\tau)$ its level and graph level are equal, then the depth and the graph depth of τ are equal and we are done.

Now let w be an occurrence of generator of τ of minimal level (say i) such that its level differs from its graph level (that can be at most i-1). Informally, the operator occurrence $\tau(w)$ appearing in τ_i has all its argument operators at most in τ_{i-2} (because by hypothesis their levels and graph levels are equal), which implies that operator $\tau(w)$ composes only with identities of level i-1. Thus it is possible to shift $\tau(w)$ (and the associated multiplicator) to level i-1 by readjusting the identities of levels i-1 and i, as well as all involved symmetries; in particular, if $\tau(w)$ was the only operator in τ_i , to obtain the desired structure it is necessary to merge the new level i (which is equivalent to a symmetry) with the symmetry of level i-1, and the resulting term will have a depth smaller than τ

(in fact, note that in the structure of terms considered in point 1, there must be at least one operator per level). This process can be iterated and eventually the desired equivalent term σ is obtained; the termination of this algorithm follows by the observation that the quantity $\sum_{w \in \mathcal{O}_{\Sigma}(\tau)} level(w) - graph_level(w)$ decreases at each iteration, and it must be positive.

We have now all the ingredients needed to prove Lemma 22.

Proof of the Faithfulness Lemma (Lemma 22). Let s and t be two arrows of $\mathbf{GS-Th}(\Sigma)$, such that $F_{\Sigma}(s) = F_{\Sigma}(t)$. We have to show that s = t, or, equivalently, that there exist two terms over $\Psi(\Sigma)$, $\sigma \in s$ and $\tau \in t$, which can be proved equivalent using the axioms of Examples 11 and 14.

By Lemma 26, we can assume that both $\sigma \in s$ and $\tau \in t$ have the structure described in point 3. Let $\delta(\sigma)$ and $\delta(\tau)$ be the dag's associated with σ and τ , respectively, by Definition 25. By hypothesis they belong to the same term graph, thus there exists an isomorphism $\eta:\delta(\sigma)\to\delta(\tau)$. By composing it with the functions ξ_{σ} and ξ_{τ} of Definition 25, we get an isomorphism $\chi\stackrel{def}{=} \xi_{\tau}^{-1} \circ \eta \circ \xi_{\sigma}: \mathcal{O}_{\Sigma}(\sigma) \to \mathcal{O}_{\Sigma}(\tau)$ relating occurrences of generators in σ and in τ .

By the choice of σ and τ , since by hypothesis they are mapped to the same term graph, we know that they have the same depth, the same number of generators in the corresponding levels, and that this number is equal to the number of nonempty nodes of that level of the term graph. Furthermore, by inspection one may verify that σ and τ have exactly the same set of occurrences of generators. As a consequence the isomorphism χ can be split into a family of isomorphisms $\{\chi_i\}_{i\leq l}$ where l is the depth, and χ_i relates generators of level i in σ and τ . Now it is not difficult to transform such isomorphisms into identities, by transforming for example τ into an equivalent term τ' having exactly the same structure of τ . The idea is that two adjacent generators (at once with the corresponding multiplicators) having the same level i can be exchanged by modifying only the symmetries of levels i-1 and i.

At this point we are left with terms σ and τ' having the same set of occurrences of operators and the same generators in the corresponding occurrences. By inspecting the structure and the relationship with the associated isomorphic dag's, it is easy to see that also all multiplicators in the corresponding occurrences have to be identical, because each of them is determined by the degree of sharing of a specific node of the term graph (an empty node for the multiplicators of level 0, a nonempty one for the others). So the only thing that can differ in the terms σ and τ' are the symmetries of the various levels. Let us show, by induction on

the depth, how τ' can be transformed into σ via suitable manipulations that preserve the equivalence: This concludes the proof.

If the depth of σ and τ' is 0, then we are done by Proposition 24 because they are discrete. Now assume that their depth is l+1. Let Π_{σ} and $\Pi_{\tau'}$ be the permutations corresponding to the symmetries of level l+1 of σ and τ' , respectively. If they are equal, then σ and τ' are identical at level l+1, and we are done because we can make identical their sub-terms of depth l by induction hypothesis. Now suppose that x is the least argument on which Π_{σ}^{-1} and $\Pi_{\tau'}^{-1}$ differ. x represents a root of the corresponding term graph. Now either this root is a node of level l+1 or it is of a lower level. In the first case, we could show that by composing τ' with a symmetry that exchanges $\Pi_{\sigma}^{-1}(x)$ and $\Pi_{\tau'}^{-1}(x)$, we obtain an arrow τ'' equivalent to τ' such that Π_{σ}^{-1} and $\Pi_{\tau''}^{-1}$ agree on all $y \leq x$ (because both $\Pi_{\sigma}^{-1}(x)$ and $\Pi_{\tau'}^{-1}(x)$ must be in the target of the same multiplicator of level l+1, and by applying the last equation of Corollary 16). In the second case, if x is a root corresponding to a node in a level lower than l+1, then both $\Pi_{\sigma}^{-1}(x)$ and $\Pi_{\tau'}^{-1}(x)$ must be the target of identities of level l+1. In this case we can exchange them by inserting a symmetry at level l+1, and the inverse symmetry at level l. The identities of level l+1 are clearly not affected, and we have obtained an arrow satisfying the same properties as τ'' above.

By iterating this procedure we finally get a term having the same symmetry of level l+1 as σ , and the statement follows by induction hypothesis.

