Volume 3 Issue 2 ISSN 2631-4444

Compositionality of Rewriting Rules with Conditions

Nicolas Behr and Jean Krivine

Université de Paris, CNRS, IRIF, F-75205, Paris, France

We extend the notion of compositional associative rewriting as recently studied in the rule algebra framework literature to the setting of rewriting rules with conditions. Our methodology is category-theoretical in nature, where the definition of rule composition operations encodes the non-deterministic sequential concurrent application of rules in Double-Pushout (DPO) and Sesqui-Pushout (SqPO) rewriting with application conditions based upon $\mathcal{M}$-adhesive categories. We uncover an intricate interplay between the category-theoretical concepts of conditions on rules and morphisms, the compositionality and compatibility of certain shift and transport constructions for conditions, and thirdly the property of associativity of the composition of rules.

1 Introduction and relation to previous work

Graph rewriting has emerged as a powerful formalism to represent complex systems whose dynamics can be captured by a finite set of rules. The *rule-based modeling* approach, originally introduced by V. Danos and C. Laneve in the early 2000s [28–30], has developed into one of the main frameworks for the study of biochemical reaction systems (in the form of the two main frameworks Kappa [20, 31] and BioNetGen[18, 50]). The approach proposes to model protein-protein interaction networks using graph rewriting models, in which proteins are the vertices of a graph whose connected components denote molecular complexes. A paradigmatic application scenario for rule-based methods in the systems biology community is the study of signaling pathways [31, 33], which are highly intricate biochemical reaction networks ubiquitous in living cells.

While the algorithmic aspects of graph rewriting are well-studied, programming language approaches to modeling with graphs are to date still a comparatively underdeveloped topic. Contrary to classical term rewriting, the notion of a *match* of a graph rewriting rule and its *effects* on a term (a graph) is subject to various definitions, allowing more or less control over possible rewrites. In addition, the mere nature of the graphs that are being rewritten impacts both the algorithmic design and expressiveness of graph rewriting. Category theory is a practical toolkit for equipping graphs with well-defined operational semantics. *Double-Pushout (DPO) rewriting* [25] is a widespread technique, partly because it does not yield side effects when rules are applied (which makes it amenable to static analysis for instance). However, when a graph rewriting rule entails node deletion, DPO semantics will not allow a match of such a rule to trigger if the node that is deleted is connected outside the domain of the match (which would yield side-effects). This has limited the practicality of DPO semantics in the context of biological modeling, where more permissive techniques have been employed. *Sesqui-Pushout (SqPO) rewriting* [26] in particular is the technique that is used to rewrite Kappa graphs [31].

Quite orthogonal to the issue of defining rule matches and effects, having access to a fine-grained control over rule triggering is a key issue when graph rewriting is used as a modeling language. To this aim, graph rewriting rules have been equipped with *application conditions* [42, 46], which can be seen as constraints that need to be checked "on the fly" when a rewrite rule is applied.

This paper presents a *compositional* variant of DPO and SqPO-type rewriting for rules with conditions in a general category-theoretical setting. From a mathematical perspective, while the

Nicolas Behr: nicolas.behr@irif.fr, http://nicolasbehr.com, corresponding author; the work of N.B. was supported by a *Marie Skłodowska-Curie Individual fellowship* (Grant Agreement No. 753750 – RaSiR)

Jean Krivine: jean.krivine@irif.fr, https://www.irif.fr/~jkrivine/homepage/Home.html framework of both types of rewriting developed here relies upon the original definitions of DPO-rewriting (see e.g. [42]) and of SqPO-rewriting [26], new developments are necessary in order to obtain the desired compositionality properties. For rules without conditions, one of the core technical obstacles has proved to be establishing a *compositional associativity* property for sequential compositions of rewriting rules, which for the DPO-type case has been achieved in [11–13], and for the SqPO-type case in [8]. The latter work also established a novel *compositional concurrency* property for SqPO-type rewriting theories. Lifting these results to the settings of rules with application conditions is the core contribution of this work.

The main motivation for our search for compositional rewriting theories is two-fold: in the setting of rewriting without conditions, the notion of *rule algebras* [8, 11, 13, 14] has been developed as a new mathematical framework to encode the concurrent and combinatorial interactions of rewriting rules, which in particular allows one to develop principled and novel analysis techniques for stochastic rewriting systems [11, 13, 15]. Especially the recent results of [15] hint at the intimate interplay of design choices in constructing rewriting systems with regards to their dynamical properties, and with the potential of greatly improving the tractability of the analysis of such systems via judiciously chosen *constraints* on objects and rewriting rules (such as implemented in the form of rigidity constraints in the Kappa formalism [32]).

