Università di Pisa

Dipartimento di Informatica

Technical Report : TR-98-05

Rewriting on cyclic structures

Andrea Corradini, Fabio Gadducci

June 12, 1998

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

Rewriting on cyclic structures*

Andrea Corradini[1] and Fabio Gadducci[2]

[1] Dipartimento di Informatica, Corso Italia 40, 56125, Pisa, Italy, (andrea@di.unipi.it).
[2] TUB, Fachbereich 13 Informatik, Franklinstraße 28/29, 10587 Berlin, Germany, (gfabio@cs.tu-berlin.de).

Abstract. We present a categorical formulation of the rewriting of *possibly cyclic* term graphs, and the proof that this presentation is equivalent to the well-accepted operational definition proposed in [3]—but for the case of *circular redexes*, for which we propose (and justify formally) a different treatment. The categorical framework, based on suitable 2-categories, allows to model also automatic garbage collection and rules for sharing/unsharing and folding/unfolding of structures. Furthermore, it can be used for defining various extensions of term graph rewriting, and for relating it to other rewriting formalisms.

Keywords: term graph rewriting, categorical models, rewriting logics.

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 [27] 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 subterms 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 subterm. Also, 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 [31]). For these reasons term graph rewriting is often used, for example, in the implementation of functional languages [23]

However, in our opinion there is an unsatisfactory gap between the achievements of the theories of term and of term graph rewriting. In particular, a solid ground for the theory of term rewriting is provided by the existence of three different but equivalent characterizations of rewriting, namely the *operational*, classical one (described in terms of redexes and substitutions [18]), the *logical* one (we think of the *rewriting logic* formalism [20]), and the *categorical* one, based on *algebraic (cartesian) 2-theories* [24]. While the operational description

* Research partly supported by the EC TMR Network GETGRATS (General Theory of Graph Transformation Systems) through the University of Pisa and the Technical University of Berlin.

is the most 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 from representation details, 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". It is worth noting that a similar "triangular" correspondence also exists for other rule based formalisms, like for example *Petri nets*, where the operational definition of, say, [25] parallels a categorical counterpart based on *symmetric strict monoidal categories* [21] and a logical one based on (the multiplicative fragment of) linear logic (see [19] for a survey).

The situation is not so lucky for term graph rewriting, which has been defined in the literature in many –yet operational in flavour– ways, representing term graphs for example as directed acyclic graphs satisfying a number of constraints [3], as suitable labelled hypergraphs called *jungles* [12], or as sets of recursive equations [1]. Only recently we have shown in [7] that the rewriting of *acyclic* term graphs can also be presented in a categorical way similar to the 2-categorical presentation of term rewriting. In fact, (ranked, acyclic) term graphs over a signature Σ are in one-to-one correspondence with the arrows of the free *gs-monoidal* category generated by Σ [8]. And term graph rewriting sequences over a rewriting system $\mathcal{R}$ (according to the definition in [3], the most widely accepted in the literature) are faithfully represented by the cells of the free gs-monoidal 2-category generated by a suitable representation of the rules in $\mathcal{R}$.

The gs-monoidal categories are symmetric strict monoidal categories equipped with two transformations, the *duplicator* "∇" and the *discharger* "!", from which cartesian categories can be recovered requiring their naturality. As shown in [8], but already argued in [28], the non-naturality of ∇ is related to the fact that term graphs with different degree of sharing are distinct, while that of ! allows for the presence of garbage in a term graph (i.e., nodes not reachable from the roots).

The main contribution of this paper is the generalization of the results of [7] to the categorical representation of *possibly cyclic* term graph rewriting. In Section 2 we first introduce *(possibly cyclic) ranked term graphs*, and three operations on them: *Composition* (a counterpart of term substitution), *(disjoint) union* and *feedback*; next we introduce our definition of term graph rewriting, which is a little variation of that in [3]. Section 3 introduces *traced gs-monoidal 2-categories*, an enrichment of the above mentioned categories with the "traced" structure firstly proposed in [13] and few other needed categorical notions.

In Section 4 we first show that cyclic term graphs over Σ can be represented as the arrows of the free traced gs-monoidal category generated by Σ; admittedly, this result is derivable from recent works on *flownomial algebras* [28, 29], even if our formulation looks original. More interestingly, we then show that if we represent the term graph rules of a rewriting system as cells over this category, and we generate the free traced gs-monoidal 2-category from it, the resulting cells faithfully represent the term graph rewriting sequences of the system satisfying a mild restriction.

An immediate consequence of this result is that by representing (cyclic) term graphs as arrows of a suitable free category, we highlight the intrinsic algebraic structure of such term graphs, independently of a specific representation: as "cartesianity" is the essential structure of terms, "traced gs-monoidality" turns out to be the essential structure of (cyclic) term graphs. Even if 2-categories equivalent to ours have been considered independently in [22, 11], in both works the focus is the interaction between such categorical structures and a *logical* presentation of term graph rewriting; while to our knowledge, our correspondence result is the first that explicitly and formally relates a categorical and an operational description of term graph rewriting.

In Section 5 we first explain the differences between our definition of term graph rewriting and that originally proposed in [3], related to *garbage collection* and to the different handling of *circular redexes*. In particular, we show that small modifications of the formal definitions allow us to capture term graph rewriting with automatic garbage collection as well, and justify the disagreement with [3] on circular redexes (the paradigmatical example is the application of rule $R_I : I(x) \to x$ to the graph having one node labelled I and one looping edge) in terms of the categorical structure. Next we discuss how to introduce rules for the automatic folding or unfolding of structures, showing that this allows to relax a technical restriction on redexes in the main correpondence result. Finally, we summarize the relationship between traced gs-monoidal 2-categories and other 2-categorical models of rewriting proposed in the literature using suitable adjunctions.

2 (Cyclic) Term Graphs and Term Graph Rewriting

This section introduces (ranked, possibly cyclic) term graphs as isomorphism classes of (ranked) labeled graphs, as well as the basic definitions related to Term Graph Rewriting. Since our main concern is to stress the underlying algebraic structure, the following presentation of term graphs slightly departs from the standard definition of [3]: See [8] for a precise comparison.

Definition 1 (graphs). *Let Σ be a (one sorted) signature, i.e., a ranked set of operator symbols, with* $\mathrm{arity}(f) = n$ *iff* $f \in \Sigma_n$. *A* (labeled, directed) graph d (over Σ) *is a triple* $d = \langle N, l, s\rangle$, *where* N *is a set of* nodes *(ranged over by* $a, b, \ldots$*),* $l : N \to \Sigma$ *is a partial function called the* labeling function, $s : N \to N^*$ *is a partial function called the* successor function, *and such that* $dom(l) = dom(s)$, *and for each node* $a \in dom(l)$, $arity(l(a)) = length(s(a))$.

A node $a \in N$ *is called* empty *if* $a \notin dom(l)$. *If* $s(a) = \langle a_1, \ldots, a_k\rangle$, *we say that* a_i *is the* i-th successor *of* a *and denote it by* $s(a)_i$. *A graph is* discrete *if all its nodes are empty. A* path *in* d *is a sequence* $\langle a_0, i_0, a_1, \ldots, i_{m-1}, a_m\rangle$, *where* $m \geq 0$, $a_0, \ldots, a_m \in N$, $i_0, \ldots, i_{m-1} \in \mathbb{N}$ *(the natural numbers), and* a_k *is the*

i_{k-1}*-th successor of* a_{k-1} *for* $k \in \underline{m}$.[1] *The* length *of this path is* m*; if* $m = 0$*, the path is* empty. *A* cycle *is a path like above where* $a_0 = a_m$.