5. On the Relation between Term Graphs and Terms

This section is devoted to the analysis of the relationship between the algebraic theory of a signature, a category whose arrows represent (tuples of) finite terms and whose definition dates back to Lawvere's PhD thesis [43], and our definition of gs-monoidal theory. We show that they are related by a functor, which can be interpreted as a categorical formulation of the unraveling function defined in many papers on term graph rewriting to extract a term from a given term graph.

It is difficult to give a full account of the relevance of Lawvere's thesis for both the fields of mathematics and computer science. As Kock and Reyes summarised, the right way of conceiving the totality of operations for an equational theory was found by Lawvere, who realized that substitution should be viewed as the composition of arrows on a certain kind of category [40].

DEFINITION 27. (Algebraic theories). An algebraic theory \mathbf{C} is a category whose objects are underlined natural numbers, and which for

each \underline{n} is equipped with an n-tuple of maps $\{\pi_i^n : \underline{n} \to \underline{1} \mid i = 1 \dots n\}$, making \underline{n} the n-fold Cartesian product of $\underline{1}$, that is, $\underline{n} = \underline{1}^n$.

Algebraic theories were originally introduced as a suitable generalisation of universal algebras, in order to provide a 'presentation independent' description of equationally definable classes of algebras, and of their relationships.

DEFINITION 28. (Algebraic theory over a signature). Given a signature Σ , the algebraic theory over Σ is defined as the category $\mathbf{Th}(\Sigma)$ such that the objects are natural numbers, and the hom-set $\mathbf{Th}(\Sigma)[\underline{n},\underline{m}]$ is the set of m-tuples of terms over variables $\{x_1,\ldots,x_n\}$, with term substitution as arrow composition.

By definition, an arrow in $\mathbf{Th}(\Sigma)[\underline{n},\underline{m}]$ identifies m terms t_1,\ldots,t_m , built over variables among (the somewhat canonical tuple) x_1,\ldots,x_n . In particular, note that for every m, the hom-set $\mathbf{Th}(\Sigma)[\underline{m},\underline{0}]$ contains only one arrow, because there is only one empty tuple of terms (over x_1,\ldots,x_m): This arrow is denoted by $!_{\underline{m}}$. The composition of $!_{\underline{m}}$ with $t:\underline{n}\to\underline{m}$ yields $!_{\underline{n}}$, because intuitively it discards all the terms of the m-tuple t. Therefore $\underline{0}$ is a terminal object in $\mathbf{Th}(\Sigma)$.

PROPOSITION 29. (Algebraic theories and term algebras). For any signature Σ , the category $\mathbf{Th}(\Sigma)$ is an algebraic theory: For each $n \in \mathbb{N}$, the requested n-tuple of maps $\pi_1^n \dots \pi_n^n$ is given by the tuple of variables x_1, \dots, x_n .

The previous result shows the relevant property (as far as our paper is concerned) of $\mathbf{Th}(\Sigma)$: There is no need to force any identification between terms, in order to get an algebraic theory. In other words (which will become clearer after introducing alternative descriptions for such theories), the extra structure induced by the categorical construction does not 'add' information to the original signature.

EXAMPLE 30. (The algebraic theory for a first-order signature). Let us consider the signature $\Sigma_e = \Sigma_0 \cup \Sigma_1$, where $\Sigma_0 = \{a\}$ and $\Sigma_1 = \{f\}$. By definition, the hom-set $\mathbf{Th}(\Sigma_e)[\underline{2},\underline{1}]$ contains all the elements of the term algebra $T_{\Sigma_e}(x_1,x_2)$, e.g., the variables x_1 and x_2 (regarded as the projections π_1^2, π_2^2), the terms $a, f(x_1, x_2), f(a, x_1),$ and so on. Composition is term substitution. As an example, consider the pair $\langle a, x_1 \rangle$ (an element of $\mathbf{Th}(\Sigma_e)[\underline{2},\underline{2}]$); composing it with arrow $f(x_1,x_2)$ (that is, applying the substitution $\{x_1/a, x_2/x_1\}$ to the term $f(x_1, x_2)$), we obtain $f(a, x_1)$; composing it with the projection π_1^2 (that is, applying the substitution $\{x_1/a, x_2/x_1\}$ to the term x_1), we obtain a; and so on.

Such a concrete description of an algebraic theory is well-suited for model-theoretical purposes (which were the main focus of Lawvere's thesis), but it lacks the usual *generative* aspects of the well-known notion of term algebra. Note that, for any algebraic theory \mathbf{C} , a relevant role is played by the arrows of the hom-set $\mathbf{C}[\underline{n},\underline{1}]$, since each other arrow $t:\underline{n}\to\underline{m}$ can be obtained as a suitable m-tuple of arrows $t_i:\underline{n}\to\underline{1}$, and this property suggests the following Axiomatic Presentation \mathbf{AP} : See for example [23].

PROPOSITION 31. (Equational characterisation of algebraic theories). For a given signature Σ , let us consider the category $\mathbf{AP}(\Sigma)$ having as objects the underlined natural numbers, containing the basic arrow $f_{\Sigma}: \underline{n} \to \underline{1}$ for each $f \in \Sigma_n$, and which for each $n, m \in \mathbb{N}$ is equipped with an n-tuple of maps $\{\pi_i^n: \underline{n} \to \underline{1} \mid i = 1...n\}$ and a tupling function $\langle \ldots \rangle : \mathbf{AP}(\Sigma)[\underline{n},\underline{1}]^m \to \mathbf{AP}(\Sigma)[\underline{n},\underline{m}]$, satisfying for all m-tuples $t_1 \ldots t_m \in \mathbf{AP}(\Sigma)[\underline{n},\underline{1}]$ and for all i = 1...m

$$\langle t_1, \dots, t_m \rangle; \pi_i^m = t_i, \qquad \langle t; \pi_1^m, \dots, t; \pi_m^m \rangle = t.$$

Then $\mathbf{AP}(\Sigma)$ is isomorphic to $\mathbf{Th}(\Sigma)$.

