Electronic Communications of the EASST
Volume 41 (2011)

Proceedings of the Tenth International Workshop on Graph Transformation and Visual Modeling Techniques (GTVMT 2011)

Categorical Abstract Rewriting Systems and Functoriality of Graph Transformation

Dominique Duval, Rachid Echahed and Frédéric Prost

17 pages

Guest Editors: Fabio Gadducci, Leonardo Mariani
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Categorical Abstract Rewriting Systems and Functoriality of Graph Transformation

Dominique Duval[1], **Rachid Echahed**[2] **and Frédéric Prost**[3]

[1] Dominique.Duval@imag.fr
Laboratoire LJK – Université de Grenoble
B. P. 53, F-38041 Grenoble, France

[2] Rachid.Echahed@imag.fr [3] Frederic.Prost@imag.fr
Laboratoire LIG – Université de Grenoble
B. P. 53, F-38041 Grenoble, France

Abstract: Abstract rewriting systems are often defined as binary relations over a given set of objects. In this paper, we introduce a new notion of abstract rewriting system in the framework of categories. Then, we define the *functoriality* property of rewriting systems. This property is sometimes called *vertical composition*. We show that most graph transformation systems are functorial and provide a counter-example of graph transformation system which is not functorial.

Keywords: Graph Transformation, Abstract rewriting.

1 Introduction

Various properties of rewriting systems can be defined on an abstract level by using the notion of abstract rewriting systems (see e.g., [1]). In this paper we focus on categorical rewriting systems, that is to say rewriting systems defined by means of category theory, and we define them in an abstract manner. We consider rule-based frameworks in which the rewrite step is defined relatively to a match. The aim is to be able to reason abstractly about rewriting systems which are defined categorically. There are many such systems which underly graph transformation, following the seminal work of [10]. In general, a graph rewriting system consists of a set of graph rewrite rules with a left-hand side L and a right-hand side R (where both are graphs). When a graph rewrite rule is applied to an instance of the graph L in a graph L_1, it replaces this instance of L by an instance of R, resulting in a new graph R_1. We introduce categorical rewriting systems in Section 2, they provide an abstract framework for dealing with such notions of rewrite rules, instances and rewrite steps. Moreover, in a graph rewriting system, usually the given graph L_1 and the modified graph R_1 can be seen as the left-hand side and right-hand side of a new rule, from which the process can be repeated. Then the functoriality problem appears: from an instance of L in L_1 and an instance of L_1 in L_2, do we get the same graph R_2 when proceeding in two steps as when proceeding in one step? The functoriality property is sometimes called the vertical composition. It is similar to the contextual closure of term rewriting systems. It ensures also the soundness of replacement of equals by equals. This is a very desirable property of programming languages, since it allows one to instantiate programs in a safe manner as in partial evaluation techniques [2]. A recent work of M. Löwe [13] adresses a similar issue in a different setting in which matches are spans instead of morphisms. In Section 3 we check that the functoriality property holds for many usual algebraic graph transformation approaches like double pushouts (DPO) [4], single pushouts (SPO) [12], sesqui-pushouts (SqPO) [3] and heterogeneous pushouts (HPO) [6]. Then in Section 4 we look at garbage removal as a categorical rewriting system, in two different ways. This yields a categorical rewriting system which is functorial, and another one which is not functorial. We refer to [14] for categorical notions: mainly commutative diagrams, functors, pushouts and pullbacks, comma categories. The class of objects of a category $\mathscr{C}$ is denoted as $|\mathscr{C}|$. A subcategory $\mathscr{M}$ of a category $\mathscr{C}$ is called a wide subcategory of $\mathscr{C}$ if it has the same objects as $\mathscr{C}$.

2 Categorical rewriting systems

2.1 Definition of categorical rewriting systems

Definition 1 A *categorical rewriting system* $(\mathscr{L} : \mathscr{M}_L \leftarrow \mathscr{P} \rightarrow \mathscr{M}_R : \mathscr{R}, \mathbf{S})$ is made of a span of categories

$$\begin{tikzcd} & \mathscr{P} \arrow[dl, "\mathscr{L}"'] \arrow[dr, "\mathscr{R}"] & \\ \mathscr{M}_L & & \mathscr{M}_R \end{tikzcd}$$

and a family of partial functions

$$\mathbf{S} = (\mathbf{S}_\rho)_{\rho \in |\mathscr{P}|}$$

where for each object ρ in $\mathscr{P}$, the partial function $\mathbf{S}_\rho$, from the set of morphisms in $\mathscr{M}_L$ with source $\mathscr{L}(\rho)$ to the set of morphisms in $\mathscr{P}$ with source ρ, is such that $\mathscr{L}(\mathbf{S}_\rho(f)) = f$ for every f in the domain of $\mathbf{S}_\rho$. The objects of $\mathscr{P}$ are the *rewrite rules* or *productions*, the morphisms of $\mathscr{M}_L$ and $\mathscr{M}_R$ are the left-hand side and right-hand side *matches*, and the partial function $\mathbf{S}_\rho$ is the *rewriting process* function with respect to ρ; its domain is denoted as $\mathrm{Dom}(\mathbf{S}_\rho)$. Given a rule ρ, the *rewrite step applying* ρ is the partial function from the set of morphisms in $\mathscr{M}_L$ with source $\mathscr{L}(\rho)$ to the set of morphisms in $\mathscr{M}_R$ with source $\mathscr{R}(\rho)$ which maps every match f in $\mathrm{Dom}(\mathbf{S}_\rho)$ to the match $g = \mathscr{R}(\mathbf{S}_\rho(f))$. The target R_1 of g may be called the *derived* object, with respect to the rule ρ and the match f.

Remark 1 *Many categorical rewriting systems are such that $\mathscr{M}_L = \mathscr{M}_R$, then this category is denoted as $\mathscr{M}$. For the interested reader we refer to [7] as an example of a rewriting system defined by composition of rewriting systems (such composition is defined in Section 2.3) in which $\mathscr{M}_L \neq \mathscr{M}_R$.*

Remark 2 *Each categorical rewriting system with $\mathscr{M}_L = \mathscr{M}_R = \mathscr{M}$ determines an* abstract rewriting system *on the objects of $\mathscr{M}$, i.e., a binary relation $\rightsquigarrow$ on $|\mathscr{M}|$, defined by $L \rightsquigarrow R$ if and only if there is some ρ in $\mathscr{P}$ such that $L = \mathscr{L}(\rho)$ and $R = \mathscr{R}(\rho)$.*

In a categorical rewriting system, the matches introduce a "vertical dimension", in addition to the "horizontal dimension" provided by the rules. A rule ρ with $\mathscr{L}(\rho) = L$ and $\mathscr{R}(\rho) = R$ is denoted as $\rho : L \rightsquigarrow R$. It should be noted that, although ρ is an *object* in the category $\mathscr{P}$, it is represented as an *arrow* from its left-hand side L to its right-hand side R; this refers to the usual notation for rewriting systems. Whenever $\mathscr{P}$ is a category of arrows, it may happen that ρ actually is a morphism in some category $\mathscr{D}$, with either $\rho : L \to R$ (as in Sections 3.1 and 3.2) or $\rho : R \to L$ (as in Sections 3.3 and 3.4). A morphism $\pi : \rho \to \rho_1$ in $\mathscr{P}$, with $\mathscr{L}(\pi) = f : L \to L_1$ and $\mathscr{R}(\pi) = g : R \to R_1$, is illustrated as follows:

$$\begin{tikzcd} L \arrow[r, squiggly, "\rho"] \arrow[d, "f"'] \arrow[dr, phantom, "\pi"] & R \arrow[d, "g"] \\ L_1 \arrow[r, squiggly, "\rho_1"'] & R_1 \end{tikzcd}$$

Then, each rewriting process $\mathbf{S}_\rho$ can be illustrated as:

$$\begin{tikzcd} L \arrow[r, squiggly, "\rho"] \arrow[d, "f"'] & R \\ L_1 & \end{tikzcd} \quad \overset{\mathbf{S}_\rho}{\longmapsto} \quad \begin{tikzcd} L \arrow[r, squiggly, "\rho"] \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_\rho(f)"] & R \arrow[d, "g"] \\ L_1 \arrow[r, squiggly, "\rho_1"'] & R_1 \end{tikzcd}$$

For instance, Definition 2 below provides categorical rewriting systems based on pushouts. As usual a category *with pushouts* is a category $\mathscr{C}$ such that for every morphisms f and ρ in $\mathscr{C}$ with the same source, the pushout of ρ and f exists in $\mathscr{C}$. The *category of arrows* of any category $\mathscr{C}$ is denoted $\mathscr{C}^{\to}$: its objects are the morphisms of $\mathscr{C}$ and its morphisms are the commutative squares in $\mathscr{C}$.

