

Bisimilarity in Term Graph Rewriting¹

Zena M. Ariola

Computer and Information Science Department, University of Oregon, Eugene, Oregon 97401
E-mail: ariola@cs.uoregon.edu

Jan Willem Klop

*Department of Mathematics and Computer Science, Vrije Universiteit, 1081 HV Amsterdam,
The Netherlands; and Computing Science Institute, Katholieke Universiteit Nijmegen,
Postbus 9010, 6500 GL Nijmegen, The Netherlands*
E-mail: jwk@cs.vu.nl

and

Detlef Plump²

Fachbereich Mathematik und Informatik, Universität Bremen, Postfach 33 04 40, 28334 Bremen, Germany
E-mail: det@informatik.uni-bremen.de

We present a survey of confluence properties of (acyclic) term graph rewriting. Results and counterexamples are given for different kinds of term graph rewriting; besides plain applications of rewrite rules, extensions with the operations of collapsing and copying, and both operations together are considered. Collapsing and copying together constitute bisimilarity of term graphs. We establish sufficient conditions for—and counterexamples to—confluence modulo bisimilarity, and the Church–Rosser property modulo bisimilarity. Moreover, we address rewriting modulo bisimilarity, that is, rewriting of bisimilarity classes of term graphs. © 2000 Academic Press

1. INTRODUCTION

Computations with term rewrite rules play an important role in areas like functional programming, symbolic computation, and theorem proving. Such computations are commonly implemented on graph-like data structures for expressions.

¹ This paper is a revised and extended version of [AKP97].

² Part of the research of this author was performed while he was on leave at CWI, Amsterdam, by a grant of the HCM network EXPRESS.

This makes it possible to *share* common subexpressions, thereby avoiding repeated evaluations of the same subexpression.

Term graph rewriting originates from the demand for a computational model that allows us to reason about implementations with sharing. In this model, rewrite rules operate on graphs rather than on trees. Although term graph rewriting is closely related to term rewriting, the two models differ with respect to important properties like termination and confluence. The reason is that sharing common subexpressions excludes certain rewrite sequences.

In this paper, we consider acyclic term graph rewriting according to the approach of [Plu93b, Plu98]. The definition of rewrite steps in this setting is—as far as acyclic term graphs are concerned—equivalent to the corresponding definitions in [BvEG⁺87, KKSdV94, AK96]. We remark, however, that this equivalence fails for cyclic graphs. In particular, a “collapsing” term rewrite rule like $\text{id}(x) \rightarrow x$ yields, when applied to certain cyclic graphs, different results in the mentioned approaches (see [KKSdV94, CD97]).

We are mainly interested, in this paper, in confluence properties of term graph rewriting. We will address not only rewriting by applications of term rewrite rules, but also extensions with the operations of collapsing and copying, and with both operations together. These operations are important for completeness reasons; while collapsing allows us to cope with term rewrite rules having repeated variables in their left-hand sides, copying allows us to simulate certain term rewriting derivations that are otherwise prevented by sharing. Moreover, collapsing increases the degree of sharing and thus can, in certain cases, considerably speed up evaluation processes.

When collapsing and copying are present together, the (reflexive-transitive closure of the) term graph rewrite relation contains *bisimilarity* of term graphs. We call two term graphs bisimilar if they represent the same term. Equivalently, both graphs collapse to a common term graph or yield a common term graph by copying. We investigate, in addition to confluence, under which conditions term graph rewriting is confluent modulo bisimilarity or even Church–Rosser modulo bisimilarity. Moreover, we characterize confluence and termination of term graph rewriting modulo bisimilarity, that is, of rewriting bisimilarity classes.

The rest of this paper is organized as follows. In Section 2, we introduce term graphs, collapsing, copying, and bisimilarity. Section 3 contains a review of term graph rewriting and motivates the use of collapsing and copying. The relation between confluence, confluence modulo bisimilarity, and the Church–Rosser property modulo bisimilarity is clarified in Section 4. In Section 5, we recall some confluence results for nonoverlapping rewrite rules and show that the full substitution strategy is cofinal. Examples demonstrate that the addition of collapsing or copying causes nonconfluence. Orthogonal rewrite systems are treated in Section 6. It is shown that collapsing may still result in nonconfluence, while plain term graph rewriting is shown to be confluent modulo bisimilarity. Section 7 is devoted to general systems with possibly overlapping rules. We present conditions under which confluence of term rewriting induces confluence of term graph rewriting, or even the Church–Rosser property modulo bisimilarity. In Section 8, rewriting of bisimilarity classes is addressed. Finally, in Section 9, we summarize our positive and negative results in two tables.

2. TERM GRAPHS AND BISIMILARITY

Let Σ be a set of *function symbols*, where each $f \in \Sigma$ comes with a natural number $\text{arity}(f) \geq 0$. Function symbols of arity 0 are called *constants*. We further assume that there is an infinite set X of *variables* such that $X \cap \Sigma = \emptyset$, and we set $\text{arity}(x) = 0$ for each variable x .

A *hypergraph* over Σ and X is a system $G = \langle V_G, E_G, \text{lab}_G, \text{att}_G \rangle$ consisting of two finite sets V_G and E_G of *nodes* (or *vertices*) and *hyperedges*, a labelling function $\text{lab}_G : E_G \rightarrow \Sigma \cup X$, and an attachment function $\text{att}_G : E_G \rightarrow V_G^*$ assigning a string of nodes to a hyperedge e such that the length of $\text{att}_G(e)$ is $1 + \text{arity}(\text{lab}_G(e))$. In the following, we call hypergraphs and hyperedges simply graphs and edges.

Given a graph G and an edge e with $\text{att}_G(e) = v v_1 \cdots v_n$, node v is the *result node* of e while v_1, \dots, v_n are the *argument nodes*. The result node v is denoted by $\text{res}(e)$.

A node v' is *reachable* from a node v if $v = v'$ or there is an edge e such that $v = \text{res}(e)$ and v' is reachable from an argument node of e . A graph is *acyclic* if for each edge, its result node is not reachable from any of its argument nodes. Given a node v in a graph G , we write $G|_v$ for the subgraph consisting of all nodes that are reachable from v and all edges having these nodes as result nodes.

DEFINITION 2.1 (Term graph). A graph G is a *term graph* if

- (1) there is a node root_G from which each node is reachable,
- (2) G is acyclic, and
- (3) each node is the result node of a unique edge.

Figure 1 shows three term graphs with binary function symbols f , g , and h , and a constant a . Edges are depicted as boxes with inscribed labels, and bullets represent nodes. A line connects each edge with its result node, while arrows point to the argument nodes. The order in the argument string is given by the left-to-right order of the arrows leaving the box.

Instead of using hypergraphs, term graphs can alternatively be defined as directed acyclic graphs consisting of a set of labelled nodes, together with a successor function from nodes to strings of nodes (see, for example [BvEG⁺87, KKSdV94]). That kind of definition is equivalent to the present one, since every term graph defined in that way can easily be transformed into a hypergraph conforming to Definition 2.1, and vice versa. In this paper we use the hypergraph framework in order to be consistent with [Plu93a, Plu93b].

A *term* over Σ and X is a variable, a constant, or a string $f(t_1, \dots, t_n)$, where f is a function symbol of arity $n \geq 1$ and t_1, \dots, t_n are terms.

DEFINITION 2.2 (Term representation). A node v in a term graph G represents the term

$$\text{term}_G(v) = \text{lab}_G(e)(\text{term}_G(v_1), \dots, \text{term}_G(v_n)),$$

where e is the unique edge with $\text{res}(e) = v$, and where $\text{att}_G(e) = v v_1 \cdots v_n$. It is understood that if $n = 0$, this means $\text{term}_G(v) = \text{lab}_G(e)$. We denote $\text{term}_G(\text{root}_G)$ also by $\text{term}(G)$.

A *graph morphism* $f: G \rightarrow H$ between two graphs G and H consists of two functions $f_V: V_G \rightarrow V_H$ and $f_E: E_G \rightarrow E_H$ that preserve labels and attachment to nodes, that is, $\text{lab}_H \circ f_E = \text{lab}_G$ and $\text{att}_H \circ f_E = f_V^* \circ \text{att}_G$ (where $f_V^*: V_G^* \rightarrow V_H^*$ maps a string $v_1 \cdots v_n$ to $f_V(v_1) \cdots f_V(v_n)$). The morphism f is *injective* (*surjective*) if f_V and f_E are. If f is injective and surjective, then it is an *isomorphism*. In this case G and H are *isomorphic*.

In the sequel we will not distinguish between isomorphic term graphs, to facilitate the presentation. More precisely, we select from every isomorphism class of term graphs a unique *standard term graph* and tacitly assume that we are dealing with standard term graphs only. Standard term graphs can be constructed by a canonical numbering of nodes, similar to the numbering of positions in terms. Details are in [Plu98].

DEFINITION 2.3 (Collapsing and copying). Given two term graphs G and H , G *collapses* to H if there is a graph morphism $G \rightarrow H$ mapping root_G to root_H . This is denoted by $G \succcurlyeq H$ or, if the morphism is noninjective, by $G \succ H$. The latter kind of collapsing is said to be *proper*. The inverse relation of collapsing is called *copying* and is denoted by \preccurlyeq . Proper copying, denoted by \prec , is the inverse relation of proper collapsing.