The above axiomatisation is largely responsible for the well-known, $na\ddot{i}ve$ presentation of the algebraic theory over Σ as the free Cartesian category with finite, strictly associative products, generated from the signature Σ .

During the Seventies, algebraic theories received a lot of attention from the computer science community, due in large part to the seminal work of the New York–Stanford based ADJ group on the *algebraic semantics* for programming languages (see, e.g., [23, 24]).

EXAMPLE 32. (Explicit construction of an algebraic theory). Let us consider again the signature Σ_e . The basic arrows are $a_{\Sigma_e}: \underline{0} \to \underline{1}$ and $f_{\Sigma_e}: \underline{2} \to \underline{1}$. Now consider for example the hom-set $\mathbf{AP}(\Sigma_e)[\underline{2},\underline{1}]$. By Proposition 31 it contains arrows f_{Σ_e}, π_1^2 and π_2^2 , and it must be isomorphic to $T_{\Sigma_e}(x_1, x_2)$. Therefore all the terms in $T_{\Sigma_e}(x_1, x_2) \setminus \{f_{\Sigma_e}, x_1, x_2\}$ have to be obtained as suitable compositions. For example, we have $a = \underline{1}_{\underline{2}}$; a_{Σ_e} (where $\underline{1}_{\underline{2}}$ is the empty tuple); and $f(a, x_1) = \langle \underline{1}_{\underline{2}}; a_{\Sigma_e}, \pi_1^2 \rangle$; f_{Σ_e} .

Note that in the concrete description of $\mathbf{Th}(\Sigma_e)$ in Example 30 all the terms were overloaded (as an example, term a is an element for each hom-set of the form $\mathbf{Th}(\Sigma_e)[\underline{n},\underline{1}]$). Instead, $\mathbf{AP}(\Sigma)$ provides a syntax that makes explicit the source object: For example, the same term a, when considered as an element of T_{Σ_e} , is represented by the arrow a_{Σ_e} ; while if considered as an element of $T_{\Sigma_e}(x_1,x_2)$, it is represented by the arrow a_{Σ_e} :

Yet another presentation for algebraic categories, having the advantage of separating in a better way the Σ -structure of the basic operators from the additional algebraic structure enjoyed by the meta-operators induced by the structure of the category at hand, can be traced back to the early Seventies work of former East Germany algebraists on dht-categories (see Section 6.6 of this paper). It turns out that gs-monoidal categories are indeed Cartesian if the naturality of the two auxiliary transformations in enforced, thus providing an easy, equational characterisation of the notion of finite products. This is shown by the following result, which has become folklore in recent years (see, e.g., [33, 41, 56]).

PROPOSITION 33. (Algebraic as gs-monoidal plus naturality). For a given signature Σ , the category $\mathbf{C}\text{-}\mathbf{Th}(\Sigma)$ obtained by quotienting the associated gs-monoidal theory $\mathbf{GS}\text{-}\mathbf{Th}(\Sigma)$ with the axioms: for all $n, m \in \mathbb{N}$, for all $t: n \to m$

$$t; \nabla_{\underline{m}} = \nabla_{\underline{n}}; (t \otimes t), \qquad t; !_{\underline{m}} = !_{\underline{n}}$$

is isomorphic to the algebraic theory $\mathbf{Th}(\Sigma)$.

Let us denote as **C-Cat** the category of Cartesian categories with strictly associative products, that is, of those gs-monoidal categories verifying the additional naturality axioms, and gs-monoidal functors. Then there is an obvious forgetful functor $U: \mathbf{C-Cat} \hookrightarrow \mathbf{GSM-Cat}$ having a left adjoint $C: \mathbf{GSM-Cat} \to \mathbf{C-Cat}$. Thanks to the characterisation of term graphs as arrows of gs-monoidal theories, the component of the unit of the above adjunction on a gs-monoidal theory, i.e., $\eta_{\mathbf{GS-Th}(\Sigma)}: \mathbf{GS-Th}(\Sigma) \to U(C(\mathbf{GS-Th}(\Sigma))) = \mathbf{Th}(\Sigma)$, maps each term graph of rank (m,n) to an m-tuple of terms with variables in x_1, \ldots, x_n . Such a mapping can be regarded as a clean categorical version of the unraveling function defined in many papers on term graph rewriting to extract a term from a term graph. Also, it provides a clear interpretation of the relationship between terms and term graphs: a term is an equivalence class of term graphs, modulo the naturality of ∇ and !.

EXAMPLE 34. (On gs-monoidal and algebraic theories). Let us consider for the last time the signature Σ_e . Now, the associated gs-monoidal theory contains the three different arrows (from $\underline{0}$ to $\underline{1}$) $t_1 = (a \otimes a)$; f, $t_2 = a$; $\nabla_{\underline{1}}$; f and $t_3 = (a \otimes a)$; $(\nabla_{\underline{1}} \otimes !_{\underline{1}})$; f, corresponding to three different term graphs (as shown in Figure 6). Those elements belong instead to the same equivalence class in $\mathbf{C}\text{-}\mathbf{Th}(\Sigma)$ (since $t_1 = t_2$ and $t_2 = t_3$ due to the naturality of, respectively, ∇ and !), that characterises the term f(a, a).

$$\emptyset \quad a \xrightarrow{1} f \cdots 1 \qquad \qquad \emptyset \quad a \xrightarrow{1} f \cdots 1 \qquad \qquad \emptyset \quad a \xrightarrow{1} f \cdots 1 \qquad \qquad a \qquad \qquad$$

Figure 6. Three term graphs denoting the same term.