Definition 2 Let $\mathscr{C}$ be a category with chosen pushouts. The *categorical rewriting system based on pushouts in* $\mathscr{C}$, denoted as $\mathrm{RS}_{PO,\mathscr{C}}$, is made of the categories $\mathscr{M}_L = \mathscr{M}_R = \mathscr{C}$ and $\mathscr{P} = \mathscr{C}^{\to}$, the source functor $\mathscr{L} = Src : \mathscr{C}^{\to} \to \mathscr{C}$, the target functor $\mathscr{R} = Tgt : \mathscr{C}^{\to} \to \mathscr{C}$, and the family of functions $\mathbf{S}_{PO}$ such that for each rule ρ the function $\mathbf{S}_{PO,\rho}$ is total and for each match f the commutative square $\mathbf{S}_{PO,\rho}(f)$ is defined as the chosen pushout of ρ and f in $\mathscr{C}$.

$$\begin{tikzcd} L \arrow[r, "\rho"] \arrow[d, "f"'] & R \\ L_1 & \end{tikzcd} \quad \overset{\mathbf{S}_{PO,\rho}}{\longmapsto} \quad \begin{tikzcd} L \arrow[r, "\rho"] \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_{PO,\rho}(f)"] & R \arrow[d, "g"] \\ L_1 \arrow[r, "\rho_1"'] & R_1 \arrow[ul, phantom, "\ulcorner", very near start] \end{tikzcd}$$

In Section 3, we consider categorical rewriting systems which generalize the pushout rewriting systems. There is a need for these generalizations, since there may be restrictions (e.g., injectivity conditions or gluing conditions) on the morphisms used for rules and for matches. These generalizations are built according to the following patterns.

Definition 3 Let $\mathscr{C}$ be a category with two wide subcategories $\mathscr{M}$ and $\mathscr{D}$. The *generalized arrow category* $\mathscr{D}^{\to\mathscr{M}}$ (in $\mathscr{C}$) is the following category: the objects in $\mathscr{D}^{\to\mathscr{M}}$ are the morphisms in $\mathscr{D}$, and the morphisms from ρ to ρ_1 in $\mathscr{D}^{\to\mathscr{M}}$, where $\rho : L \to R$ and $\rho_1 : L_1 \to R_1$ in $\mathscr{D}$, are the pairs $(f : L \to L_1, g : R \to R_1)$ of morphisms in $\mathscr{M}$ such that $g \circ \rho = \rho_1 \circ f$ in $\mathscr{C}$. The *source functor* $Src : \mathscr{D}^{\to\mathscr{M}} \to \mathscr{M}$ and the *target functor* $Tgt : \mathscr{D}^{\to\mathscr{M}} \to \mathscr{M}$ map each object ρ in $\mathscr{D}^{\to\mathscr{M}}$ to its source and target, when ρ is seen as a morphism in $\mathscr{D}$; they map each morphism (f,g) in $\mathscr{D}^{\to\mathscr{M}}$ to the morphisms f and g in $\mathscr{M}$, respectively.

This situation yields two spans of categories where $\mathscr{M}_L = \mathscr{M}_R = \mathscr{M}$ and $\mathscr{P} = \mathscr{D}^{\to\mathscr{M}}$, as defined below; these spans will be used for describing graph transformation systems as categorical rewriting systems in Sections 3 and 4.

Definition 4 Let $\mathscr{C}$ be a category with two wide subcategories $\mathscr{M}$ and $\mathscr{D}$. Let $\mathscr{D}^{\to\mathscr{M}}$ denote the corresponding generalized arrow category and $Src, Tgt : \mathscr{D}^{\to\mathscr{M}} \to \mathscr{M}$ the source and target functors.

- The *direct arrows-based span on $\mathscr{C}$ with rules in $\mathscr{D}$ and matches in $\mathscr{M}$* is the span of categories $(Src : \mathscr{M} \leftarrow \mathscr{D}^{\to\mathscr{M}} \to \mathscr{M} : Tgt)$. This means that a rule $\rho : L \rightsquigarrow R$ is a morphism $\rho : L \longrightarrow R$ in $\mathscr{D}$, a match is a morphism in $\mathscr{M}$ and a morphism of rules (from ρ to ρ_1) is a commutative square in $\mathscr{C}$ with f, g in $\mathscr{M}$:

$$\begin{tikzcd} L \arrow[r, "\rho"] \arrow[d, "f"'] \arrow[dr, phantom, "="] & R \arrow[d, "g"] \\ L_1 \arrow[r, "\rho_1"'] & R_1 \end{tikzcd}$$

- The *inverse arrows-based span on $\mathscr{C}$ with rules in $\mathscr{D}$ and matches in $\mathscr{M}$* is the span of categories $(Tgt : \mathscr{M} \leftarrow \mathscr{D}^{\to\mathscr{M}} \to \mathscr{M} : Src)$. This means that a rule $\rho : L \rightsquigarrow R$ is a morphism $\rho : R \longrightarrow L$ in $\mathscr{D}$, a match is a morphism in $\mathscr{M}$ and a morphism of rules (from ρ to ρ_1) is a commutative square in $\mathscr{C}$ with f, g in $\mathscr{M}$:

$$\begin{tikzcd} L \arrow[d, "f"'] \arrow[dr, phantom, "="] & R \arrow[l, "\rho"'] \arrow[d, "g"] \\ L_1 & R_1 \arrow[l, "\rho_1"] \end{tikzcd}$$

2.2 Functoriality of categorical rewriting systems

A categorical rewriting system, when it is seen as an abstract rewriting system, is read "horizontally": it maps the left-hand side match $f : L \to L_1$ to the right-hand side match $g : R \to R_1$. But it may also be read "vertically": it maps the rule $\rho : L \rightsquigarrow R$ to the rule $\rho_1 : L_1 \rightsquigarrow R_1$. In this section we study a functoriality property of categorical rewriting systems from this "vertical" point of view; a similar property is called "vertical composition" in [13]. The statements and results below are given up to isomorphism.

Definition 5 A categorical rewriting system $(\mathscr{L} : \mathscr{M}_L \leftarrow \mathscr{P} \to \mathscr{M}_R : \mathscr{R}, \mathbf{S})$ is *functorial* if for each rule $\rho : L \rightsquigarrow R$ the partial function $\mathbf{S}_\rho$ satisfies:

- the identity id_L is in the domain of $\mathbf{S}_\rho$ and

$$\mathbf{S}_\rho(\mathrm{id}_L) = \mathrm{id}_\rho\ .$$

$$\begin{tikzcd} L \arrow[r, rightsquigarrow, "\rho"] \arrow[d, "\mathrm{id}_L"'] & R \\ L & \end{tikzcd} \quad \overset{\mathbf{S}_\rho}{\longmapsto} \quad \begin{tikzcd} L \arrow[r, rightsquigarrow, "\rho"] \arrow[d, "\mathrm{id}_L"'] \arrow[dr, phantom, "\mathrm{id}_\rho"] & R \arrow[d, "\mathrm{id}_R"] \\ L \arrow[r, rightsquigarrow, "\rho"'] & R \end{tikzcd}$$

- and for each pair of consecutive morphisms $f_1 : L \to L_1$ and $f_2 : L_1 \to L_2$ in $\mathscr{M}_L$, if $f_1 \in \mathrm{Dom}(\mathbf{S}_\rho)$ and $f_2 \in \mathrm{Dom}(\mathbf{S}_{\rho_1})$, where ρ_1 denotes the target of $\mathbf{S}_\rho(f_1)$, then $f_2 \circ f_1 \in \mathrm{Dom}(\mathbf{S}_\rho)$ and
$$\mathbf{S}_{\rho_1}(f_2) \circ \mathbf{S}_\rho(f_1) = \mathbf{S}_\rho(f_2 \circ f_1) \; .$$

$$\begin{tikzcd}
L \arrow[r, rightsquigarrow, "\rho"] \arrow[d, "f_1"'] & R & & L \arrow[r, rightsquigarrow, "\rho"] \arrow[d, "f_1"'] \arrow[dr, phantom, "\mathbf{S}_\rho(f_1)"] & R \arrow[d, "g_1"] & & L \arrow[r, rightsquigarrow, "\rho"] \arrow[dd, "f_2 \circ f_1"'] \arrow[ddr, phantom, "\mathbf{S}_\rho(f_2 \circ f_1)"] & R \arrow[dd, "g_2 \circ g_1"] \\
L_1 \arrow[d, "f_2"'] & & \arrow[r, mapsto, "\mathbf{S}_\rho"] & L_1 \arrow[r, rightsquigarrow, "\rho_1"] \arrow[d, "f_2"'] \arrow[dr, phantom, "\mathbf{S}_{\rho_1}(f_2)"] & R_1 \arrow[d, "g_2"] & = & & \\
L_2 & & & L_2 \arrow[r, rightsquigarrow, "\rho_2"'] & R_2 & & L_2 \arrow[r, rightsquigarrow, "\rho_2"'] & R_2
\end{tikzcd}$$

For instance, using Definition 2, the next result is due to the well-known compositionality property of pushouts.