Our second main motivation originates in the desire to analyze rewriting systems *statically*, rather than via simulation-based techniques. While the traditional rewriting theory generally approaches this problem from the viewpoint of *derivation traces* (i.e. sequences of rule applications to a given input graph), we posit that a viable alternative approach may consist in focusing instead on *sequential rule compositions*, which in particular when combined with application conditions is anticipated to yield a powerful framework to study the *causality* of rewriting systems. For certain specialized applications of DPO-type graph rewriting, namely those in the well-established field of *chemical graph rewriting* [3, 7, 17], such types of analyses have already proven very fruitful [4–6, 43]. Therefore, we believe that our compositional refinements described in the present paper can provide a significant contribution to future algorithm developments in this field.

1.1 Related work

1.1.1 Traditional rewriting literature

First and foremost, the work presented in this paper relies heavily upon the rich literature on categorical rewriting theories with its almost 50 year long history. The introduction of the seminal concept of *adhesive categories* by Lack and Sobociński in the early 2000s [52, 53] and its refinement to $\mathcal{M}$-adhesive categories in the work of Ehrig et al. [22, 39, 41, 42, 47] resulted in a powerful mathematical framework for *Double-Pushout (DPO)* rewriting, and in particular permitted to unify numerous concepts from the earlier graph rewriting literature [63]. While $\mathcal{M}$-adhesive categories have been demonstrated to be capable of describing a broad range of data structures of relevance in applications of rewriting theories (see e.g. [37] and [12] for curated lists of examples), it is only via a second key refinement of the rewriting theory that more elaborate data structures such as e.g. chemical molecules may be encoded: the theory of *constraints* and *application conditions*. This theory had been a well-established component of some of the earliest categorical formulations of graph rewriting theories since the 1980s by Habel et al. [35, 49], and was later generalized to the $\mathcal{M}$-adhesive setting as utilized in the present paper in [42, 46, 60].

Building upon this extensive body of work, and motivated by applications to modeling in the life sciences (cf. [10] and comments below), we demonstrate in the present paper that by requiring the underlying $\mathcal{M}$-adhesive category to satisfy certain additional technical assumptions, one obtains a refined variant of DPO-semantics that is well suited for developing novel static analysis techniques for rewriting systems. Referring to Remarks 1 and 2 in the main text for the precise technical details of the relationship between our “refined” and the “traditional” DPO semantics, our work moreover extends the traditional DPO-theory by providing an *associativity theorem* which had not been previously known in the literature (cf. Theorem 6).

The second type of categorical rewriting semantics considered in this paper is the *Sesqui-Pushout (SqPO)* rewriting theory introduced in [26]. Despite the importance of this type of semantics especially in the modeling of biochemical reaction systems (cf. Section 1.1.2), the approach had been considerably less well-developed than the DPO-variant, lacking in particular a theory of constraints and application conditions as well as notions of *concurrency* and *associativity* theorems. While in [34] an attempt had been made to identify a special sub-class of SqPO-type rewriting systems over $\mathcal{M}$-adhesive categories in which the rewriting semantics coincides with that of the DPO-setting, and thus allowing to reuse the theory of application condition from this setting, this special case of *"reversible"* SqPO-rules constitutes too strong a restriction in order to study the rewriting systems of relevance e.g. in biochemistry (Section 1.1.2). Building upon our earlier work on SqPO-rewriting in the setting of rules without conditions [8], and taking advantages of the close analogies with our refined DPO-type framework as presented in comprehensive technical detail in the present paper, we develop these missing elements of SqPO-rewriting theory (cf. Section 5). Referring to [26] for an extended discussion, it might finally be worthwhile to note that SqPO-semantics in the setting of ($\mathcal{M}$-) linear rules and ($\mathcal{M}$-) monic matches (as is the case in the present paper) is well known to coincide with *single-pushout (SPO)* semantics [54].

