Università di Pisa

Dipartimento di Informatica

Technical Report : TR-00-04

Term Graph Syntax for Multi-Algebras

A. Corradini, F. Gadducci, W. Kahl

April 05, 2000

ADDR: Corso Italia 40,56125 Pisa,Italy. TEL: +39-50-887111. FAX: +39-50-887226

Term Graph Syntax for Multi-Algebras*

A. Corradini[1], F. Gadducci[2], W. Kahl[3]

[1] Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa, Italy, andrea@di.unipi.it

[2] Division of Informatics, University of Edinburgh, Mayfield Road, EH9 3JZ Edinburgh, Great Britain, fabio@dcs.ed.ac.uk

[3] Fakultät für Informatik, Universität der Bundeswehr München, 85577 Neubiberg, Germany, kahl@informatik.unibw-muenchen.de

Abstract Multi-algebras allow to model nondeterminism in an algebraic framework by interpreting operators as functions from individual arguments to sets of possible results. Starting from a functorial presentation of multi-algebras based on *gs-monoidal theories*, we argue that specifications for multi-algebras should be based on the notion of *term graphs* instead of on standard terms. We consider the simplest case of (term graph) equational specification, showing that it enjoys an unrestricted form of substitutivity. We discuss the expressive power of equational specification for multi-algebras, and we sketch possible extensions of the calculus.

1 Introduction

The extension of algebraic specification techniques for the treatment of nondeterminism is the topic of numerous contributions to the literature in the last two decades (see [WM97a] for an overview). It is recognised that nondeterminism shows up not only in the specification of intrinsically nondeterministic systems, like concurrent systems, but also when the system is deterministic and one wants to abstract away from implementation details or, in general, from information that would be necessary for a complete description of the system (like timing considerations, internals of the states, etc.).

Most algebraic approaches to the specification of nondeterminism make use of models that generalise standard (partial) algebras, namely *multi-algebras*, where an operator of the signature is interpreted as a function from individual arguments to sets of possible results. Less frequently, *power algebras* are used as well, where operators are interpreted as functions between powersets (thus arguments are sets, too), but we will not be concerned with power algebras here.

Many extensions of specification techniques to nondeterminism are *non-intrusive* or *conservative*, namely, they do not change the existing framework for deterministic specifications, and in particular they reduce to the standard theory when only deterministic operations are considered [WM97b]. Possibly, the success of multi-algebras in the algebraic specification community (with respect to power algebras, but also with respect to (heterogeneous) relations, which are essentially the same as multi-algebras) can be explained in this perspective by the fact that they are closer to (partial and total) algebras.

The goal of the present paper is to discuss a "syntactical issue" in the field of multi-algebraic specifications. We start from the observation that a persistent technical problem

* Research partly supported by the EC TMR Network GETGRATS and the Esprit WG APPLIGRAPH.

arises in the design of calculi for nondeterminism, namely the lack of full substitutivity, which is a fundamental, good property of deterministic calculi. Then we note that among the various solutions proposed to address this problem, many resort to some notion of "sharing" in terms, which allows to restrict nondeterminism in some form [AC79,Hes88,KB99]. Such a solution forces one to depart from the standard syntax of terms, which is universally accepted in algebraic specifications, and the new syntax making use of sharing is usually justified by pragmatical considerations only.

The main point that we make here is that, in specifications with nondeterminism, the use of terms with sharing (formalised later by the notion of *term graphs*) is justified also by more foundational, semantical motivations. Essentially, our analysis leads to the conclusion that term graphs play the same rôle for multi-algebras that standard terms play for total algebras, and this fact is justified in terms of a functorial semantics of multi-algebras proposed by two of the authors [CG99b,CG99c].

Next we elaborate on this theme, by considering the simplest specification language one may conceive using term graphs, namely term graph equational logic. We present a calculus for equational deduction which includes substitutivity and which is shown to be sound, and for the time being we only claim its completeness. The calculus is non-standard, as it includes checking for the existence of a term graph morphism in the premises of a rule. We discuss the expressive power of the calculus, recognizing its weakness, and we sketch possible extensions which fit smoothly in our framework. The first one consists in allowing for inequations in the specification, and we elaborate on the corresponding theory.

The paper is organised as follows: In the next section we introduce the basics of multi-algebras. In Sect. 3 we discuss in detail the lack of substitutivity in calculi for nondeterminism, reviewing some of the solutions proposed in the literature, as well as some closely related concepts. In Sect. 4 we introduce term graphs and some operations on them. Next in Sect. 5 we present the functorial semantics for multi-algebras of [CG99b] where the classifying category, the *gs-monoidal theory*, has term graphs as arrows, and we extend the theory with a natural ordering which we characterise in two different ways. Based on this result, and by analogy with the deterministic case, we argue in Sect. 6 that term graphs should play the rôle of terms, as they are suitable to denote derived relations of a multi-algebra. Section 7 introduces term graph equational deduction, proves its soundness and discusses its expressive power. Finally we hint at further developments including the extension of equational deduction with inequations.

2 Multi-algebras and relations

By interpreting the operators of a signature as relations instead of as functions, one easily models in an algebraic framework some sort of nondeterministic behaviour [WM97a]. For *multi-algebras*, the relation associated with an operator is regarded as a function mapping each input value to a *set* of (possible) output values. Quite obviously, this is equivalent to the standard definition of relations as subsets of the direct product.

Definition 1 (relations). *Let A and B be two sets. A* relation $R : A \leftrightarrow B$ *is a function $R : A \to \mathcal{P}(B)$, where $\mathcal{P}$ is the power-set operator. A relation is* total *if $|R(a)| \geq 1$ for all $a \in A$; it is* univalent (functional) *if $|R(a)| \leq 1$ for all $a \in A$. Given two relations $P : A \leftrightarrow B$, $R : B \leftrightarrow C$, their* composition $P; R : A \leftrightarrow C$ *denotes the relation obtained by composing P with $R^+ : \mathcal{P}(B) \to \mathcal{P}(C)$, the point-wise extension of R.*

Therefore a univalent relation is a partial function $R : A \rightharpoonup B$, while a univalent and total relation is just a function $R : A \to B$. In the following, we denote with **Rel** and **Set** the category having sets as objects and relations or functions, respectively, as arrows.

We assume the reader to be familiar with the usual definition of signature, (term) algebra, and so on. For the sake of simplicity, throughout the paper we will restrict our analysis to the one-sorted case: we leave the generalisation to the many-sorted case as a matter for future research.

Definition 2 (multi-algebras). *Let Σ be a signature. A* multi-algebra *A over Σ is a pair $\langle |A|, \rho_A \rangle$, where the* carrier *$|A|$ is a set, and $\rho_A = \{f_A \mid f \in \Sigma\}$ is a family of relations such that, for every $f \in \Sigma_n$, $f_A : |A|^n \leftrightarrow |A|$.*

Unlike the case of algebras, for multi-algebras many different definitions of homomorphisms are possible [WM97a]: in this paper we stick to the notion that we find most natural (and which better fits with the categorical framework we shall use), namely *tight, point-to-set* homomorphisms.