Proposition 1 *Let $\mathscr{C}$ be a category with pushouts. The categorical rewriting system $\mathrm{RS}_{PO,\mathscr{C}}$ is functorial.*

Remark 3 *For any category $\mathscr{C}$ and any object X in $\mathscr{C}$, let $X \downarrow \mathscr{C}$ denote the coslice category of objects of $\mathscr{C}$ under X. Then the objects of $X \downarrow \mathscr{C}$ are the morphisms in $\mathscr{C}$ with source X. Let $\mathrm{RS} = (\mathscr{L} : \mathscr{M}_L \leftarrow \mathscr{P} \to \mathscr{M}_R : \mathscr{R}, \mathbf{S})$, be a categorical rewriting system. For each rule $\rho : L \rightsquigarrow R$ let $\mathscr{L}_\rho : \rho \downarrow \mathscr{P} \to L \downarrow \mathscr{M}_L$ denote the functor induced by $\mathscr{L}$. Then $\mathbf{S}_\rho$ can be seen as a partial function $\mathbf{S}_\rho : |L \downarrow \mathscr{M}_L| \rightharpoonup |\rho \downarrow \mathscr{P}|$ such that $\mathscr{L}_\rho \circ \mathbf{S}_\rho$ is the identity of $\mathrm{Dom}(\mathbf{S}_\rho)$. The name "functorial" comes from this interpretation of categorical rewriting systems: let $\mathrm{RS} = (\mathscr{L} : \mathscr{M}_L \leftarrow \mathscr{P} \to \mathscr{M}_R : \mathscr{R}, \mathbf{S})$ be a categorical rewriting system, and let us assume that for each rule $\rho : L \rightsquigarrow R$ the rewriting process $\mathbf{S}_\rho$ is total, which means that it is a total function $\mathbf{S}_\rho : |L \downarrow \mathscr{M}_L| \rightharpoonup |\rho \downarrow \mathscr{P}|$ such that $\mathscr{L}_\rho \circ \mathbf{S}_\rho$ is the identity of $|L \downarrow \mathscr{M}_L|$. For each morphism $h : f_1 \to f_2$ in $L \downarrow \mathscr{M}_L$, i.e., for each morphism $h : L_1 \to L_2$ in $\mathscr{M}_L$ such that $h \circ f_1 = f_2$, let us define $\mathbf{S}_\rho(h : f_1 \to f_2) = \mathbf{S}_{\rho_1}(h)$ where ρ_1 is the target of $\mathbf{S}_\rho(f_1)$ in $\mathscr{P}$. Then it can be proved that RS is functorial if and only if for each rule $\rho : L \rightsquigarrow R$, $\mathbf{S}_\rho(\mathrm{id}_L) = \mathrm{id}_\rho$ and $\mathbf{S}_\rho$ is a functor $\mathbf{S}_\rho : L \downarrow \mathscr{M}_L \to \rho \downarrow \mathscr{P}$.*

2.3 Composition of categorical rewriting systems

In order to compose ("horizontally") categorical rewriting systems, we use composition of spans: given two spans of categories $\mathscr{L} : \mathscr{M}_L \leftarrow \mathscr{P} \to \mathscr{M}_R : \mathscr{R}$ and $\mathscr{L}' : \mathscr{M}'_L \leftarrow \mathscr{P}' \to \mathscr{M}'_R : \mathscr{R}'$ which are consecutive, in the sense that $\mathscr{M}_R = \mathscr{M}'_L$, the composed span $\mathscr{L}'' : \mathscr{M}_L \leftarrow \mathscr{P}'' \to \mathscr{M}'_R : \mathscr{R}''$ is obtained from the pullback of $\mathscr{R}$ and $\mathscr{L}'$, as follows:

$$\begin{tikzcd}
& & \mathscr{P}'' \arrow[dl] \arrow[dr] \arrow[ddll, bend right, "\mathscr{L}''"'] \arrow[ddrr, bend left, "\mathscr{R}''"] & & \\
& \mathscr{P} \arrow[dl, "\mathscr{L}"'] \arrow[dr, "\mathscr{R}"] & & \mathscr{P}' \arrow[dl, "\mathscr{L}'"'] \arrow[dr, "\mathscr{R}'"] & \\
\mathscr{M}_L & & \mathscr{M}_R = \mathscr{M}'_L & & \mathscr{M}'_R
\end{tikzcd}$$

The objects of $\mathscr{P}''$ are the pairs (ρ,ρ') with ρ in $\mathscr{P}$ and ρ' in $\mathscr{P}'$ such that $\mathscr{R}(\rho) = \mathscr{L}'(\rho')$. The morphisms from $\rho'' = (\rho,\rho')$ to $\rho''_1 = (\rho_1,\rho'_1)$ in $\mathscr{P}''$ are the pairs $\pi'' = (\pi,\pi')$ where $\pi : \rho \to \rho_1$ in $\mathscr{P}$ and $\pi' : \rho' \to \rho'_1$ in $\mathscr{P}'$ are such that $\mathscr{R}(\pi) = \mathscr{L}'(\pi')$.

Definition 6 Let $\mathrm{RS} = (\mathscr{L} : \mathscr{M}_L \leftarrow \mathscr{P} \to \mathscr{M}_R : \mathscr{R}, \mathbf{S})$ and $\mathrm{RS}' = (\mathscr{L}' : \mathscr{M}'_L \leftarrow \mathscr{P}' \to \mathscr{M}'_R : \mathscr{R}', \mathbf{S}')$ be two categorical rewriting systems which are consecutive, in the sense that $\mathscr{M}_R = \mathscr{M}'_L$. The *composition* of RS and RS′ is the categorical rewriting system

$$\mathrm{RS}' \circ \mathrm{RS} = (\mathscr{L}'' : \mathscr{M}_L \leftarrow \mathscr{P}'' \to \mathscr{M}'_R : \mathscr{R}'', \mathbf{S}''_{(\rho,\rho')})$$

where $\mathscr{L}'' : \mathscr{M}_L \leftarrow \mathscr{P}'' \to \mathscr{M}'_R : \mathscr{R}''$ is the composition of the spans in RS and RS′ and where the family of partial functions $\mathbf{S}'' = (\mathbf{S}''_{\rho''})_{\rho'' \in |\mathscr{P}''|}$ is defined as follows, for each $\rho'' = (\rho,\rho')$ in $\mathscr{P}''$: the domain of $\mathbf{S}''_{\rho''}$ is made of the morphisms f in $\mathrm{Dom}(\mathbf{S}_\rho)$ such that $\mathscr{R}(\mathbf{S}_\rho(f))$ is in $\mathrm{Dom}(\mathbf{S}'_{\rho'})$, and for each $f \in \mathrm{Dom}(\mathbf{S}''_{\rho''})$:

$$\mathbf{S}''_{(\rho,\rho')}(f) = (\mathbf{S}_\rho(f), \mathbf{S}'_{\rho'}(f')) \ \text{ where } \ f' = \mathscr{R}(\mathbf{S}_\rho(f)) \; .$$

$$\begin{tikzcd}
L \arrow[r, rightsquigarrow, "\rho"] \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_\rho(f)"] & R = L' \arrow[r, rightsquigarrow, "\rho'"] \arrow[d, "f'"] \arrow[dr, phantom, "\mathbf{S}_{\rho'}(f')"] & R' \arrow[d, "f''"] \\
L_1 \arrow[r, rightsquigarrow, "\rho_1"'] & R_1 = L'_1 \arrow[r, rightsquigarrow, "\rho'_1"'] & R'_1
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
L \arrow[r, rightsquigarrow, "\rho''"] \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_{(\rho,\rho')}(f)"] & R' \arrow[d, "f''"] \\
L_1 \arrow[r, rightsquigarrow, "\rho''_1"'] & R'_1
\end{tikzcd}$$

This composition gives rise to the bicategory of categorical rewriting systems (as for spans, we get a bicategory rather than a category, because the uniqueness of pushouts is only up to isomorphim). The next result follows easily from the definitions.