We conclude this section with a final remark on the terminology we introduced so far. Yet another description for $\mathbf{Th}(\Sigma)$ could be given in terms of substitutions: the elements of each hom-set $\mathbf{Th}(\Sigma)[\underline{n},\underline{m}]$ are the term substitutions, i.e., the functions $\{x_1,\ldots,x_m\}\to T_\Sigma(x_1,\ldots,x_n)$. This was the property motivating the terminology for s-monoidal categories (where s stands for substitution, see [51]), as categories containing (at least one of) the natural transformations $\{\rho,\nabla,!\}$. In fact, each of those transformations is needed to characterise a different class of term substitutions, i.e., those bijective, surjective and injective, respectively. In our categories, having ruled out naturality for ∇ and !, we can regard the elements of a hom-set $\mathbf{GS-Th}(\Sigma)[\underline{n},\underline{m}]$ as \mathbf{Graph} Substitutions, i.e., homomorphisms from the discrete graph $\{x_1,\ldots,x_m\}$ to a term graph with exactly \underline{n} empty nodes. Whence the prefix 'gs-' which qualifies many of the notions we introduced.

6. Extensions and Related Literature

This section is devoted to a discussion about various possible extensions of the main characterisation result presented in the paper, as well as about the related literature. We consider in turn the extension to term graphs over many-sorted and polyadic signatures, to cyclic term graphs, and finally to term graph rewriting. Next in Sections 6.5 and 6.6 we briefly overview the related literature, considering separately the approaches based on a logical presentation of term graphs (or similar structures), and those based more directly on categorical structures.

6.1. Many-sorted signatures

A many-sorted signature is a pair $\langle S, \Sigma \rangle$, where S is a set of sorts, and Σ is a family of sets of operators indexed by $S^* \times S$. If $w = s_1 \dots s_n$, then an operator $f \in \Sigma_{w,s}$ is intended to take n arguments of sorts s_1, \ldots, s_n , and to deliver a result of sort s.

As far as the relationship between terms over a signature and arrows of the corresponding algebraic theory is concerned, this relationship generalises smoothly to the many-sorted case [6]. The objects of the associated algebraic theory are the elements of the free monoid generated by the set of sorts S (i.e., finite sequences of elements of S), and the various presentations of the algebraic theory of the many-sorted signature $\langle S, \Sigma \rangle$ will be adjusted to consider generators of the form $f: w \to s$, for $f \in \Sigma_{w,s}$.

The definitions and results of this paper can be generalised to many-sorted signatures exactly in the same straightforward way, obtaining, as expected, a characterisation result of term graphs over a many-sorted signature as arrows of the corresponding gs-monoidal theory. In particular, Lemmas 12 and 15 will be replaced by more general statements. For example, the statement of Lemma 15 will become: For any $v, w \in S^*$ with length(v) = n and length(w) = m, the set of arrows from v to w in GSM(S) is isomorphic to the set of all functions g from $\{1, \ldots, m\}$ to $\{1, \ldots, n\}$ such that $w|_i = v|_{g(i)}$, where $w|_i$ denotes the i-th element of sequence w.

6.2. Polyadic signatures

A polyadic (one-sorted) signature Σ is a family of sets of operators indexed by $\mathbb{N} \times \mathbb{N}$. A polyadic operator $f \in \Sigma_{n,m}$ is intended to take n arguments, and to return (an ordered list of) m results. The algebraic theory of a polyadic signature Σ can be defined as in Proposition 31 for the standard, monadic case, i.e., as a free Cartesian category over a set of generators that in this case may have a more general shape, for example $f: \underline{n} \to \underline{m}$. Now it is easily shown that for each polyadic signature Σ there is a (monadic) signature Σ' such that the algebraic theories $\mathbf{Th}(\Sigma)$ and $\mathbf{Th}(\Sigma')$ are isomorphic. Simply, for each polyadic operator $f \in \Sigma_{n,m}$, add operators $\{f_1, \ldots, f_m\}$ to Σ'_n . Then the isomorphism will relate the tuple $\langle f_1, \ldots, f_m \rangle$ to f, and each f_i to f; π_i .

This shows that polyadic operators do not add expressive power, at least as far as algebraic theories are concerned, and this is due to the properties of projections, thus to Cartesianity. In fact, quite interestingly, in the case of gs-monoidal theories such a property does not hold, i.e., there are polyadic signatures such that their gs-monoidal theory is not isomorphic to any gs-monoidal theory of a monadic signature. And even more interestingly, our main result, Theorem 23, can be generalised to the case where Σ is a polyadic signature.

Unfortunately, however, the definition of term graph used in this paper does not work for a polyadic signature, because the fact that an operator returns a single result is, in a sense, wired in the representation of term graphs, where the operators are used as node labels. Therefore

for the mentioned generalisation of our main result we need to pass to a different presentation of term graphs, the so-called *jungles* [32].

Without giving the formal details, a jungle over a (monadic) signature Σ is a hyper-graph where the hyper-edges are labeled by operators of Σ , and each hyper-edge labeled by $f \in \Sigma_n$ has one source node and exactly n target nodes. The category of jungles over Σ is equivalent to the category \mathbf{Dag}_{Σ} of dag's over Σ [15], and this is sufficient to ensure that all definitions and results of this paper can be recasted by replacing dag's with jungles. Now also the case of polyadic signatures can be dealt with uniformly, because it is sufficient to adjust the definition of jungle by requiring that if an hyper-edge is labeled by $f \in \Sigma_{n,m}$ then it must have m sources and n targets. At this point most definitions, statements and proofs of the paper should be adjusted accordingly in a straightforward manner, and a characterisation result for term graphs over a polyadic signature would be obtained. The extension to polyadic signatures has been used for example in [20].