If $a \in N$ *is a node of a graph* $d = \langle N, l, s \rangle$*, then by* $d|a$ *we denote the full subgraph of* d *containing all the nodes reachable from* n *via a path. For a graph* d *we often denote its components by* $N(d)$*,* l_d *and* s_d*, respectively. Moreover,* $N_\emptyset(d)$ *and* $N_\Sigma(d)$ *denote the set of empty and non-empty nodes of* d*, respectively (thus* $N(d) = N_\Sigma(d) \uplus N_\emptyset(d)$*, for* $\uplus$ *disjoint union).*

Let d *and* d' *be two graphs. A* (graph) morphism $f : d \to d'$ *is a function* $f : N(d) \to N(d')$ *that preserves labeling and successors. A morphism* $f : d \to d'$ *is* Σ-injective *if its restriction to* $N_\Sigma(d)$ *is injective. Graphs over* Σ *and graph morphisms form a category denoted* $\mathbf{G}_\Sigma$. □

Definition 2 (ranked graphs and term graphs). *An* (i, j)-ranked graph *(or also, a* graph of rank (i, j)*) is a triple* $g = \langle r, d, v \rangle$*, where* d *is a graph with at least* j *empty nodes,* $r : \underline{i} \to N(d)$ *is a function called the* root mapping*, and* $v : \underline{j} \to N_\emptyset(d)$ *is an injective function from* $\underline{j}$ *to 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 graphs* $g = \langle r, d, v \rangle$ *and* $g' = \langle r', d', v' \rangle$ *are* isomorphic *if there exists a* ranked graph isomorphism $\phi : g \to g'$*, i.e., a graph 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 graphs. We shall often write* G^i_j *to recall that* G *has rank* (i, j). □

We introduce now three operations on ranked term graphs which provide them with a traced monoidal categorical structure that will be made explicit in Section 4.

Definition 3 (operations on ranked term graphs). Composition. *Let* ${G'}^j_k = [\langle r', d', v' \rangle]$ *and* $G^i_j = [\langle r, d, v \rangle]$ *be two term graphs. Their* composition *is the term graph* ${G'}^j_k ; G^i_j$ *of rank* (i, k)*, defined as* $[\langle in_d \circ r, d'', in_{d'} \circ v' \rangle]$*, where* $\langle d'', in_d : d \to d'', in_{d'} : d' \to d'' \rangle$ *is the pushout of* $\langle v : \underline{j} \to d, r' : \underline{j} \to d' \rangle$ *in category* $\mathbf{G}_\Sigma$ *(set* $\underline{j}$ *is regarded as a discrete graph).*[2]

Union. *The* union *or* parallel composition *of two term graphs* $G^i_j = [\langle r, d, v \rangle]$ *and* ${G'}^k_l = [\langle r', d', v' \rangle]$ *is the term graph* $G^i_j \oplus {G'}^k_l$ *of rank* $(i + k, j + l)$*, defined as* $[\langle r'', d \uplus d', v'' \rangle]$*, where* $r'' : \underline{i + k} \to d \uplus d'$ *is defined as* $r''(x) = r(x)$ *if* $x \in \underline{i}$*, and* $r''(x) = r'(x - i)$ *if* $x \in \{i + 1, \ldots, i + k\}$*; and* $v'' : \underline{j + l} \to d \uplus d'$ *is defined similarly.*

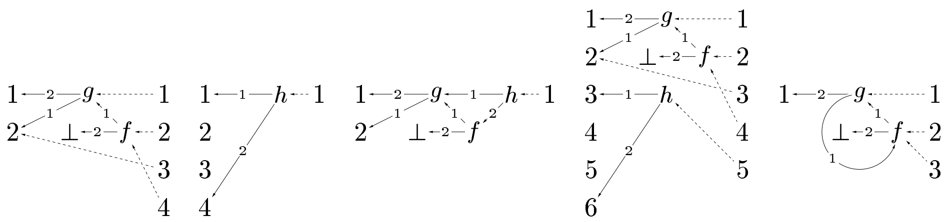
Feedback. *The* feedback *over a term graph* $G = [\langle r, d, v \rangle]$ *of rank* $(i + 1, j + 1)$ *is the term graph* $G^\dagger = [\langle r', d', v' \rangle]$ *of rank* (i, j) *such that* $r'(x) = r(x)$ *and*

[1] For each $i \in \mathbb{N}$ we denote by $\underline{i}$ the set $\underline{i} = \{1, \ldots, i\}$ (thus $\underline{0} = \emptyset$).

[2] Actually, the pushout of two arrows in $\mathbf{G}_\Sigma$ does not always exist: it does 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 [10] for necessary and sufficient conditions for their existence in the equivalent category of *jungles*.

$v'(y) = v'(y)$ for $x \in \underline{i}$ and $y \in \underline{j}$, respectively, and $d' = \langle N(d)/_{\approx}, l', s' \rangle$, where $\approx$ is the least equivalence relation such that $v(j+1) \approx r(i+1)$, and l' and s' are determined by l_d and s_d, respectively. We shall use the following notation for repeated applications of feedback: $G^{\dagger^0}$ denotes G, and $G^{\dagger^{n+1}}$ denotes $(G^{\dagger^n})^\dagger$ □

Example 1 (term graphs, composition, union and feedback). In the term graphs shown below variable 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; empty nodes which are not variables are drawn as $\perp$; non-empty nodes are represented by their label, 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 $(4, 2)$, five nodes (three empty (1, 2, and $\perp$) and two non-empty, f and g), the successors of g are the variables 2 and 1 (in this order), the successors of f are g and $\perp$, and the four roots are g, f, 2, and f. Composition of term graphs can be performed by matching the roots of the first graph with the variables of the second one, and then by eliminating them, as shown by $G_1 \,;\, G_2$. The last two term graphs show the result of union and feedback. □



G_1 $\qquad$ G_2 $\qquad$ $G_1 \,;\, G_2$ $\qquad$ $G_1 \oplus G_2$ $\qquad$ $G_1^\dagger$

It is possible to show that every ranked term graph can be obtained as the value of an expression containing only suitable *atomic* term graphs as constants, and composition, union and feedback as operators (it is implicit in Theorem 14). We introduce now the basics of term graph rewriting.

Definition 4 (rule, redex, term graph rewriting system). *A* rule *(over Σ) is a triple $\mathbf{R} = \langle d, r_l, r_r \rangle$, where d is an* acyclic *graph (over Σ), and r_l, r_r are two distinguished nodes of d, called the* left root *and the* right root*, respectively. Furthermore, we require that (1) node r_l is not empty, (2) all nodes of d are reachable from r_l or r_r, and (3) all empty nodes in d are reachable from r_l. A* redex *in a graph d_0 is a pair $\Delta = \langle \mathbf{R}, f \rangle$ where $\mathbf{R}$ is a rule, and $f : d|r_l \to d_0$ is a graph morphism Node $f(r_l)$ of d_0 is called the* root *of the redex Δ. Redex $\langle \mathbf{R}, f \rangle$ is* Σ-injective *if so is f. A* (term graph) rewriting system *is a pair $\mathcal{R} = \langle \{\mathbf{R}_i\}_{i \in I}, \Sigma \rangle$ such that for all $i \in I$, $\mathbf{R}_i$ is a rule over Σ.* □

Definition 5 (term graph rewriting). *Let $g_0 = \langle r_0, d_0, v_0 \rangle$ be an (i, j)-ranked graph, and let $\Delta = \langle \langle d, r_l, r_r \rangle, f : d|r_l \to d_0 \rangle$ be a redex. We say that g_0* rewrites to g_4 via Δ*, denoted as $g_0 \to_\Delta g_4$, if g_4 is the (i, j)-ranked graph obtained through the following construction*

[Root undefine phase] *Let d_1 be the graph obtained from d_0 by making the root of the redex empty; that is, l_{d_1} and s_{d_1} are undefined on $f(r_l)$, and agree with l_{d_0} and s_{d_0} elsewhere.*

[Build phase] *Define d_2 as the graph obtained by adding to d_1 (a copy of) the part of the rule graph d not reachable from r_l. Thus $N(d_2) = N(d_1) \uplus (N(d) - N(d|r_l))$; and l_{d_2} and s_{d_2} are determined in the expected way from l_{d_1}, l_d, and s_{d_1}, s_d, respectively.*

Let $\hat{f} : N(d) \to N(d_2)$ be the obvious extension of function $f : N(d|r_l) \to N(d_0)$.