Proposition 2 *Let* RS *and* RS′ *be two consecutive categorical rewriting systems. If* RS *and* RS′ *are functorial then* $\mathrm{RS}' \circ \mathrm{RS}$ *is functorial.*

3 Functoriality of graph transformations

Since [10], several graph transformation systems have been studied following the algebraic approach. We show that many of them can be seen as categorical rewriting systems which satisfy the functoriality property. A direct arrows-based span is used in Sections 3.1 and 3.2 for single pushout and heterogeneous pushout rewriting systems. In Sections 3.3 and 3.4, for double pushout and sesqui-pushout rewriting systems, an inverse arrows-based span is used, then a direct one, and finally both are composed according to Definition 6.

Definition 7 A *graph* is a set of *nodes* and a set of *edges* with two functions from edges to nodes called the *source* and the *target* functions. A *morphism of graphs* is made of a function on nodes and a function on edges which preserve the sources and targets. This provides the category of graphs, denoted as **Graph**.

3.1 Single Pushout rewriting

In this section we show that, under suitable assumptions, the single pushout approach to graph transformation (SPO) [9] can be seen as a categorical rewriting system. Let $\mathscr{M}_{SPO} = \mathbf{Graph}$ be the category of graphs. Let $\mathscr{C}_{SPO} = \mathbf{Graph^p}$ be the category of graphs with partial morphisms, so that $\mathscr{M}_{SPO}$ can be seen as a wide subcategory of $\mathscr{C}_{SPO}$. Let $\mathscr{D}_{SPO} = \mathbf{Graph^p_m}$ be the wide subcategory of $\mathscr{C}_{SPO}$ with partial monomorphisms. We consider the direct arrows-based span on $\mathscr{C}_{SPO}$ with rules in $\mathscr{D}_{SPO}$ and matches in $\mathscr{M}_{SPO}$. Following [9, Definition 7], given a rule $r : L \rightarrowtail R$, we say that a match $f : L \longrightarrow L_1$ is *conflict-free* with respect to r when f does not identify any item (node or edge) in the domain of r with an item outside this domain. For each rule $r : L \rightarrowtail R$, we define $\mathbf{S}_{SPO,r}$ as the partial function with domain the conflict-free matches with respect to r, such that $\mathbf{S}_{SPO,r}(f)$ is the pushout of f and r in $\mathbf{Graph^p}$ for each f in $\mathrm{Dom}(\mathbf{S}_{SPO,r})$. It follows from [9, Proposition 5 and Lemma 8] that this pushout exists, that r_1 is a partial monomorphism and that g is a total morphism.

$$\begin{tikzcd}
L \arrow[r, tail, "r"] \arrow[d, "f"'] & R & {} \arrow[r, mapsto, "\mathbf{S}_{SPO,r}"] & {} & L \arrow[r, tail, "r"] \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_{SPO,r}(f)"] & R \arrow[d, "g"] \\
L_1 & & & & L_1 \arrow[r, tail, "r_1"'] & R_1
\end{tikzcd}$$

Definition 8 The *categorical rewriting system for graphs based on single pushouts*, denoted as RS_{SPO}, is made of the direct arrows-based span on $\mathscr{C}_{SPO} = \mathbf{Graph^p}$ with rules in $\mathscr{D}_{SPO} = \mathbf{Graph}$ and matches in $\mathscr{M}_{SPO} = \mathbf{Graph^p_m}$ together with the family of partial functions $\mathbf{S}_{SPO}$ defined as above from pushouts in $\mathbf{Graph^p}$.

Lemma 1 *Let us consider the categorical rewriting system* RS_{SPO}*. Let* $r : L \rightarrowtail R$ *be a rule and* $f_1 : L \longrightarrow L_1$ *a match which is conflict-free with respect to* r*. Let* R_1 *with* $r_1 : L_1 \rightarrowtail R_1$ *and* $g_1 : R \longrightarrow R_1$ *be the pushout of* r *and* f_1 *in* $\mathbf{Graph^p}$*. Let* $f_2 : L_1 \longrightarrow L_2$ *be a match which is conflict-free with respect to* r_1*. Then* $f_2 \circ f_1$ *is conflict-free with respect to* r*.*

Proof. Let $f = f_2 \circ f_1 : L \longrightarrow L_2$. The proof is done by contradiction. Let us assume that there are two items x and y in L such that $f(x) = f(y)$, with $x \in \mathrm{Dom}(r)$ and $y \notin \mathrm{Dom}(r)$. Then there are two cases:

1. If $f_1(x) = f_1(y)$ then f_1 is not conflict-free with respect to r.
2. Otherwise let $x_1 = f_1(x)$ and $y_1 = f_1(y)$, so that $f_2(x_1) = f_2(y_1)$. The commutativity of the square $\mathbf{S}_{SPO,r}(f_1)$ is written as $g_1 \circ r = r_1 \circ f_1$. This implies that $g_1 \circ r$ and $r_1 \circ f_1$ have the same domain, and since f_1 and g_1 are total this means that for each item x in L, $x \in \mathrm{Dom}(r)$ if and only if $f_1(x) \in \mathrm{Dom}(r_1)$. Thus, $x_1 \in \mathrm{Dom}(r_1)$ and $y_1 \notin \mathrm{Dom}(r_1)$, so that f_2 is not conflict-free with respect to r_1.

$\square$

Proposition 3 *The categorical rewriting system* RS_{SPO} *is functorial.*

Proof. This is due to Lemma 1 and to the well-known compositionality property of pushouts. $\square$

3.2 Heterogeneous pushout rewriting

We now consider the heterogeneous pushout framework (HPO) presented in [6], which allows some deletion and cloning in the context of termgraph rewriting. Given a set called the set of *labels*, with an *arity* (a natural number) for each label, a *termgraph* is a graph where some nodes are labeled, when a node n has a label ℓ then the successors of n form a totally ordered set and their number is the arity of ℓ, and when a node n is unlabeled then it has no successor. If G is a termgraph then $|G|$ denotes the set of nodes of G. A *morphism of termgraphs* (respectively a *partial morphism of termgraphs*) is a morphism of graphs (respectively a partial morphism of graphs) which maps labeled nodes to labeled nodes, preserving the labels and the ordering of the successors. This provides the category of termgraphs **TermGraph**. Let $\mathscr{M}_{HPO} = \mathbf{TermGraph_m}$ be the wide subcategory of $\mathbf{TermGraph_m}$ with monomorphisms. Let $\mathscr{C}_{HPO}$ be the category with the termgraphs as objects and with morphisms from L to R the pairs (τ, σ) of partial termgraph morphisms $\tau : L \rightharpoonup R$ and $\sigma : R \rightharpoonup L$. Then $\mathscr{M}_{HPO}$ is considered as a wide subcategory of $\mathscr{C}_{HPO}$ by identifying each total morphism of termgraphs $f : L \longrightarrow L_1$ to the pair (f, ω) where $\omega : L_1 \rightharpoonup L$ is nowhere defined. Let $\mathscr{D}_{HPO}$ be the wide subcategory of $\mathscr{C}_{HPO}$ with morphisms the pairs $\rho = (\tau, \sigma) : L \hookrightarrow R$ such that the domain of τ is the set of nodes of L and the domain of σ is a subset of the set of nodes of R. Moreover, every node $p \in |R|$ in the domain of σ is either unlabelled or such that the node $q = \sigma(p) \in |L|$ is such that p and q share the same label and the successors of p in R are the image by τ of the successors of q in L.