Definition 3 (homomorphism). *Let A and B be multi-algebras over Σ. A* (tight, point-to-set) homomorphism *$\varphi : A \to B$ is a relation $\varphi : |A| \leftrightarrow |B|$ such that $\varphi^n; f_B = f_A; \varphi$ for all $f \in \Sigma_n$. We will denote by $\mathbf{MAlg}_\Sigma$ the category of multi-algebras over Σ and tight homomorphisms.*

3 On substitutivity

Since the interpretations of the operators of the given signature Σ in a multi-algebra A are relations, given a term t from $T_\Sigma(X)$, the term algebra of Σ, it is natural to consider t as denoting a derived relation in A: informally, this makes use of the fact that if $f, g : S \leftrightarrow S$ are set-valued functions $f, g : S \to \mathcal{P}(S)$, then we can compose them as $f; g^+ : S \to \mathcal{P}(S)$, where g^+ is the pointwise extension of g. Still, we can choose how to interpret variables. In the *singular* interpretation they denote individuals, while in the *plural* interpretation they denote arbitrary sets. As explained neatly in [WM95], the plural interpretation of variables is consistent with the choice of power algebras as models, while the singular interpretation is consistent with multi-algebras, and therefore we will stick to the latter.

This choice immediately brings us to face the problem of lack of substitutivity, which we explain with the following example borrowed from [KB99].

Example 1 (lack of substitutivity). Let $\Sigma = \{c, one, g\}$ be a signature with constants c and one, and g a binary operator. Furthermore, let $A \in \mathbf{MAlg}_\Sigma$ be as follows: $|A| = \{0,1\}$, $c_A = \{0,1\}$, $one_A = \{1\}$, $g_A(x,y) = \{1\}$ if $x = y$, and $g_A(x,y) = \{0\}$ otherwise.

Now consider the equation $g(x,x) = one$. For any valuation of the variable x the equation is satisfied in A, because since x is a singular variable, the valuation associates a single value to it, so that $g_A(x,x)$ always returns $\{1\}$. Let us substitute c for x in the equation. It is easy to see that $g(c,c) = one$ does not hold in A, since $one_A = \{1\}$ which is different from

$$g_A(c_A, c_A) = g_A^+(\{0,1\},\{0,1\}) = g_A(0,0) \cup g_A(0,1) \cup g_A(1,0) \cup g_A(1,1) = \{0,1\}.$$

A closely related problem is put in evidence in [Hus92], where it is shown that the classical term rewriting calculus (in particular the substitutivity rule) is not sound in the presence of nondeterministic operators.

The problems related to substitutivity just sketched are also related to (but distinct from) the well-known fact that the *call-time-choice* and the *run-time-choice* semantics of parameter passing (in the presence of nondeterminism) in general differ. In the first case the actual parameter (which can be a nondeterministic expression) is evaluated before the execution of the procedure (as in a call-by-value semantics), while with run-time-choice the actual parameter is evaluated every time it appears in the body of the procedure (like in call-by-name semantics). For example, the function

$$g(x) \stackrel{def}{=} \textit{if } x = 0 \textit{ then } 0 \textit{ elsif } x \neq 0 \textit{ then } 1 \textit{ else } 2$$

can return 2 in the run-time-choice semantics only. Another related observation is the fact that the *inside-out* substitution (corresponding to call-time-choice) is not associative in the presence of nondeterminism [ES77,ES78].

Quite obviously, all these phenomena, which arise only in the presence of nondeterminism, are due to the fact that a nondeterministic expression, if evaluated more than once, can deliver different results. Many authors have obtained handy techniques for reasoning about nondeterminism by introducing some notion of *sharing*.

For example, Astesiano and Costa [AC79] extend the lambda-calculus with sharing, by modifying the β-rule in such a way that the actual substitution is deferred until when the term to be substituted is in normal form: this approach has the merit of combining the good properties of the call-time-choice semantics with the better termination properties of call-by-need parameter passing. Konikowska and Białasik [KB99] introduce a binding operator **let** which realises sharing: they explain the difference between the term **let** $x := c$ **in** $g(x, x)$ **ni** and $g(x, x)[c/x]$ by showing that in the multi-algebra of Example 1 the former is evaluated to $\{1\}$, while the latter to $\{0, 1\}$. This is a concrete example of the well-known fact that sharing constrains nondeterminism.

Hesselink [Hes88] introduces *accumulated arrows* as a generalisation of the term algebra concept. An *arrow* is an operation of the signature which can return more than one value, and accumulated arrows are built from arrows by *partial composition* and *rearrangements*. Given a multi-algebra for the signature, each accumulated arrow is associated inductively with a derived operation (such operations are used for example in the formal definition of the *implementation equivalence* between models and of the *extraction equivalence* between values). Even if this point is not made explicit in [Hes88], it is easy to see that rearrangements can be used to introduce a form of sharing, and indeed it can be shown that in the case of an ordinary signature, accumulated arrows are essentially (a more concrete version of) the term graphs we are going to introduce.

Therefore on the one hand the use of sharing when reasoning about nondeterministic operations is quite well understood; on the other hand many other approaches stick to standard terms and address the problems sketched above directly in the design of the logical system. As a typical example, substitutivity is only admitted for deterministic terms in the calculi proposed, e.g., in [WM97b,KB99] and in the variation of term rewriting discussed in [Hus92].

We understand that there are many good reasons for refraining from using terms with sharing in the formal specification of nondeterministic systems: they are intrinsically more complex than standard terms; they force one to depart from the standard syntax used in the deterministic case; when using terms a much richer theory and tool support (for example for rewriting) can be exploited; and the use of terms with sharing is usually justified by pragmatical considerations, which may look weak if compared with the central rôle that standard terms play for total algebras, as elements of the initial (or free) model.

Still admitting that the term graphs we are going to introduce are definitely more complex than terms, we disagree with the other points. The next two sections will show that, comparing the functorial presentations of algebras and multi-algebras (the former based on the Lawvere theory of Σ, the latter on the *gs-monoidal theory* [CG99a]), term graphs play the same rôle for multi-algebras that standard terms play for total algebras. Furthermore, the theory of term graphs and of term graph rewriting is nowadays well-developed (see the book [SPvE93] and the references therein), and term graph rewriting is used, as a matter of fact, in many implementations of functional languages [PvE93].

4 Term Graphs

This section introduces term graphs and some operations on them following [CG99a]. The presentation departs slightly from the standard definition (see e.g. [BvEG+87]) because we are more interested in algebraic properties of term graphs. In the following we assume that Σ is a standard one-sorted signature.