6.3. Cyclic term graphs

In the present paper we only considered acyclic term graphs. By allowing term graphs with cycles, one can represent in a finitary way certain structures that arise dealing with recursive definitions (as for the implementation of the fixed point combinator Y proposed in [60]). Therefore it is certainly interesting to consider the extension of our results to the possibly cyclic case. Preliminary results [12] show that this goal can be achieved by enriching gs-monoidal categories with a feedback structure (already studied in the algebraic specification community: See [2] for earlier references, and especially the work of Stefanescu [19, 18]) using the recently introduced notion of trace [35], thus yielding traced gsmonoidal categories. In this case, the correspondence with terms has to be reconsidered, because in general a cyclic term graph unravels to a rational term, i.e., a possibly infinite term with a finite number of distinct sub-terms. The relationship between rational terms and cyclic term graphs is addressed in [9, 38], while in [11] rational terms are related to μ -terms (which are essentially a subclass of term graphs): None of these papers, however, phrases this relationship categorically as a functor between suitable theories, as we did for algebraic and gsmonoidal theories in Section 5. This is certainly an interesting topic for future research.

6.4. Term graph rewriting

Our characterisation of term graphs allows to put on the same ground and compare the techniques developed respectively for term rewriting

and for term graph rewriting, obtaining a description of the latter in an intuitive equational way. More to the point, the categorical representation of terms over a signature Σ as arrows of the algebraic theory of Σ can be lifted to a categorical representation of term rewriting, using 2-categories. In fact, a term rewrite rule $R = l \rightarrow r$ (where l and r are two terms over Σ) can be represented as a cell, i.e., a vertical arrow from the arrow representing l to the arrow representing r. This situation can be denoted by $R: l \Rightarrow r: \underline{n} \rightarrow \underline{1}$, which also states that l and r are arrows from n to 1, i.e., that they have at most nvariables. Given a term rewriting system, the structure obtained in this way, i.e., the algebraic theory of Σ enriched with one cell for each rule, is called a *computad*. The interesting fact is that from such a computad, a free construction can generate a (Cartesian) 2-category by adding all identity cells, and closing cells with respect to horizontal and vertical composition. The resulting 2-category faithfully represents all the possible rewriting sequences of the original system. In fact, horizontal composition of cells generates all the possible instantiations of the rules and, at the same time, places rules in all possible contexts. Vertical composition acts instead as sequential composition. Furthermore, the generated rewriting sequences are subject to an equivalence that coincides with the so-called permutation equivalence [7], due to the axioms of 2-categories [21, 42].

A similar construction can be followed for term graph rewriting as well, as it is shown in [10]. The idea is to add to the gs-monoidal category generated by Σ cells representing the rules of a term graph rewriting system, and to consider the (gs-monoidal) 2-category freely generated by such cells. The main result is that the cells in the resulting 2-category represent term graph rewriting sequences of the original system satisfying a mild restriction on redexes. In [12] the overall approach is further extended by relating cyclic term graph rewriting and traced gs-monoidal 2-categories.

An interesting topic that remains to be explored is the analysis of the equivalence on rewriting sequences induced by the axioms of (traced, gs-monoidal) 2-categories. Not a secondary point, since permutation equivalence has proved too coarse (due to the 'implicit execution' of duplication and garbage collection) to be considered as a faithful description of the actual behaviour of a concurrent machine implementing the rewriting process over a distributed structure [14, 21].

6.5. A LOGICAL VIEW OF TERM GRAPHS

In the introduction we mentioned three equivalent presentations of term rewriting, namely the *operational*, based on redexes and substitutions,

on which most of the theory is based; the *categorical*, discussed to some extent in the previous subsection; and the *logical* one, for which we refer to [47, 48], but also [29, 55].

It is worth stressing that the logical and the categorical views are very strongly related, the former providing often a more syntactical presentation of the latter. Various papers in the literature discuss term graph rewriting using a logical presentation in terms of inference rules and axioms, including [26, 27, 49].

As far as the topic of the present paper is concerned, we just want to recall the inference rule that defines term graphs as an extension of standard first order terms. Roughly, the usual syntax is enriched with an explicit 'let' operator, whose introduction rule is given by:

$$\frac{\Gamma \vdash t : \tau \quad \Gamma, x : \tau \vdash s : \sigma}{\Gamma \vdash (\mathbf{let} \ x \ \mathbf{be} \ t \ \mathbf{in} \ s) : \sigma}$$

The intuitive meaning is that given a context Γ (i.e., a suitable tuple of typed variables), the structure of the resulting term is the same as that of s, after replacing each occurrence of x by a pointer to t. Obviously, whether s can be considered as a true term graph or just a term depends on the axioms imposed. For example, if the axiom let x be t in s = s[x/t] is required to hold (where s[x/t] is obtained by substituting each occurrence of x in s with t), then the let operator becomes (almost) syntactic sugar for terms. Therefore, when let is used for the syntax of term graphs, the above axiom cannot hold in general.

Similarly, the 'discard' operator can provide a syntax for explicit garbage. The introduction rule is:

$$\frac{\Gamma \vdash t : \tau}{\Gamma \vdash discard(t) : \bullet}$$

where • is the unit type. Clearly, implicit garbage collection is obtained by imposing an axiom that equates any two terms of unit type.

6.6. HISTORICAL REMARKS ON CATEGORICAL MODELS

The observation that Cartesian categories are monoidal categories equipped with additional natural transformations is quite old, and indeed many authors studied categorical structures that are weaker than the former and richer than the latter. The motivations, however, cannot be reduced easily to a single topic or intuition.