We consider the direct arrows-based span on $\mathscr{C}_{HPO}$ with rules in $\mathscr{D}_{HPO}$ and matches in $\mathscr{M}_{HPO}$. Following [6, Definitions 6 and 7], for each rule $\rho : L \hookrightarrow R$ and each match $f : L \rightarrowtail L_1$, a *heterogeneous cocone over* ρ *and* f is made of a rule $\rho_1 : L_1 \hookrightarrow R_1$ and a match $g : R \rightarrowtail R_1$ such that $\rho_1 \circ f = g \circ \rho$ in $\mathscr{C}_{HPO}$. A *morphism of heterogeneous cocones over* ρ *and* f, say $h : (\rho_1, g) \to (\rho_1', g')$, is a morphism $h : R_1 \longrightarrow R_1'$ in $\mathscr{M}_{HPO}$ such that $h \circ \rho_1 = \rho_1'$ and $h \circ g = g'$ in $\mathscr{C}_{HPO}$. This yields the category of heterogeneous cocones over ρ and f, and a *heterogeneous pushout of* ρ *and* f is defined as an initial object in this category. The uniqueness of the heterogeneous pushout, up to isomorphism, is a consequence of its initiality property. Its existence is proven in [6, theorem 1] by providing an explicit construction. For each rule $\rho : L \hookrightarrow R$ let us define $\mathbf{S}_{HPO,\rho}$ as the total function such that $\mathbf{S}_{HPO,\rho}(f)$ is the heterogeneous pushout of f and ρ for each match f, which is denoted as:

$$\begin{tikzcd} L \arrow[r, hook, "\rho"] \arrow[d, tail, "f"'] & R & & L \arrow[r, hook, "\rho"] \arrow[d, tail, "f"'] \arrow[dr, phantom, "\mathbf{S}_{HPO,\rho}(f)"] & R \arrow[d, tail, "g"] \\ L_1 & & \arrow[r, mapsto, "\mathbf{S}_{HPO,\rho}"] & \; & L_1 \arrow[r, hook, "\rho_1"'] & R_1 \end{tikzcd}$$

It follows from [6, Proposition 1] that this construction provides a rule ρ_1 and a match g, so that we get a categorical rewriting system.

Definition 9 The *categorical rewriting system for termgraphs based on heterogeneous pushouts*, denoted as RS_{HPO}, is made of the direct arrows-based span on $\mathscr{C}_{HPO}$ with rules in $\mathscr{D}_{HPO}$ and matches in $\mathscr{M}_{HPO} = \mathbf{TermGraph}$ together with the family of partial functions $\mathbf{S}_{HPO}$ defined as above from heterogeneous pushouts.

Proposition 4 *The categorical rewriting system* RS_{HPO} *is functorial.*

Proof. The compositionality property of heterogeneous pushouts, similar to the compositionality property of pushouts, follows easily from their initiality property. Proposition 4 is a consequence of this property. □

3.3 Double pushout rewriting

In this section we check that under suitable assumptions the graph transformation based on double pushouts (DPO) [4] can be considered as a categorical rewriting system which is composed, in the sense of Definition 6, of a categorical rewriting system based on pushout complements (as defined below) followed by a categorical rewriting system based on pushouts (Definition 2). We restrict our study to cases where the pushout complement is unique. Let $\mathscr{M}_{POC} = \mathscr{C}_{POC} = \mathbf{Graph}$ be the category of graphs. Let $\mathscr{D}_{POC} = \mathbf{Graph_m}$ be the wide subcategory of $\mathscr{C}_{POC}$ with injective morphisms. We consider the inverse arrows-based span on $\mathscr{C}_{POC}$ with rules in $\mathscr{D}_{POC}$ and matches in $\mathscr{M}_{POC}$. This means that a rule $\rho : L \rightsquigarrow R$ is a monomorphism of graphs $\rho : R \rightarrowtail L$, or (according to the usual notations) $l : K \rightarrowtail L$. Following [15], we define a *partial graph* as a graph in which there may be edges without a source or target node. Given a graph G and a subgraph H of G, we denote as $G - H$ the partial graph made of the nodes and edges in G which are not in H, with the restriction of the source and target functions. In general $G - H$ is not a graph, since it can have *dangling* edges, i.e., edges which are not in H but which have their source or target in H. Following [4, Proposition 9], given a rule $l : K \rightarrowtail L$ we say that a match $f : L \longrightarrow L_1$ satisfies the *gluing condition* with respect to l if:

- *Dangling condition.* If an edge e_1 in L_1 is incident to a node in $f(L - l(K))$ then e_1 is in $f(L)$.
- *Identification condition.* If two nodes (respectively two edges) x and y in L are such that $x \neq y$ and $f(x) = f(y)$ then x and y are in $l(K)$.

One can remark that if the dangling condition is satisfied then $L_1 - f(L - l(K))$ is a graph. It is proven in [4, Proposition 9] that when f satisfies the gluing condition with respect to l then the graph $K_1 = L_1 - f(L - l(K))$ together with the inclusion $l_1 : K_1 \rightarrowtail L_1$ and the morphism $g : K \longrightarrow K_1$ which maps each node or edge x to $f(l(x))$ forms a pushout complement of l and f in **Graph**, and in addition this pushout complement is unique up to isomorphism. For each rule $l : K \rightarrowtail L$ we define $\mathbf{S}_{POC,l}$ as the partial function with domain the matches with source L which satisfy the gluing condition with respect to l, such that $\mathbf{S}_{POC,\rho}(f)$ is the pushout complement of l and f for each f in $dom(\mathbf{S}_{POC,l})$:

$$\begin{tikzcd} L \arrow[d, "f"'] & K \arrow[l, tail, "l"'] \\ L_1 & \end{tikzcd} \quad \overset{\mathbf{S}_{POC,\rho}}{\longmapsto} \quad \begin{tikzcd} L \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_{POC,\rho}(f)"] & K \arrow[l, tail, "l"'] \arrow[d, "g"] \\ L_1 & K_1 \arrow[l, tail, "l_1"] \end{tikzcd}$$

Definition 10 The *categorical rewriting system for graphs based on pushout complements*, denoted as RS_{POC}, is made of the inverse arrows-based span on $\mathscr{C}_{POC} = \mathbf{Graph}$ with rules in

$\mathscr{D}_{POC} = \mathbf{Graph_m}$ and matches in $\mathscr{M}_{POC} = \mathbf{Graph}$ together with the family of partial functions $\mathbf{S}_{POC}$ defined as above from pushout complements in **Graph**. The *categorical rewriting system for graphs based on double pushouts*, denoted as RS_{DPO}, is the composition of RS_{POC} and $\mathrm{RS}_{PO,\mathbf{Graph}}$ (from Definition 2).

Lemma 2 *Let us consider the categorical rewriting system* RS_{POC}. *Let* $l : K \rightarrowtail L$ *be a rule and* $f_1 : L \longrightarrow L_1$ *a match which satisfies the gluing condition with respect to* l. *Let* (K_1, l_1, g_1) *be the pushout complement of* l *and* f_1. *Let* $f_2 : L_1 \to L_2$ *be a match which satisfies the gluing condition with respect to* l_1. *Then* $f_2 \circ f_1$ *satisfies the gluing condition with respect to* l.

Proof. Let $f = f_2 \circ f_1 : L \to L_2$ We have to prove that f satisfies the dangling condition and the identification conditions with respect to l.

- *Dangling condition.* Suppose that f_1 and f_2 verify the identification condition. Let e_2 be an edge in L_2 which is incident to a node x_2 in $f(L - l(K))$. We have to prove that e_2 is in $f_2(f_1(L)) = f(L)$. There are two cases:

  1. There exists an edge e_1 in L_1 such that $e_2 = f_2(e_1)$. Hence there is a node x_1' in L_1 such that e_1 is incident to x_1' and $f_2(x_1') = x_2$. On the other hand, let x be a node in $L - l(K)$ such that $x_2 = f(x)$, and let $x_1 = f_1(x)$, so that $x_1 \in f_1(L - l(K))$. Since l_1 is the inclusion of $K_1 = L_1 - f_1(L - l(K))$ in L_1, it follows that x_1 is not in $l_1(K_1)$. Hence, the identification condition of f_2 ensures that $x_1 = x_1'$. Now, as f_1 satisfies the dangling condition with respect to l, we have that e_1 is in $f_1(L)$, thus $f_2(e_1) = e_2$ is in $f_2(f_1(L)) = f(L)$.
  2. The edge e_2 has no f_2-antecedent in L_1. Let x be a node of $L - l(K)$ such that $f(x) = x_2$. Let $x_1 = f_1(x)$, then $x_1 \in L_1 - l_1(K_1)$ because (K_1, l_1, g_1) being the pushout complement of l and f_1, it is unique and K_1 is the subgraph of L obtained by removing all items that are in the image of f_1 but not in the image of $f_1 \circ l$ (see [4, Proposition 9]). Thus e_2 is an edge incident to a node of $f_2(L_1 - l_1(K_1))$. Since f_2 satisfies the dangling condition with respect to l_1, we know that e_2 is in $f_2(L_1)$, which contradicts our hypothesis that e_2 has no f_2-antecedent. Thus, this case cannot occur.