Definition 4 (term graphs and morphisms). *A* directed acyclic graph (dag) *(over Σ) is a triple $d = \langle N(d), l_d, s_d \rangle$, where $N(d)$ is a set of* nodes, *$l_d : N(d) \rightharpoonup \Sigma$ is a partial function called the* labelling function, *and $s_d : N(d) \rightharpoonup N^*$ is a partial function called the* successor function, *such that the following conditions are satisfied:*

- *$dom(l_d) = dom(s_d)$; a node $n \in N$ is called* empty *if $n \notin dom(l_d)$.*
- *for each node $n \in dom(l_d)$, $arity(l_d(n)) = length(s_d(n))$;*
- *d has no cycles (defined in the expected way).*

A (j,i)-ranked dag *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*[1] *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.
A (dag) morphism *$f : d \to d'$ is a function $f : N(d) \to N(d')$ that preserves labelling and successors, and if $g = \langle r, d, v \rangle$ and $g' = \langle r', d', v' \rangle$ are two ranked dags having the same rank, then a* ranked dag morphism *$\varphi : g \to g'$, is a dag morphism $\varphi : d \to d'$ such that $\varphi \circ r = r'$ and $\varphi \circ v = v'$.*

A (j,i)-ranked term graph *G is an isomorphism class of (j,i)-ranked dags. Sometimes we write G^i_j if G has rank (j,i).*

There are two natural operations on term graphs, defined as follows.

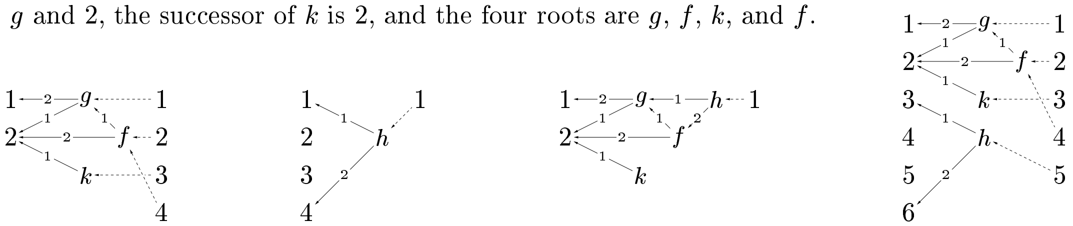
Definition 5 (Composition of ranked term graphs). *Let ${G'}^j_k = [\langle r', d', v' \rangle]$ and $G^i_j = [\langle r, d, v \rangle]$ be two ranked term graphs. Their* composition *is the ranked term graph $H^i_k = {G'}^j_k ; G^i_j$ defined as $H^i_k = [\langle in_d \circ r, d'', in_{d'} \circ v' \rangle]$, 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 $v(x) \approx r'(x)$ for $x \in \underline{j}$, 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.*

[1] For a natural number n we assume $\underline{n} = \{1, \dots, n\}$, thus $\underline{0} = \emptyset$.

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

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

The drawing below 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 $(2, 4)$, five nodes (two empty, 1 and 2, and three nonempty, f, g and k), the successors of g are the variables 2 and 1 (in this order), the successors of f are g and 2, the successor of k is 2, and the four roots are g, f, k, and f.

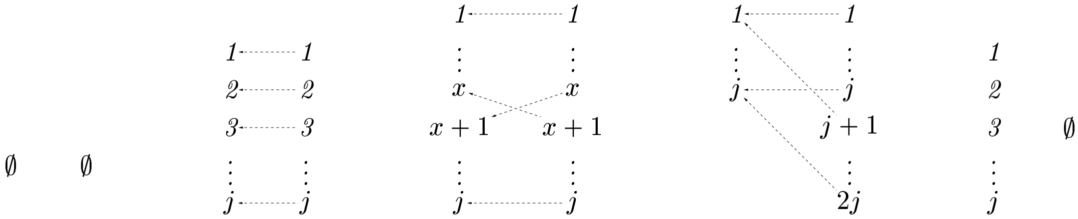


G_1 $\quad$ G_2 $\quad$ $G_3 = G_1 \,;\, G_2$ $\quad$ $G_4 = G_1 \oplus G_2$

These graphical conventions facilitate the understanding of the two basic operations on term graphs. The composition of two term graphs is obtained by first matching the roots of the first graph with the variables of the second, and then eliminating these variables. For example, term graph $G_3 = G_1 \,;\, G_2$ is the composition of G_1 and of G_2 of rank $(4, 1)$. Note that in this term graph node k is not reachable from the root: as such it is called *garbage*. The union of two graphs is essentially their disjoint union, where the lists of variables and roots are concatenated. The last term graph $G_4 = G_1 \oplus G_2$ is the union of G_1 and G_2, of rank $(6, 5)$.

We introduce now some term graphs which later will play a distinguished rôle.

Definition 7 (auxiliary term graphs). *For each* $j \in \mathbb{N}$ *and for each permutation* Π *on* $\underline{j}$, *the distinguished term graphs* $G_\emptyset$, $G^{\underline{j}}_{id}$, G_Π, $G^{\underline{j}}_\nabla$, *and* $G^{\underline{j}}_!$ *are defined as follows:*[2]



$G_\emptyset$ $\quad$ $G^{\underline{j}}_{id}$ $\quad$ $G_{\Pi^j_x}$ $\quad$ $G_{\nabla(\underline{j})}$ $\quad$ $G^{\underline{j}}_!$

[2] Since every node of such term graphs is a root or a variable, these term graphs are defined using, a bit improperly, r, v, and s for the root, variable, and successor functions, respectively.

[***Empty term graph***] *$G_\emptyset$ is the empty graph having rank $(0,0)$.*

[***Identities***] *Term graph G^{j}_{id} has rank (j,j) and exactly j empty nodes, with $r(x) = v(x)$ for each $x \in \underline{j}$.*

[***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 $x \in \underline{j}$, $v(x) = r(\Pi(x))$. The drawing above shows term graphs $G_{\Pi^j_x}$ for the basic permutation Π^j_x which exchanges x and $x+1$.*

[***Duplicators***] *Term graph $G_{\nabla(j)}$ for $j > 0$ has rank $(j, 2j)$ and it is defined as $r(x) = v(x)$ if $x \in \underline{j}$, and $r(x) = v(x-j)$ if $x \in \{j+1, \ldots, 2j\}$.*

[***Dischargers***] *Term graph $G^{j}_{!}$ has rank $(j,0)$, j empty nodes and no roots.*

5 Functorial semantics: from algebras to multi-algebras

Before summarising the functorial presentation of multi-algebras proposed in [CG99b,CG99c], we briefly recall the classical definition of *algebraic theory* over a signature —a category whose arrows represent (tuples of) terms, introduced in Lawvere's thesis [Law63]— and its use in providing a functorial presentation of (total) algebras.