As far as our work is concerned, let us summarise the motivations that led us to consider a categorical presentation of term graphs. When introducing gs-monoidal categories (see [21, 22]: We called them s-monoidal, since we were not yet aware of Pfender's work cited below),

our intention was to enrich symmetric monoidal categories in order to model a notion of local sharing: Our starting points were the already mentioned [33, 41]. Next, in [14], when analysing two categorical models for term rewriting, we observed that neither the standard 2-categorical presentation of [46, 48] (see Section 6.4), nor the formulation based on the weaker notion of sesqui-category by Stell [57, 59] were completely satisfactory from the perspective of a truly concurrent semantics. In fact, from that point of view, the equivalence on rewriting sequences (i.e., cells) holding in 2-categories (which coincides with the permutation equivalence) is too coarse, while that of sesqui-categories is too fine (see [14] for a detailed comparison). Actually, we realized that standard terms were non suitable for describing the states of a distributed system, because the usual notion of reduction can have nonlocal effects (think for example of a rule having an occurrence of a variable x in the left-hand side, and none in the right-hand side: Its application can delete a sub-term of unbounded size). This motivated our study of structures related to terms, but more concrete, i.e., term graphs.

After the characterisation of term graphs using gs-monoidal theories presented here and the 2-categorical presentation of their rewriting proposed in [10], we intend to study to which extent the resulting equivalence on rewriting sequences is satisfactory for a concurrent semantics.

Other authors have considered categories similar to ours with different motivations. The most related to ours, in our opinion, is the study of the algebra of flow graphs, a topic that has surfaced many times in the literature on algebraic specifications. A comprehensive contribution to this topic is given by the works on flownomial algebras by \$\text{Stefănescu}\$ (see [19, 17], and [18] for a survey). As flow graphs, he considers structures that are richer than ours, namely, using our terminology, cyclic hyper-graphs (see Sections 6.2 and 6.3); on the categorical side, he relates such graphs to symmetric monoidal categories enriched with (non-natural) transformations that correspond to our ∇ and !, as well as their duals, and with a traced structure in the spirit of [35], and he proposes a corresponding axiomatisation. To some extent his results subsume ours, but his treatment lacks an explicit, independent definition of the graphical structures he characterises.

A more semantical point of view is taken by other authors, which considered various kind of monoidal categories arising from the study of certain classes of algebras and from the analysis of the laws they satisfy. Among them, Pfender [51] introduced s-monoidal categories for characterising the algebraic structure of classes of substitutions. As noticed at the end of Section 5, such categories contain at least one of the natural transformations in $\{\rho, \nabla, !\}$, which are needed to characterise

bijective, surjective and injective term substitutions, respectively. Since the transformations are always assumed to be natural, only (variations of) standard terms are obtained.

The analysis of the categorical structure of classes of algebras of partial maps is the main topic of the work by Hoenke and other former East Germany algebraists [8, 30, 31]. They introduce dht-symmetric monoidal categories, which can be regarded as gs-monoidal ones with a zero-object, and where transformation ∇ is natural while! in general is not. A functorial presentation of partial algebras is obtained by considering suitable functors from a theory having that dht-symmetric monoidal structure to the category Par of sets and partial functions. In [13] we prove that, similarly, multi-algebras 10 over a signature Σ can be characterised as gs-monoidal functors from the gs-monoidal theory of Σ to Rel, the category of sets and relations. This result is based on the observation (already made by Hoenke) that, informally, a relation f is a partial function if and only if ∇ is natural with respect to it (i.e., f; $\nabla = \nabla$; $f \otimes f$).

Robinson and Rosolini have similar goals in [56], where they survey the approaches that try to recover which is the extra structure we need on a category for it to be a 'well-behaved' category of partial maps. To this aim, they introduce p-categories, which were very close to ours if we required ∇ to be natural. Also related to ours are the copy-categories introduced in [28].

Walters et alii investigate the algebraic structure of relations, regarded as arrows of suitable categories of spans [36, 37]. They have on the operational side an interpretation in terms of processes, and on the categorical side suitable bicategories. Interestingly, the axiomatisation of relations they propose makes use of ∇ , !, and their duals: Since their axioms are slightly different from those by Ştefănescu, in their framework a traced structure is not explicitly needed but it is derivable.

Finally, also related to our work are premonoidal categories, introduced by Power and Robinson [54]. These categories are extensively used as a semantical model in the community of action calculi, since they provide a categorical characterisation of some sort of data flow structures (see Hasegawa [26] and Jeffrey [34]): we refer to Pavlovic [50] and to Barber et alii [3] for an overview. A combination of their approach and ours, with a specific application to term rewriting, is due to Miyoshi [49].

 $[\]overline{}^{10}$ A multi-algebra \mathcal{A} over a one-sorted signature Σ has a carrier A and, for each operator $f \in \Sigma_n$, a function $f_{\mathcal{A}} : A^n \to \mathcal{P}(A)$, where $\mathcal{P}(A)$ is the power-set of A, which can be regarded as a relation $f_{\mathcal{A}} \subseteq A^{n+1}$ [61]. Clearly, partial algebras are multi-algebras where $f_{\mathcal{A}}(\overline{a})$ is either a singleton or the empty set for each $\overline{a} \in A^n$.

