Sesqui-pushout rewriting*

Andrea Corradini[1], Tobias Heindel[1],
Frank Hermann[2], and Barbara König[3]

[1] Dipartimento di Informatica, Università di Pisa, Italy
[2] Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany
[3] Institut für Informatik und interaktive Systeme, Univ. Duisburg-Essen, Germany

Abstract. *Sesqui-pushout* (SqPO) *rewriting*—"sesqui" means "one and a half" in Latin—is a new algebraic approach to abstract rewriting in any category. SqPO rewriting is a deterministic and conservative extension of double-pushout (DPO) rewriting, which allows to model "deletion in unknown context", a typical feature of single-pushout (SPO) rewriting, as well as cloning.
After illustrating the expressiveness of the proposed approach through a case study modelling an access control system, we discuss sufficient conditions for the existence of final pullback complements and we analyze the relationship between SqPO and the classical DPO and SPO approaches.

1 Introduction

In the area of graph transformation the two main categorical approaches used to describe the effect of applying a rule to a graph are the double-pushout approach (DPO [2, 7]) and the single-pushout approach (SPO [5, 17]). Both approaches use concepts of category theory to obtain an elegant and compact description of graph rewriting, but they differ with respect to the kind of morphisms under consideration, the form of the rules, and the diagrams the rewriting steps are based on. The aim of this paper is to propose a new categorical approach to rewriting that combines the good properties of both approaches and improves them by allowing to model cloning of structures in a natural way.

In the DPO approach [2, 7] a rule q is a span $q = L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ of arrows in a category of graphs and *total* graph morphisms. Given an *occurrence* of q in a graph G, i.e., a *match morphism* $m\colon L \to G$ from the *left-hand side* L to G, to apply q to G one first deletes from G the part of the occurrence of L which is not present in the *interface* K, and then one adds to the resulting graph those parts of the *right-hand side* R which are not in the image of K.
This construction is described by a double-pushout diagram as in (1) which, given q and m, can be constructed if there exists a *pushout complement* of α and m, i.e., arrows $A \xleftarrow{\gamma} D \xleftarrow{i} K$ making the resulting square a pushout.

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "\lrcorner" near start] & K \arrow[l, "\alpha"'] \arrow[r, "\beta"] \arrow[d, "i"] \arrow[dr, phantom, "\ulcorner" near end] & R \arrow[d, "c"] \\ A & D \arrow[l, "\gamma"] \arrow[r, "\delta"'] & B \end{tikzcd} \quad (1)$$

* Research partially supported by EPSRC grant GR/T22049/01, DFG project SANDS, EC RTN 2-2001-00346 SegraVis, EU IST-2004-16004 SEnSOria, and MIUR PRIN 2005015824 ART.

Since the pushout complement is not characterised by a universal property, the DPO approach is in general non-deterministic: given a rule and a match, there could be several (possibly non-isomorphic) resulting graphs. To guarantee determinism, one usually sticks to *left-linear* rules, i.e., α must be injective. In this case, it is known that a pushout complement of α and m exists if and only if m satisfies the so-called *dangling* and *identification condition* with respect to α.

In the SPO approach [5, 17], instead, a rule is an arrow $q \colon L \rightharpoonup R$ in a category of graphs and *partial* graph morphisms. The application of the rule q to a match m is modelled by a single pushout in this category and thus, by the universal property of pushouts, SPO rewriting is instrinsically deterministic. It is well known that a *partial* map $q \colon L \rightharpoonup R$ can be represented, in a category with *total* maps as arrows, as a span $L \longleftarrow\!\!\prec dom(q) \xrightarrow{q} R$, where $dom(q)$ is the domain of definition of q. Given an SPO rule and a match, the result of SPO rewriting is isomorphic to the result of DPO rewriting using the corresponding span and the same match, provided that the pushout complement exists. Thus, as explained in [5, 17], SPO rewriting on graphs (or similar structures) subsumes DPO rewriting: this fact is exploited in practice by the AGG system[1], which implements SPO rewriting but offers to developers both SPO and DPO.

Unlike DPO, SPO rewriting is possible even if the match does not satisfy the dangling or identification condition w.r.t. the rule. The dangling condition requires that if a node of G is to be deleted, then any arc incident to it is deleted as well. If this condition does not hold, both the node and all incident arcs are deleted in the graph resulting from SPO rewriting. Thus SPO rewriting allows to model *deletion in unknown context*, and this is recognized as a useful feature in several applications. The identification condition does not allow the match to identify in G an item to be preserved by the rule with one to be deleted. If this condition does not hold, the SPO construction deletes that item from G, and thus the morphism from R to the resulting graph is partial. Most often this feature (called "precedence of deletion over preservation") is ruled out by restricting the class of allowed matches.

The use of categorical machinery made possible, along the years, the generalization of basic definitions and main results of both the DPO and the SPO approach to a more abstract setting, where the structures on which rewriting is performed are objects of a generic category satisfying suitable properties. The characterization of such properties has been the main topic of the theory of High Level Replacement (HLR) systems [4, 6], and recently the definition of adhesive categories [16] (and their variants) provided a more manageable definition of them for the DPO case. The generalization of the SPO approach to suitable categories of spans has been elaborated in [14, 18].

Here we propose a new categorical approach to rewriting, called *sesqui-pushout* (SqPO) *rewriting*. Following the trend described in the last paragraph, the approach is presented abstractly in an arbitrary category. Rules are DPO-like spans of arrows, and a rewriting step is defined as for DPO rewriting, but the left pushout is replaced by a pullback satisfying a certain universal property:

[1] See http://tfs.cs.tu-berlin.de/agg/.

the role of the pushout complement is played now by the so-called *final pullback complement* (PBC).

Since final pullback complements are unique up to isomorphism, SqPO rewriting is deterministic. For *left-linear* rules, the final pullback complement coincides with the pushout complement, if the latter exists, and in this sense SqPO rewriting subsumes DPO rewriting. When the pushout complement does not exist but the final pullback complement does, SqPO rewriting models faithfully deletion in unknown context, like the SPO: dangling edges are removed. Strictly speaking, however, sesqui-pushout rewriting does not subsume SPO rewriting fully. In fact, by construction, there is always a total morphism from the right-hand side of the rule to the result of a sesqui-pushout rewriting step; thus if the match identifies items to be deleted with items to be preserved then the final pullback complement does not exist and rewriting is not allowed.

Interestingly, the final pullback complement is unique (if it exists) even for rules which are not left-linear, unlike the pushout complement. In this case, the final pullback complement *is not* a pushout complement in general, but it models faithfully the effect of *cloning*, at least for some concrete structures where the details have been worked out.

Based on the above discussion, we can explain the name we chose for the proposed approach. "Sesqui" is the latin word for "one and a half" and suggests that our approach is placed halfway between the single-pushout and the double-pushout approach. In fact, metodologically, it is based on a construction similar to the DPO, but it captures essential features of the SPO as well.