Two examples of collapsing and copying are given in Fig. 1. It is easy to see that the collapse morphisms are the surjective graph morphisms between term graphs, and that the following fact holds.

FACT 2.4. For all term graphs G and H , $G \succcurlyeq H$ implies $\text{term}(G) = \text{term}(H)$.

In the following, we will frequently use term graphs with minimal or maximal sharing.

DEFINITION 2.5 (Tree and fully collapsed term graph). A term graph G is a *tree* if there is no H with $H \succ G$, while G is *fully collapsed* if there is no H with $G \succ H$.

For example, the middle graph in Fig. 1 is fully collapsed. The following is shown in [Plu93b].

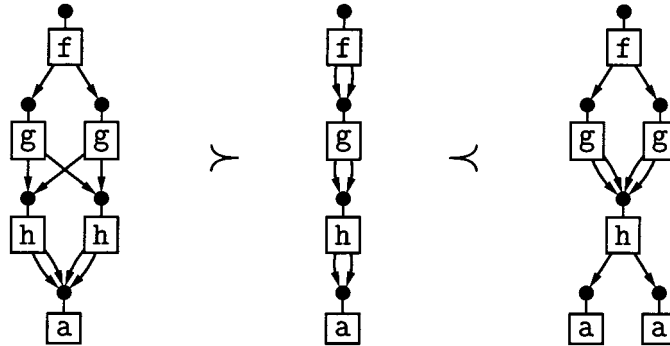


FIG. 1. Collapsing and copying.

LEMMA 2.6. *For every term graph G , there is a unique tree ΔG and a unique fully collapsed term graph ∇G such that*

$$\Delta G \succcurlyeq G \succcurlyeq \nabla G.$$

DEFINITION 2.7 (Bisimilarity). Two term graphs G and H are *bisimilar*, denoted by $G \sim H$, if $\text{term}(G) = \text{term}(H)$.

The three graphs in Fig. 1, for instance, are bisimilar. Note that the two outer graphs are neither related by collapsing nor by copying.

Originally, the notion of bisimilarity and bisimulation was formulated in the theory of concurrent or communicating systems, also called process algebra. As it turned out, the notion applies directly and elegantly to term graphs, in order to give an equivalent formulation of “tree equivalence,” that is, identity of the possibly infinite trees arising after unwinding possibly cyclic term graphs (see [AK96]). Bisimilarity and bisimulations are in the term graph setting much simpler than in process algebra, and our present setting of acyclic term graphs is even more simple, enabling us to define bisimilarity directly without mentioning the notion of bisimulation.

Given a term graph G , the set $[G] = \{G' \mid G \sim G'\}$ is the *bisimilarity class* of G . This class is partially ordered by \succcurlyeq .

THEOREM 2.8 [AK96]. *For every term graph G , $\langle [G], \succcurlyeq \rangle$ is a complete lattice.*

Clearly, ΔG and ∇G are the greatest and the least elements in $[G]$, respectively. So we can characterize bisimilarity as follows.

COROLLARY 2.9. *For all term graphs G and H , the following are equivalent:*

- (1) $G \sim H$.
- (2) $\Delta G = \Delta H$.
- (3) $\nabla G = \nabla H$.

3. TERM GRAPH REWRITING

In this section, we review how term graphs are transformed by applications of term rewrite rules, and we motivate the use of collapsing and copying in term graph rewriting.

A *term rewrite rule* $l \rightarrow r$ consists of two terms l and r over Σ and X such that l is not a variable and all variables in r occur also in l . A set \mathcal{R} of term rewrite rules is a *term rewriting system*. We assume that the reader is familiar with basic concepts of term rewriting. (For an introduction, see the textbook [BN98] or one of the surveys [DJ90, Klo92].) For the following we fix an arbitrary term rewriting system \mathcal{R} . The term rewrite relation associated with \mathcal{R} is denoted by \rightarrow , its transitive closure by \rightarrow^+ , and its reflexive-transitive closure by \rightarrow^* .

Given a term t , we write Δt for the tree representing t . Moreover, $\diamond t$ denotes the term graph representing t such that only variables are shared, that is, each node

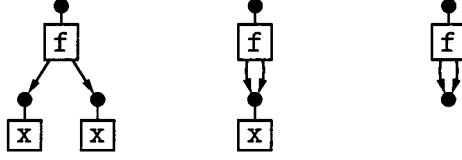


FIG. 2. The graphs $\Delta^f(x, x)$, $\diamond f(x, x)$ and $\underline{\underline{\diamond f(x, x)}}$.

v with an indegree greater than one satisfies $\text{term}_{\diamond t}(v) \in X$, and for each variable x in t there is a unique node v with $\text{term}_{\diamond t}(v) = x$. The graph resulting from $\diamond t$ after removing all edges labelled with variables is denoted by $\underline{\underline{\diamond t}}$. As an example, Fig. 2 shows the graphs $\Delta^f(x, x)$, $\diamond f(x, x)$, and $\underline{\underline{\diamond f(x, x)}}$.

DEFINITION 3.1 (Instance and redex). A term graph T is an *instance* of a term t if there is graph morphism $\underline{\underline{\diamond t}} \rightarrow T$ sending $\text{root}_{\diamond t}$ to root_T . Given a node v in a term graph G and a rule $l \rightarrow r$ in \mathcal{R} , the pair $\langle v, l \rightarrow r \rangle$ is a *redex* if $G|_v$ is an instance of l .

DEFINITION 3.2 (Term graph rewriting). Let G be a term graph containing a redex $\langle v, l \rightarrow r \rangle$. Then there is a *proper rewrite step* $G \Rightarrow_{v, l \rightarrow r} H$, where H is the term graph constructed as follows:

- (1) $G_1 = G - \{e\}$ is the graph obtained from G by removing the unique edge e satisfying $\text{res}(e) = v$.
- (2) G_2 is the graph obtained from the disjoint union $G_1 + \underline{\underline{\diamond r}}$ by
 - identifying v with $\text{root}_{\diamond r}$,
 - identifying the image of $\text{res}(e_1)$ with $\text{res}(e_2)$, for each pair $\langle e_1, e_2 \rangle \in E_{\diamond l} \times E_{\diamond r}$ with $\text{lab}_{\diamond l}(e_1) = \text{lab}_{\diamond r}(e_2) \in X$.
- (3) $H = G_2|_{\text{root}_G}$ is the term graph obtained from G_2 by removing all nodes and edges not reachable from root_G (“garbage collection”).

We denote such a rewrite step also by $G \Rightarrow_v H$ or simply by $G \Rightarrow H$, and we write $G \Rightarrow^* H$ if there are graphs G_0, \dots, G_n ($n \geq 0$) such that $G = G_0 \Rightarrow G_1 \Rightarrow \dots \Rightarrow G_n = H$.

EXAMPLE 3.3. Figure 3 shows the three intermediate steps in the construction of a term graph rewrite step. The term rewrite rule applied to G is $x \times (y + z) \rightarrow (x \times y) + (x \times z)$. In G and H , shaded nodes and edges belong to the occurrences of $\underline{\underline{\diamond x \times (y + z)}}$ and $\underline{\underline{\diamond (x \times y) + (x \times z)}}$, respectively. Note that the variables y and z correspond to the same node; that is, the graph morphism $\underline{\underline{\diamond x \times (y + z)}} \rightarrow G$ identifies the nodes representing y and z .

Given a term graph rewrite step $G \Rightarrow H$ and a node v in G , v either has a unique image in H or is removed by garbage collection. We use a partial function $\text{tr}_{G \Rightarrow H} : V_G \rightarrow V_H$, the *track function* for $G \Rightarrow H$, to assign to each node in G its corresponding node in H .

DEFINITION 3.4 (Track function). Let $G \Rightarrow_{v, l \rightarrow r} H$ be a proper term graph rewrite step. Let $\text{in} : G_1 \rightarrow G_1 + \underline{\underline{\diamond r}}$ be the injective graph morphism associated with

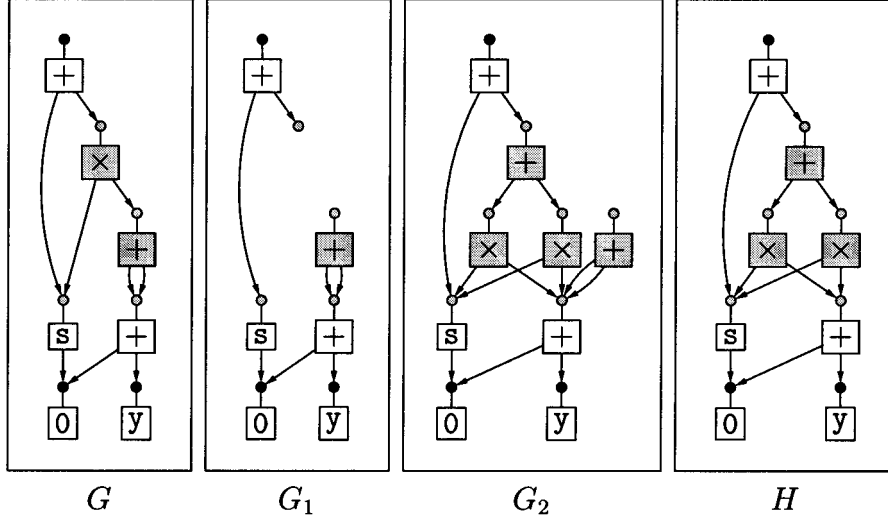


FIG. 3. The construction of a term graph rewrite step.

the disjoint union and $ident : G_1 + \diamond r \rightarrow G_2$ be the surjective morphism associated with the identification, in the construction of Definition 3.2. Then the *track function* for this rewrite step is the partial function $tr_{G \Rightarrow H} : V_G \rightarrow V_H$ defined as

$$tr_{G \Rightarrow H}(v) = \begin{cases} ident(in(v)), & \text{if } ident(in(v)) \in V_H, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

The track function is extended to rewrite sequences as

$$tr_{G \Rightarrow^* H}(v) = \begin{cases} v, & \text{if } n = 0 \\ tr_{G_{n-1} \Rightarrow G_n}(\dots tr_{G_0 \Rightarrow G_1}(v) \dots), & \text{otherwise,} \end{cases}$$

where $G \Rightarrow^* H$ denotes the sequence $G = G_0 \Rightarrow G_1 \Rightarrow \dots \Rightarrow G_n = H$.