[Redirection phase] *Define d_3 as the graph obtained by replacing all references to the root of the redex by references to the image of the right root. More precisely, $d_3 = \langle N(d_2), l_{d_2}, s_{d_3} \rangle$, where $s_{d_3}(a)_i = s_{d_2}(a)_i$ if $s_{d_2}(a)_i \neq f(r_l)$, and $s_{d_3}(a)_i = \hat{f}(r_r)$ if $s_{d_2}(a)_i = f(r_l)$.*

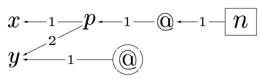
[Root removal phase] *Let d_4 be the graph obtained from d_3 by removing node $f(r_l)$—unless $f(r_l) = \hat{f}(r_r)$, in which case $d_4 = d_3$—and let the* track function *$tr_\Delta : N(d_0) \to N(d_4)$ be defined as $tr_\Delta(a) = \hat{f}(r_r)$ if $a = f(r_l)$, and $tr_\Delta(a) = a$ otherwise.*

Then the ranked graph g_4 is defined as $\langle tr_\Delta \circ r_0, d_4, tr_\Delta \circ v_0 \rangle$.

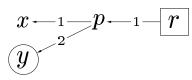
We will say that $\Delta = \langle \mathbf{R}, f \rangle$ is a redex in a term graph G if it is a redex in one of its elements, say $g = \langle r, d, v \rangle$. If $g \to_\Delta g'$, then we write $G \to_\Delta G'$, where $G' = [g']$.

A term graph rewriting sequence *$G_0 \to^*_{\mathcal{R}} G_n$ over a rewriting system $\mathcal{R}$ is a sequence of $n \geq 0$ rewriting steps $G_0 \to_{\Delta_1} G_1 \ldots G_{n-1} \to_{\Delta_n} G_n$, where at each step a rule of $\mathcal{R}$ is used. A rewriting sequence is* Σ-injective *if so are all redexes in it.* □

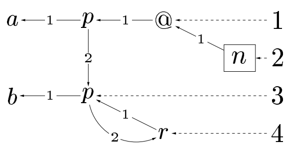
Example 2 (rewriting sequence). Let Σ be the signature containing the constants $\{a, b, \emptyset\}$, the unary operators $\{@, n, l, r\}$ and the binary operator p. Term graphs over Σ are intended to represent manipulations on lists of a's and b's, where $\emptyset$ is the empty list, *p(air)* is the pairing operator, *l(eft)* and *r(ight)* are the obvious selectors, @ is a pointer and *n(ext)* moves a pointer. The figure below shows first the term graphs representing rule $\mathbf{R_1} : n(@(p(x, y))) \to @(y)$ and the collapsing rule $\mathbf{R_2} : r(p(x, y)) \to y$; then the rewriting sequence $G_0 \Rightarrow_{\mathbf{R_1}} G_3 \Rightarrow_{\mathbf{R_2}} G_4$ is shown, where also some intermediate states are depicted. A nodes enclosed in a square is the left root (in the rules) or the root of the redex (in the other graphs), while a node enclosed in a circle is the right root (in the rules) or its image. Term graphs G_1 and G_2 are the results of the *root undefine* and *build* phase, respectively, of the rewriting $G_0 \Rightarrow_{\mathbf{R_1}} G_3$. Note furthermore that G_4 is obtained form G_3 via *root undefine*, and the next phases of rewriting $G_3 \Rightarrow_{\mathbf{R_2}} G_4$ leave it unchanged because the images of the left and right root coincide. □



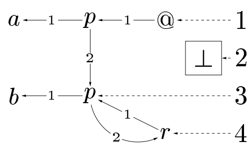
$\mathbf{R_1}$



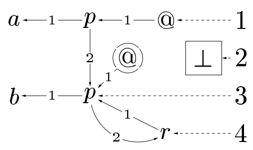
$\mathbf{R_2}$



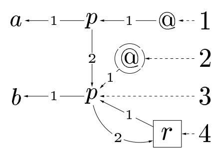
G_0



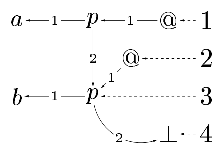
G_1



G_2



G_3



G_4

3 Traced GS-Monoidal 2-Categories

In this section we first recall some basic definitions about 2-categories [16, 15], then we introduce *traced gs-monoidal 2-categories*, to be used in our characterization of term graph rewriting.

Definition 6 (2-categories). *A* 2-category $\underline{\mathbf{C}}$ *is a structure* $\langle Ob_c, \mathbf{C}, *, id\rangle$ *such that* Ob_c *is a set of* 2-objects *and, indexed by elements in* Ob_c, $\mathbf{C}$ *is a family of categories* $\mathbf{C}[a, b]$ *(the* hom-categories *of* $\underline{\mathbf{C}}$*),* $*$ *is a family of functors* $*^b_{a,c}$: $\mathbf{C}[a, b] \times \mathbf{C}[b, c] \to \mathbf{C}[a, c]$ *and* id *is a family of objects* $id_a \in |\mathbf{C}[a, a]|$*, satisfying for each* $\alpha \in \mathbf{C}[a, b]$

[category] $\qquad id_a * \alpha = \alpha = \alpha * id_b \qquad (\alpha * \beta) * \gamma = \alpha * (\beta * \gamma)$

where for the sake of readability the indexes of $*$ *are dropped, and the arrow* id_{id_a} *corresponding to the identity on the object* id_a *is denoted by the object itself.* □

The *category* axioms impose a categorical structure to the pair $\mathbf{C}_u = \langle Ob_c, |\mathbf{C}[a, b]|\rangle$ (denoted as the *underlying category* of $\underline{\mathbf{C}}$), where the objects are the elements of the set of 2-objects, and the elements of the hom-set $\mathbf{C}_u[a, b]$ are the objects of the hom-categories $\mathbf{C}[a, b]$. In fact, we denote as *arrows* and *cells* of the 2-category $\underline{\mathbf{C}}$ the objects and arrows of the hom-categories, respectively: By $\alpha : f \Rightarrow g : a \to b$ we mean that α is a cell in $\mathbf{C}[a, b]$ from f to g.

For the sake of brevity, we present all definitions directly at the 2-categorical level: Since each category can be seen as a 2-category whose hom-categories are discrete (i.e., the only cells are identities), the analogous definitions for (normal) categories are easily obtained as a special case.