After introducing the basic definitions and properties of SqPO rewriting in Sec. 2, we demonstrate the expressiveness of the approach in Sec. 3 by modelling the access control problem described in [11]. In Sec. 4 we show how to construct the final pullback complement in some concrete categories, and we discuss its existence in general. Sec. 5 is dedicated to the comparison of SqPO rewriting with DPO and SPO rewriting, and also presents a Local Church Rosser theorem for parallel independent direct derivations. A concluding section summarizes the results of the paper and discusses further topics of investigation.

2 Defining sesqui-pushout rewriting

In this section we present only the fundamentals of sesqui-pushout rewriting: an example illustrating the expressiveness of this new approach is presented in the next section.

Let $\mathbb{C}$ be a category, about which we do not make any assumption, for the time being: objects and arrows belong to $\mathbb{C}$ if not specified otherwise. As in the DPO approach, a *rule* q is a span of arrows $q = L \stackrel{\alpha}{\longleftarrow} K \stackrel{\beta}{\longrightarrow} R$. Now the general idea of SqPO-rewriting is to replace the left square of a DPO rewriting diagram by a "most general" pullback complement with respect to the rule and the matching arrow, called a *final pullback complement* [3,18]. We will discuss in Section 4 how the universal property characterizing this construction is related to the right adjoint to the pullback functor.

After the general definition, we list some basic properties of final pullback complements and we present a simpler characterization which is applicable when the rule q is left-linear, i.e., α is mono.

Definition 1 (Sesqui-pushout rewriting). *Let $q = L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ be a rule, and $m\colon L \to A$ be an arrow, called a* match. *Then we write $A \xRightarrow{\langle m,q\rangle} B$, and we say that there is a* direct derivation from A to B (using m and q) *if we can construct a diagram such as* (2) *where the following conditions hold:*

- *the right square is a pushout, and*
- *the arrows $A \xleftarrow{\gamma} D \xleftarrow{i} K$ form a final pullback complement of $A \xleftarrow{m} L \xleftarrow{\alpha} K$, (this is indicated by the sign $\llcorner$ in Diagram* (2)*)*

$$\begin{tikzcd} L \arrow[d, "m"'] & K \arrow[l, "\alpha"'] \arrow[r, "\beta"] \arrow[d, "i"] \arrow[dl, phantom, "\llcorner"] & R \arrow[d, "c"] \\ A & D \arrow[l, "\gamma"] \arrow[r, "\delta"'] & B \arrow[ul, phantom, "\ulcorner"] \end{tikzcd} \tag{2}$$

where a final pullback complement *of $A \xleftarrow{m} L \xleftarrow{\alpha} K$ is defined to be a pair of arrows $A \xleftarrow{\gamma} D \xleftarrow{i} K$ such that*

1. *the square $K \xrightarrow{\alpha} L \xrightarrow{m} A \xleftarrow{\gamma} D \xleftarrow{i} K$ is a pullback, and*
2. *for each pullback $K' \xrightarrow{\alpha'} L \xrightarrow{m} A \xleftarrow{\gamma'} D' \xleftarrow{i'} K'$, and for each $\boldsymbol{f}\colon K' \to K$ such that $\alpha \circ \boldsymbol{f} = \alpha'$, there exists a unique $\widehat{\boldsymbol{f}}\colon D' \to D$ such that $\gamma \circ \widehat{\boldsymbol{f}} = \gamma'$ and $i \circ \boldsymbol{f} = \widehat{\boldsymbol{f}} \circ i'$ (see the right hand diagram).*

$$\begin{tikzcd} & & K' \arrow[dll, "\alpha'"'] \arrow[dl, bend left, "\boldsymbol{f}"] \arrow[dd, "i'"] \\ L \arrow[d, "m"'] & K \arrow[l, "\alpha"] \arrow[d, "i"] & \\ A & D \arrow[l, "\gamma"'] & \\ & & D' \arrow[ull, "\gamma'"] \arrow[ul, dashed, "\widehat{\boldsymbol{f}}"] \end{tikzcd}$$

It immediately follows from the defining properties that the final pullback complement of any pair of composable arrows is unique up to isomorphism, if it exists. Additionally, if the rule is left-linear, i.e., $L \xleftarrow{\alpha} K$ is mono, then also γ is mono, and it can be characterized as the largest pullback complement, where *largest* is interpreted in the poset of subobjects of A. These facts are formalized in the following lemma.

Lemma 2 (Properties of final pullback complements).

In the square on the right let $A \xleftarrow{\gamma} D \xleftarrow{i} K$ be a final pullback complement. Then the following facts hold.

$$\begin{tikzcd} L \arrow[d, "m"'] & K \arrow[l, "\alpha"'] \arrow[d, "i"] \arrow[dl, phantom, "\llcorner"] \\ A & D \arrow[l, "\gamma"] \end{tikzcd}$$

1. *If $A \xleftarrow{\gamma'} D' \xleftarrow{i'} K$ is another final pullback complement of $A \xleftarrow{m} L \xleftarrow{\alpha} K$, then there is a unique isomorphism $\phi\colon D' \to D$ such that $\phi \circ i' = i$ and $\gamma \circ \phi = \gamma'$. Thus final pullback complements are unique up to iso.*

$$\begin{tikzcd} L \arrow[d, "m"'] & K \arrow[l, "\alpha"'] \arrow[d, "i'"] \arrow[dr, bend left, "i"] & \\ A & D' \arrow[l, "\gamma'"'] \arrow[r, dotted, "\cong"] & D \arrow[ll, bend left, "\gamma"] \end{tikzcd}$$

2. *If additionally α is monic then*
   (a) *the arrow γ is monic*
   (b) *the arrow γ can be characterized as the largest among the subobjects of A that provide a pullback complement of $A \xleftarrow{m} L \xleftarrow{\alpha} K$, i.e.*

   for every pullback complement $A \xleftarrow{\delta} E \xleftarrow{j} K$ there exists a unique arrow $\boldsymbol{\ell}\colon E \to D$ such that $i = \boldsymbol{\ell} \circ j$ and $\delta = \gamma \circ \boldsymbol{\ell}$.

$$\begin{tikzcd} L \arrow[d, "m"'] & K \arrow[l, tail, "\alpha"'] \arrow[d, "j"] \\ A & E \arrow[l, tail, "\delta"] \end{tikzcd} \Rightarrow \begin{tikzcd} L \arrow[d, "m"'] & K \arrow[l, tail, "\alpha"'] \arrow[d, "i"] \arrow[dr, bend left, "j"] & \\ A & D \arrow[l, tail, "\gamma"'] & E \arrow[l, dashed, tail, "\boldsymbol{\ell}"'] \arrow[ll, bend left, "\delta"] \end{tikzcd}$$

It is worth stressing that, by the first point of Lemma 2, uniqueness of final pullback complements holds in any category, even if α is not monic: this fact guarantees that the result of SqPO rewriting is determined (up to iso), also in situations where DPO rewriting is "ambiguous" because of the existence of several pushout complements.