References

- Z.M. Ariola and J.W. Klop. Equational term graph rewriting. Fundamenta Informaticae, 26:207-240, 1996.
- E. S. Bainbridge. Feedback and generalized logic. Information and Control, 31:75-96, 1976.
- 3. A. Barber, Ph. Gardner, M. Hasegawa, and G. Plotkin. From action calculi to linear logic. In M. Nielsen and W. Thomas, editors, *Computer Science Logic*, volume 1414 of *LNCS*, pages 78–97. Springer Verlag, 1998.
- H.P. Barendregt, M.C.J.D. van Eekelen, J.R.W. Glauert, J.R. Kennaway, M.J. Plasmeijer, and M.R. Sleep. Term graph reduction. In J.W. de Bakker, A.J. Nijman, and P.C. Treleaven, editors, *Parallel Architectures and Languages Europe*, volume 259 of *LNCS*, pages 141–158. Springer Verlag, 1987.
- 5. M. Bauderon and B. Courcelle. Graph expressions and graph rewritings. *Mathematical Systems Theory*, 20:83–127, 1987.
- S.L. Bloom and E.G. Wagner. Many-sorted theories and their algebras with some applications to data types. In M. Nivat and J.C. Reynolds, editors, Algebraic methods in semantics, pages 133–168. Cambridge University Press, 1985.
- G. Boudol. Computational semantics of term rewriting systems. In M. Nivat and J. Reynolds, editors, Algebraic Methods in Semantics, pages 170–235. Cambridge University Press, 1985.
- L. Budach and H.-J. Hoenke. Automaten und Functoren. Akademie-Verlag, 1975
- A. Corradini and F. Drewes. (Cyclic) term graph rewriting is adequate for rational parallel term rewriting. Technical Report TR-97-14, Dipartimento di Informatica, Pisa, 1997.
- 10. A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In E. Moggi and G. Rosolini, editors, *Category Theory and Computer Science*, volume 1290 of *LNCS*, pages 87–105. Springer Verlag, 1997.
- A. Corradini and F. Gadducci. Rational term rewriting. In M. Nivat, editor, Foundations of Software Science and Computation Structures, volume 1378 of LNCS, pages 156-171. Springer Verlag, 1998.
- 12. A. Corradini and F. Gadducci. Rewriting on cyclic structures. Technical Report TR-98-05, Dipartimento di Informatica, Pisa, 1998. To appear in TIA.
- A. Corradini and F. Gadducci. Functorial semantics for multi-algebras. In J. L. Fiadeiro, editor, Recent Trends in Algebraic Development Techniques, volume 1589 of LNCS, pages 78-90. Springer Verlag, 1999.
- 14. A. Corradini, F. Gadducci, and U. Montanari. Relating two categorical models of term rewriting. In J. Hsiang, editor, *Rewriting Techniques and Applications*, volume 914 of *LNCS*, pages 225–240. Springer Verlag, 1995.
- 15. A. Corradini, U. Montanari, F. Rossi, H. Ehrig, and M. Löwe. Logic programming and graph grammars. In H. Ehrig, H.-J. Kreowski, and G. Rozenberg, editors, *Graph-Grammars and Their Application to Computer Science*, volume 532 of *LNCS*, pages 221–237. Springer Verlag, 1991.
- A. Corradini and F. Rossi. Hyperedge replacement jungle rewriting for term rewriting systems and logic programming. Theoret. Comput. Sci., 109:7–48, 1993.
- 17. Gh. Ştefănescu. On flowchart theories: Part II. The nondeterministic case. Theoret. Comput. Sci., 52:307–340, 1987.

- Gh. Ştefănescu. Algebra of flownomials. Technical Report SFB-Bericht 342/16/94 A, Technical University of München, Institut für Informatik, 1994.
- 19. V.-E. Căzănescu and Gh. Ştefănescu. Towards a new algebraic foundation of flowchart scheme theory. Fundamenta Informaticae, 13:171–210, 1990.
- 20. G. Ferrari and U. Montanari. Tiles for concurrent and located calculi. In C. Palamidessi and J. Parrow, editors, *Expressiveness in Concurrency*, volume 7 of *Electronic Notes in Computer Science*. Elsevier, 1997.
- 21. F. Gadducci. On the Algebraic Approach to Concurrent Term Rewriting. PhD thesis, University of Pisa, Department of Computer Science, 1996.
- 22. F. Gadducci and U. Montanari. The tile model. In G. Plotkin, C. Stirling, and M. Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1999. To appear.
- J.A. Goguen, J.W. Tatcher, E.G. Wagner, and J.R Wright. Some fundamentals
 of order-algebraic semantics. In A. Mazurkiewicz, editor, *Mathematical Foundations of Computer Science*, volume 45 of *LNCS*, pages 153–168. Springer Verlag, 1976.
- 24. J.A. Goguen, J.W. Tatcher, E.G. Wagner, and J.R Wright. Initial algebra semantics and continuous algebras. *Journal of the ACM*, 24:68–95, 1977.
- A. Habel. Hyperedge replacement: Grammars and languages, volume 643 of LNCS. Springer Verlag, 1992.
- M. Hasegawa. Models of Sharing Graphs. PhD thesis, University of Edinburgh, Department of Computer Science, 1997.
- M. Hasegawa. Recursion from cyclic sharing: Traced monoidal categories and models of cyclic lambda-calculus. In Ph. de Groote and R. Hindly, editors, Typed Lambda Calculi and Applications, volume 1210 of LNCS, pages 196–213. Springer Verlag, 1997.
- 28. U. Hensel and D. Spooner. A view on implementing processes: Categories of circuits. In M. Haveraaen, O. Owe, and O. Dahl, editors, *Recent Trends in Data Types Specification*, volume 1130 of *LNCS*, pages 237–255. Springer Verlag, 1995.
- 29. B. P. Hilken. Towards a proof theory of rewriting: The simply typed $2-\lambda$ -calculus. Theoret. Comput. Sci., 170:407-444, 1996.
- 30. H.-J. Hoenke. On partial algebras. In B. Csákány, E. Fried, and E.T. Schmidt, editors, *Universal Algebra*, volume 29 of *Colloquia Mathematica Societatis János Bolyai*, pages 373–412. North-Holland, 1977.
- H.-J. Hoenke. On partial recursive definitions and programs. In M. Karpiński, editor, Fundamentals of Computation Theory, volume 56 of LNCS, pages 260– 274. Springer Verlag, 1977.
- 32. B. Hoffmann and D. Plump. Implementing term rewriting by jungle evaluation. Informatique Théorique et Applications/Theoretical Informatics and Applications, 25:445-472, 1991.
- B. Jacobs. Semantics of weakening and contraction. Annals of Pure and Applied Logic, 69:73-106, 1994.
- A. Jeffrey. Premonoidal categories and a graphical view of programs. Technical report, COGS, University of Sussex, 1997.
- 35. A. Joyal, R.H. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119:425–446, 1996.
- P. Katis, N. Sabadini, and R.F.C. Walters. Bicategories of processes. *Journal of Pure and Applied Algebra*, 115:141–178, 1997.
- P. Katis, N. Sabadini, and R.F.C. Walters. SPAN(Graph): A categorical algebra of transition systems. In M. Johnson, editor, Algebraic Methodology and