Term graph rewriting is sound with respect to term rewriting in the following sense.

THEOREM 3.5 (Soundness, [BvEG⁺87, HP88]). *For all term graphs G and H ,*

$$G \Rightarrow H \text{ implies } term(G) \rightarrow^+ term(H).$$

In the sequel we consider not only term graph rewriting by \Rightarrow but also extensions with collapsing and copying.

DEFINITION 3.6 (\Rightarrow_{coll} , \Rightarrow_{copy} , \Rightarrow_{bi}). The relations \Rightarrow_{coll} , \Rightarrow_{copy} and \Rightarrow_{bi} on term graphs are defined as

$$\begin{aligned} \Rightarrow_{coll} &= \Rightarrow \cup \succ, \\ \Rightarrow_{copy} &= \Rightarrow \cup \prec, \\ \Rightarrow_{bi} &= \Rightarrow \cup \succ \cup \prec. \end{aligned}$$

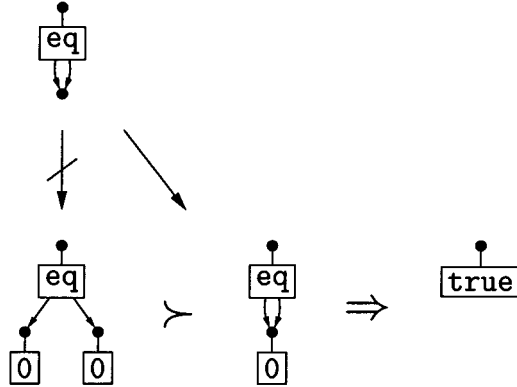


FIG. 4. Collapsing to enable a rule application.

We refer to \Rightarrow , $\Rightarrow_{\text{coll}}$, $\Rightarrow_{\text{copy}}$, and \Rightarrow_{bi} as *plain term graph rewriting*, *term graph rewriting with collapsing*, *term graph rewriting with copying*, and *term graph rewriting with collapsing and copying*, respectively.

Given a binary relation \Rightarrow on term graphs, its inverse is denoted by \Leftarrow , and we write \Leftrightarrow , \Rightarrow^{λ} , and \Rightarrow^* for the symmetric, reflexive, and reflexive-transitive closure of \Rightarrow , respectively.

Note that the relation $\Rightarrow_{\text{bi}}^*$ contains bisimilarity since $G \sim H$ implies $G \Leftarrow \Delta G \Rightarrow H$ (see Lemma 2.6 and Corollary 2.9). Moreover, $\Rightarrow_{\text{coll}}$, $\Rightarrow_{\text{copy}}$, and \Rightarrow_{bi} are sound in the sense of Theorem 3.5 if we replace \rightarrow^+ by \rightarrow^* (collapse and copy steps do not change the represented term).

Collapsing allows us to cope with term rewrite rules having repeated variables in their left-hand sides. For instance, the rule $\text{eq}(x, x) \rightarrow \text{true}$ cannot be applied to the tree $\Delta \text{eq}(0, 0)$ because there is no graph morphism $\diamond_{\text{eq}(x, x)} \rightarrow \Delta \text{eq}(0, 0)$ (see Fig. 4). This problem is overcome by first collapsing $\Delta \text{eq}(0, 0)$ so that subsequently the rule can be applied.

Another advantage of collapsing is that, in certain cases, it can speed up evaluation processes drastically. A prime example is the specification of the Fibonacci function:

$$\begin{aligned} \text{fib}(0) &\rightarrow 0 \\ \text{fib}(s(0)) &\rightarrow s(0) \\ \text{fib}(s(s(x))) &\rightarrow \text{fib}(s(x)) + \text{fib}(x). \end{aligned}$$

Using these three rules, evaluating a term of the form $\text{fib}(s^n(0))$ by term rewriting requires a number of rewrite steps exponential in n (see [AS85]). One easily observes that the same number of steps is needed for plain term graph rewriting. After replacing \Rightarrow by $\Rightarrow_{\text{coll}}$, however, it is possible to evaluate $\text{fib}(s^n(0))$ in a linear number of steps. The evaluation strategy can be described as follows: (1) Collapse steps have priority over proper rewrite steps and produce fully collapsed term graphs. (2) Out of two *fib*-redexes, the one representing the greater number is reduced. See Fig. 5 for an illustration of this strategy. It is not difficult to verify

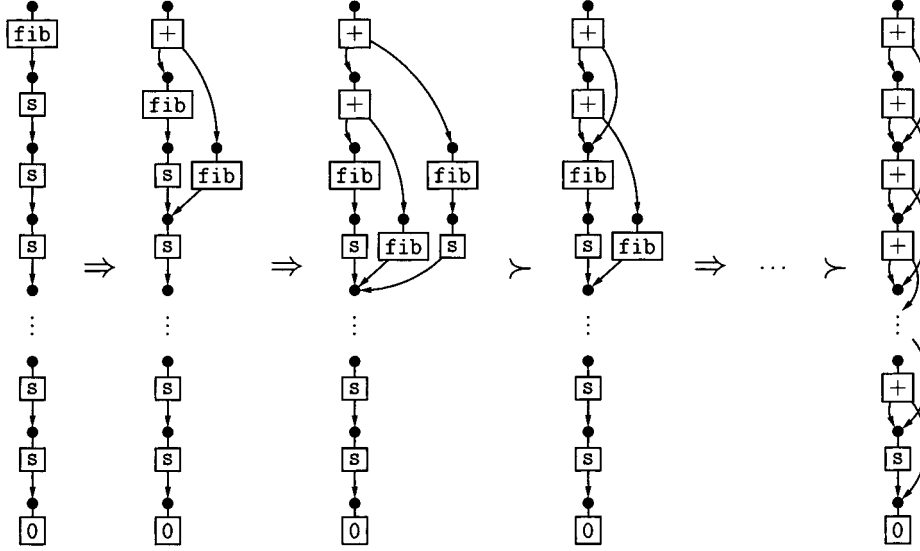


FIG. 5. Collapsing to speed up evaluation.

that, for $n \geq 2$, this procedure evaluates $\text{fib}(s^n(0))$ in $2n + 1$ steps (viz., $n + 1$ proper rewrite steps and n collapse steps).

The benefit of copying is that it makes term graph rewriting complete with respect to term rewriting; Theorem 7.1 will show that every term rewriting sequence can be simulated if both collapsing and copying are present. Moreover, if there are no repeated variables in the left-hand sides of rules, then copying alone guarantees completeness (see Theorem 7.3).

4. NOTIONS OF CONFLUENCE

In this section we define confluence, confluence modulo bisimilarity and the Church–Rosser property modulo bisimilarity. It turns out that for $\Rightarrow_{\text{coll}}$, $\Rightarrow_{\text{copy}}$ and \Rightarrow_{bi} , these three properties are equivalent. For plain term graph rewriting, however, the Church–Rosser property modulo bisimilarity is strictly stronger than confluence modulo bisimilarity, and confluence is incomparable with the two other properties.

DEFINITION 4.1 (Confluence properties). A binary relation \Rightarrow on term graphs is

- (1) *confluent* if for every constellation $G_1 \Leftarrow^* G \Rightarrow^* G_2$ there is a term graph G_3 such that $G_1 \Rightarrow^* G_3 \Leftarrow^* G_2$,
- (2) *confluent modulo bisimilarity* if whenever $G_1 \Leftarrow^* G \sim H \Rightarrow^* H_1$, there are term graphs G_2 and H_2 such that $G_1 \Rightarrow^* G_2 \sim H_2 \Leftarrow^* H_1$,
- (3) *Church–Rosser modulo bisimilarity* if whenever $G \approx H$, there are term graphs G_1 and H_1 such that $G \Rightarrow^* G_1 \sim H_1 \Leftarrow^* H$. Here \approx is the transitive closure of the relation $\Leftarrow \cup \sim$.

See Fig. 6 for illustrations of these properties.