3 Modelling the Access Control problem

To show the expressive power of SqPO rewriting, we model the basic Access Control systems of [11]; for this we use *simple graphs*, i.e., graphs with at most one edge of each type between two nodes. This category has all pullbacks and pushouts, and hence a rule is applicable at a match if the relevant final pullback complement exists. Unlike pullbacks, pushouts are not computed componentwise on nodes and edges, because multiple parallel edges are not allowed. As discussed in [16], the category of simple graphs is *quasi-adhesive*; this implies, among other things, that pushout complements along *regular monos* are unique (if they exists).[2]

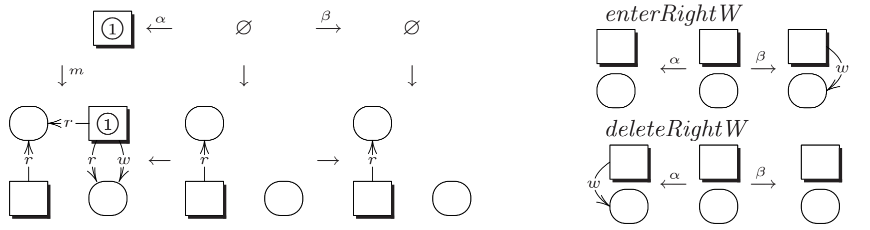
The Discrete Access Control model [11] considers a *protection system*, which controls the access of a set of *subjects* to a set of *objects*. Moreover suitable commands can change the state of the system. The corresponding decision problem consists in deciding whether a subject can obtain a certain right after applying a sequence of commands to a given initial state. Commands are sequences of *primitive operations* guarded by a Boolean condition: such operations model elementary changes of the system. Here we shall model the configurations of a system and the primitive operations. We also introduce a new operation called `clone`, which allows us to show a non-left-linear rule at work.

Definition 3 (Protection system). *A* protection system $P = (R, C)$ *consists of a finite set of* rights R *and a finite set of* commands C*. A* configuration *of a protection system is a triple* $c = (S, O, A)$*, where* S *is a set of* current subjects*,* O *is a set of* current objects *and* A *is an* access matrix $A[s, o] \subseteq R$*, with* $s \in S, o \in O$*.*

We model the configurations of a protection system as simple graphs, and the primitive operations as SqPO-rules. We depict subjects by shaded boxes $\square$, objects by rounded boxes $\bigcirc$, and if a subject possesses a right $i \in R$ to an object, we draw a labelled arc between them: $\square \xrightarrow{i} \bigcirc$. In the examples we use the common rights for "read" and "write" access, which are labelled by r and w, respectively.

The transformations of a configuration are defined by six primitive operations. They are `create subject` X_s and `create object` X_o for creating subjects and objects, `destroy subject` X_s and `destroy object` X_o for destroying subjects and objects, and finally `enter` i `into`(X_s, X_o) and `delete` i `from`(X_s, X_o)

[2] A *regular* mono is an arrow which is an equalizer of some pair of parallel arrows. In the category of simple graphs, an injective morphism is regular if it reflects edges.



(a) Application of *destroySubject*

(b) Rules

Fig. 1. Basic rules for transforming an Access Control configuration

for entering and deleting rights. Figure 1(a) shows the destruction of a subject by the application of the rule *destroySubject*, i.e., how the corresponding node and its incident edges in the graph are deleted. The morphisms are defined by mappings according to the numbers within the boxes. The left square is clearly a final pullback with the bottom right graph being its final pullback complement object. In fact, the square is a pullback, and it satisfies Condition *2(b)* of Lemma 2. Note that the effect of SqPO rewriting is similar to SPO, while DPO rewriting would not be applicable here.

Figure 1(b) depicts the rules which correspond to the operations of establishing and deleting the "write" access to a subject.

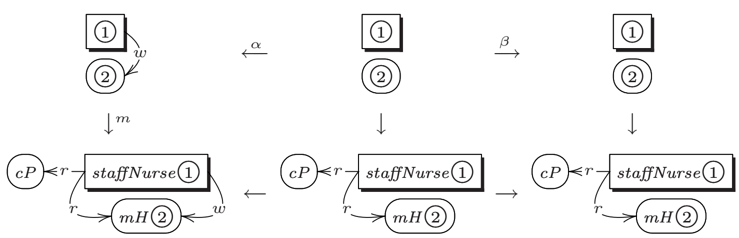


Fig. 2. Application of *deleteRightW*

Figure 2 shows the application of *deleteRightW* using SqPO rewriting: the left square is clearly a final pullback. Notice that in this case the DPO approach would be non-deterministic, as there are two non-isomorphic pushout complements for the given α and $L \xrightarrow{m} G$: the shown final pullback complement and G itself. Indeed, the category of simple graphs is quasi-adhesive, and uniqueness of pushout complements is guaranteed along *regular* monos only, i.e., morphisms reflecting edges: but α is not regular.

A new and interesting aspect of non-linear rules is used in Figure 3, where a subject is *cloned.* The final pullback complement construction automatically generates copies of the adjacent edges (cf. Construction 5). The rule *cloneSubject* is applied to a configuration representing a staff nurse, which has access to two objects, namely *cP* (representing contact information of patients) and *mH* (de-

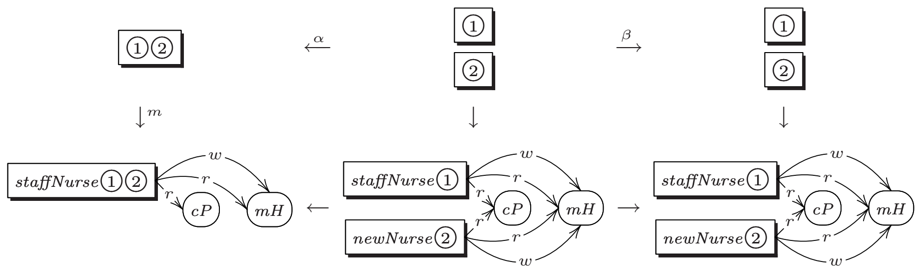


Fig. 3. Application of *cloneSubject*

noting medical history information of patients). When a hospital employs a new nurse, the administrator might not want to define all rights separately again. In general, in systems with complex configurations, operations which model cloning are of great help.

The role based model for Access Control RBAC described in [8] and [20] is widely used, and it can be considered as an extension of the model in [11]. Graph transformation is used in [15] for defining and verifying an RBAC model: unlike in our approach, negative application conditions are needed there to avoid multiple edges.