Definition 7 (2-functors and (natural) 2-transformations). *Let $\underline{\mathbf{C}}$, $\underline{\mathbf{D}}$ be 2-categories. A* 2-functor $F : \underline{\mathbf{C}} \to \underline{\mathbf{D}}$ *is a pair $\langle F_o, F_m \rangle$, where $F_o : Ob_c \to Ob_d$ is a function and F_m is a family of functors $F_{a,b} : \mathbf{C}[a,b] \to \mathbf{C}[F_o(a), F_o(b)]$.*

Let $F, G : \underline{\mathbf{C}} \to \underline{\mathbf{D}}$ be 2-functors. A 2-transformation $\eta : F \Rightarrow G$ *is a family of arrows of $\underline{\mathbf{D}}$, indexed by objects of $\underline{\mathbf{C}}$, such that $\eta_a \in |\mathbf{D}[F_o(a), G_o(a)]|$; it is* natural *if moreover it verifies $F_C(\alpha) * \eta_b = \eta_a * G_C(\alpha)$ for every cell $\alpha \in \mathbf{C}[a,b]$.* □

Definition 8 (traced gs-monoidal 2-categories). *A* traced gs-monoidal 2-category *is a structure $\underline{\mathbf{C}} = \langle \underline{\mathbf{C}}_0, \otimes, e, \rho, \nabla, !, tr \rangle$, where $\underline{\mathbf{C}}_0$ is a 2-category and*

- *e is a distinguished object in $\underline{\mathbf{C}}_0$ and $\otimes : \underline{\mathbf{C}}_0 \times \underline{\mathbf{C}}_0 \to \underline{\mathbf{C}}_0$ is a 2-functor, satisfying*
  [monoid] $id_e \otimes \alpha = \alpha = \alpha \otimes id_e \qquad (\alpha \otimes \beta) \otimes \gamma = \alpha \otimes (\beta \otimes \gamma)$
- *$\rho : \otimes \Rightarrow \otimes \circ \mathrm{X} : \underline{\mathbf{C}}_0 \times \underline{\mathbf{C}}_0 \to \underline{\mathbf{C}}_0$ is a natural 2-transformation (where* X *is the 2-functor that swaps its two arguments), such that $\rho_{e,e} = id_e$ and satisfying*
  [symmetry] $(id_a \otimes \rho_{b,c}) * (\rho_{a,c} \otimes id_b) = \rho_{a \otimes b, c} \qquad \rho_{a,b} * \rho_{b,a} = id_{a \otimes b}$
- *$\nabla : Id_{\underline{\mathbf{C}}_0} \Rightarrow \otimes \circ \Delta : \underline{\mathbf{C}}_0 \to \underline{\mathbf{C}}_0$ and $! : Id_{\underline{\mathbf{C}}_0} \Rightarrow e : \underline{\mathbf{C}}_0 \to \underline{\mathbf{C}}_0$ are two 2-transformations (Δ is the diagonal 2-functor), such that $!_e = \nabla_e = id_e$ and satisfying*
  [duplication] $\nabla_a * (id_a \otimes \nabla_a) = \nabla_a * (\nabla_a \otimes id_a) \qquad \nabla_a * (id_a \otimes !_a) = id_a \qquad \nabla_a * \rho_{a,a} = \nabla_a$
  [monoidality] $\nabla_{a \otimes b} * (id_a \otimes \rho_{b,a} \otimes id_b) = \nabla_a \otimes \nabla_b \qquad !_{a \otimes b} = !_a \otimes !_b$
- *$tr^u_{a,b} : \mathbf{C}[a \otimes u, b \otimes u] \to \mathbf{C}[a,b]$ is a family of functors indexed by objects in $\underline{\mathbf{C}}_0$, satisfying*
  [naturality] $tr^u_{a,b}(\alpha * (id_b \otimes \beta)) = tr^u_{a,b}((id_a \otimes \beta) * \alpha)$
  $tr^u_{c,d}((\alpha \otimes id_u) * \beta * (\gamma \otimes id_u)) = \alpha * tr^u_{a,b}(\beta) * \gamma$
  [vanishing] $tr^e_{a,b}(\alpha) = \alpha \qquad tr^u_{a \otimes v, b \otimes v}(tr^v_{a,b}(\alpha)) = tr^{v \otimes u}_{a,b}(\alpha)$
  [superposing] $tr^u_{a \otimes c, b \otimes d}((id_a \otimes \rho_{c,u}) * (\alpha \otimes \beta) * (id_b \otimes \rho_{u,d})) = tr^u_{a,b}(\alpha) \otimes \beta$
  [yanking] $tr^u_{u,u}(\rho_{u,u}) = id_u$.

A traced gs-monoidal 2-functor $F : \underline{\mathbf{C}} \to \underline{\mathbf{C}}'$ *is a 2-functor which preserves all the relevant structure "on the nose", i.e., such that $F(e) = e'$, $F(\alpha \otimes \beta) = F(\alpha) \otimes' F(\beta)$, and so on.*

We denote as **TGS-2Cat** *the category of small traced gs-monoidal 2-categories and corresponding 2-functors, and as* **TGS-Cat** *its full subcategory including only small categories.* □

Various substructures of such 2-categories have benn studied in literature. For example, structures as $\langle \underline{\mathbf{C}}_0, \otimes, e, \rho \rangle$ are *symmetric strict monoidal 2-categories*; those as $\langle \underline{\mathbf{C}}_0, \otimes, e, \rho, \nabla, ! \rangle$ are *gs-monoidal 2-categories* [7] (and we will denote **GS-2Cat** and **GS-Cat** the corresponding category of such small 2-categories and categories, respectively); and those as $\langle \underline{\mathbf{C}}_0, \otimes, e, \rho, tr \rangle$ are *traced 2-categories* (the obvious 2-categorical version of those in [13]). We refer the reader to those papers for an intuitive explanation of the axioms listed above, and we sketch now how to generate from a signature a free traced gs-monoidal category, called its *traced gs-monoidal theory*.

Definition 9 (polyadic signatures). *A* polyadic 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 polyadic 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. Polyadic signatures and their morphisms form a category, denoted* **PS**. $\square$

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

Proposition 1 (free theories). *Let* V_t : **TGS-Cat** $\to$ **GS-Cat** *be the functor mapping a traced gs-monoidal category to the underlying gs-monoidal one (forgetting the traced structure), and let* V_{gs} : **GS-Cat** $\to$ **PS** *be the functor mapping a gs-monoidal category to the underlying polyadic signature (forgetting all the structure but the monoidal structure on objects). Both functors admit a left (free) adjoint, denoted* GS : **PS** $\to$ **GS-Cat** *and* T : **GS-Cat** $\to$ **TGS-Cat***, respectively.*

The gs-monoidal theory **GS-Th***(*Σ*) of a given (one-sorted) signature* Σ *is defined as* $GS(\Sigma)$*; its* traced gs-monoidal theory **TGS-Th***(*Σ*) as* $T(GS(\Sigma))$.

$\square$

A *computad* [30] is a category equipped with a graph structure over hom-sets (informally, a set of cells not closed under composition), which intuitively represent a rule based system, the states being the arrows and the rules the cells. The main fact for our analysis is that from a computad a (structured) 2-category is generated in a free way, by closing the cells under all relevant operations.

Definition 10 (computads). *A* gs-monoidal computad $\mathbf{C}_S$ *is a pair* $\langle \mathbf{C}, S\rangle$*, where* $\mathbf{C}$ *is a gs-monoidal category and* S *is a set of "cells" over the hom-sets of* $\mathbf{C}$*. A* gs-monoidal c-morphism $\langle F, h\rangle : \mathbf{C}_S \to \mathbf{D}_T$ *is a pair such that* $F : \mathbf{C} \to \mathbf{D}$ *is a gs-monoidal functor and* $h : S \to T$ *is a function, preserving source and target of the cells in the expected way. The gs-monoidal computads and their morphisms form a category, denoted* **GS-Comp**. $\square$

Proposition 2 (free traced gs-monoidal 2-categories). *Let* V_2 : **TGS-2Cat** $\to$ **GS-Comp** *be the forgetful functor mapping a traced gs-monoidal 2-category to the underlying gs-monoidal computad (forgetting cell composition and trace structure): It admits a left adjoint* TGS_2. $\square$

Intuitively, TGS_2 composes the cells of a computad in all the possible ways, both horizontally and vertically, imposing further equalities in order to satisfy the axioms of a traced 2-category and preserving the gs-monoidal structure on the underlying category.