- *Identification condition.* Suppose that there are two items $x, y \in L$ such that $x \neq y$ and $f(x) = f(y)$. We have to prove that x and y are in $l(K)$. Then there are two cases:

  1. If $f_1(x) = f_1(y)$, the identification condition of f_1 with respect to l implies that x and y are in $l(K)$.
  2. If $f_1(x) \neq f_1(y)$, let $x_1 = f_1(x)$ and $y_1 = f_1(y)$, so that $x_1 \neq y_1$ and $f_2(x_1) = f_2(y_1)$. The identification condition of f_2 with respect to l_1 implies that x_1 and y_1 are in $l_1(K_1)$. Now since $K_1 = L_1 - f_1(L - l(K))$ and x_1, y_1 are in $f_1(L)$, it implies that they are in $l(K)$.

$\square$

Proposition 5 *The categorical rewriting systems* RS_{POC} *and* RS_{DPO} *are functorial.*

Proof. The functoriality of RS_{POC} follows from Lemma 2 and the compositionality property of pushouts. Then the functoriality of RS_{DPO} follows from the functoriality of $\mathrm{RS}_{PO,\mathbf{Graph}}$ (Proposition 1) and from Proposition 2. □

3.4 Sesqui-pushout rewriting

Similarly to Section 3.3, under suitable assumptions the graph transformation based on sesqui-pushouts (SqPO) [3] can be considered as a categorical rewriting system which is composed of a categorical rewriting system based on final pullback complements (as defined below) followed by a categorical rewriting system based on pushouts. Final pullback complements are defined in [8, Theorem 4.4] as follows. For each match $f : L \to L_1$ let us consider the slice categories $\mathscr{D}\!\uparrow\!L$ and $\mathscr{D}\!\uparrow\!L_1$ of objects of $\mathscr{D}$ over L and L_1, respectively. Let $f^* : \mathscr{D}\!\uparrow\!L_1 \to \mathscr{D}\!\uparrow\!L$ denote the pullback functor, which maps each $l_1 : K_1 \to L_1$ to $f^*(l_1) : K \to L$ such that there is a pullback square:

$$
\begin{tikzcd}
L \arrow[d, "f"'] & K \arrow[l, "f^*(l_1)"'] \arrow[d] \arrow[dl, phantom, "\lrcorner", very near start] \\
L_1 & K_1 \arrow[l, "l_1"]
\end{tikzcd}
$$

The *Dyckhoff-Tholen condition* for f states that the pullback functor f^* has a right adjoint f_* such that $f^* \circ f_*$ is the identity. This last condition implies that the functor $f_* : \mathscr{D}\!\uparrow\!L \to \mathscr{D}\!\uparrow\!L_1$ provides a pullback complement for f and l, for every $l : K \to L$, which is called the *final pullback complement* (FPBC) of f and l. The definition of the final pullback complement of f and l implies that, when it does exist, it is unique. Let $\mathscr{C}_{FPBC} = \mathbf{Graph}$ be the category of graphs, and let $\mathbf{Graph_m}$ be the category of graphs with monomorphisms, seen as a wide subcategory of $\mathbf{Graph}$. Following [3], we define two kinds of rewriting systems based on FPBCs. In the first one the rules are monomorphisms, in the second one the matches are monomorphisms. In both cases we consider an inverse arrows-based span on $\mathbf{Graph}$.

1. *Left-linear rules.* Let $\mathscr{D}_{FPBC,1} = \mathbf{Graph_m}$ and $\mathscr{M}_{FPBC,1} = \mathbf{Graph}$. Following [3, definition 4], given a rule $l : K \rightarrowtail L$ we say that a match $f : L \longrightarrow L_1$ is *conflict-free* with respect to l when f does not identify any item in the image of l with an item outside this image (note the similarity with the definition of conflict-free matches for SPO). For each rule $l : K \rightarrowtail L$ we define $\mathbf{S}_{FPBC,1,l}$ as the partial function with domain the conflict-free matches with respect to l, such that $\mathbf{S}_{FPBC,1,l}(f)$ is the final pullback complement of l and f in $\mathbf{Graph}$, for each f in $\mathrm{Dom}(\mathbf{S}_{FPBC,1,l})$. It is proved in [3, construction 5] that this final pullback complement exists, and that it yields $l_1 : K_1 \rightarrowtail L_1$ and $g : K \longrightarrow K_1$.

$$
\begin{tikzcd}
L \arrow[d, "f"'] & K \arrow[l, tail, "l"'] & \arrow[r, mapsto, "\mathbf{S}_{FPBC,1,l}"] & {} & L \arrow[d, "f"'] & K \arrow[l, tail, "l"'] \arrow[d, "g"] \arrow[dl, phantom, "\lrcorner", very near start] \arrow[dl, phantom, "\mathbf{S}_{FPBC,1,l}(f)"] \\
L_1 & & & & L_1 & K_1 \arrow[l, tail, "l_1"]
\end{tikzcd}
$$

2. *Monic matches.* Let $\mathscr{D}_{FPBC,2} = \mathbf{Graph}$ and $\mathscr{M}_{FPBC,2} = \mathbf{Graph_m}$. Given a rule $l : K \longrightarrow L$ we define $\mathbf{S}_{FPBC,2,l}$ as the total function on $\mathbf{Graph_m}$ such that $\mathbf{S}_{FPBC,2,l}(f)$ is the final

pullback complement of l and f in **Graph**, for each f in $\mathbf{Graph_m}$. It is proved in [3, construction 6] that this final pullback complement exists, and that it yields $l_1 : K_1 \longrightarrow L_1$ and $g : K \rightarrowtail K_1$.

$$\begin{tikzcd} L & K \arrow[l, tail, "l"'] \\ L_1 \arrow[from=u, "f"] & \end{tikzcd} \quad \overset{\mathbf{S}_{FPBC,2,l}}{\longmapsto} \quad \begin{tikzcd} L \arrow[d, "f"'] \arrow[dr, phantom, "\mathbf{S}_{FPBC,2,l}(f)"] & K \arrow[l, tail, "l"'] \arrow[d, tail, "g"] \\ L_1 & K_1 \arrow[l, "l_1"] \end{tikzcd}$$

Definition 11 The *categorical rewriting systems for graphs based on final pullback complements*, denoted as $\mathrm{RS}_{FPBC,i}$ with $i = 1$ or $i = 2$, are made of the inverse arrows-based span on **Graph** with rules in $\mathbf{Graph_m}$ and matches in **Graph** when $i = 1$, and with rules in **Graph** and matches in $\mathbf{Graph_m}$ when $i = 2$, together with the family of functions $\mathbf{S}_{FPBC,i}$ defined as above from final pullback complements in **Graph**, so that $\mathbf{S}_{FPBC,1}$ is partial and $\mathbf{S}_{FPBC,2}$ is total. For each $i \in \{1,2\}$, the *categorical rewriting systems for graphs based on sesqui-pushouts*, denoted as $\mathrm{RS}_{SqPO,i}$, is the composition of $\mathrm{RS}_{FPBC,i}$ and $\mathrm{RS}_{PO,\mathbf{Graph}}$ (from Definition 2).