4 Existence and construction of final pullback complements

In this section we first give, as it is usual for algebraic approaches to rewriting, a concrete set-based description of SqPO-rewriting steps in the category of graphs; since pushouts are treated as usual, we only provide a construction for final pullback complements. In the sequel we address the question under which conditions final pullback complements exist and how they may be constructed "abstractly" in categories where right adjoints to pullback functors exist (most of the categories used in practice are of this kind).

4.1 Constructing final pullback complements in Graph

Since in practice one usually works with concrete objects, i.e., with objects that are represantable by structured sets, it is useful to present a set-theoretical construction of final pullback complements in a sample category of this kind. We consider here directed (multi-)graphs, but the construction can be generalized easily to algebras over an arbitrary *graph structure*, i.e., a signature with unary operator symbols only [17]. We present explicit constructions of the final pullback complement in **Graph** for the case in which either the left morphism of the rule or the match are monic, i.e., injective. For left-linear rules, we also provide a necessary and sufficient condition for its existence.

Recall that a *graph* is a tuple $G = \langle \mathsf{V}_G, \mathsf{E}_G, \mathsf{src}_G \colon \mathsf{E}_G \to \mathsf{V}_G, \mathsf{tgt}_G \colon \mathsf{E}_G \to \mathsf{V}_G \rangle$ where V_G and E_G are disjoint sets, which are called *vertices* and *edges*, respectively; the latter are connected according to the *source* and *target* functions src_G and tgt_G, respectively. A *graph morphism* $f \colon G \to H$ is a pair $\langle f_{\mathsf{V}} \colon \mathsf{V}_G \to \mathsf{V}_H, f_{\mathsf{E}} \colon \mathsf{E}_G \to \mathsf{E}_H \rangle$ such that $\mathsf{src}_H \circ f_{\mathsf{E}} = f_{\mathsf{V}} \circ \mathsf{src}_G$ and $\mathsf{tgt}_H \circ f_{\mathsf{E}} = f_{\mathsf{V}} \circ \mathsf{tgt}_G$. The category of graphs and graph morphisms is denoted by **Graph**.

Left-linear rules. Given a rule $q = L \overset{\alpha}{\hookleftarrow} K \xrightarrow{\beta} R$ with monic α (that we assume w.l.o.g. to be an inclusion) and a match $m \colon L \to A$, it is easy to show that there does not exist any pullback complement (and thus, *a fortiori*, any *final* PBC) of $A \xleftarrow{m} L \overset{\alpha}{\hookleftarrow} K$ if the match m is not *conflict-free* with respect to α.

Definition 4 (Conflict freeness). *A match* $m \colon L \to A$ *is* conflict-free *with respect to* $L \overset{\supseteq}{\hookleftarrow} K$ *if* $m(L \setminus K) \cap m(K) = \varnothing$.

For example, for the non-conflict-free match m shown to the right, any graph closing the square and making it commutative should contain at least one node (the image of ① under the vertical arrow), but in this case the resulting square would not be a pullback.

$$\begin{tikzcd} \{①, ②\} \arrow[d, "m"', "{\substack{1\mapsto 0\\ 2\mapsto 0}}"] & \{①\} \arrow[l, tail, "\supseteq"'] \arrow[d] \\ \{⓪\} & ? \arrow[l] \end{tikzcd}$$

It is worth observing that conflict-freeness is weaker than other conditions that are often imposed on matches in the framework of algebraic graph rewriting. For example, if the match m is monic, *d-injective* [17], or it satisfies the *identification condition* of DPO rewriting, then it is conflict-free.

Assuming conflict-freeness, the final pullback complement exists in **Graph**, and it can be described as follows as a subgraph of A (according to Condition *2(b)* of Lemma 2).

Construction 5 (Final PBC for left-linear rules in Graph) *Let be given a rule* $L \overset{\supseteq}{\hookleftarrow} K \xrightarrow{\beta} R$ *and a conflict-free match* $m \colon L \to A$*. Then the final* PBC *for* $A \xleftarrow{m} L \overset{\supseteq}{\hookleftarrow} K$ *is given by* $A \overset{\supseteq}{\hookleftarrow} D \xleftarrow{m|_K} K$*, where* D *is defined as*

$$\begin{aligned} \mathsf{V}_D &= \mathsf{V}_A \setminus m(\mathsf{V}_L \setminus \mathsf{V}_K) \\ \mathsf{E}_D &= \{ e \in \mathsf{E}_A \setminus m(\mathsf{E}_L \setminus \mathsf{E}_K) \mid \mathsf{src}_A(e) \in \mathsf{V}_D \wedge \mathsf{tgt}_A(e) \in \mathsf{V}_D \} \end{aligned}$$

It is evident from the construction that all the edges of A that are connected to deleted nodes are deleted as well, thus D is a well-defined graph; it is easily shown that it is indeed a pullback complement of the given arrows, and that no larger subgraph of A would be a PBC.

General rules, monic matches. If the left-hand side of the rule is not monic but the match is, the final pullback complement exists in **Graph** and it can be described as follows.

Construction 6 (Final PBC for monic matches in Graph) *Let be given a rule $L \xleftarrow{\alpha} K \xrightarrow{\beta} R$ and a monic match $m \colon L \rightarrowtail A$. Then a final pullback complement $A \xleftarrow{\gamma} D \xleftarrow{i} K$ can be constructed as follows.*

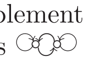
$$\mathsf{V}_D = \big(\mathsf{V}_A \setminus m(\mathsf{V}_L)\big) \uplus \mathsf{V}_K, \qquad \gamma_{\mathsf{V}}(u) = \begin{cases} m(\alpha_{\mathsf{V}}(u)) & \text{if } u \in \mathsf{V}_K \\ u & \text{if } u \in \mathsf{V}_A \end{cases}$$

$$\mathsf{E}_D = \left\{ \begin{array}{r} \langle e, u, v\rangle \mid e \in \mathsf{E}_A \setminus m(\mathsf{E}_L) \wedge u, v \in \mathsf{V}_D \wedge \\ \mathsf{src}_A(e) = \gamma_{\mathsf{V}}(u) \wedge \mathsf{tgt}_A(e) = \gamma_{\mathsf{V}}(v) \end{array} \right\} \uplus \mathsf{E}_K$$

$$\gamma_{\mathsf{E}}(e) = \begin{cases} e' & \text{if } e = \langle e', u, v\rangle \\ m(\alpha_{\mathsf{E}}(e)) & \text{otherwise} \end{cases}$$

$$\mathsf{src}_D(e) = \begin{cases} u & \text{if } e = \langle e', u, v\rangle \\ \mathsf{src}_K(e) & \text{otherwise} \end{cases} \qquad \mathsf{tgt}_D(e) = \begin{cases} v & \text{if } e = \langle e', u, v\rangle \\ \mathsf{tgt}_K(e) & \text{otherwise} \end{cases}$$

In words, the resulting graph D contains a copy of K, a copy of the largest subgraph of A which is not in the image of m, and a suitable number of copies of each arc of A incident to a node in $m(\alpha(K))$: this has the effect of "cloning" part of A. The proof that D is indeed a final PBC is omitted for space reasons.