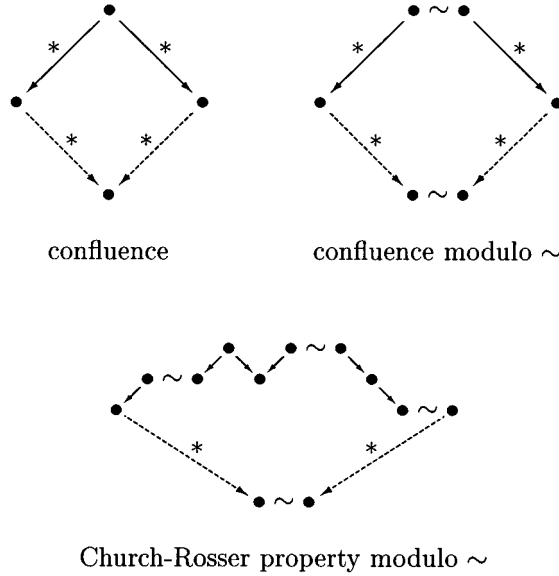


FIG. 6. Confluence properties.

An important consequence of confluence is that rewriting yields deterministic results. Call a term graph N a *normal form* with respect to \Rightarrow if there is no N' with $N \Rightarrow N'$. The relation \Rightarrow is *normalizing* if for every term graph G there is a normal form N such that $G \Rightarrow^* N$. *Uniqueness of normal forms* means that whenever $N_1 \Leftarrow^* G \Rightarrow^* N_2$ for normal forms N_1 and N_2 , then $N_1 = N_2$. While confluence implies uniqueness of normal forms, confluence modulo bisimilarity implies uniqueness of normal forms up to bisimilarity.

From Definition 4.1 it is clear that “Church–Rosser modulo \sim ” implies “confluent modulo \sim .” The following lemma is a specialization of a lemma of Huet [Hue80] to the term graph setting.

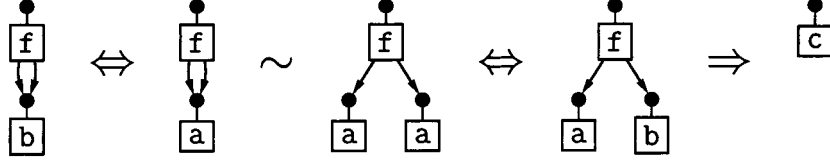
LEMMA 4.2. *A normalizing relation \Rightarrow on term graphs is Church–Rosser modulo bisimilarity if and only if it is confluent modulo bisimilarity.*

For plain term graph rewriting, the Church–Rosser property modulo bisimilarity is strictly stronger than confluence modulo bisimilarity. This will become apparent by Example 6.2 in conjunction with Theorem 6.4. (The two outer graphs in Fig. 11 are related by \approx but cannot be reduced to bisimilar graphs by \Rightarrow .)

The next two examples show that for plain term graph rewriting, confluence is, in general, incomparable with both the Church–Rosser property modulo bisimilarity and confluence modulo bisimilarity.

EXAMPLE 4.3. Consider the system:

$$\begin{aligned} a &\rightarrow b \\ b &\rightarrow a \\ f(a, b) &\rightarrow c. \end{aligned}$$

FIG. 7. Confluence without confluence modulo \sim .

It is easy to check that \Rightarrow is confluent, but Fig. 7 shows that confluence modulo bisimilarity (and hence the Church–Rosser property modulo bisimilarity) fails.

EXAMPLE 4.4. Plain term graph rewriting may be Church–Rosser modulo bisimilarity (and hence, confluent modulo bisimilarity) without being confluent. This is demonstrated by the system:

$$\begin{aligned} g(x) &\rightarrow f(x, x) \\ g(a) &\rightarrow f(a, a). \end{aligned}$$

Theorem 7.7 will show that \Rightarrow is Church–Rosser modulo \sim , since \Rightarrow is normalizing and term rewriting is confluent. However, the term graph representing $g(a)$ has two distinct normal forms (see Fig. 8); hence \Rightarrow is not confluent.

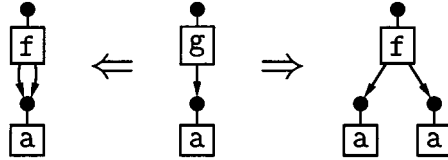
In the presence of collapsing or copying, the three confluence properties become equivalent. Thus, in particular, these properties are equivalent for the relations $\Rightarrow_{\text{coll}}$, $\Rightarrow_{\text{copy}}$, and \Rightarrow_{bi} .

THEOREM 4.5. *Let \Rightarrow be a binary relation on term graphs such that $\succ \subseteq \Rightarrow^*$ or $\prec \subseteq \Rightarrow^*$. Then the following are equivalent:*

- (1) \Rightarrow is confluent.
- (2) \Rightarrow is confluent modulo bisimilarity.
- (3) \Rightarrow is Church–Rosser modulo bisimilarity.

Proof. Suppose that $\succ \subseteq \Rightarrow^*$; the case $\prec \subseteq \Rightarrow^*$ is treated analogously. We show the implications (1) \rightarrow (3) \rightarrow (2) \rightarrow (1).

(1) \rightarrow (3) Consider term graphs G and H with $G \approx H$. Then there are term graphs G_0, \dots, G_n , $n \geq 1$, such that $G_0 = G$, $G_n = H$, and for $i = 1, \dots, n$, $G_{i-1} \Leftrightarrow G_i$ or $G_{i-1} \sim G_i$. By Lemma 2.6 and Corollary 2.9, each constellation $G_{i-1} \sim G_i$ satisfies $G_{i-1} \leq \Delta G_i \geq G_i$. Hence, $G_{i-1} \Leftarrow^* \circ \Rightarrow^* G_i$ for $i = 1, \dots, n$. By induction on n , using confluence, we see that there is a term graph M such that $G = G_0 \Rightarrow^* M \Leftarrow^* G_n = H$. Thus, \Rightarrow is Church–Rosser modulo \sim .

FIG. 8. Church–Rosser property modulo \sim without confluence.

(3) \rightarrow (2) Immediate consequence of Definition 4.1.

(2) \rightarrow (1) Assume that $G_1 \Leftarrow^* G \Rightarrow^* G_2$. By confluence modulo \sim , there are term graphs G_3 and G_4 such that $G_1 \Rightarrow^* G_3 \sim G_4 \Leftarrow^* G_2$. Then $G_3 \succcurlyeq \nabla G_3 \preccurlyeq G_4$ and, hence, $G_1 \Rightarrow^* \nabla G_3 \Leftarrow^* G_2$. So \Rightarrow is confluent. \blacksquare

5. NONOVERLAPPING SYSTEMS

It is known that plain term graph rewriting is confluent if the left-hand sides of the given term rewrite rules do not overlap. After recalling this and a related result about the uniqueness of complete developments, we show that the reduction strategy of full substitution is cofinal. Then counterexamples are given demonstrating that confluence fails as soon as term graph rewriting is extended with copying or collapsing.

For the next definition we need the notion of a *substitution*, which is a mapping σ on the set of terms over Σ and X such that $\sigma(c) = c$ for every constant c , and $\sigma(f(t_1, \dots, t_n)) = f(\sigma(t_1), \dots, \sigma(t_n))$ for every composite term $f(t_1, \dots, t_n)$.

DEFINITION 5.1 (Non-overlapping). A term s *overlaps* a term t in a subterm u of t if u is not a variable and there are substitutions σ and τ such that $\sigma(s) = \tau(u)$. The term rewriting system \mathcal{R} is *nonoverlapping* if for all rules $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ in \mathcal{R} , l_1 overlaps l_2 in a subterm u only if $u = l_2$ and $(l_1 \rightarrow r_1) = (l_2 \rightarrow r_2)$.

THEOREM 5.2. *Let \mathcal{R} be nonoverlapping and G, G_1 and G_2 be term graphs such that $G_1 \Leftarrow_{v_1} G \Rightarrow_{v_2} G_2$. Then there is a term graph G_3 such that $G_1 \Rightarrow_{\text{tr}(v_2)}^\lambda G_3 \Leftarrow_{\text{tr}(v_1)}^\lambda G_2$, where $\text{tr}_{G \Rightarrow G_1}^\lambda G_3 = \text{tr}_{G \Rightarrow G_2}^\lambda G_3$.*

Proof. A proof is already given in [Sta80], in a slightly different technical framework. A proof conforming to the present setting can be found in [Plu93b], as part of the proof of the so-called Critical Pair Lemma.

Call the relation \Rightarrow *subcommutative* if whenever $G_1 \Leftarrow G \Rightarrow G_2$, there is a term graph G_3 such that $G_1 \Rightarrow^\lambda G_3 \Leftarrow^\lambda G_2$. It is well known that subcommutativity implies confluence (for arbitrary binary relations; see [Klo92]).

COROLLARY 5.3. *If \mathcal{R} is nonoverlapping, then \Rightarrow is subcommutative.*

For the rest of this section we assume that \mathcal{R} is an arbitrary nonoverlapping system. The following property of subcommutative relations will be needed in showing that the full substitution strategy is cofinal.