4 Rewriting Sequences as Cells of a 2-Theory

This section presents the main result of the paper. We first show how to generate in a free way a traced gs-monoidal 2-category from a given rewriting system $\mathcal{R}$; next we prove that the cells of this category faithfully correspond to Σ-injective rewriting sequences using the rules of $\mathcal{R}$.

Part of the construction of the mentioned traced gs-monoidal 2-category is based on the fact that term graphs over Σ can be equipped with a traced gs-monoidal structure, and actually there is an isomorphism between the resulting category and the traced gs-monoidal theory of Σ (see Proposition 1). This result is derivable from recent works on *flownomial algebras* [28, 29]; it is also closely related to [14] and [11], even if the following formulation is, to our knowledge, original (and generalizes the correspondence between *acyclic* term graphs and gs-monoidal theories of [8]).

Definition 11 (the category of term graphs). *For a given signature Σ, $\mathbf{TG}_\Sigma$ denotes the structure $\langle TG_\Sigma, \oplus, \underline{0}, G_{\Pi(\_,\_)}, G_{\nabla(\_)}, G_!^{\_}, tr\rangle$, where*

- *TG_Σ is 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 term graph composition (Definition 3), and the identity on $\underline{i}$ is the discrete term graph $G_{id}^{\underline{i}}$ of rank (i,i) where the k-th root is also the k-th variable, for all $k \in \underline{i}$.*
- *Functor $\oplus : \mathbf{TG}_\Sigma \times \mathbf{TG}_\Sigma \to \mathbf{TG}_\Sigma$ is defined on arrows as in Definition 3, and on objects as $\underline{n} \oplus \underline{m} = \underline{n+m}$;*
- *For each $n, m \in \mathbb{N}$, $G_{\Pi(\underline{n},\underline{m})}$ is the discrete term graph of rank $(n+m, n+m)$ with $n+m$ nodes, such that (denoting by r and v the root and variable functions of one of its representatives) $r(x) = v(n+x)$ if $x \leq m$, and $r(x) = v(x-m)$ if $m < x \leq n+m$.*
- *For each $n \in \mathbb{N}$, $G_{\nabla(\underline{n})}$ is the discrete term graph of rank $(2*n, n)$ with n nodes, such that $r(x) = x$ if $x \leq n$, and $r(x) = x - n$ if $n < x \leq 2*n$.*
- *For each n, $G_!^{\underline{n}}$ is the discrete term graph of rank $(0, n)$ with n nodes.*
- *For each $n, m, k \in \mathbb{N}$, the function $tr^{\underline{k}}_{\underline{n},\underline{m}} : TG_\Sigma[\underline{n+k}, \underline{m+k}] \to TG_\Sigma[\underline{n}, \underline{m}]$ maps a term graph G of rank $(n+k, m+k)$ to $G^{\dagger^k}$ (see Definition 3).*

Moreover, $\mathbf{ATG}_\Sigma$ denotes the structure obtained restricting the pair $\langle TG_\Sigma, \oplus\rangle$ to $\langle ATG_\Sigma, \oplus'\rangle$, where term graphs are acyclic *and all empty nodes are variables.* □

Theorem 1 (categories of term graphs as theories). *The structure $\mathbf{ATG}_\Sigma$ introduced in Definition 11 is a gs-monoidal category isomorphic to $\mathbf{GS\text{-}Th}(\Sigma)$ [8]. The structure $\mathbf{TG}_\Sigma$ is also traced, and it is isomorphic to $\mathbf{TGS\text{-}Th}(\Sigma)$.* □

The traced gs-monoidal 2-category generated by a rewriting system over a signature Σ will have $\mathbf{TGS\text{-}Th}(\Sigma)$ as underlying category. The result just presented allows us to use term graphs over Σ to denote the arrows of such a category, and to make free use of its equational presentation when reasoning about them. Now we first show how to associate (in a quite straightforward way) with each rewriting system $\mathcal{R}$ its *tgs-monoidal 2-theory* $\mathbf{TGS\text{-}2Th}(\mathcal{R})$, and then we proceed by relating cells in this 2-category and rewriting sequences over $\mathcal{R}$.

Definition 12 (rule as cell). *Let* $\mathbf{R} = \langle d, r_l, r_r \rangle$ *be a rule, as in Definition 4,* such that there is no path from r_r to r_l.[3] *Let* n *be the number of empty nodes of* d*, and let* m *be the number of nodes of* $d|r_l$*. Then we define the term graphs* $L_{\mathbf{R}}$ *and* $R_{\mathbf{R}}$ *of rank* (m, n) *as follows*

- $L_{\mathbf{R}} = [\langle r, d|r_l, v \rangle]$*, where* $r : \underline{m} \to N(d|r_l)$ *is a bijection, and* $v : \underline{n} \to N_\emptyset(d|r_l)$ *is a bijection between* $\underline{n}$ *and the set of empty nodes of* $d|r_l$*. Note that* r *and* v *are well-defined by the definition of* n *and* m*; moreover they could be fixed in a canonical way, but we don't need this.*
- $R_{\mathbf{R}} = [\langle r', d - \{r_l\}, v \rangle]$*, where* $d - \{r_l\}$ *is the graph obtained by removing from* d *the node* r_l*,* v *is the same as before, and for all* $x \in \underline{m}$*,* $r'(x) = r_r$ *if* $r(x) = r_l$*, and* $r'(x) = r(x)$ *otherwise.*
  Note that $R_{\mathbf{R}}$ *is a well-defined term graph of rank* (m, n) *because, by condition* (2) *of Definition 4 and the above assumption,* r_l *is not the successor of any node in* d*.*

The cell representing rule $\mathbf{R}$ *is the cell* $\alpha(\mathbf{R}) : L_{\mathbf{R}} \Rightarrow R_{\mathbf{R}} : \underline{n} \to \underline{m}$*.* □

Definition 13 (the tgs-monoidal 2-theory of a rewriting system). *Let* $\mathcal{R} = \langle \{\mathbf{R}_i\}_{i \in I}, \Sigma \rangle$ *be a rewriting system, and let* $\mathbf{GS\text{-}Th}(\Sigma)$ *be the gs-monoidal theory of* Σ*. The* gs-monoidal computad representing $\mathcal{R}$ *is defined as the computad* $\mathbf{C}_{\mathcal{R}} = \langle \mathbf{GS\text{-}Th}(\Sigma), \{\alpha(\mathbf{R}_i)\}_{i \in I} \rangle$*, where for each rule* $\mathbf{R}_i$ *in* $\mathcal{R}$ *the cell* $\alpha(\mathbf{R}_i) : L_{\mathbf{R}_i} \Rightarrow R_{\mathbf{R}_i}$ *is as in Definition 12. The* tgs-monoidal 2-theory of $\mathcal{R}$*, denoted by* $\mathbf{TGS\text{-}2Th}(\mathcal{R})$*, is the free traced gs-monoidal 2-category generated by* $\mathbf{C}_{\mathcal{R}}$*, as described in Proposition 2.* □

We need now a few technical definitions and lemmas before presenting the main result relating cells of $\mathbf{TGS\text{-}2Th}(\mathcal{R})$ and rewriting sequences over $\mathcal{R}$.