Example 7 (Final PBC *of a non-left-linear rule).* According to Construction 6, the final pullback complement for  $\xleftarrow{m} \circ \xleftarrow{\alpha} \circ\ \circ$ in **Graph** is . Notice that there are four pushout complements of the given arrows: , , , and ; hence in this case the final PBC is not a pushout complement. Incidentally, it can be shown that also in the category of *simple* graphs  is a final PBC of the given arrows, but in this case it is a pushout complement as well.

Interestingly, note that one can derive from  a clique with n nodes by $n-1$ consecutive applications of the rule $\circ \xleftarrow{\alpha} \circ\ \circ \xrightarrow{id} \circ\ \circ$.

General rules, general matches. In the case of non-left-linear rules and non-injective matches, the exact conditions for the existence of final pullback complements in **Graph** and the details of its construction are rather involved, and go beyond the scope of this paper; the interested reader is encouraged to use the constructions of [9] to specialize the results in Section 4.2 to the category of graphs. One of the main issues of this general case is that the final pullback construction cannot be performed componentwise on nodes and edges.

Recall that in the case of DPO-rewriting, restricting to monic matches actually enhances expressiveness [10], in the sense of modelling power. It is left as future work to check if a similar result holds for SqPO-rewriting as well.

4.2 Final pullback complements in arbitrary categories

In this section we provide sufficient conditions for the existence of final pullback complements in a category. We first need to introduce some categorical concepts. We assume a fixed category $\mathbb{C}$, to which all mentioned objects and arrows belong unless we say otherwise.

Definition 8 (Slice category and pullback functor). *Let A be an object. The* slice category over A, *denoted by $\mathbb{C}{\downarrow}A$, has all $\mathbb{C}$-arrows $(B \xrightarrow{\beta} A)$ with codomain A as objects, and given two objects $(B \xrightarrow{\beta} A)$ and $(C \xrightarrow{\gamma} A)$ of $\mathbb{C}{\downarrow}A$ each $\mathbb{C}$-arrow $\boldsymbol{f}\colon B \to C$ satisfying the equality $\gamma \circ \boldsymbol{f} = \beta$ is an arrow $\boldsymbol{f}\colon \beta \to \gamma$ in $\mathbb{C}{\downarrow}A$.*

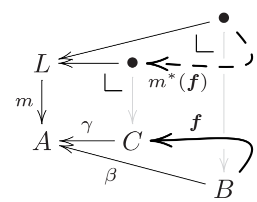
A pullback functor *along an arrow $m\colon L \to A$ is a functor $m^*\colon \mathbb{C}{\downarrow}A \to \mathbb{C}{\downarrow}L$ which maps each object*

$$\beta \in \mathsf{ob}(\mathbb{C}{\downarrow}A) \quad \textit{to} \quad m^*(\beta) \in \mathsf{ob}(\mathbb{C}{\downarrow}L)$$

and provides an additional arrow $m'_\beta\colon m^*(B) \to B$ such that the right hand diagram is a pullback. Further each arrow $\boldsymbol{f}\colon \beta \to \gamma$ in $\mathbb{C}{\downarrow}A$ is mapped to the unique arrow $m^*(\boldsymbol{f})\colon m^*(\beta) \to m^*(\gamma)$ such that the following is true, by the universal property of pullbacks:

$$m^*(\gamma) \circ m^*(\boldsymbol{f}) = m^*(\beta) \wedge m'_\gamma \circ m^*(\boldsymbol{f}) = \boldsymbol{f} \circ m'_\beta.$$

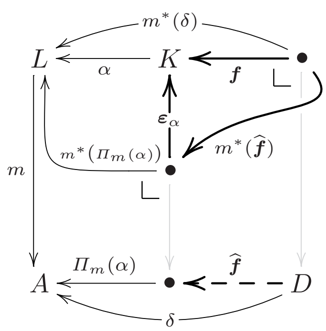
$$\begin{tikzcd} L \arrow[d, "m"'] & m^*(B) \arrow[l, "m^*(\beta)"'] \arrow[d, "m'_\beta"] \\ A & B \arrow[l, "\beta"] \end{tikzcd}$$



Given a category $\mathbb{C}$ and an arrow m such that the pullback functor $m^*\colon \mathbb{C}{\downarrow}A \to \mathbb{C}{\downarrow}L$ exists, the right adjoint to m^*, if it exists, is usually denoted by $\Pi_m\colon \mathbb{C}{\downarrow}L \to \mathbb{C}{\downarrow}A$. Even if Π_m does not exist, it might exist *partially* at an object $\alpha \in \mathsf{ob}(\mathbb{C}{\downarrow}L)$. In this case $\Pi_m(\alpha)$ satisfies a univeral property which can be described as follows.

Definition 9 (Right adjoints (partial)).

Let $m\colon L \to A$ be an arrow, let $m^*\colon \mathbb{C}{\downarrow}A \to \mathbb{C}{\downarrow}L$ be a pullback functor, and let $(K \xrightarrow{\alpha} L) \in \mathsf{ob}(\mathbb{C}{\downarrow}L)$ be an object. Then the right adjoint $\Pi_m\colon \mathbb{C}{\downarrow}L \to \mathbb{C}{\downarrow}A$ to m^* exists partially *at α if there is an object $\Pi_m(\alpha) \in \mathbb{C}{\downarrow}A$ and an arrow $\boldsymbol{\varepsilon}_\alpha\colon m^*\big(\Pi_m(\alpha)\big) \to \alpha$ in $\mathbb{C}{\downarrow}L$ such that for every $(D \xrightarrow{\delta} A) \in \mathsf{ob}(\mathbb{C}{\downarrow}A)$ and each $\boldsymbol{f}\colon m^*(\delta) \to \alpha$ there exists a unique $\widehat{\boldsymbol{f}}\colon \delta \to \Pi_m(\alpha)$ such that $\boldsymbol{\varepsilon}_\alpha \circ m^*(\widehat{\boldsymbol{f}}) = \boldsymbol{f}$.*



To illustrate these definitions we give an example, based on [9], where we talk about the simpler right adjoint to the preimage functor in **Set**.