COROLLARY 5.4. *For all term graphs G, G_1 , and G_2 , $G_1 \Leftarrow G \Rightarrow^* G_2$ implies that there is a term graph G_3 such that $G_1 \Rightarrow^* G_3 \Leftarrow^\lambda G_2$.*

Proof. By induction on the length of $G \Rightarrow^* G_2$, using subcommutativity. \blacksquare

We are going to show that complete developments of sets of redexes yield unique results. This fact allows us to define the full substitution strategy. In the next section, the cofinality property of this strategy will be used to prove that \Rightarrow is confluent modulo bisimilarity over orthogonal rewrite systems.

Since \mathcal{R} is nonoverlapping, every redex $\langle v, l \rightarrow r \rangle$ is uniquely determined by the node v . Hence, in this section, we treat redexes as nodes.

DEFINITION 5.5 (Residuals). Let Π be a set of redexes in a term graph G . The set $\rho(\Pi)$ of *residuals* of Π with respect to a rewrite sequence $G \Rightarrow^* H$ is defined as follows. If $G \Rightarrow^* H$ has length 0, then $\rho(\Pi) = \Pi$. If $G \Rightarrow^* H$ has the form $G \Rightarrow_v G' \Rightarrow^* H$, then $\rho(\Pi)$ is the set of residuals of $\text{tr}_{G \Rightarrow G'}(\Pi - \{v\})$ with respect to $G' \Rightarrow^* H$.

By the assumption that \mathcal{R} is nonoverlapping, the residuals of a redex set are again redexes. Note that this is different in term rewriting; there this property may fail when rules are present that have repeated variables in their left-hand sides.

DEFINITION 5.6 (Development). A *development* of a set Π of redexes in a term graph G is either a derivation $G \Rightarrow^* H$ of length 0, or a derivation of the form $G \Rightarrow_v G' \Rightarrow^* H$ such that $v \in \Pi$ and such that $G' \Rightarrow^* H$ is a development of the residuals of Π in G' . The development is *complete* if Π has no residuals in H .

The following result was given in [BvEG⁺87], without proof.

THEOREM 5.7 (Uniqueness of developments). *Given a set Π of redexes in a term graph G , all complete developments of Π end in the same term graph.*

Proof. Consider two complete developments $G \Rightarrow^* H_1$ and $G \Rightarrow^* H_2$ of Π . We proceed by induction on the number of redexes in Π . If Π is empty, then $H_1 = G = H_2$ by the definition of complete development. Otherwise, there are (not necessarily distinct) nodes v_1 and v_2 in Π such that for $i = 1, 2$, $G \Rightarrow^* H_i$ has the form $G \Rightarrow_{v_i} G_i \Rightarrow^* H_i$. By Theorem 5.2, there are steps

$$G_1 \xrightarrow[\text{tr}(v_2)]{\lambda} G_3 \xleftarrow[\text{tr}(v_1)]{\lambda} G_2$$

such that $\text{tr}_{G \Rightarrow G_1 \Rightarrow^\lambda G_3} = \text{tr}_{G \Rightarrow G_2 \Rightarrow^\lambda G_3}$. Now consider the redex set

$$\Pi' = \text{tr}_{G \Rightarrow G_1 \Rightarrow^\lambda G_3}(\Pi - \{v_1, v_2\}) = \text{tr}_{G \Rightarrow G_2 \Rightarrow^\lambda G_3}(\Pi - \{v_1, v_2\}).$$

Let $G_3 \Rightarrow^* H_3$ be a complete development of Π' . It is not difficult to show that

$$\Pi' = \text{tr}_{G_1 \Rightarrow^\lambda G_3}(\text{tr}_{G \Rightarrow G_1}(\Pi - \{v_1\}) - \{\text{tr}(v_2)\}),$$

hence, $G_1 \Rightarrow_{\text{tr}(v_2)}^\lambda G_3 \Rightarrow^* H_3$ is a complete development of $\text{tr}_{G \Rightarrow G_1}(\Pi - \{v_1\})$. By assumption, $G_1 \Rightarrow^* H_1$ is a complete development of this set as well. Hence, by induction hypothesis, $H_3 = H_1$. Analogously one shows $H_3 = H_2$. Thus $H_1 = H_2$. ■

Given a term graph G , we denote by $\text{Cpl}(G)$ the term graph that results from a complete development of all redexes in G . The process of repeatedly developing all redexes is called the *full substitution* or *Gross–Knuth* strategy in the context of term rewriting systems (see [Klo92]). We show that this strategy is “cofinal” for term graph rewriting over nonoverlapping systems.

THEOREM 5.8 (Cofinality). *For all term graphs G and H , $G \Rightarrow^* H$ implies that there is $n \geq 0$ such that $H \Rightarrow^* \text{Cpl}^n(G)$.*

Proof. By induction on the length of $G \Rightarrow^* H$. Suppose that $G \Rightarrow^* H' \Rightarrow H$ for some term graph H' . By induction hypothesis, $H' \Rightarrow^* \text{Cpl}^n(G)$ for some $n \geq 0$. Then, by Corollary 5.4, there is a term graph H'' such that $H \Rightarrow^* H'' \Leftarrow^\lambda \text{Cpl}^n(G)$. Thus, by the definition of complete development, $H'' \Rightarrow^* \text{Cpl}^{n+1}(G)$. It follows $H \Rightarrow^* \text{Cpl}^{n+1}(G)$.

Confluence of \Rightarrow no longer holds if collapsing or copying is added, as the following two counterexamples demonstrate. Moreover, the examples show that none of the four relations \Rightarrow , $\Rightarrow_{\text{coll}}$, $\Rightarrow_{\text{copy}}$, and \Rightarrow_{bi} is confluent modulo bisimilarity for nonoverlapping systems in general.

EXAMPLE 5.9. Consider the nonoverlapping system of Huet [Hue80]:

$$\begin{aligned} f(x, x) &\rightarrow a \\ f(x, g(x)) &\rightarrow b \\ c &\rightarrow g(c) \end{aligned}$$

Figure 9 demonstrates that the tree representing $f(c, c)$ has two distinct normal forms with respect to $\Rightarrow_{\text{coll}}$, so $\Rightarrow_{\text{coll}}$, and \Rightarrow_{bi} are neither confluent nor confluent modulo bisimilarity. Note that in the left-hand sides of the first two rewrite rules, the variable x occurs twice.

EXAMPLE 5.10. The rule

$$f(x, x) \rightarrow a$$

also contains two occurrences of x in its left-hand side. It shows that, for nonoverlapping systems, $\Rightarrow_{\text{copy}}$ need neither be confluent nor confluent modulo bisimilarity. To see this, observe that the graph on the right in Fig. 10 is a normal form with respect to $\Rightarrow_{\text{copy}}$.

Figure 10 also demonstrates that plain term graph rewriting is not confluent modulo bisimilarity for nonoverlapping systems in general. This is because the rewrite step on the left is proper, and the graphs in the middle and on the right are bisimilar.

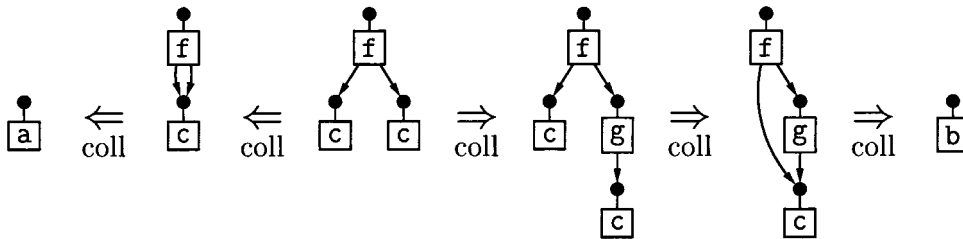
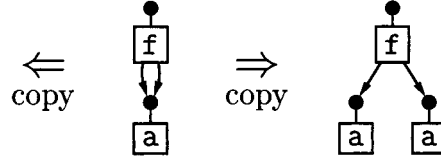


FIG. 9. Nonconfluence of $\Rightarrow_{\text{coll}}$ and \Rightarrow_{bi} .

FIG. 10. Nonconfluence of $\Rightarrow_{\text{copy}}$.

6. ORTHOGONAL SYSTEMS

The counterexamples of the previous section show that for nonoverlapping systems, \Rightarrow , $\Rightarrow_{\text{coll}}$, $\Rightarrow_{\text{copy}}$, and \Rightarrow_{bi} need not be confluent modulo bisimilarity, and the last three relations need not be confluent either. In this section and the next it will become clear that this failure is caused, with the exception of $\Rightarrow_{\text{coll}}$, by rewrite rules with repeated variables in their left-hand sides.

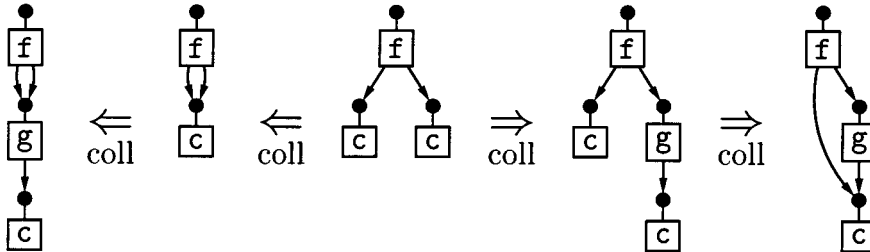
DEFINITION 6.1 (Orthogonal). The term rewriting system \mathcal{R} is *left-linear* if for each rewrite rule $l \rightarrow r$ in \mathcal{R} , no variable occurs more than once in l . The system \mathcal{R} is *orthogonal* if it is left-linear and nonoverlapping.