Definition 8 (algebraic theories). *An* algebraic theory $\mathbf{C}$ *is a category whose objects are (underlined) natural numbers, such that for each $\underline{n}$ it is equipped with an n-tuple of* distinguished morphisms *(or* projections*)* $\{\pi^n_i : \underline{n} \to \underline{1} \mid i = 1 \ldots n\}$, *making $\underline{n}$ the n-fold categorical product of $\underline{1}$, that is, $\underline{n} = \underline{1}^n$.*

Given a signature Σ, the algebraic theory over Σ, $\mathbf{Th}(\Sigma)$, *is defined as follows. For each pair of objects $\underline{n}$ and $\underline{m}$, the hom-set $\mathbf{Th}(\Sigma)[\underline{n},\underline{m}]$ is the set of m-tuples of terms over Σ with variables in $\{x_1, \ldots, x_n\}$; arrow composition is term substitution; and for each $n \in \mathbb{N}$, $i \in \{1, \ldots, n\}$, the distinguished morphism π^n_i is $\langle x_i \rangle : \underline{n} \to \underline{1}$.*

Theorem 1 (algebras as functors). *The category $[\mathbf{Th}(\Sigma) \to \mathbf{Set}]^c$ of* cartesian *functors*[3] *from the algebraic theory over Σ to the category of sets, having natural transformations as arrows,*[4] *is equivalent to $\mathbf{Alg}_\Sigma$, the category of Σ-algebras and Σ-homomorphisms.* □

It is worth recalling how the correspondence between algebras and functors works. A Σ-algebra $A = \langle |A|, \{f_A \mid f \in \Sigma\}\rangle$ determines a cartesian functor $M_A : \mathbf{Th}(\Sigma) \to \mathbf{Set}$ as follows. On objects, let $M_A(\underline{1}) = |A|$; this determines $M_A(\underline{n})$ for each $n \in Nat$, as $M_A(\underline{n}) = M_A(\underline{1})^n = |A|^n$. On arrows, for each $f \in \Sigma_n$ let $M_A(\langle f(x_1, \ldots, x_n)\rangle : \underline{n} \to \underline{1}) = f_A : |A|^n \to |A|$; then M_A is determined uniquely on all arrows of $\mathbf{Th}(\Sigma)$ by closing under composition and tupling, and observing that the distinguished morphisms have to be mapped to projections. Vice-versa, and similarly, given a cartesian functor $M : \mathbf{Th}(\Sigma) \to \mathbf{Set}$, a Σ-algebra A_M is easily defined as $A_M = \langle M(\underline{1}), \{M(\langle f(x_1, \ldots, x_n)\rangle) \mid f \in \Sigma\}\rangle$.

Evidently, in this functorial presentation of algebras every arrow of the algebraic theory is mapped to an arrow of $\mathbf{Set}$, i.e., to a function. In particular, arrows of the form

[3] We denote as *cartesian* those functors F that preserve the n-fold product of $\underline{1}$ *on the nose*, that is, such that $F(\underline{n}) = F(\underline{1})^n$, and each distinguished morphism π^n_i is mapped to the corresponding i-th projection of $F(\underline{n})$.

[4] 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 $\mathbf{A}$, $\tau_a; G(f) = F(f); \tau_{a'}$.

$\langle f(x_1,\ldots,x_n)\rangle$, which faithfully represent operators of Σ, are interpreted as n-ary functions on the carrier $M(\underline{1})$.

As we have argued in [CG99b,CG99c], if one wants to interpret the operators of the signature as relations (as for multi-algebras), one should use **Rel** instead of **Set** as the target category of the functors. In particular, for technical reasons which are explained in detail in the cited papers, a correct functorial presentation of multi-algebras is obtained by considering a theory which has a structure weaker than cartesian, and functors to **Rel** which preserve such structure. This structure is the following:

Definition 9 (gs-monoidal categories). *A* strict 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 strict monoidal category (see [ML71]) and* $! : Id \Rightarrow e : \mathbf{C}_0 \to \mathbf{C}_0$, $\nabla : Id \Rightarrow \otimes \circ D : \mathbf{C}_0 \to \mathbf{C}_0$ *are two transformations (D is the diagonal functor), such that* $!_e = \nabla_e = id_e$ *and they satisfy the* coherence *axioms:*

$$\nabla_a; (id_a \otimes \nabla_a) = \nabla_a; (\nabla_a \otimes id_a), \quad \nabla_a; (id_a \otimes !_a) = id_a, \quad \nabla_a; \rho_{a,a} = \nabla_a$$

and the monoidality *axioms*

$$\nabla_{a\otimes b}; (id_a \otimes \rho_{b,a} \otimes id_b) = \nabla_a \otimes \nabla_b, \quad !_{a\otimes b} = !_a \otimes !_b$$

A gs-monoidal functor $\langle F, \varphi, \varphi_e\rangle : \mathbf{C} \to \mathbf{C}'$ *is a symmetric monoidal functor (that is, a functor F equipped with two natural isomorphisms* $\varphi_e : F(e) \to e'$ *and* $\varphi : F(a \otimes b) \to F(a) \otimes' F(b)$*) such that* $F(!_a); \varphi_e = !'_{F(a)}$ *and* $F(\nabla_a); \varphi = \nabla'_{F(a)}$*; it is* strict *if* φ *and* φ_e *are identities. The category of small strict gs-monoidal categories and their strict functors is denoted by* **GSM-Cat**.

Informally speaking, gs-monoidal categories have a structure which is in the middle between cartesian and symmetric monoidal categories:[5] it can be considered categorical folklore [Jac94,Laf95] that equipping a monoidal category with suitable *natural* transformations, one obtains a cartesian category (see for example [CG99a] for a recollection). Using our terminology, a cartesian category is just a gs-monoidal category where transformations ∇ and ! are natural.

The algebraic theory of a signature Σ (Def. 8) can also be defined as the "free cartesian category" generated by Σ. Similarly, the *gs-monoidal theory of* Σ, $\mathbf{GS\text{-}Th}(\Sigma)$, can be defined as the free gs-monoidal category generated by Σ. This category has a key rôle in the main result of [CG99b], which is the following:

Theorem 2 (functorial representation of multi-algebras). *Let* Σ *be a signature. Then there exists an equivalence of categories between the category* $\mathbf{MAlg}_\Sigma$ *of multi-algebras and tight homomorphisms and the category of gs-monoidal functors* $[\mathbf{GS\text{-}Th}(\Sigma) \to \langle \mathbf{Rel}, \times\rangle]^{gs}$ *and natural transformations.*[6]

Let us now come back to term graphs. The main results of [CG99a] are summarised in the following theorem, which states that term graphs over Σ can be put in one-to-one correspondence with the arrows of the gs-monoidal theory of Σ.

[5] We consider here categories where the tensor product is strictly associative.

[6] In the statement we indicate explicitely that the relevant gs-monoidal structure of category **Rel** is the one based on the direct product $\times$, because **Rel** enjoys another such structure based on the categorical product (which coincides with the coproduct, and it is disjoint union). In the following, $\times$ will be understood.