Definition 14 (term graph contexts). *A* (term graph) context $C[i, j]$ *is a well-formed, ranked term graph expression containing exactly one (ranked) placeholder* $[i, j]$ *(for some* $i, j \in \mathbb{N}$*). We write* $C[i, j] : (n, m)$ *to say that assuming that the place-holder has rank* (i, j)*, then the rank of* $C[i, j]$ *is* (n, m)*; and by* $C[G]$ *we denote the term graph obtained by evaluating the expression* C *after replacing* $[i, j]$ *by a term graph* G *of rank* (i, j)*.* □

Lemma 1 (Σ-injective morphisms and contexts). *Let* G *be an acyclic term graph of rank* (m, n) *such that all its nodes are roots, and* H *be a term graph.*

[3] This restriction is not essential: The general case just requires more care in the representation of a rule as a cell, and will be treated in the full paper. Note that if in $\mathbf{R}$ there is a non-empty path from r_r to r_l, then any application of the rule results in a cyclic graph.

1. *If $C[m,n]$ is a context such that $H = C[G]$, then for every acyclic graph $d_G \in G$ and graph $d_H \in H$ there is a Σ-injective graph morphism $f_C : d_G \to d_H$.*
2. *If d_G is an acyclic graph in G, d_H is a graph in H, and there is a Σ-injective graph morphism $f : d_G \to d_H$, then there is a context $C_f[m,n]$ such that $C_f[G] = H$.* □

Lemma 2 (term graph rewriting as contextualization). *Let $\mathbf{R} = \langle d, r_l, r_r \rangle$ be a rule and $\alpha : L_{\mathbf{R}} \Rightarrow R_{\mathbf{R}} : \underline{n} \to \underline{m}$ the associated cell by Definition 12. Then for every context $C[m,n]$ and for every acyclic graph d_0 in $C[L_{\mathbf{R}}]$ there is a Σ-injective redex $\Delta = \langle \mathbf{R}, f : d|r_l \to d_0 \rangle$ such that $C[L_{\mathbf{R}}] \to_{\Delta} C[R_{\mathbf{R}}]$.*

Proof. Let $f : d|r_l \to d_0$ be the Σ-injective graph morphism whose existence is ensured by Lemma 1. The fact that $C[L_{\mathbf{R}}] \to_{\langle \mathbf{R}, f \rangle} C[R_{\mathbf{R}}]$ can be shown by comparing the construction in Definition 5 with the definition of $L_{\mathbf{R}}$ and $R_{\mathbf{R}}$. □

Theorem 2 (rewriting sequences as cells of the traced gs-monoidal 2-theory). *Let G and H be two term graphs having the same rank (m,n). Then there is a cell $\alpha : G \Rightarrow H : \underline{n} \to \underline{m}$ in* **TGS-2Th**$(\mathcal{R})$ *if and only if there is a Σ-injective rewriting sequence $G \to_{\mathcal{R}}^{*} H$.*

Proof (outline). Only if part. Note that cells in **TGS-2Th**$(\mathcal{R})$ are generated (according to Proposition 2) from the cells of computad $\mathbf{C}_{\mathcal{R}}$ (the *2-generators*) by closing them with respect to all the operations of traced gs-monoidal categories, and by imposing the corresponding axioms. By structural induction, any cell α can then be proved decomposable into an equivalent cell $\alpha_1 \cdot \alpha_2 \cdot \ldots \cdot \alpha_k$ such that each α_i contains exactly one 2-generator. Let us restrict our attention to a single cell $\alpha : G \Rightarrow H$ containing only one 2-generator: It can be decomposed as $C[\alpha(\mathbf{R})]$ for a suitable context C and a rule $\mathbf{R} \in \mathcal{R}$. By the definition of $\alpha(\mathbf{R})$, we have that $G = C[L_{\mathbf{R}}]$ and $H = C[R_{\mathbf{R}}]$, and by Lemma 2 there is a Σ-injective redex Δ such that $G \to_{\Delta} H$. The statement follows.
If Part. If $G \to_{\Delta} H$ using a Σ-injective redex $\langle \mathbf{R}, f \rangle$ with $\mathbf{R} \in \mathcal{R}$, then by Lemma 1 there is a context C_f such that $G = C_f[L_{\mathbf{R}}]$; and moreover, $H = C_f[R_{\mathbf{R}}]$ by Lemma 2. By Definitions 12 and 13 the cell $\alpha(\mathbf{R}) : L_{\mathbf{R}} \Rightarrow R_{\mathbf{R}}$ is a 2-generator of **GS-2Th**$(\mathcal{R})$, and thus (by the generation rules for cells) $C_f[\alpha(\mathbf{R})] : C_f[L_{\mathbf{R}}] \Rightarrow C_f[R_{\mathbf{R}}]$ is a cell. The statement follows. □

5 Discussion and Conclusions

In this section we first discuss the relationship between our definition of term graph rewriting and that originally proposed in [3], and then we comment on the main result of the paper, sketching some possible extensions and related future work.

With respect to the corresponding definition in [3], our definition of term graph rewriting (Definition 5) differs for two main aspects: the absence of a final *garbage collection phase*, removing from the resulting term graph all the non-variable nodes not reachable from the roots; and the presence of an initial phase (*root undefine*) where we make the root of the redex undefined, i.e., we transform it in an empty node. We discuss these two facts in turn.

The absence of garbage collection in our definition is due to the fact that, on the categorical side, the 2-transformation "!" is not required to be natural in (traced) gs-monoidal categories: As argued in [8], the naturality of ! is indeed the categorical counterpart of *automatic garbage collection*. More explicitly, by naturality we would have $g{*}!_{\underline{1}} = !_{\underline{n}} : \underline{n} \to \underline{0}$ for an n-ary operator g, meaning that a term graph with a single garbage node g is "the same" as the term graph obtained by removing it. Also, we would have for each term graph G of rank $(1, n+1)$, $tr^{\underline{1}}_{\underline{n},\underline{0}}(G){*}!_{\underline{0}} = tr^{\underline{1}}_{\underline{n},\underline{0}}(G) = !_{\underline{n}}$ (because $!_{\underline{0}} = id_{\underline{0}}$), meaning that all garbage nodes can be safely removed at once from a cyclic term graph without roots. As a consequence, we easily obtain a categorical representation (analogous to Theorem 1) of "term graphs without garbage nodes", and a correspondence between Σ-injective rewrites *with* garbage collection and cells of the 2-theory, as in Theorem 2, simply by requiring the naturality of ! in the definition of traced gs-monoidal 2-category.

The categorical framework makes easy also to consider an alternative solution, consisting of adding a set of "garbage collection rules" which incrementally delete the garbage nodes produced during the rewriting. For this, it is sufficient to require that "!" is an *op-lax natural* transformation.