Example 10 (The adjunction $m^{-1} \dashv \forall_m$). Consider a function $m\colon L \to A$ and the pre-image functor $m^{-1}\colon \langle \wp(A), \subseteq \rangle \to \langle \wp(L), \subseteq \rangle$ (recall that every poset gives rise to a category). The functor m^{-1} is essentially the restriction of a pullback functor $m^*\colon \mathbf{Set}{\downarrow}A \to \mathbf{Set}{\downarrow}L$, since given a subset $D \in \wp(A)$, m^* maps the inclusion morphism $D \overset{\subseteq}{\rightarrowtail} A$ to some mono $m^*(D) \rightarrowtail L$, such that $m^*(D) \cong m^{-1}(D)$. For each subset $K \in \wp(L)$ we define the set $\forall_m(K) \subseteq A$ by

$$\forall_m(K) = \{a \in A \mid \forall \ell \in m^{-1}(\{a\}).\, \ell \in K\}.$$

In fact, this definition of $\forall_m$ makes it a functor $\forall_m \colon \langle \wp(L), \subseteq \rangle \to \langle \wp(A), \subseteq \rangle$. Note that $\forall_m$ can be seen as the restriction of $\Pi_m \colon \mathbf{Set}\!\downarrow\! L \to \mathbf{Set}\!\downarrow\! A$ to the subcategory $\langle \wp(L), \subseteq \rangle$, since Π_m maps monos into L to monos into A.

Further one verifies that for all subsets $D \in \wp(A)$

$$m^{-1}(D) \subseteq K \quad \text{ if and only if } \quad D \subseteq \forall_m(K). \tag{3}$$

To make the link to Definition 9 more precise, note that the co-unit for K corresponds to the inclusion $m^{-1}\big(\forall_m(K)\big) \subseteq K$; further the Equivalence (3) implies that for all sets $D \in \wp(A)$, if the inclusion $m^{-1}(D) \subseteq K$ holds then $D \subseteq \forall_m(K)$ and hence also $m^{-1}(D) \subseteq m^{-1}\big(\forall_m(K)\big)$ hold.

The above definitions provide a sufficient condition for the existence of final pullback complements in an arbitrary category, as stated by the following lemma.

Lemma 11 (Existence and construction of final PBC). *Let $A \xleftarrow{m} L \xleftarrow{\alpha} K$ be a pair of composable arrows. Assume that the pullback functor $m^* \colon \mathbb{C}\!\downarrow\! A \to \mathbb{C}\!\downarrow\! L$ exists, that the right adjoint Π_m to it exists partially at α, and that the arrow $\boldsymbol{\varepsilon}_\alpha \colon m^*\big(\Pi_m(\alpha)\big) \to \alpha$ satisfies the conditions of Definition 9. Then*

1. *There exists a final pullback complement for $A \xleftarrow{m} L \xleftarrow{\alpha} K$ iff ε_α is iso.*
2. *If ε_α is iso, then the pair of composable arrows $\langle \Pi_m(\alpha), m'_{\Pi_m(\alpha)} \circ \varepsilon_\alpha^{-1} \rangle$ is a final pullback complement.*

5 Putting SqPO into context

This section is dedicated to the relation of sesqui-pushout rewriting to the double- and single-pushout approach, which are the most widely used categorical approaches to rewriting. It should be mentioned that SqPO rewriting can also be seen as a "conceptual instance" of the very general categorical approach proposed by Wolfram Kahl [13], which is based on fibred categories, but space limitations prevent us to discuss the relationship to the latter.

For the case of left-linear rules, we will, in a certain sense, locate SqPO in between SPO and DPO. In fact we will see that SqPO rewriting coincides with DPO rewriting under mild assumptions, but its deletion mechanism is more general and closer to the one of SPO rewriting.

5.1 Relation between the SqPO and the DPO approach

The definition of SqPO rewriting differs from that of DPO rewriting only in the construction of the left square, which is a final pullback in the former case, and a pushout in the latter. Therefore, whenever the pushout complement of a match with respect to (the left-hand side of) a rule exists and it is also a final pullback complement, then the results of both constructions is the same. This holds in a very general case, namely for *left-regular rules* in *quasi-adhesive categories*.[3]

[3] Hence it also holds for *left-linear rules* in *adhesive categories*, which include **Set**, **Graph**, and several categories of graph-like objects. In fact, an adhesive category is a quasi-adhesive one where all monos are regular (see [16]).

Proposition 12 (DPO **vs.** SqPO). *Let $\mathbb{C}$ be a quasi-adhesive category, let $q = L \overset{\alpha}{\leftarrowtail} K \xrightarrow{\beta} R$ be a left-regular rule (i.e., such that α is regular mono) and let $A \xleftarrow{m} L$ be a match in $\mathbb{C}$. Then any pushout complement $A \xleftarrow{\gamma} D \xleftarrow{i} K$ for $A \xleftarrow{m} L \overset{\alpha}{\leftarrowtail} K$ is a final pullback complement. As a consequence, the following hold.*

1. *If $A \underset{\mathrm{DPO}}{\overset{\langle m,q\rangle}{\Longrightarrow}} B$ then also $A \overset{\langle m,q\rangle}{\Longrightarrow} B$.*
2. *If $A \overset{\langle m,q\rangle}{\Longrightarrow} B$ and a pushout complement of $A \xleftarrow{m} L \overset{\alpha}{\leftarrowtail} K$ exists, then also $A \underset{\mathrm{DPO}}{\overset{\langle m,q\rangle}{\Longrightarrow}} B$.*

Proof. In [16] it is shown (Lemma 2.3) that in a quasi-adhesive category pushouts along regular monos are pullbacks.
Furthermore, it is proved (Lemma 2.8) that if the square to the right is a pushout and α is regular, then $\gamma\colon D \to A$ enjoys the universal property of the right adjoint to the pullback functor m^* at α, i.e., $\gamma \cong \Pi_m(\alpha)$.

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "\lrcorner"] & K \arrow[l, tail, "\alpha"'] \arrow[d, "i"] \\ A & D \arrow[l, "\gamma"] \end{tikzcd}$$

Thus by Lemma 11 $A \xleftarrow{\gamma} D \xleftarrow{i} K$ is a final pullback complement. □

For non-left-regular rules, as shown by Example 7, there exist in general several pushout complements and hence DPO rewriting is ambiguous. In contrast, SqPO rewriting is always deterministic, and its result models cloning, which cannot be obtained with DPO.

5.2 Relation between the SqPO and the SPO approach

We discuss now the relation between the SPO and the SqPO approach. First we concentrate on algebras for a graph structure, where the SPO approach coincides with the SqPO approach when we restrict the first to conflict-free matches and the latter to left-linear rules. Then we briefly discuss that a similar result holds for non-left-linear rules, in the context of the categorial generalization of the SPO approach presented in [18].

SPO **over graph structures.** Single-pushout rewriting has been defined in [5, 17] for categories of algebras over *graph structures*, i.e., over signatures with unary operator symbols only.[4] For example, **Graph** can be seen as the category of algebras for the signature including two sorts, V and E, and two operator symbols, $\mathsf{src}, \mathsf{tgt}\colon \mathsf{E} \to \mathsf{V}$.