Theorem 3 (category of term graphs as gs-monoidal theory). *For a given signature Σ, let $\mathbf{TG}_\Sigma = \langle TG_\Sigma, \oplus, G_\emptyset, G_{\Pi(\_,\_)}, G_{\nabla(\_)}, G_{!}^{-} \rangle$ be the structure defined as follows:*

- *TG_Σ is the category having as objects underlined natural numbers, and as arrows from $\underline{j}$ to $\underline{i}$ all (j,i)-ranked term graphs. Arrow composition is defined as in Def. 5, and the identity on $\underline{j}$ is the identity term graph G_{id}^{j} of Def. 7. The axioms of categories are easily shown to hold.*
- *$\oplus : TG_\Sigma \times TG_\Sigma \to TG_\Sigma$ is a strictly associative bifunctor, such that $\underline{i} \oplus \underline{j} = \underline{i+j}$ on objects, and defined as in Def. 6 on arrows. The empty term graph $G_\emptyset$ is the unit for $\oplus$. Thus $\langle TG_\Sigma, \oplus, G_\emptyset \rangle$ is a strict monoidal category.*
- *For each n, m, let $\Pi(\underline{n}, \underline{m})$ be 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. Then term graph $G_{\Pi(\underline{n},\underline{m})}$ is as in Def. 7.*
- *For each j, $G_{\nabla(\underline{j})}$ and $G_{!}^{j}$ are as in Def. 7.*

Then $\mathbf{TG}_\Sigma$ is a gs-monoidal category. Furthermore, it is isomorphic to the gs-monoidal theory $\mathbf{GS\text{-}Th}(\Sigma)$. □

We conclude this section by exploring further the structure of the gs-monoidal category $\mathbf{TG}_\Sigma$. The leading intuition is that ranked dag morphisms equip the homsets of $\mathbf{TG}_\Sigma$ with a preorder, making it order-enriched. However, we do not start from dag morphisms for a reason which will be explained later.

Definition 10 (preorder $\sqsubseteq$ on term graphs). *Let the relation $\sqsubseteq$ on term graphs be defined as the smallest reflexive and transitive relation such that*[7]

1. *if F is a (i,j)-ranked term graph, then $F; !_j \sqsubseteq !_i$,*
2. *if F is a (i,j)-ranked term graph, then $F; \nabla_j \sqsubseteq \nabla_i; (F \otimes F)$,*
3. *for any four term graphs G_1, G_2, H_1, H_2, $G_1 \sqsubseteq G_2$ and $H_1 \sqsubseteq H_2$ implies $G_1 \otimes H_1 \sqsubseteq G_2 \otimes H_2$,*
4. *for any four term graphs G_1, G_2, H_1, H_2 such that $G_1; H_1$ and $G_2; H_2$ exist, $G_1 \sqsubseteq G_2$ and $H_1 \sqsubseteq H_2$ implies $G_1; H_1 \sqsubseteq G_2; H_2$.*

Also category **Rel** has a natural order-enriched structure, given by inclusions of relations. Interestingly, all gs-monoidal functors from $\mathbf{TG}_\Sigma$ to **Rel** preserve the order enriched structure.

Theorem 4 (gs-monoidal functors from the theory are order-enriched). *The structure $\langle \mathbf{TG}_\Sigma, \sqsubseteq \rangle$ is an order-enriched category. Moreover, every gs-monoidal functor M : $\mathbf{TG} \to \mathbf{Rel}$ is order-enriched, i.e., $F \sqsubseteq G$ implies $M(F) \subseteq M(G)$.*

Proof. The first point is obvious by the definition of $\sqsubseteq$. For the second point, since $M(!_j)$ and $M(\nabla_j)$ are total relations for every such functor (see [CG99b]), it follows that for each term graph F, $M(F; !_j) \subseteq M(!_i)$, and $M(F; \nabla_j) \subseteq M(\nabla_i; (F \otimes F))$; then the statement follows because also conditions 3 and 4 for $\sqsubseteq$ as listed in Def. 10 obviously hold for $\subseteq$ in *Rel* with conventional cartesian products. □

[7] For the sake of simplicity, in the following we will denote the term graph $G_{!}^{j}$ by $!_j$, and, similarly, G_{∇}^{j} by ∇_j.

The next result shows that relation $\sqsubseteq$ is exactly the preorder induced on term graphs by the existence of ranked morphims.

Theorem 5 (relation $\sqsubseteq$ is the preorder induced by morphisms). *Let G and H be two term graphs, and let $g \in G$, $h \in H$ be two arbitrary ranked dags. Then $G \sqsubseteq H$ if and only if there exists a ranked dag morphism $f : h \to g$.*

Proof (outline). **(Only if part)** Easy, by induction on the clauses defining $\sqsubseteq$, also observing that morphisms include identities and are closed by composition.

(If part) Let $f : h \to g$. We consider an epi-mono factorisation $h \xrightarrow{f'} g' \overset{f''}{\hookrightarrow} g$. For the injective morphism f'', $G \sqsubseteq [g']$ can be shown using only reflexivity of $\sqsubseteq$ and the first clause. For the surjective morphism f' the proof proceeds by induction on "the degree of non-injectivity", defined as $dni(f') = \#\{n \in N(h) \mid \exists n' \neq n \,.\, f'(n) = f'(n')\}$. If $dni(f) = 0$ then f' is an isomorphism and we are done. For the inductive case, suppose $dni(f') > 0$. Then select a node n of h which is not mapped injectively by f' and of "maximal depth" (i.e., such that all nodes from which n is reachable are mapped injectively). Factorise f' as $f_1; f_2$, where: the intermediate graph is obtained from g' by adding a fresh node n'; morphism f_1 is defined as $f_1(n) = n'$ and $f_1(x) = f'(x)$ elsewhere; and f_2 is defined as $f_2(n') = f'(n)$ and $f_2(x) = x$ elsewhere (requiring that f_1 and f_2 are dag morphisms uniquely determines the successor and labeling functions of the intermediate graph). Observe that $dni(f_2) = 2$ and $dni(f_1) < dni(f')$. Therefore the induction hypothesis can be applied to f_1, while the statement for f_2 follows essentially by a single application of the second clause of the definition of $\sqsubseteq$, using the third and fourth clauses as congruence rules. $\square$

6 Using term graphs to denote relations

A *derived operation* in a (total) Σ-algebra A is a function obtained by composing arbitrarily (using both functional application and pairing) the fundamental operations, which are the functions providing the interpretation of the operators of Σ in A. For a given algebra $A : \mathbf{Th}(\Sigma) \to \mathbf{Set}$, the derived operations of A are one-to-one with the arrows of the algebraic theory, essentially because such arrows are freely generated from the operators of the signature (as it can be grasped from the characterisation of the algebraic theory as a free construction, see, e.g., [CG99a]), and because A maps them to arrows of **Set**, which are indeed functions.