Definition 15 (lax, op-lax, and pseudo-natural 2-transformations [16]). *Let $\underline{\mathbf{C}}$ be a 2-category. We denote with $\underline{\mathbf{C}}^{co}$ the dual 2-category with reversed cells, namely $\mathbf{C}^{co}[a,b] = (\mathbf{C}[a,b])^{op}$, with the obvious extension to 2-functors and 2-transformations.*

Let $F, G : \underline{\mathbf{C}} \to \underline{\mathbf{D}}$ be 2-functors. A 2-transformation $\eta : F \Rightarrow G$ is lax natural *if it comes equipped with a family of cells of $\underline{\mathbf{D}}$, indexed by arrows $f \in \underline{\mathbf{C}}[a,b]$, such that $\eta_f : \eta_a * G_{a,b}(f) \Rightarrow F_{a,b}(f) * \eta_b \in \mathbf{D}[F_o(a), G_o(b)]$; it is* op-lax natural *if the dual $\eta^{co} : F^{co} \Rightarrow G^{co}$ is lax natural; it is* pseudo-natural *if it is both lax and op-lax natural, and for each arrow f the associated cells η_f and η^{co}_f form an isomorphic pair.* □

Theorem 3 (rewriting sequences with garbage collection as cells). *Let $\mathcal{R} = \langle \{\mathbf{R}_i\}_{i \in I}, \Sigma \rangle$ be a rewriting system, and let $\mathbf{TGS\text{-}2Th}_{gc}(\mathcal{R})$ be the free traced gs-monoidal 2-category generated by the gs-computad $\langle \mathbf{GS\text{-}Th}(\Sigma), S \rangle$, requiring "!" to be an* op-lax natural *transformation.*

Now let G and H be two term graphs with no garbage nodes. Then there is a cell $\alpha : G \Rightarrow H$ in $\mathbf{TGS\text{-}2Th}_{gc}(\mathcal{R})$ if and only if there is a Σ-injective rewriting sequence $G \to^*_{gc,\mathcal{R}} H$, where the subscript gc *means that we add after the* root removal *phase a* garbage collection *one.* □

This result follows easily from Theorem 2, by observing that the additional rewrites can only be applied to garbage nodes. The op-lax naturality breaks down to the presence of cells $g{*}!_{\underline{1}} \Rightarrow !_{\underline{n}}$ and $tr^{\underline{1}}_{\underline{n},\underline{0}}(G) \Rightarrow !_{\underline{n}}$ for each n-ary operator g and each graph G of rank $(1, n+1)$. Therefore in this formulation garbage collection is executed concurrently with other rewrites, without affecting the correctness of the system.

As far as the presence of the initial *root undefine phase* in Definition 5 is concerned, it is easy to check that it affects the result of the rewriting only in the case of *circular redexes*, that is, when the image of the left and of the right roots of the rule coincide. For Σ-injective redexes, this can only happen with *collapsing rules*, i.e., those where the right root is a variable.

The paradigmatical example is the *circular* I, i.e., the application of the identity rule $\mathbf{R_I} : I(x) \to x$ to the cyclic graph having one node labelled I and one looping edge. Using the definition in [3] (or ours without the first phase), this redex reduces to itself, while according to our construction it reduces to an empty node which is not a variable (another example of reduction of a circular redex is the rewriting $G_3 \Rightarrow_{\mathbf{R_2}} G_4$ at the end of Section 2). This result is consistent with the categorical model: According to Definition 12, the cell representing $\mathbf{R_I}$ is $\alpha(\mathbf{R_I}) = (\nabla_{\underline{1}} * (I \otimes id_{\underline{1}})) \Rightarrow \nabla_{\underline{1}} : \underline{2} \to \underline{1}$, and since cells are closed under traces, there is a cell $tr^{\underline{1}}_{\underline{0},\underline{1}}(\alpha(\mathbf{R_I})) = tr^{\underline{1}}_{\underline{0},\underline{1}}(\nabla_{\underline{1}} * (I \otimes id_{\underline{1}})) \Rightarrow tr^{\underline{1}}_{\underline{0},\underline{1}}(\nabla_{\underline{1}}) : \underline{1} \to \underline{0}$, which goes from the arrow representing the "I-loop" to that representing $\perp$.

It is worth mentioning that other definitions of term graph rewriting proposed in the literature agree with ours on circular redexes (see [17, 1, 6][4]), and that many authors already discussed the disagreement of various definitions of term graph rewriting in the handling of circular redexes. However, to our knowledge it is the first time that the correctness of the proposed construction is verified against a more declarative, categorical counterpart. Informally, we can say that a circular redex reduces to $\perp$ because the categorical structure forces the rewrite relation to be a *pre-congruence*, that is, to be closed with respect to the all operators, including feedback.

We discuss now some possible extensions and applications of our main result.

Remember from Theorem 2 that cells in the 2-theory of $\mathcal{R}$ only represent Σ-injective rewriting sequences. This restriction can be relaxed by adding to the system explicit rules which change locally the degree of sharing in a term graph (for the acyclic case, such rules have been considered e.g. in [12, 2]). Categorically, this is obtained requiring the transformation "∇" to be *pseudo-natural*. Then, for each n-ary operator f, the cell $\nabla_f : f * \nabla_{\underline{1}} \Rightarrow \nabla_{\underline{n}} * (f \otimes f)$ allows one to create two copies of f if it is shared, while the inverse $\nabla_f^{co} : \nabla_{\underline{n}} * (f \otimes f) \Rightarrow f * \nabla_{\underline{1}}$ merges together two copies of f if they share all their successors. Furthermore, derived cells allow one to "unfold" or "fold" cyclic structures. For example, if g is of arity 1, then $tr^{\underline{1}}_{\underline{0},\underline{1}}(g * \nabla_{\underline{1}})$ represents the term graph having one node labeled by g, which is the only root, and one looping edge (a g-loop). By pseudo-naturality of ∇ and since cells are closed under traces, there is a cell $tr^{\underline{1}}_{\underline{0},\underline{1}}(\nabla_g) : tr^{\underline{1}}_{\underline{0},\underline{1}}(g * \nabla_{\underline{1}}) \Rightarrow$

[4] Even if equivalent to ours, we refrained from using the definition of term graph rewriting of [6] (based on the "Double-Pushout Approach" to graph rewriting) in order to ease the comparison with [3].

$tr^{\underline{1}}_{\underline{0},\underline{1}}(\nabla_{\underline{1}} * (g \otimes g))$, and this last term represents the graph containing a g-node pointing to a g-loop, i.e., a partial unfolding of the g-loop.

Theorem 4 (rewriting sequences with unsharing as cells). *Let $\mathcal{R} = \langle \{\mathbf{R}_i\}_{i \in I}, \Sigma \rangle$ be a rewriting system, and let $\mathbf{TGS\text{-}2Th}_s(\mathcal{R})$ be the free traced gs-monoidal 2-category generated by the gs-computad $\langle \mathbf{GS\text{-}Th}(\Sigma), S \rangle$, requiring "$\nabla$" to be a* pseudo-natural *transformation.*

Now let G and H be two term graphs such that there is a rewriting sequence using non-self-overlapping *redexes*[5] *$G \to^*_{\mathcal{R}} H$: Then there exists a cell $\alpha : G \Rightarrow H$ in $\mathbf{TGS\text{-}2Th}_s(\mathcal{R})$.* □

In the extended 2-theory, $\mathbf{TGS\text{-}2Th}_s(\mathcal{R})$ a rewriting step via a non-Σ-injective redex can be simulated by a cell composed of a sequence of unsharing/unfolding cells (that transform the redex into a Σ-injective one), of the 2-generator corresponding to the rule, and finally of a sequence of sharing/folding cells (that restore the original degree of sharing). For acyclic graphs, the theorem is a corollary of Theorem 7.1 in [2]. The converse of the theorem clearly does not hold.

The categorical presentation can be exploited to relate formalisms via suitable adjunctions. The diagram below summarizes the relationships between traced gs-monoidal 2-categories and other 2-categories used to represent other rewriting formalisms based on terms or term graphs. Arrows represent adjunctions in the direction of the left adjoint. Adjunction (1) essentially models the classical 2-categorical presentation of term rewriting using algebraic 2-theories [26, 24, 20], where a rewriting system is represented by a *cartesian* computad, and its free cartesian 2-category has cells for all rewriting sequences. Adjunction (2) models acyclic term graph rewriting, and it is described in [8], where also the commutativity of the top square of adjuctions is discussed. Adjunction (3) models *rational term rewriting* (RTR), i.e., the extension of standard term rewriting obtained by allowing for possible infinite terms with a finite number of subterms, and for *infinite parallel rewriting*, as originally defined in [5]. The categorical presentation of RTR based on adjunction (3) and on *iteration 2-theories* [4] will appear in the full version of [9], where the operational and logical presentations are introduced. Adjunction (4) models possibly cyclic term graph rewriting, as described in this paper. The vertical adjunctions (7,8) add traced or feedback structure, resulting in possibly cyclic term graphs to the left, and in rational terms to the right. Finally, the horizontal adjunctions (0,5,6) are obtained by enforcing the naturality of ∇ and !.