For the rest of this subsection let $\mathbb{C}$ be the category of algebras and total homomorphisms of an arbitrary but fixed graph structure, and let $\mathbb{C}^{\mathsf{p}}$ be the category having the same objects and *partial morphisms* as arrows: that is, an arrow $f\colon X \rightharpoonup Y$ of $\mathbb{C}^{\mathsf{p}}$ is a total homomorphism $f\colon \mathit{dom}(f) \to Y$ from a subalgebra $\mathit{dom}(f) \subseteq X$.

[4] Note that all such categories can be seen as categories of set-valued functors, and therefore they are adhesive (see [16]).

As recalled in the introduction, according to the SPO approach a rule is an arrow $q\colon L \rightharpoonup R$ of $\mathbb{C}^{\mathsf{p}}$, and it is applied to a *total* match $m\colon L \to A$ by constructing a pushout in $\mathbb{C}^{\mathsf{p}}$. This is always possible, because $\mathbb{C}^{\mathsf{p}}$ is co-complete. To simulate such a direct derivation using the SqPO approach, we consider the rule as a span $\hat{q} = L \leftarrowtail dom(q) \xrightarrow{q} R$ in $\mathbb{C}$, and look for the final pullback complement of $A \xleftarrow{m} L \leftarrowtail dom(q)$ in $\mathbb{C}$. Then, as summarized by the next proposition, it is possible to show that the results of the two constructions are equal if and only if the final pullback complement exists, i.e., by Construction 5, if and only if match m is conflict-free with respect to $L \leftarrowtail dom(q)$.

Proposition 13 (SPO vs. SqPO). *Let L, K, R and A be objects; let $q\colon L \rightharpoonup R$ be an arrow of $\mathbb{C}^{\mathsf{p}}$, and $\hat{q} = L \overset{\supseteq}{\leftarrowtail} dom(q) \xrightarrow{q} R$ be the corresponding span in $\mathbb{C}$, and let $m\colon L \to A$ be a total match morphism. Then the following are true*

1. *If $A \overset{\langle m,\hat{q}\rangle}{\Longrightarrow} B$ then $A \underset{\text{SPO}}{\overset{\langle m,q\rangle}{\Longrightarrow}} B$.*
2. *If $A \underset{\text{SPO}}{\overset{\langle m,q\rangle}{\Longrightarrow}} B$ and m is conflict-free then $A \overset{\langle m,\hat{q}\rangle}{\Longrightarrow} B$.*

The square to the right shows the result of SPO rewriting with a rule q and a non-conflict-free match m in category $\mathbf{Set}^{\mathsf{p}}$. Note that the function from the right-hand side of the rule to the resulting set is partial: this effect is often considered as unintuitive, and it is ruled out by imposing suitable constraints on the matches.

$$\begin{tikzcd}
\{①, ②\} \arrow[r, "q", "1\mapsto 1"'] \arrow[d, "m"', "\substack{1\mapsto 0\\ 2\mapsto 0}"] & \{①\} \arrow[d] \\
\{⓪\} \arrow[r] & \varnothing
\end{tikzcd}$$

As shown in [17] the morphism from the right-hand side to the resulting object is total *if and only if* the match is conflict-free, thus SqPO rewriting rules out exactly the SPO direct derivations where this unintuitive effect shows up.

SPO over arbitrary categories The SPO approach has been lifted to an abstract, categorical setting in [14, 18]. Following the approach of [19], in [14] a *partial morphism* in a category $\mathbb{C}$ is defined as an equivalence class of spans of $\mathbb{C}$, where the left arrows are monic. Generalizing even further, in [18] rules are defined as spans like $L \xleftarrow{m} K \xrightarrow{h} R$, where $m \in \mathcal{M}$ and $h \in \mathcal{H}$ are required to belong to two classes of arrows of $\mathbb{C}$ satisfying suitable properties: in particular, it is not required that arrows in $\mathcal{M}$ are mono. Even if the technical details of this analysis are beyond the scope of the present paper, it turns out that for these classes of rules, every SqPO-derivation is a SPO-derivation. Moreover the reverse holds, whenever there exists a final pullback complement for the involved matching morphism. In other words, the statement of Proposition 13 holds true in the more general framework of [18] by replacing in point *2* the condition of conflict-freeness with that of existence of final pullback complements.

5.3 Parallelism

After having discussed which fragments of the classical algebraic approaches are subsumed by the new one, we present the local Church-Rosser theorem as evidence that (part of) the existing parallelism theory can be transferred to the realm of SqPO rewriting. Also a theorem concerning sequential commutativity holds true for SqPO rewriting, but we do not present it because of space limitations. We assume here that objects and arrows belong to a fixed quasi-adhesive category $\mathbb{C}$, and that rules are spans of regular monos.

Definition 14 (Parallel Independence). *Let there be two direct derivations* $G \xRightarrow{\langle m_1,p_1\rangle} H_1$ *and* $G \xRightarrow{\langle m_2,p_2\rangle} H_2$. *Then they are* parallel independent *if there exist morphisms* $u\colon L_1 \to D_2$ *and* $v\colon L_2 \to D_1$, *such that* $\gamma_2 \circ u = m_1$ *and* $\gamma_1 \circ v = m_2$.

$$\begin{tikzcd}
 & R_1 \arrow[dl, "n_1"'] & K_1 \arrow[l, tail, "\beta_1"'] \arrow[r, tail, "\alpha_1"] \arrow[d, tail, "k_1"] & L_1 \arrow[dr, "m_1"] \arrow[rrrd, "u"] & & L_2 \arrow[dl, "m_2"'] \arrow[llld, "v"'] & K_2 \arrow[l, tail, "\alpha_2"'] \arrow[r, tail, "\beta_2"] \arrow[d, tail, "k_2"] & R_2 \arrow[dr, "n_2"] & \\
H_1 & & D_1 \arrow[ll, tail, "\delta_1"'] \arrow[rr, tail, "\gamma_1"'] & & G & & D_2 \arrow[ll, tail, "\gamma_2"'] \arrow[rr, tail, "\delta_2"'] & & H_2
\end{tikzcd}$$

Definition 14 can be seen as a conservative extension of the definitions given in the literature for SPO and DPO. More precisely, if two SPO direct derivations are also SqPO-derivations, then they are parallel independent in the SPO sense if and only if they are so according to Definition 14. The same holds for DPO-parallel independence as well, obviously.