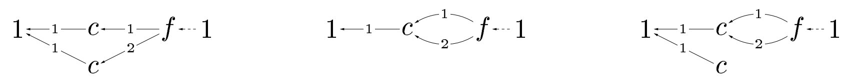
Now, the fact that each (possibly non-ground) term over Σ denotes a derived operation becomes obvious by the fact that arrows of the algebraic theory from $\underline{n}$ to $\underline{m}$ are one-to-one with m-tuples of terms with at most n variables. We made this obvious observation just to stress the fact that the central rôle played by terms in the syntax of (total) algebraic specification can be put in connection with the structure of the theory used in the functorial presentation of total algebras. This suggests that for other classes of algebras, having a different functorial presentation, other syntactical structure may play the rôle of terms, i.e., may faithully denote derived operators.

In particular, in the case of multi-algebras, Theorem 3 above motivates the choice of term graphs as a concrete syntax for derived relations. The usefulness of (suitable kinds of) diagrams to denote relations and to calculate on them, is witnessed not only by the use of sharing in papers like [AC79,Hes88,KB99], as discussed in Sect. 3, but also, for example, by the graphical calculi of [CL95,Kah97], and by applications to circuit design as in [BH94].

Given a multi-algebra $A : \mathbf{TG}_\Sigma \to \mathbf{Rel}$, it is not difficult to present the set-valued function associated with a given term graph: we show this by an example. Let us consider graph G_3 of the drawing after Def. 6. Then G_3^A is a function from $A(\underline{1})^2$ to $\mathcal{P}(A(\underline{1}))$, and it is defined as

$$a \in G_3^A(a_1, a_2) \Leftrightarrow \begin{cases} \exists x_g, x_f, x_h, x_k \in A(\underline{1}) \,.\, x_g \in g^A(a_2, a_1) \wedge x_f \in f^A(x_g, a_2) \wedge \\ \wedge x_h \in h^A(x_g, x_f) \wedge x_k \in k^A(a_2) \wedge a = x_h \end{cases}$$

where f^A, g^A, h^A and k^A are the fundamental relations corresponding to the operators of the signature. Note that in the definition of relation G_3^A the variable x_k is not used elsewhere. In fact, the garbage of a term graphs has in the denoted relation a rôle of "domain restriction".



F_1 $\qquad\qquad$ F_2 $\qquad\qquad$ F_3

As a further example, consider the term graphs of the drawing above. There, c is a unary and f a binary operator, and they correspond to the arrows $\alpha_1 = \nabla_{\underline{1}}; c \otimes c; f$, $\alpha_2 = c; \nabla_{\underline{1}}; f$, and $\alpha_3 = \nabla_{\underline{1}}; ((c; \nabla_{\underline{1}}; f) \otimes (c; !_{\underline{1}}))$ all of them in $\mathbf{GS\text{-}Th}(\Sigma)[\underline{1}, \underline{1}]$. Note that these arrows are distinct in $\mathbf{GS\text{-}Th}(\Sigma)$, but they are equivalent in the algebraic theory $\mathbf{Th}(\Sigma)$ because there, ∇ and ! are natural (naturality of ∇ implies $\alpha_1 = \alpha_2$, and naturality of ! implies $\alpha_2 = \alpha_3$). More intuitively, this means that "unravelling" F_1, F_2 and F_3 from the root node f we obtain the same term $f(c(1), c(1))$.

For a given multi-algebra A, the three term graphs represent the following relations:

$a \in F_1^A(a') \Leftrightarrow \exists x_1, x_2 \,.\, x_1 \in c^A(a') \wedge x_2 \in c^A(a') \wedge a \in f^A(x_1, x_2)$,
$a \in F_2^A(a') \Leftrightarrow \exists x_1 \,.\, x_1 \in c^A(a') \wedge a \in f^A(x_1, x_1)$, and
$a \in F_3^A(a') \Leftrightarrow \exists x_1, x_2 \,.\, x_1 \in c^A(a') \wedge x_2 \in c^A(a') \wedge a \in f^A(x_1, x_1)$.

Clearly, the first two relations are in general different, because if $c^A(a')$ has at least two elements, then f^A could be applied to two different arguments in F_1^A, but not in F_2^A. However, the second and third relations coincide for every possible multi-algebra, because the domain restriction forced by the garbage node c ($\exists x_2 \,.\, x_2 \in c^A(a')$) is already subsumed in F_2^A. This shows that term graphs are slightly redundant as far as one is concerned in denoting the derived relations of a multi-algebra. Let us make this point more precise. From Theorems 4 and 5 we immediately have

Corollary 1. *If $f : g \to h$ is a ranked dag morphism, and M is a gs-monoidal functor from $\mathbf{TG}_\Sigma$ to Rel, then $M([h]) \subseteq M([g])$.*

As an aside, we may note that the statement corresponding to Corollary 1 fails to hold in an alternative set of relational models for gs-monoidal theories, namely where *demonic composition* "$\sqsubset$" is used instead of standard relational composition, and the ordering used is the *refinement ordering* "$\sqsubseteq$" (see [DMN97]). In that context, sequential composition and cartesian products are still monotonic, and $F \,\square\, \nabla \sqsubseteq \nabla \,\square\, (F \times F)$ also holds, but for the discharger the ordering goes the other way around: in general, we have $! \sqsubseteq F \,\square\, !$ (for further discussion, see [Kah00]). For this reason, we did not give Def. 10 the status of *the definition* of order-enriched gs-monoidal categories, since the example of the demonic calculus of relations shows that sensible orderings in gs-monoidal categories can be given in quite different ways.

Definition 11 (garbage equivalence). *Two ranked term graphs G_1 and G_2 are called* garbage equivalent*, written $G_1 \underset{g}{=} G_2$, iff for every ranked dags $g_1 \in G_1$ and $g_2 \in G_2$ there are two ranked dag homomorphisms $f : g_1 \to g_2$ and $g : g_2 \to g_1$.*

As an example, term graphs F_2 and F_3 of figure 3 are garbage equivalent. The name we have given to the relation $\underset{g}{=}$ may seem strange at first, but is justified by the following:

Lemma 1. *Let the* skeleton *of a term graph be its subgraph obtained by removing all garbage, namely, the subgraph containing all empty nodes and all nodes reachable from the roots. If $G_1 \underset{g}{=} G_2$, then the skeletons of G_1 and G_2 are isomorphic.*

We conclude this section by showing that garbage equivalence equates exactly the term graphs that denote the same relation in all multi-algebras. The details of the proof will appear elsewhere.

Theorem 6 (soundness and completeness of garbage equivalence). *Let G_1 and G_2 be term graphs. Then $G_1 \underset{g}{=} G_2$ if and only if for every gs-monoidal functor $M : \mathbf{TG}_\Sigma \to \mathbf{Rel}$, we have $M(G_1) = M(G_2)$.*

Proof (outline). The *only if part* (soundness) immediately follows from Corollary 1. For the *if part* (completeness), suppose that $M(G_1) = M(G_2)$ in all Σ-multi-algebras. Then one builds a multi-algebra M_{G_1} whose carrier contains all *pointed, standard dags*[8] which are garbage equivalent to G_1, and the basic operations are defined in a suitable way. Next one can show that each term graph H such that $M_{G_1}(H) = M_{G_1}(G_1)$ is garbage equivalent to G_1, and this concludes the proof. □

7 Term Graph Equational Logic

Having chosen term graphs to denote derived operators in a multi-algebra, we can explore their use in specification. We consider here the simplest case, namely equational logic. The formal framework sketched in Section 5 gives us precise guidelines for the basic definitions of the calculus. In particular, equality is interpreted as standard equality of relations.