[5] A redex is *self-overlapping* if the left root and another non-variable node are mapped to the same node; this restriction will be discussed in the full paper.

$$
\begin{tikzcd}
\textit{(acyclic term graphs)} & \mathbf{GS\text{-}2Cat} \arrow[rrr, "5"] \arrow[dd, "7"] & & & \mathbf{C\text{-}2Cat} \arrow[dd, "8"] & \textit{(finite terms)} \\
& & \mathbf{GS\text{-}Comp} \arrow[ul, "2"] \arrow[r, "0"] \arrow[dl, "4"] & \mathbf{C\text{-}Comp} \arrow[ur, "1"] \arrow[dr, "3"] & & \\
\textit{(term graphs)} & \mathbf{TGS\text{-}2Cat} \arrow[rrr, "6"] & & & \mathbf{It\text{-}2Cat} & \textit{(rational terms)}
\end{tikzcd}
$$

In particular, it is possible to show that adjunction (6) maps the traced gs-monoidal theory of a signature $\mathbf{TGS\text{-}Th}(\Sigma)$ to the *iteration theory* $\mathbf{It\text{-}Th}(\Sigma)$, and that the unit of the adjunction maps a term graph G of rank (n, m) to an n-tuple of rational terms over m variables, essentially obtained by "unraveling" G from its roots. It is especially interesting to note that this framework helps us to clarify the rôle of empty non-variable nodes of term graphs, i.e., those resulting from the reduction of a circular redex. We argued in [5, 6] that such a node corresponds to the completely undefined term, i.e., the bottom element of the CPO of possibly infinite, *possibly partial* terms. This turns out to be correct, because rational terms in the iteration theory of Σ are natually endowed with a CPO structure, and the above mentioned unit maps such nodes to the bottom element there (which explains why we denoted them as $\perp$). This intepretation compares favourably with the more *ad hoc* treatment of such nodes in [1], for which a fresh constant $\bullet$ is added to the signature.

Let us conclude by observing that some interesting extensions of term graph rewriting can be modeled without additional effort in the proposed categorical framework, and will be considered in the full paper. These include the generalization to *many-sorted* signatures, to *polyadic* signatures (i.e., where operators return more values instead of exactly one), and to possibly cyclic rules.

References

1. Z.M. Ariola and J.W. Klop. Equational Term Graph Rewriting. *Fundamenta Informaticae*, 26:207–240, 1996.
2. Z.M. Ariola, J.W. Klop, and D. Plump. Confluent rewriting of bisimilar term graphs. In C. Palamidessi and J. Parrow, editors, *Proceedings EXPRESS'97*, volume 7 of *Electronic Notes in Theoretical Computer Science*. Elsevier Sciences, 1997. `http://www.elsevier.nl/locate/entcs/volume7.html`.
3. 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 *Proceedings PARLE*, volume 259 of *LNCS*, pages 141–158. Springer Verlag, 1987.
4. S.L. Bloom, Z. Ésik, A. Labella, and E.G. Manes. Iteration 2-theories. In *Proceedings AMAST'97*, 1997. To appear.
5. A. Corradini. Term rewriting in CT_Σ. In *Proceedings CAAP '93*, volume 668 of *LNCS*, pages 468–484. Springer Verlag, 1993.
6. 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.

7. A. Corradini and F. Gadducci. A 2-categorical presentation of term graph rewriting. In *Proceedings CTCS'97*, volume 1290 of *LNCS*. Springer Verlag, 1997.
8. A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 1998. To appear. Available at `http://www.di.unipi.it/~gadducci/papers/aptg.ps`.
9. A. Corradini and G Gadducci. Rational term rewriting. In M. Nivat, editor, *Proceedings FoSSaCS'98*, volume 1378 of *LNCS*, pages 156–171. Springer Verlag, 1998.
10. A. Corradini and F. Rossi. Hyperedge Replacement Jungle Rewriting for Term Rewriting Systems and Logic Programming. *Theoret. Comput. Sci.*, 109:7–48, 1993.
11. M. Hasegawa. *Models of Sharing Graphs.* PhD thesis, University of Edinburgh, 1997.
12. B. Hoffmann and D. Plump. Implementing Term Rewriting by Jungle Evaluation. *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 25:445–472, 1991.
13. A. Joyal, R. Street, and D. Verity. Traced Monoidal Category. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119:425–446, 1996.
14. P. Katis, N. Sabadini, and R.F.C. Walters. Span(graph): A categorical algebra of transition systems. to appear Proceedings AMAST'97, 1997.
15. G. Kelly. *Basic Concepts of Enriched Category Theory.* Cambridge University Press, 1982.
16. G.M. Kelly and R.H. Street. Review of the elements of 2-categories. In G.M. Kelly, editor, *Proceedings of the Sydney Category Seminar*, volume 420 of *Lecture Notes in Mathematics*, pages 75–103. Springer Verlag, 1974.
17. J.R. Kennaway. On 'On Graph Rewritings'. *Theoret. Comput. Sci.*, 52:37–58, 1980.
18. 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.
19. N. Martí-Oliet and J. Meseguer. From Petri nets to linear logic through categories: A survey. *International Journal of Foundations of Computer Science*, 4:297–399, 1991.
20. J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoret. Comput. Sci.*, 96:73–155, 1992.
21. J. Meseguer and U. Montanari. Petri nets are monoids. *Information and Computation*, 88:105–155, 1990.
22. H. Miyoshi. Rewriting logic for cyclic sharing structures. In M. Sato and Y. Toyama, editors, *Proceedings FLOPS'97*. World Scientific, 1997.
23. M.J. Plasmeijer and M.C.J.D. van Eekelen. *Functional Programming and Parallel Graph Rewriting*. Addison Wesley, 1993.
24. A.J. Power. An abstract formulation for rewrite systems. In *Proceedings Category Theory in Computer Science*, volume 389 of *LNCS*, pages 300–312. Springer Verlag, 1989.
25. W. Reisig. *Petri Nets: An Introduction.* EACTS Monographs on Theoretical Computer Science. Springer Verlag, 1985.
26. D.E. Rydehard and E.G. Stell. Foundations of equational deductions: A categorical treatment of equational proofs and unification algorithms. In *Proceedings Category Theory in Computer Science*, volume 283 of *LNCS*, pages 114–139. Springer Verlag, 1987.
27. M.R. Sleep, M.J. Plasmeijer, and M.C. van Eekelen, editors. *Term Graph Rewriting: Theory and Practice.* Wiley, London, 1993.

28. G. Stefanescu. On flowchart theories: Part II. The nondeterministic case. *Theoret. Comput. Sci.*, 52:307–340, 1987.
29. G. Stefanescu. Algebra of flownomials. Technical Report SFB-Bericht 342/16/94 A, Technical University of München, Institut für Informatik, 1994.
30. R. Street. Categorical structures. In M. Hazewinkel, editor, *Handbook Of Algebra*, pages 529–577. Elsevier, 1996.
31. D. A. Turner. A new implementation tecnique for applicative languages. *Software: Practice and Experience*, 9:31–49, 1979.