The main result of this section is that for orthogonal systems, plain term graph rewriting is confluent modulo bisimilarity. As far as confluence is concerned, we know from the previous section that \Rightarrow is confluent for nonoverlapping systems and, hence, in particular, for orthogonal systems. In the next section we show that $\Rightarrow_{\text{copy}}$ and \Rightarrow_{bi} are confluent for classes of systems that properly include all orthogonal systems. In contrast, term graph rewriting with collapsing need not be confluent even for orthogonal systems.

EXAMPLE 6.2. Consider the single rule

$$c \rightarrow g(c)$$

and suppose that Σ contains a binary function symbol f . Figure 11 shows two $\Rightarrow_{\text{coll}}$ -derivations starting from $\Delta f(c, c)$ such that the resulting graphs do not have a common reduct under $\Rightarrow_{\text{coll}}$: the graphs derivable on the left represent the terms $f(g^n(c), g^n(c))$, $n \geq 1$, while the graphs derivable on the right represent $f(g^n(c), g^{n+1}(c))$, $n \geq 0$. Thus, $\Rightarrow_{\text{coll}}$ is nonconfluent. Notice also that \Rightarrow is not Church–Rosser modulo \sim .

FIG. 11. Nonconfluence of $\Rightarrow_{\text{coll}}$.

LEMMA 6.3. *Let \mathcal{R} be orthogonal. Then for all term graphs G and H , $G \sim H$ implies $\text{Cpl}(G) \sim \text{Cpl}(H)$.*

Proof. Given a complete development $T \Rightarrow^* \text{Cpl}(T)$ of all redexes in a term graph T , there is a corresponding complete development $\text{term}(T) \rightarrow^* \text{term}(\text{Cpl}(T))$ of all redexes in $\text{term}(T)$ (see [KKSdV94] for a proof in a slightly different technical setting). Since for orthogonal term rewriting systems, all complete developments of a set of redexes yield the same result [HL91], $\text{term}(G) = \text{term}(H)$ implies $\text{term}(\text{Cpl}(G)) = \text{term}(\text{Cpl}(H))$. ■

It is worth mentioning that this lemma does not hold for nonoverlapping systems. A counterexample is again $\mathcal{R} = \{f(x, x) \rightarrow a\}$; in Fig. 10, the graph in the middle is bisimilar to the graph on the right which is a normal form with respect to \Rightarrow .

THEOREM 6.4. *If \mathcal{R} is orthogonal, then \Rightarrow is confluent modulo bisimilarity.*

Proof. Suppose that $G_1 \leftarrow^* G \sim H \Rightarrow^* H_1$. By Theorem 5.8, there are $m, n \geq 0$ such that $G_1 \Rightarrow^* \text{Cpl}^m(G)$ and $H_1 \Rightarrow^* \text{Cpl}^n(H)$. Hence, choosing $p = \max(m, n)$, we obtain $G_1 \Rightarrow^* \text{Cpl}^p(G)$ and $H_1 \Rightarrow^* \text{Cpl}^p(H)$. Now $\text{Cpl}^p(G) \sim \text{Cpl}^p(H)$ follows from Lemma 6.3. ■

COROLLARY 6.5. *If \mathcal{R} is orthogonal and \Rightarrow normalizing, then \Rightarrow is Church–Rosser modulo bisimilarity.*

Proof. Combine Theorem 6.4 and Lemma 4.2. ■

Corollary 6.5 will be generalized by Theorem 7.7, where orthogonality is weakened to left-linearity in conjunction with confluence of \rightarrow .

7. GENERAL SYSTEMS

In this section we drop the assumption of the two previous sections that \mathcal{R} is nonoverlapping. Instead, we infer confluence of \Rightarrow_{bi} , $\Rightarrow_{\text{copy}}$, and $\Rightarrow_{\text{coll}}$ from confluence of term rewriting. In the case of $\Rightarrow_{\text{copy}}$ and $\Rightarrow_{\text{coll}}$, this requires suitable further conditions. Finally, we give sufficient conditions under which confluence of term rewriting makes \Rightarrow Church–Rosser modulo bisimilarity.

We first show that term graph rewriting with collapsing and copying can simulate term rewriting, following the proof of the so-called Completeness Theorem in [Plu93b].

THEOREM 7.1. *For all term graphs G and H :*

$$G \Rightarrow_{\text{bi}}^* H \quad \text{if and only if} \quad \text{term}(G) \rightarrow^* \text{term}(H).$$

Proof. (Only if) By soundness of \Rightarrow_{bi} .

(If) Suppose that for every term rewrite step $t \rightarrow u$ there are term graphs T and U such that

$$\triangle t \triangleright T \Rightarrow U \triangleleft \diamond u. \quad (1)$$

Then $\text{term}(G) \rightarrow^* \text{term}(H)$ implies $\Delta \text{term}(G) \Rightarrow_{\text{bi}}^* \Delta \text{term}(H)$, and with Lemma 2.6 follows $G \preceq \Delta \text{term}(G) \Rightarrow_{\text{bi}}^* \Delta \text{term}(H) \succeq H$. To show (1), let $l \rightarrow r$ be the rule applied in $t \rightarrow u$ and π be the associated redex position in t . Let v be the unique node in Δt specified by π . Then, there is a collapsing $\Delta t \succcurlyeq T$ such that $T|_{v'}$ is fully collapsed, where v' is the image of v in T , and such that each node of T not belonging to $T|_{v'}$ has an indegree of at most one. By the structure of T , there is a step $T \Rightarrow_{v', l \rightarrow r} U$ such that $\text{term}(U) = u$. (Since $T|_{v'}$ is fully collapsed, $l \rightarrow r$ is applicable at v' even if l contains repeated variables, and as there is a unique path from root_T to v' , $T \Rightarrow_{v', l \rightarrow r} U$ simulates $t \rightarrow u$.) Hence $U \preceq \Delta u$. ■

One should be aware that the generality of the relation \Rightarrow_{bi} has to be paid with termination and efficiency problems. (Recall that a binary relation \rightarrow on a set A is *terminating* if there does not exist an infinite sequence $a_1 \rightarrow a_2 \rightarrow \dots$.) In particular, \Rightarrow_{bi} is nonterminating for every term graph representing a term containing two or more occurrences of some subterm. This is because such a graph admits an infinite sequence of alternating collapse and copy steps. In contrast, \Rightarrow , $\Rightarrow_{\text{coll}}$, and $\Rightarrow_{\text{copy}}$ are terminating whenever the term rewrite relation \rightarrow is terminating. Apart from nontermination, the search space for computing a term normal form by \Rightarrow_{bi} may be much larger than for \Rightarrow or $\Rightarrow_{\text{coll}}$. (See [Plu98] for conditions under which $\Rightarrow_{\text{coll}}$ suffices to compute term normal forms.)

COROLLARY 7.2. *The relation \Rightarrow_{bi} is confluent if and only if \rightarrow is confluent.*

Proof. (Only if) Suppose that $t_1 \leftarrow^* t \rightarrow^* t_2$ for some terms t , t_1 , and t_2 . Then $\Delta t_1 \leftarrow_{\text{bi}}^* \Delta t \Rightarrow_{\text{bi}}^* \Delta t_2$ by Theorem 7.1. Since \Rightarrow_{bi} is confluent, there is a term graph G such that $\Delta t_1 \Rightarrow_{\text{bi}}^* G \leftarrow_{\text{bi}}^* \Delta t_2$. Hence $t_1 \rightarrow^* \text{term}(G) \leftarrow^* t_2$ by Theorem 7.1.

(If) Given derivations $G_1 \leftarrow_{\text{bi}}^* G \Rightarrow_{\text{bi}}^* G_2$, Theorem 7.1 yields $\text{term}(G_1) \leftarrow^* \text{term}(G) \rightarrow^* \text{term}(G_2)$. Then, since \rightarrow is confluent, there is a term t such that $\text{term}(G_1) \rightarrow^* t \leftarrow^* \text{term}(G_2)$. With Theorem 7.1 follows $G_1 \Rightarrow_{\text{bi}}^* \Delta t \leftarrow_{\text{bi}}^* G_2$, since $\text{term}(\Delta t) = t$.

In order to simulate term rewriting by $\Rightarrow_{\text{copy}}$, the underlying system \mathcal{R} has to be left-linear.

THEOREM 7.3. *If \mathcal{R} is left-linear, then for all terms t and u :*

$$\Delta t \Rightarrow_{\text{copy}}^* \Delta u \quad \text{if and only if} \quad t \rightarrow^* u.$$

Proof. (Only if) Immediate consequence of the soundness of $\Rightarrow_{\text{copy}}$.