Theorem 15 (Local Church-Rosser).
Given two parallel independent direct transformations $G \xRightarrow{\langle m_1,p_1\rangle} H_1$ *and* $G \xRightarrow{\langle m_2,p_2\rangle} H_2$, *there are an object* G' *and direct transformations* $H_1 \xRightarrow{\langle m_2',p_2\rangle} G'$ *and* $H_2 \xRightarrow{\langle m_1',p_1\rangle} G'$.

$$\begin{tikzcd}
 & G \arrow[dl, Rightarrow, "\langle m_1,p_1\rangle"'] \arrow[dr, Rightarrow, "\langle m_2,p_2\rangle"] & \\
H_1 \arrow[dr, Rightarrow, "\langle m_2',p_2\rangle"'] & & H_2 \arrow[dl, Rightarrow, "\langle m_1',p_1\rangle"] \\
 & G' &
\end{tikzcd}$$

The proof of this theorem is very similar to the one given in [16], the difference being that we need some additional SqPO-specific lemmas.

6 Conclusion

We have proposed a new algebraic approach to rewriting in arbitrary categories, called *sesqui-pushout rewriting*, and we discussed its basic properties. In the classical case of graphical structures and left-linear rules, its relation to the SPO and DPO approaches is summarized by the following table, where application conditions are listed below the features of the approaches.

	DPO	$\lesssim$	SqPO	$\lesssim$	SPO
deletion in unknown context	-		✓		✓
precedence of deletion over preservation	-		-		✓
indentification and dangling condition	✓		-		-
conflict-free matches	✓		✓		-

We have a chain of simulations (indicated by the symbol $\lesssim$): every DPO derivation is a SqPO derivation, and every SqPO derivation is a SPO derivation, by seeing left-linear rules as partial morphisms. Furthermore, when DPO rewriting is not possible because the dangling condition is not satisfied, if SqPO rewriting is possible then SPO is possible as well, and both model deletion in unknown context. Finally, when SqPO rewriting is not possible because the match is not conflict-free, then DPO rewriting is not possible because the identification condition is not satisfied, but SPO rewriting is possible and the conflict is resolved in favour of deletion. However, in this case there is no total morphism from the right-hand side of the rule to the resulting graph: an effect that is often considered as undesirable, and that is ruled out automatically by the new approach.

Probably the most original and interesting feature of sesqui-pushout rewriting is the fact that it can be applied to non-left-linear rules as well, and in this case it models the cloning of structures.

We presented a Local Church Rosser theorem for the new approach. We are confident that most of the parallelism and concurrency theory of the DPO and SPO approaches can be lifted smoothly to sesqui-pushout rewriting: this is a topic of ongoing research. Concluding, let us remark that we compared the new approach only with SPO and DPO because they are the most widely used categorical approach to rewriting, but there are several others to which sesqui-pushout rewriting has to be related as well, including the fibred approach by Kahl [13], the double-pullback approach by Heckel [12], and the pullback approach by Bauderon [1].

Acknowledgements. We would like to thank Paolo Baldan and Paweł Sobociński for enlightening discussions about the topic of the paper.

References

1. Michel Bauderon. A uniform approach to graph rewriting: The pullback approach. In M. Nagl, editor, *Graph-Theoretic Concepts in Computer Science*, volume 1017 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 1995.
2. Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, Reiko Heckel, and Michael Löwe. Algebraic approaches to graph transformation—part I: Basic concepts and double pushout approach. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 1: Foundations*, chapter 3. World Scientific, 1997.
3. Roy Dyckhoff and Walter Tholen. Exponentiable morphisms, partial products and pullback complements. *Journal of Pure and Applied Algebra*, 49(1-2):103–116, 1987.
4. Hartmut Ehrig, Annegret Habel, Hans-Jörg Kreowski, and Francesco Parisi-Presicce. Parallelism and concurrency in high level replacement systems. *Mathematical Structures in Computer Science*, 1:361–404, 1991.
5. Hartmut Ehrig, Reiko Heckel, Martin Korff, Michael Löwe, Leila Ribeiro, Annika Wagner, and Andrea Corradini. Algebraic approaches to graph transformation—part II: Single pushout approach and comparison with double pushout approach. In G. Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation, Vol.1: Foundations*, chapter 4. World Scientific, 1997.

6. Hartmut Ehrig and Michael Löwe. Categorical principles, techniques and results for high-level-replacement systems in computer science. *Applied Categorical Structures*, 1:21–50, 1993.
7. Hartmut Ehrig, Michael Pfender, and Hans-Jürgen Schneider. Graph grammars: An algebraic approach. In *Proc. 14th IEEE Symp. on Switching and Automata Theory*, pages 167–180, 1973.
8. David F. Ferriolo, D. Richard Kuhn, and Ramaswamy Chandramouli. *Role-Based Access Control*. Artech House computer security series. Artech House, 2003. ISBN: 1-580-53370-1.
9. Robert Goldblatt. *Topoi: The Categorial Analysis of Logic*. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, Amsterdam, The Netherlands, 1984.
10. Annegret Habel, Jürgen Müller, and Detlef Plump. Double-pushout graph transformation revisited. *Mathematical Structures in Computer Science*, 11(5):637–688, 2001.
11. Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in operating systems. *Commun. ACM*, 19(8):461–471, 1976.
12. Reiko Heckel, Hartmut Ehrig, Uwe Wolter, and Andrea Corradini. Double-pullback transitions and coalgebraic loose semantics for graph transformation systems. *Applied Categorical Structures*, 9(1), 2001.
13. Wolfram Kahl. A fibred approach to rewriting — how the duality between adding and deleting cooperates with the difference between matching and rewriting. Technical Report 9702, Fakultät für Informatik, Universität der Bundeswehr München, May 1997.
14. Richard Kennaway. Graph Rewriting in Some Categories of Partial Morphisms. In Hartmut Ehrig, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *Proc. 4th. Int. Workshop on Graph Grammars and their Application to Computer Science*, volume 532 of *Lecture Notes in Computer Science*, pages 490–504. Springer-Verlag, 1991.
15. Manuel Koch, Luigi V. Mancini, and Francesco Parisi-Presicce. A formal model for role-based access control using graph transformation. In Frédéric Cuppens, Yves Deswarte, Dieter Gollmann, and Michael Waidner, editors, *ESORICS*, volume 1895 of *Lecture Notes in Computer Science*, pages 122–139. Springer, 2000.
16. Stephen Lack and Paweł Sobociński. Adhesive and quasiadhesive categories. *Theoretical Informatics and Applications*, 39(2):511–546, 2005.
17. Michael Löwe. Algebraic approach to single-pushout graph transformation. *Theoretical Computer Science*, 109:181–224, 1993.
18. Miquel Monserrat, Francesc Rosselló, Joan Torrens, and Gabriel Valiente. Single pushout rewriting in categories of spans I: The general setting. Technical Report LSI-97-23-R, Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 1997.
19. Edmund Robinson and Giuseppe Rosolini. Categories of partial maps. *Information and Computation*, 79(2):95–130, 1988.
20. Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.