Definition 12 (term graph equations, validity).

A (term graph) equation over *Σ is a pair $\langle F, G\rangle$ of term graphs of the same rank, usually written $F = G$.*

Given a Σ-multi-algebra A, i.e., a gs-monoidal functor $A : \mathbf{TG}_\Sigma \to \mathbf{Rel}$, we say that equation $F = G$ holds in *A if $A(F) = A(G)$, i.e., if they are the same relation. In this case we write $A \models F = G$. The same equation is* valid *(written $\models F = G$) if it holds in all Σ-multi-algebras.*

These notions extend in the expected way to a set of equations E: we write $A \models E$ if $A \models e$ for all $e \in E$. Also, if $\mathcal{A}$ is a class of multi-algebras, we write $\mathcal{A} \models E$ if $A \models E$ for all $A \in \mathcal{A}$.

A (term graph equational) specification *is a pair $\langle \Sigma, E\rangle$, where E is a set of term graph equations over Σ. Given a specification $\langle \Sigma, E\rangle$, the class of its* models *is $\mathbf{MAlg}_{\Sigma,E} = \{A \in \mathbf{MAlg}_\Sigma \mid A \models E\}$*

[8] *Standard* means that the nodes are natural numbers, *pointed* means "with a distinguished node".

By the discussion in the last section it is clear that we have to take into account garbage equivalence in the deduction system. We do this in quite a rude way, with a non-standard rule that includes in the premises the existence of suitable term graph morphisms; we will discuss this point later on.

Definition 13 (rules of term graph equational deduction). *The* deduction rules of term graph equational logic *are the following, where all equations are assumed to be well-formed, i.e., the left- and right-hand-side have the same rank.*

[**reflexivity, symmetry, transitivity**] *For all term graphs G, G', G'':*

$$\frac{}{\emptyset \vdash G = G} \qquad \frac{E \vdash G = G'}{E \vdash G' = G} \qquad \frac{E_1 \vdash G = G', \quad E_2 \vdash G' = G''}{E_1 \cup E_2 \vdash G = G''}$$

where $H \to K$ holds true iff there is a term graph ranked morphism between any dag in H and any dag in K.

[**substitutivity**] *For any term graphs G_1, G_2, H_1, H_2 such that $G_1;G_2$ and $H_1;H_2$ are defined,*

$$\frac{E_1 \vdash G_1 = H_1, \quad E_2 \vdash G_2 = H_2}{E_1 \cup E_2 \vdash G_1;G_2 = H_1;H_2}$$

[**congruence**] *For any term graphs G_1, G_2, H_1, H_2,*

$$\frac{E_1 \vdash G_1 = H_1, \quad E_2 \vdash G_2 = H_2}{E_1 \cup E_2 \vdash G_1 \oplus G_2 = H_1 \oplus H_2}$$

[**garbage equivalence**] *For any term graphs F, F', G, and G':*

$$\frac{G' = G, \quad F \to G', \quad F' = F, \quad G \to F'}{\emptyset \vdash F = G}$$

Given a set of equations E and an equation e, we write $E \vdash e$ if there is a proof of $E' \vdash e$, with $E' \subseteq E$, using only the above rules and the following one:

[**axiom**]

$$\frac{e \in E}{\{e\} \vdash e}$$

It comes of no surprise that this calculus is sound, given the discussions in Section 3 about substitutivity and the soundness of garbage equivalence proved in the previous section (actually, the full proof also makes use of a lemma showing that garbage equivalence is a gs-monoidal congruence).

Theorem 7 (soundness of term graph equational deduction). *Let Σ be a signature and let E be a set of term graph equations over Σ. Then for every term graph equation $F = G$ over Σ,*

$$E \vdash F = G \quad \Longrightarrow \quad \mathbf{MAlg}_{\Sigma,E} \models F = G$$

We claim that the above calculus is also complete, i.e., that actually an "if and only if" holds in the previous statement, but we were not able to proof it yet.

Clearly, the form of the premises of the *[garbage equivalence]* rule is quite disappointing, but apparently there is no easy way to get rid of them. Informally, the rule is needed to deduce, in the presence of relations $R \subset S \subset T$ and of an equation $R = T$, that also $S = R$ and $S = T$. The rule shows that even if we are concerned in proving equalities of relations only, inequations come into play: this confirms that, as observed by many authors, a calculus based on inequalities is more natural for multi-algebras. This is also motivated by the fact that (standard) equations are in general very poor for specifying nondeterminism, and this leads to the introduction of logics containing not only inequalities as atomic formulas (interpreted as inclusions of the denoted relations), but also some limited form of negation, or even full first order logic (see for example [Hus92,WM95,WM97b,KB99], and [WM97a] for an overview).

We conclude this section by briefly commenting on the expressive power of term graph equations, showing three properties that we think it would not be possible to describe using standard term equations, and interpreting equality as standard equality of relations. In fact, using term graph equations we can force totality and/or univalence of a derived relation, and we can express a limited form of inclusion, using *domain restriction*.

Proposition 1 (specifying with term graph equations). *Let A be a Σ-multi-algebra, and G, F be term graphs of rank (j, i).*

- $A \models G; !_i = \ !_j$ *if and only if G^A is a total relation.*
- $A \models G; \nabla_i = \nabla_j; (G \oplus G)$ *if and only if G^A is a univalent relation, i.e., $\forall w \in |A|^j \,.\, |G^A(w)| \leq 1$.*
- $A \models \nabla_j; (G \oplus (F; !_i)) = F$ *if and only if F^A is a domain restriction of G^A, i.e., if $\forall w \in |A|^i \,.\, w \in dom(F^A) \Rightarrow F^A(w) = G^A(w)$.*

Interestingly, we can express in this purely equational framework the DET predicate of [Hus92], where $DET(t)$ means "t is total and univalent", and therefore also the *necessary equality* used in various calculi [WM97b,WM95]. There, $F \doteq G$ holds in A if both F^A and G^A are total and univalent, and they are equal as functions. By the last proposition we clearly have

$$\begin{array}{rcl} A \models F \doteq G & \Leftrightarrow & A \models F{=}G \ \wedge\ G; !_i = \ !_j \ \wedge\ F; !_i = \ !_j \ \wedge \\ & & G; \nabla_i = \nabla_j; (G \oplus G) \ \wedge\ F; \nabla_i = \nabla_j; (F \oplus F) \ . \end{array}$$

Despite this fact, the expressive power of term graph equations is quite limited. One cannot specify general inclusion between relations, and one cannot force partiality or forbid univalence. For example, it is not possible to specify a binary choice operator which returns a non-singleton set of results in all models.