(If) It suffices to show that for every term rewrite step $t \rightarrow u$ there is a term graph U such that

$$\Delta t \Rightarrow U \preceq \Delta u.$$

Let $l \rightarrow r$ be the rule applied in $t \rightarrow u$. Then, since \mathcal{R} is left-linear, $\Delta t \Rightarrow_{v, l \rightarrow r} U$ for some term graph U , where v is the node corresponding to the redex position in t . As there is no sharing in Δt , we have $\text{term}(U) = u$. Thus, $U \preceq \Delta u$. ■

COROLLARY 7.4. *If \mathcal{R} is left-linear, then $\Rightarrow_{\text{copy}}$ is confluent if and only if \rightarrow is confluent.*

Proof. (Only if) Easy consequence of Theorem 7.3 and soundness of $\Rightarrow_{\text{copy}}$.

(If) Consider derivations $G_1 \leftarrow^*_{\text{copy}} G \Rightarrow^*_{\text{copy}} G_2$. By soundness of $\Rightarrow_{\text{copy}}$, we get $\text{term}(G_1) \leftarrow^* \text{term}(G) \rightarrow^* \text{term}(G_2)$. Confluence of \rightarrow implies that there is a term t such that $\text{term}(G_1) \rightarrow^* t \leftarrow^* \text{term}(G_2)$. With Theorem 7.3 follows $\Delta \text{term}(G_1) \Rightarrow^*_{\text{copy}} \Delta t \leftarrow^*_{\text{copy}} \Delta \text{term}(G_2)$. Hence, using Lemma 2.6,

$$G_1 \preceq \Delta \text{term}(G_1) \Rightarrow^*_{\text{copy}} \Delta t \leftarrow^*_{\text{copy}} \Delta \text{term}(G_2) \succeq G_2. \quad \blacksquare$$

Corollary 7.4 implies that $\Rightarrow_{\text{copy}}$ is confluent, in particular, for orthogonal systems. For it is well known that orthogonality implies confluence of term rewriting (see, for example, [BN98]).

An analogue to Corollary 7.4 for the case of $\Rightarrow_{\text{coll}}$ can be obtained by replacing the condition of left-linearity with normalization of $\Rightarrow_{\text{coll}}$.

THEOREM 7.5 [Plu93a]. *Suppose that $\Rightarrow_{\text{coll}}$ is normalizing. Then $\Rightarrow_{\text{coll}}$ is confluent if and only if \rightarrow is confluent.*

In general, normalization of $\Rightarrow_{\text{coll}}$ neither implies nor follows from normalization of \Rightarrow . If all rules are left-linear, however, the two properties are equivalent.

LEMMA 7.6. *If \mathcal{R} is left-linear, then $\Rightarrow_{\text{coll}}$ is normalizing if and only if \Rightarrow is normalizing.*

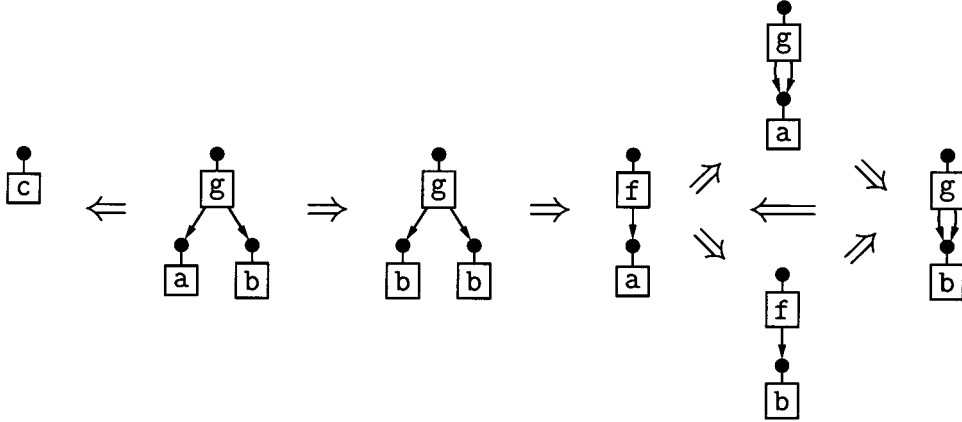
Proof. (If) Left-linearity implies that for every normal form N with respect to \Rightarrow , ∇N is a normal form with respect to $\Rightarrow_{\text{coll}}$.

(Only if) In [HP96] it is shown that every derivation $G \Rightarrow^*_{\text{coll}} H$ can be transformed into a “minimally collapsing” derivation $G \Rightarrow^*_{\text{coll}} H'$ such that $H' \succeq H$. If \mathcal{R} is left-linear, this implies $G \Rightarrow^* H'$. Moreover, if H is a normal form with respect to $\Rightarrow_{\text{coll}}$, then $\text{term}(H') = \text{term}(H)$ is a normal form with respect to \rightarrow . Hence, H' is a normal form with respect to \Rightarrow . \blacksquare

We conclude this section by giving conditions under which confluence of \rightarrow guarantees that \Rightarrow is Church–Rosser modulo bisimilarity. It turns out that both left-linearity of \mathcal{R} and normalization of \Rightarrow are needed.

THEOREM 7.7. *If \mathcal{R} is left-linear, \rightarrow confluent, and \Rightarrow normalizing, then \Rightarrow is Church–Rosser modulo bisimilarity.*

Proof. By Lemma 4.2, it suffices to show that \Rightarrow is confluent modulo bisimilarity. Given a constellation $G_1 \leftarrow^* G \sim H \Rightarrow^* H_1$, consider normal forms G_2 and H_2 of G_1 and H_1 , respectively. Then $\text{term}(G_2) \leftarrow^* \text{term}(G) = \text{term}(H) \rightarrow^* \text{term}(H_2)$ by soundness of \Rightarrow . Since G_2 and H_2 are normal forms and \mathcal{R} is left-linear, $\text{term}(G_2)$ and $\text{term}(H_2)$ are normal forms with respect to \rightarrow . (Left-linearity implies that $\diamond l$ is a tree for every term rewrite rule $l \rightarrow r$; hence, given a term graph T , there is a graph morphism $\diamond l \rightarrow T$ if and only if $l \rightarrow r$ is applicable to $\text{term}(T)$.) Now confluence of \rightarrow yields $\text{term}(G_2) = \text{term}(H_2)$, thus $G_2 \sim H_2$. \blacksquare

FIG. 12. Nonconfluence of \Rightarrow .

The premise of Theorem 7.7 cannot be relaxed by dropping left-linearity or normalization, as is witnessed by Example 5.10 and 4.3, respectively. In these examples, \Rightarrow is not even confluent modulo bisimilarity. Moreover, normalization of \Rightarrow cannot be replaced by normalization of \rightarrow . We demonstrate this by a counterexample from [Plu93a].

EXAMPLE 7.8. Suppose that \mathcal{R} consists of the rules:

$$\begin{aligned} f(x) &\rightarrow g(x, x) \\ a &\rightarrow b \\ g(a, b) &\rightarrow c \\ g(b, b) &\rightarrow f(a). \end{aligned}$$

Using structural induction on terms, it is easy to verify that every term has a unique normal form. Hence, \rightarrow is normalizing and confluent. But Fig. 12 shows that \Rightarrow is neither confluent nor confluent modulo bisimilarity. (Notice that there is no graph rewrite step $\nabla g(a, a) \Rightarrow \Delta g(a, b)$ corresponding to the term rewrite step $g(a, a) \rightarrow g(a, b)$.)

We finally remark that the assumptions of Theorem 7.7 do not guarantee that \Rightarrow is confluent. This can be seen from Example 4.4. There, \Rightarrow is even terminating and \mathcal{R} is “almost orthogonal”; that is, every two overlapping term rewrite steps $t_1 \leftarrow t \rightarrow t_2$ satisfy $t_1 = t_2$ and the overlap occurs at the roots of the left-hand sides of the applied rules.

8. REWRITING MODULO BISIMILARITY

We have seen that the relation \Rightarrow_{bi} behaves nicely with respect to confluence in that it is confluent if and only if term rewriting is confluent. A drawback of \Rightarrow_{bi} , as remarked in the previous section, is that it is nonterminating whenever there is a term graph containing two different nodes representing the same term. We show

that this problem disappears when moving from \Rightarrow_{bi} to rewriting of bisimilarity classes. It turns out that term graph rewriting modulo bisimilarity behaves with respect to confluence and termination exactly like term rewriting.

DEFINITION 8.1 (\Rightarrow_{\sim}). The relation \Rightarrow_{\sim} on bisimilarity classes is defined as $[G] \Rightarrow_{\sim} [H]$ if $G' \Rightarrow H'$ for some $G' \in [G]$ and $H' \in [H]$. The reflexive-transitive closure of \Rightarrow_{\sim} is denoted by \Rightarrow_{\sim}^* . We refer to \Rightarrow_{\sim} as *term graph rewriting modulo bisimilarity*.

LEMMA 8.2. *For all term graphs G and H ,*

$$[G] \Rightarrow_{\sim}^* [H] \text{ if and only if } G \Rightarrow_{\text{bi}}^* H.$$

Proof. (If) By a straightforward induction on the length of the derivation $G \Rightarrow_{\text{bi}}^* H$.