Lemma 3 *Let us consider the categorical rewriting system $\mathrm{RS}_{FPBC,1}$. Let $l : K \rightarrowtail L$ be a rule and $f_1 : L \longrightarrow L_1$ a match which is conflict-free with respect to l. Let (K_1, l_1, g_1) be the final pullback complement of l and f_1. Let $f_2 : L_1 \to L_2$ be a match which is conflict-free with respect to l_1. Then $f_2 \circ f_1$ is conflict-free with respect to l.*

Proof. Let $f = f_2 \circ f_1 : L \longrightarrow L_2$. The proof is done by contradiction. Let us assume that there are two items x and y in L such that $f(x) = f(y)$, with $x \in l(K)$ and $y \notin l(K)$. Then there are two cases:

1. If $f_1(x) = f_1(y)$ then f_1 is not conflict-free with respect to l.
2. Otherwise let $x_1 = f_1(x)$ and $y_1 = f_1(y)$, so that $f_2(x_1) = f_2(y_1)$. The commutativity of the square $\mathbf{S}_{FPBC,l}(f_1)$ implies that $x_1 \in l_1(K_1)$. Moreover, the construction of the final pullback complement in [3, construction 6] shows that $y_1 \notin l_1(K_1)$ since $y \notin l(K)$. Thus, $x_1 \in l_1(K_1)$ and $y_1 \notin l_1(K_1)$, so that f_2 is not conflict-free with respect to l_1.

□

Proposition 6 *The categorical rewriting systems $\mathrm{RS}_{FPBC,i}$ and $\mathrm{RS}_{SqPO,i}$, for $i = 1$ and $i = 2$, are functorial.*

Proof. Similar to the proof of Proposition 5. □

A similar result (vertical composition of sesqui-pushout graph transformations) is stated in [13, proposition 5].

4 A non-functorial graph transformation system

We define two *garbage removal rewriting systems*, as two attempts to formalize the process of removing unreachable nodes from a given graph. One of these rewriting systems is not functorial, but the other is. Let $\mathbf{Graph}_\subseteq$ be the category of graphs with inclusions; it is a preorder, thus every diagram in $\mathbf{Graph}_\subseteq$ is commutative. In both rewriting systems, the underlying span is the inverse arrows-based span on $\mathbf{Graph}_\subseteq$ with rules and matches in $\mathbf{Graph}_\subseteq$.

4.1 Garbage removal

Definition 12 Let L_1 be a graph and A a subgraph of L_1. The set of nodes of L_1 which are *reachable from* A (A stands for *A*live nodes) is defined recursively, as follows: a node of A is reachable from A, and the successors of a node reachable from A are reachable from A. The subgraph of L_1 generated by the nodes reachable from A is called the *maximal subgraph of* L_1 *reachable from* A, it is denoted as $\Lambda_A(L_1)$.

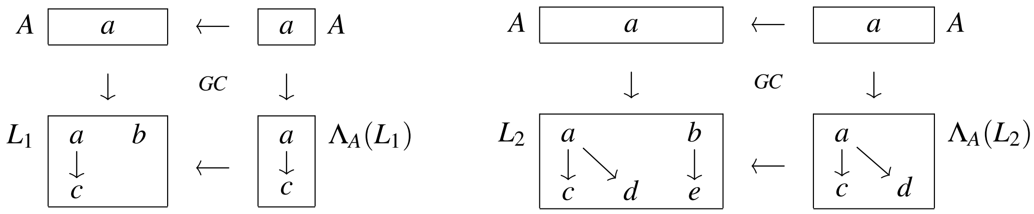
The aim of garbage removal is the determination of $\Lambda_A(L_1)$. In fact, $\Lambda_A(L_1)$ does not depend on the edges of A, only on its nodes. The nodes of A play the role of *roots* for the graph L_1, with $\Lambda_A(L_1)$ as the result of garbage removal from these roots. There are several categorical characterizations of $\Lambda_A(L_1)$, see for instance [5], but they are not used in this paper. Garbage removal provides a factorization of the inclusion $A \subseteq L_1$ in two inclusions $A \subseteq \Lambda_A(L_1) \subseteq L_1$. This is denoted:

$$\begin{tikzcd} A \arrow[d] & & A \arrow[d] \arrow[dr] & \\ L_1 & \arrow[r, mapsto, "GC"] & L_1 & \Lambda_A(L_1) \arrow[l, "GC"'] \end{tikzcd}$$

This "triangular" diagram is equivalent to the "rectangular" one:

$$\begin{tikzcd} A \arrow[d] & A \arrow[l] & & A \arrow[d] \arrow[dr, phantom, "GC"] & A \arrow[l] \arrow[d] \\ L_1 & & \arrow[r, mapsto, "GC"] & L_1 & \Lambda_A(L_1) \arrow[l] \end{tikzcd}$$

Example 1 *Here are two simple examples, where A is made of a single node.*



We generalize this situation by allowing the rules to be any inclusions $R \subseteq L$, not only identities; thus for instance the inclusion $\Lambda_A(L_1) \subseteq L_1$ can be seen as a rule. Then, garbage removal can be seen as a categorical rewriting system with respect to the inverse arrows-based span on $\mathbf{Graph}_\subseteq$ with rules and matches in $\mathbf{Graph}_\subseteq$. This can be done in two ways: in Section 4.2 the alive subgraph A is the left-hand side L while in Section 4.3 it is the right-hand side R.

4.2 Garbage removal as a non-functorial graph rewriting system

Definition 13 The *L-garbage removal rewriting system* RS_{LGC} is defined as the inverse arrows-based span on $\mathbf{Graph}_\subseteq$ with rules and matches in $\mathbf{Graph}_\subseteq$ together with the total functions

$\mathbf{S}_{LGC,\rho}$, for every $\rho : R \subseteq L$, which map each inclusion $L \subseteq L_1$ to the commutative square in $\mathbf{Graph}_{\subseteq}$ with vertices L, R, L_1 and $\Lambda_L(L_1)$.

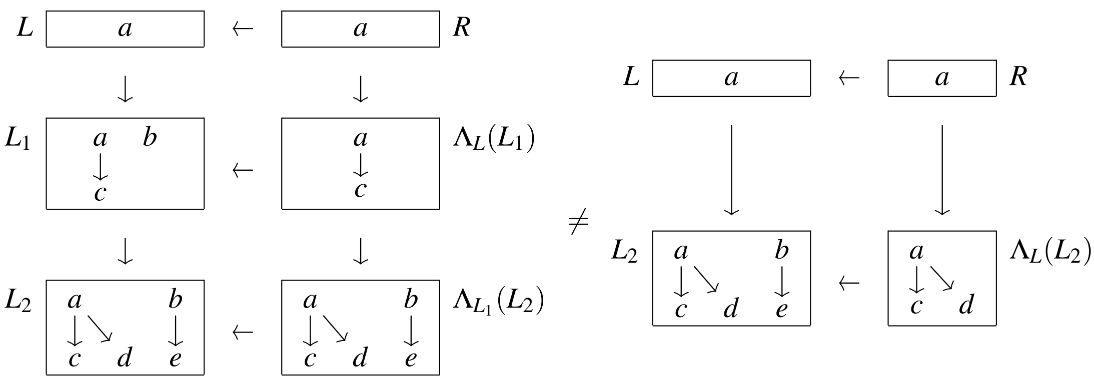
$$\begin{tikzcd} L \arrow[d, "f"'] & R \arrow[l, "\rho"'] \\ L_1 & \end{tikzcd} \quad \overset{\mathbf{S}_{LGC,\rho}}{\longmapsto} \quad \begin{tikzcd} L \arrow[d] \arrow[dr] \arrow[dr, phantom, "GC"', shift right=3] \arrow[dr, phantom, "=", shift left=3] & R \arrow[l] \arrow[d] \\ L_1 & \Lambda_L(L_1) \arrow[l] \end{tikzcd}$$

Proposition 7 *The categorical rewriting system* RS_{LGC} *is not functorial.*

Proof. In general $\Lambda_{L_1}(L_2)$ is not the same as $\Lambda_L(L_2)$, see Example 2 below. □

$$\begin{tikzcd} L \arrow[d] \arrow[dr] \arrow[dr, phantom, "GC"', shift right=3] \arrow[dr, phantom, "=", shift left=3] & R \arrow[l] \arrow[d] \\ L_1 \arrow[d] \arrow[dr] \arrow[dr, phantom, "GC"', shift right=3] \arrow[dr, phantom, "=", shift left=3] & R_1 = \Lambda_L(L_1) \arrow[l] \arrow[d] \\ L_2 & R_2 = \Lambda_{L_1}(L_2) \arrow[l] \end{tikzcd} \quad \neq \quad \begin{tikzcd} L \arrow[dd] \arrow[ddr] \arrow[ddr, phantom, "GC"', shift right=3] \arrow[ddr, phantom, "=", shift left=3] & R \arrow[l] \arrow[dd] \\ & \\ L_2 & \Lambda_L(L_2) \arrow[l] \end{tikzcd}$$

Example 2 *Let us apply* RS_{LGC} *to* $R = L \subseteq L_1 \subseteq L_2$ *and to* $R = L \subseteq L_2$*, as in Example 1. We get* $\Lambda_{L_1}(L_2) \neq \Lambda_L(L_2)$.