8 Discussion and Conclusions

Our main goal in this paper was to substantiate the claim that term graphs should play for the specification of multi-algebras the same rôle that standard terms play for total algebras. The technical justification of this comes from a functorial presentation of multi-algebras, originally proposed in [CG99b], where term graphs are shown to be in one-to-one correspondence to the arrows of the classifying theory. We also presented a calculus for term graph equational deduction proving its soundness, but only claiming its completeness. On the one hand, the calculus takes full advantage of the use of term graphs, because an unrestricted substitutivity rule holds (which does not hold for other calculi for multi-algebras based on the singular interpretation of variables). On the other hand, the calculus includes a non-standard rule used to capture the garbage equivalence on term graphs: in the absence of non-logical axioms, this equivalence was shown to identify exactly those term graphs denoting the same relation in all multi-algebras.

The next topic of research we would like to address is the study of logical calculi for multi-algebraic specification based on term graphs, and their comparison to other calculi from the literature with respect to both the complexity of rules, and the expressive power. Our very first goal is to try to improve the presentation of the equational deduction calculus, and to prove its completeness. Next we intend to extend the calculus in various ways.

The first natural extension of term graph equational logic consists of allowing for atomic formulas like $F \prec G$, which holds in a multi-algebra A if $F^A \subseteq G^A$. We think that this extension would fit smoothly in our formal framework, and it would allow to handle in a satisfactory way the garbage equivalence issue. Let us sketch how the order enriched structure of $\mathbf{TG}_\Sigma$ explored in Sect. 5 can provide us a calculus for inequations. Given a set of atomic formulas $\Phi = \{F_i \prec G_i \mid i \leq n\}$, let $\langle \mathbf{TG}_\Sigma, \sqsubseteq_\Phi \rangle$ be the order-enriched gs-monoidal category having $\mathbf{TG}_\Sigma$ as underlying category, and where $\sqsubseteq_\Phi$ is defined like $\sqsubseteq$ in Def. 10, by adding the clause

0. *if* $F \prec G$ *then* $F \sqsubseteq_\Phi G$.

Next the models of the specification $\langle \Sigma, \Phi \rangle$ are defined as all order-enriched gs-monoidal functors from $\langle \mathbf{TG}_\Sigma, \sqsubseteq_\Phi \rangle$ to $\langle \mathbf{Rel}, \subseteq \rangle$. Quite obviously, in general such functors are less than all multi-algebras, i.e., Theorem 4 does not hold for $\sqsubseteq_\Phi$. At this point the axioms and rules of Def. 10 (including reflexivity and transitivity), together with the axioms in Φ provide us a calculus for inequations which is sound by construction, but whose completeness is a matter of future research.

Further extensions we foresee include Horn clause logic, and some limited form of negation which appears to be needed in order to have that basic expressive power which is needed to specify satisfactorily, for example, a non-deterministic choice operator (see, e.g., [WM97b]).

References

[AC79] E. Astesiano and G. Costa. Sharing in Nonodeterminism. In *Automata, Languages and Programming Sixth Colloquium*, volume 71 of *LNCS*. Springer Verlag, 1979.

[BH94] C. Brown and G. Hutton. Categories, Allegories and Circuit Design. In *Proc. 9th Annual Symposium on Logic in Computer Science, Paris, France*, pages 372–381. IEEE Computer Society Press, 1994.

[BvEG+87] 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.

[CG99a] A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 1999. To appear.

[CG99b] 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.

[CG99c] A. Corradini and F Gadducci. Functorial Semantics for Multi-algebras and Partial Algebras, with Applications to Syntax. Submitted for publication, 1999.

[CL95] S. Curtis and G. Lowe. A graphical calculus. In B. Möller, editor, *Mathematics of Program Construction (MPC'95)*, volume 947 of *LNCS*, pages 214–231. Springer Verlag, 1995.

[DMN97] Jules Desharnais, Ali Mili, and Thanh Tung Nguyen. Refinement and demonic semantics. In C. Brink, W. Kahl, and G. Schmidt, editors, *Relational Methods in Computer Science*, pages 166–183, Wien, 1997. Springer-Verlag.

[ES77] J. Engelfriet and E.M. Schmidt. IO and OI.1. *J. Comput. Syst. Sci*, 15:328–353, 1977.

[ES78] J. Engelfriet and E.M. Schmidt. IO and OI.2. *J. Comput. Syst. Sci*, 16:67–99, 1978.

[Hes88] W.H. Hesselink. A Mathematical Approach to Nondeterminism in Data Types. *ACM TOPLAS*, 10:87–117, 1988.

[Hus92] H. Hussmann. Nondeterministic Algebraic Specifications and Nonconfluent Term Rewriting. *J. Logic Programming*, 12:237–255, 1992.

[Jac94] B. Jacobs. Semantics of weakening and contraction. *Annals of Pure and Applied Logic*, 69:73–106, 1994.

[Kah97] W. Kahl. Algebraic graph derivations for graphical calculi. In F. D'Amore, P.G. Franciosa, and A. Marchetti-Spaccamela, editors, *Graph Theoretic Concepts in Computer Science (WG'96)*, volume 1197 of *LNCS*, pages 224–238. Springer Verlag, 1997.

[Kah00] Wolfram Kahl. Unsharp demonic products. Talk to be presented at the Fifth International RelMiCS Workshop, 2000.

[KB99] B. Konikowska and M. Białasik. Reasoning with first order nondeterministic specifications. *Acta Informatica*, 36:375–403, 1999.

[Laf95] 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.

[Law63] F.W. Lawvere. Functorial semantics of algebraic theories. *Proc. National Academy of Science*, 50:869–872, 1963.

[ML71] S. Mac Lane. *Categories for the working mathematician.* Springer Verlag, 1971.

[PvE93] M.J. Plasmeijer and M.C.J.D. van Eekelen. *Functional Programming and Parallel Graph Rewriting.* Addison Wesley, 1993.

[SPvE93] M.R. Sleep, M.J. Plasmeijer, and M.C. van Eekelen, editors. *Term Graph Rewriting: Theory and Practice.* Wiley, London, 1993.

[WM95] M. Walicki and S. Meldal. Multialgebras, Power Algebras, and Complete Calculi of Identities and Inclusions. In E. Astesiano, G. Reggio, and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 906 of *LNCS*, pages 453–468. Springer Verlag, 1995.

[WM97a] M. Walicki and S. Meldal. Algebraic approaches to nondeterminism: An overview. *ACM Computing Surveys*, 29:30–81, 1997.

[WM97b] M. Walicki and S. Meldal. A complete calculus for the multialgebraic and functional semantics of nondeterminism. *ACM Trans. Program. Lang. Syst.*, 17:366–393, 1997.