(Only if) By induction on the length of $[G] \Rightarrow_{\sim}^* [H]$. If $[G] = [H]$, then $G \leq \Delta G \geq H$ and, hence, $G \Rightarrow_{\text{bi}}^* H$. Suppose now that $[G] \Rightarrow_{\sim}^* [M] \Rightarrow_{\sim} [H]$ for some term graph M , where $G \Rightarrow_{\text{bi}}^* M$. Then there are term graphs M' and H' such that $M \sim M' \Rightarrow H' \sim H$. It follows $M \Rightarrow_{\text{bi}}^* M' \Rightarrow_{\text{bi}} H' \Rightarrow_{\text{bi}}^* H$; thus $G \Rightarrow_{\text{bi}}^* H$. ■

Notice that the above equivalence does not hold for the transitive closures of \Rightarrow_{\sim} and \Rightarrow_{bi} . For, if $G \Rightarrow_{\text{bi}} H$ is a collapse or copy step, then $[G] \Rightarrow_{\sim}^+ [H]$ will not hold in general.

THEOREM 8.3. *The following are equivalent:*

- (1) \Rightarrow_{\sim} is confluent.
- (2) \Rightarrow_{bi} is confluent.
- (3) \rightarrow is confluent.

Proof. The equivalence of (1) and (2) follows from Lemma 8.2, while (2) and (3) are equivalent by Corollary 7.2. ■

Now we are going to show that rewriting of bisimilarity classes terminates if and only if term rewriting terminates. Actually, the next lemma says more: term graph rewriting modulo \sim generalizes term rewriting in that every step corresponds to a nonempty sequence of term rewrite steps, while every term rewrite step corresponds to a class rewrite step.

LEMMA 8.4. *For all term graphs G and H ,*

- (1) $[G] \Rightarrow_{\sim} [H]$ implies $\text{term}(G) \rightarrow^+ \text{term}(H)$, and
- (2) $\text{term}(G) \rightarrow \text{term}(H)$ implies $[G] \Rightarrow_{\sim} [H]$.

Proof. (1) This holds by soundness of \Rightarrow , see Theorem 3.5.

(2) In the proof of Theorem 7.1 it is shown that for every term rewrite step $t \rightarrow u$ there are term graphs T and U such that $\Delta t \geq T \Rightarrow U \leq \Delta u$. Hence, there are G' and H' such that $\Delta G \geq G' \Rightarrow H' \leq \Delta H$. Then $G \sim G'$ and $H \sim H'$; hence $[G] \Rightarrow_{\sim} [H]$. ■

TABLE 1
Overview of Confluence

	\Rightarrow	$\Rightarrow_{\text{coll}}$	$\Rightarrow_{\text{copy}}$	\Rightarrow_{bi}	\Rightarrow_{\sim}
\mathcal{R} orthogonal	+	− ^{6.2}	+	+	+
\mathcal{R} non-overlapping	+ ^{5.3}	−	− ^{5.10}	− ^{5.9}	−
\rightarrow confluent	−	−	−	+ ^{7.2}	+
\mathcal{R} left-linear, \rightarrow confluent	−	−	+ ^{7.4}	+	+
\rightarrow confluent, $\Rightarrow_{\text{coll}}$ normalizing	−	+ ^{7.5}	− ^{5.10}	+	+
\mathcal{R} left-linear, \rightarrow confluent & terminating	− ^{4.4}	+	+	+	+

THEOREM 8.5. *The relation \Rightarrow_{\sim} is terminating if and only if \rightarrow is.*

Proof. Immediate consequence of Lemma 8.4. ■

9. CONCLUSION

Our positive and negative results on confluence, confluence modulo bisimilarity, and the Church–Rosser property modulo bisimilarity are summarized in Tables 1 and 2. In both tables, a “+” means that the respective confluence property holds under the given conditions, while a “−” indicates that there exists a counterexample. Exponents refer to the corresponding results and counterexamples.

TABLE 2
Confluence Properties Modulo

	\Rightarrow confluent modulo \sim	\Rightarrow Church–Rosser modulo \sim
\mathcal{R} orthogonal	+ ^{6.4}	− ^{6.2}
\mathcal{R} non-overlapping \rightarrow confluent & terminating	− ^{5.10}	−
\mathcal{R} left-linear, \rightarrow confluent & normalizing	− ^{7.8}	−
\mathcal{R} left-linear, \rightarrow confluent, \Rightarrow normalizing	+	+ ^{7.7}

The conditions for confluence considered in this paper either forbid overlaps between term rewrite rules or require confluence of the associated term rewrite relation. Another tool for analyzing confluence are *critical pairs* of term graph rewrite steps. We refer to [Plu94] for their definition and their use to decide confluence of $\Rightarrow_{\text{coll}}$ in the presence of termination.

Finally, proving confluence of term graph rewriting can be facilitated by combination results. For example, it is known for $\Rightarrow_{\text{coll}}$ that confluence, together with termination, is preserved by the disjoint union of two term rewriting systems. A summary of combination results for $\Rightarrow_{\text{coll}}$ is given in [KR98].

ACKNOWLEDGMENT

We thank the anonymous referees for their useful comments.

Received December 1997; final manuscript received June 14, 1999

REFERENCES

- [AK96] Ariola, Z. M., and Klop, J. W. (1996), Equational term graph rewriting, *Fund. Inform.* **26**, 207–240.
- [AKP97] Ariola, Z. M., Klop, J. W., and Plump, D. (1997), Confluent rewriting of bisimilar term graphs, in “Proc. Fourth Workshop on Expressiveness in Concurrency,” *Electronic Notes in Theoretical Computer Science*, Vol. 7, Elsevier, Amsterdam.
- [AS85] Abelson, H., and Sussman, G. J. (1985), “Structure and Interpretation of Computer Programs,” MIT Press, Cambridge, MA.
- [BN98] Baader, F., and Nipkow, T. (1998), “Term Rewriting and All That,” Cambridge Univ. Press, Cambridge.
- [BvEG⁺87] Barendregt, H., van Eekelen, M., Glauert, J., Kennaway, R., Plasmeijer, R., and Sleep, R. (1987), Term graph rewriting, in “Proc. Parallel Architectures and Languages Europe,” *Lecture Notes in Computer Science*, Vol. 259, p. 141–158, Springer-Verlag, Berlin.
- [CD97] Corradini A., and Drewes F. (1997), “(Cyclic) Term Graph Rewriting Is Adequate For Rational Parallel Term Rewriting,” Technical Report TR-97-14, Università di Pisa, Dipartimento di Informatica.
- [DJ90] Dershowitz N., and Jouannaud, J.-P. (1990), Rewrite systems, in “Handbook of Theoretical Computer Science” (J. van Leeuwen, Ed.), Vol. B, Chap. 6, p. 244–320, Elsevier, Amsterdam.
- [HL91] Huet G., and Lévy, J.-J. (1991), Computations in orthogonal rewrite systems, I, II, in “Computational Logic: Essays in Honor of Alan Robinson” (J. L. Lassez and G. Plotkin, Eds.) pp. 395–443, MIT Press, Cambridge, MA.
- [HP88] Hoffmann B., and Plump D. (1988), Jungle evaluation for efficient term rewriting, in “Proc. Algebraic and Logic Programming,” *Math. Res.* Vol. 49, pp. 191–203, Akad. Verlag, Berlin. [Also in “Lecture Notes in Computer Science,” Vol. 343, pp. 191–203, Springer-Verlag, Berlin, 1989].
- [HP96] Habel A., and Plump D. (1996), Term graph narrowing, *Math. Struct. Comput. Sci.* **6**, 649–676.
- [Hue80] Huet, G. (1980), Confluent reductions: Abstract properties and applications to term rewriting systems, *Assoc. Comput. Math.* **27**(4), 797–821.

- [KKSdV94] Kennaway, R., Klop, J. W., Sleep, R., and de Vries, F. J. (1994), On the adequacy of graph rewriting for simulating term rewriting, *ACM Trans. Programming Lang. Systems* **16**(3), 493–523.
- [Klo92] Klop, J. W. (1992), Term rewriting systems, in “Handbook of logic in Computer Science,” (S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, Eds), Vol. 2, pp. 1–116, Oxford Univ. Press, Oxford.
- [KR98] Krishna Rao, Ma. R. K. (1998), Modular aspects of term graph rewriting, *Theoret. Comput. Sci.* **208**(1–2), 59–86.
- [Plu93a] Plump, D. (1993), Collapsed tree rewriting: Completeness, confluence, and modularity, in “Proc. Conditional Term Rewriting Systems,” Lecture Notes in Computer Science, Vol. 656, 97–112. Springer-Verlag, Berlin.
- [Plu93b] Plump, D. (1993), Evaluation of functional expressions by hypergraph rewriting, Dissertation, Universität Bremen, Fachbereich Mathematik und Informatik.
- [Plu94] Plump D. (1994), Critical pairs in term graph rewriting, in “Proc. Mathematical Foundations of computer Science,” Lecture Notes in Computer Science, Vol. 841, pp. 556–566, Springer-Verlag, Berlin.
- [Plu98] Plump, D. (1998), Term graph rewriting, in “Handbook of Graph Grammars and Computing by graph Transformation (H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, Eds.), Vol. 2, World Scientific, Singapore.
- [Sta80] Staples J. (1980), Computation on graph-like expressions. *Theoret. Comput. Sci.* **10**, 171–185.