It turns out that if we choose the right-hand side of the rule instead of its left-hand side as the alive subgraph, the graph transformation system obtained is functorial: this is done in the next section.

4.3 Garbage removal as a functorial graph rewriting system

Definition 14 The *R-garbage removal rewriting system* RS_{RGC} is defined as the inverse arrows-based span on $\mathbf{Graph}_{\subseteq}$ with rules and matches in $\mathbf{Graph}_{\subseteq}$ together with the total functions $\mathbf{S}_{RGC,\rho}$, for every $\rho : R \subseteq L$, which map each inclusion $L \subseteq L_1$ to the commutative square in

Graph$_\subseteq$ with vertices L, R, L_1 and $\Lambda_R(L_1)$.

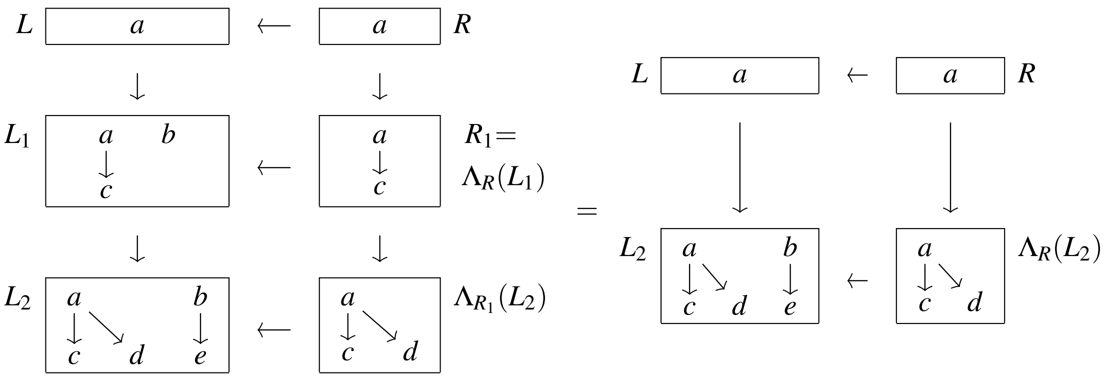
$$
\begin{tikzcd}
L \arrow[d, "f"'] & R \arrow[l, "\rho"'] \\
L_1 &
\end{tikzcd}
\quad \overset{\mathbf{S}_{RGC,\rho}}{\longmapsto} \quad
\begin{tikzcd}
L \arrow[d, "f"'] \arrow[dr, phantom, "="] & R \arrow[l, "\rho"'] \arrow[d] \arrow[dl] \\
L_1 & \Lambda_R(L_1) \arrow[l, "GC"]
\end{tikzcd}
$$

Proposition 8 *The categorical rewriting system* RS_{RGC} *is functorial.*

Proof. It is easy to check that $\Lambda_{R_1}(L_2)$, where $R_1 = \Lambda_R(L_1)$, is the same as $\Lambda_R(L_2)$. $\square$

$$
\begin{tikzcd}
L \arrow[d] \arrow[dr, phantom, "="] & R \arrow[l] \arrow[d] \arrow[dl] \\
L_1 \arrow[d] \arrow[dr, phantom, "="] & R_1 = \Lambda_R(L_1) \arrow[l, "GC"'] \arrow[d] \arrow[dl] \\
L_2 & R_2 = \Lambda_{R_1}(L_2) \arrow[l, "GC"']
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
L \arrow[d] \arrow[dr, phantom, "="] & R \arrow[l] \arrow[d] \arrow[dl] \\
L_2 & R_2 = \Lambda_R(L_2) \arrow[l, "GC"']
\end{tikzcd}
$$

Example 3 *Let us apply* RS_{RGC} *to* $R = L \subseteq L_1 \subseteq L_2$ *and to* $R = L \subseteq L_2$*, as in Example 1. We get* $\Lambda_{R_1}(L_2) = \Lambda_R(L_2)$.



5 Conclusion

We have introduced a new notion of abstract rewriting system based on categories. These systems are designed for dealing with abstract rewriting frameworks where rewrite steps are defined by means of matches. We have defined the properties of (horizontal) composition as well as functoriality of rewriting in our abstract setting and we have illustrated these properties throughout several algebraic graph rewriting systems. We plan to extend and deepen our abstract framework by investigating other instances such as [11, 13] and by allowing the rewriting processes $\mathbf{S}_\rho$ to be relations instead of partial functions.

Acknowledgements: We would like to thank Andrea Corradini and Barbara König for enlightening discussions about the Dyckhoff-Tholen condition. We also thank anonymous referees for insightful comments.

Bibliography

[1] F. Baader and T. Nipkow. *Term rewriting and all that.* Cambridge University Press, 1998.

[2] C. Consel and O. Danvy. Tutorial Notes on Partial Evaluation. In *20th Symposium on Principles of Programming Languages (POPL'93)*, pages 493–501. ACM, 1993.

[3] A. Corradini, T. Heindel, F. Hermann, and B. König. Sesqui-pushout rewriting. In *Third International Conference on Graph Transformations (ICGT 06)*, volume 4178 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2006.

[4] A. Corradini, U. Montanari, F. Rossi, H. Ehrig, R. Heckel, and M. Löwe. Algebraic approaches to graph transformation - part I: Basic concepts and double pushout approach. In *Handbook of Graph Grammars*, pages 163–246, 1997.

[5] D. Duval, R. Echahed, and F. Prost. Adjunction for Garbage Collection with Application to Graph Rewriting. In *18th International Conference on Rewriting Techniques and Applications, RTA 2007*, Springer Lecture Notes in Computer Science 4533 pages 122–136, 2007.

[6] D. Duval, R. Echahed, and F. Prost. A heterogeneous pushout approach to term-graph transformation. In *20th International Conference on Rewriting Techniques and Applications, RTA 2009*, Springer Lecture Notes in Computer Science 5595 pages 194–208, 2009.

[7] D. Duval, R. Echahed, and F. Prost. Graph rewriting with polarized cloning. *Available at http://arxiv.org/abs/0811.3400 Submitted.*

[8] R. Dyckhoff and W. Tholen. Exponentiable morphisms, partial products and pullback complements. In *Journal of Pure and Applied Algebra*, 49(1&2):103–116, 1987.

[9] H. Ehrig, R. Heckel, M. Korff, M. Löwe, L. Ribeiro, A. Wagner, and A. Corradini. Algebraic approaches to graph transformation - part II: Single pushout approach and comparison with double pushout approach. In *Handbook of Graph Grammars*, pages 247–312, 1997.

[10] H. Ehrig, M. Pfender, and H. J. Schneider. Graph-grammars: An algebraic approach. In *14th Annual Symposium on Foundations of Computer Science (FOCS), 15-17 October 1973, The University of Iowa, USA*, pages 167–180. IEEE, 1973.

[11] R. Heckel, H. Ehrig, U. Wolter and A. Corradini. Double-pullback transitions and coalgebraic loose semantics for graph transformation systems. In *Applied Categorical Structures*, 9:83–110, 1997.

[12] M. Löwe. Algebraic approach to single-pushout graph transformation. *Theor. Comput. Sci.*, 109(1&2):181–224, 1993.

[13] M. Löwe. Graph-rewriting in span-categories. In *Fifth International Conference on Graph Transformations (ICGT 10)*, volume 6372 of *Lecture Notes in Computer Science*, pages 218–233. Springer, 2010.

[14] S. Mac Lane. *Categories for the Working Mathematician.* 2nd edition. Graduate Texts in Mathematics 5, Springer-Verlag (1997).

[15] H. J. Schneider and H. Ehrig. Grammars on Partial Graphs. *Acta Informatica* 6:297–316, 1976.