- Software Technology, volume 1349 of LNCS, pages 307–321. Springer Verlag, 1997.
- J.R. Kennaway, J.W. Klop, M.R. Sleep, and F.J. de Vries. On the adequacy
 of graph rewriting for simulating term rewriting. ACM Trans. Program. Lang.
 Syst., 16:493-523, 1994.
- 39. J.W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 1, pages 1–116. Oxford University Press, 1992.
- A. Kock and G.E. Reyes. Doctrines in categorical logic. In J. Barwise, editor, Handbook of Mathematical Logic, pages 283–313. North Holland, 1977.
- 41. Y. Lafont. Equational reasoning with 2–dimensional diagrams. In H. Comon and J.-P. Jouannaud, editors, *Term Rewriting*, volume 909 of *LNCS*, pages 170–195. Springer Verlag, 1995.
- 42. C. Laneve and U. Montanari. Axiomatizing permutation equivalence in the λ -calculus. In H. Kirchner and G. Levi, editors, *Algebraic and Logic Programming*, volume 632 of *LNCS*, pages 350–363. Springer Verlag, 1992.
- 43. F.W. Lawvere. Functorial semantics of algebraic theories. *Proc. National Academy of Science*, 50:869-872, 1963.
- 44. S. Mac Lane. Categories for the working mathematician. Springer Verlag, 1971.
- S. MacLane. Natural associativity and commutativity. Rice University Studies, 49:28–46, 1963.
- N. Martí-Oliet and J. Meseguer. From Petri nets to linear logic through categories: A survey. Int. Journal of Foundations of Computer Science, 4:297–399, 1991.
- 47. N. Martí-Oliet and J. Meseguer. Rewriting logic as a logical and semantical framework. In J. Meseguer, editor, *Rewriting Logic and Applications*, volume 4 of *Electronic Notes in Computer Science*. Elsevier, 1997.
- 48. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. Theoret. Comput. Sci., 96:73-155, 1992.
- H. Miyoshi. Rewriting logic for cyclic sharing structures. In T. Sato and A. Middeldorp, editors, Fuji International Symposium on Functional and Logic Programming, pages 167–186. World Scientific, 1998.
- D. Pavlović. Categorical logic of names and abstraction in action calculi. *Mathematical Structures in Computer Science*, 7:619-637, 1997.
- 51. M. Pfender. Universal algebra in s-monoidal categories. Technical Report 95–22, University of Munich, Department of Mathematics, 1974.
- M.J. Plasmeijer and M.C.J.D. van Eekelen. Functional Programming and Parallel Graph Rewriting. Addison Wesley, 1993.
- 53. A.J. Power. An abstract formulation for rewrite systems. In D.H. Pitt, D.E. Rydehard, P. Dybjer, A.M. Pitts, and A. Poigné, editors, *Category Theory and Computer Science*, volume 389 of *LNCS*, pages 300–312. Springer Verlag, 1989.
- 54. J. Power and E. Robinson. Premonoidal categories and notions of computation.

 Mathematical Structures in Computer Science, 7:453-468, 1998.
- 55. A. Rensink. Bisimilarity of open terms. In C. Palamidessi and J. Parrow, editors, *Expressiveness in Concurrency*, volume 7 of *Electronic Notes in Computer Science*. Elsevier, 1997.
- E. Robinson and G. Rosolini. Categories of partial maps. Information and Computation, 79:95–130, 1988.
- 57. D.E. Rydehard and E.G. Stell. Foundations of equational deductions: A categorical treatment of equational proofs and unification algorithms. In D.H.

- Pitt, A. Poigné, and D.E. Rydehard, editors, Category Theory and Computer Science, volume 283 of LNCS, pages 114–139. Springer Verlag, 1987.
- 58. M.R. Sleep, M.J. Plasmeijer, and M.C. van Eekelen, editors. Term Graph Rewriting: Theory and Practice. Wiley, London, 1993.

 J.G. Stell. Categorical Aspects of Unification and Rewriting. PhD thesis,
- University of Manchester, 1992.
- 60. D. A. Turner. A new implementation technique for applicative languages. $Software:\ Practice\ and\ Experience,\ 9:31-49,\ 1979.$
- 61. M. Walicki and S. Meldal. Algebraic approaches to nondeterminism: An overview. ACM Computing Surveys, 29:30-81, 1997.