1.1.2 Bio- and organo-chemistry

An intriguing trend in modern rewriting theory over the past 15 years has been the development of rule-based modeling approaches in biochemistry (Kappa [20], BioNetGen [50]) and in organic chemistry (MØD [3]), which are based upon SqPO- and DPO-type semantics, respectively. While an detailed discussion of the numerous sophisticated theoretical and algorithmic techniques (including in particular various notions of *static analysis techniques*, cf. e.g. [1, 23, 31–33, 44, 56, 61] and [4, 6, 7, 17, 43]) is beyond the scope of the present paper, suffice it here to comment on a salient technical point: notably, many of the developments in the aforementioned frameworks were not explicitly rooted in categorical rewriting theory itself, despite the foundations of the two fields upon this type of theory. We recently demonstrated in [10] that based upon the results of the present paper, one may not only reformulate equivalently the current implementations of chemical rewriting theories, but one may also take advantage of our novel compositionality and associativity results in order to formulate rule algebra and tracelet theories (see below) to develop new approaches to the static analysis of complex reaction systems and their dynamics. The encodings of chemical molecules and their reactions as reported in [10] in both the bio- and the organo-chemical settings are based upon typed variants of *undirected simple graphs* that satisfy suitable sets of constraints, which is why we will take undirected simple graphs and rewriting rules thereof as a running example throughout this paper.

1.1.3 Rule algebra and tracelet theory

Rule-based modeling approaches for chemical reaction systems are part of a larger class of theories known as *stochastic rewriting theories*. Over the past five years, we have demonstrated that one may express such systems based upon the general theory of *continuous-time Markov chains (CTMCs)* [58], leading to the so-called *stochastic mechanics* framework [8, 10–13, 15]. At the core of this framework is the notion of *rule algebras*, which encode the non-determinism in the choices of admissible matches in sequential compositions of rules via the construction of a certain *binary operation* (on a vector space whose basis is indexed by equivalence classes of linear rules). As explained in detail in [12, 15], since the "blueprint" of this type of construction in the general mathematics and in particular combinatorics literature is provided by the operations of derivative and multiplication by formal variables acting upon the space of formal power series, the aforementioned binary operation must satisfy two fundamental technical properties: it must be *associative* (in the standard sense of associativity of binary operations), and it must possess a *neutral element*. Referring to loc. cit. for the precise technical details, the former property requires an *associativity theorem* for sequential rule compositions, while the latter property requires the underlying category to possess a (strict) $\mathcal{M}$-initial object. Finally, in order to formulate the stochastic rewriting theories themselves, it is necessary to endow the rule algebras with *canonical representations*, the construction of which in turn hinges on the *concurrency theorems* for the respective sort of rewriting. The technical results presented in the present paper have been recently applied in [10] to establish precisely the rule algebra theories for linear rules with conditions both in DPO- and in SqPO-semantics.

A second recent line of developments has been the introduction of the theory of *tracelets* in [9], which also relies heavily on the technical results of the present paper. Intuitively, while the aforementioned rule algebraic constructions aim to reason based upon rule algebra products of linear rules (which, in a sense, encode “sums over all possible ways to compose rules”), the concept of tracelets is more directly related to the classical rewriting-theoretical notion of *derivation traces* [26, 37, 63]. We refer the interested readers to [9] for an in-depth review of the relationship of tracelet theory with the traditional concepts of concurrency and related static analysis techniques, mentioning here only the fact that yet again this new theory relies upon the notion of associative compositional rewriting theories.

1.2 A motivating example: rewriting simple graphs

Since our main constructions will be somewhat technical, let us start with a simple example in order to provide the readers with an intuitive picture of the main concepts. To this end, consider the task of defining a sound notion of rewriting for *undirected simple graphs*, the type of graphs where at most one undirected edge may exist between any two given vertices of the graph, as opposed to undirected multigraphs. While there exist various attempts in the literature to encode simple graphs directly via a suitable category-theoretical construction (cf. e.g. [2, 21, 22, 26, 34]), the disadvantage of such an approach consists of the fact that these categories are not $\mathcal{M}$-adhesive, and thus do not permit to profit from the constructions for $\mathcal{M}$-adhesive categories available in the general DPO-rewriting framework. In fact, constructions of simple graph categories such as **RDFGraph** of [21] (for a notion of directed typed simple graphs called RDF graphs) or **SGraph** of [2, 26] (of undirected simple graphs) are well known to lack uniqueness of pushout complements (POCs), quintessential in defining DPO-semantics (Section 4) as well as (compositional) SqPO-semantics (Section 5) in the first place. Albeit several elaborate workarounds to this problem have been considered in the literature (such as e.g. the *minimal-POC-PO* semantics for RDF-graph rewriting in [21]), these workarounds introduce alterations to the rewriting semantics that do not necessarily reflect the intuitions one might have for simple graph rewriting from a mathematics or general applied sciences background.

To circumvent these issues, we will instead start our construction from a “host category” **uGraph** of *undirected multigraphs*[1] [10, 12, 59] that is itself $\mathcal{M}$-adhesive and satisfies all requirements to admit compositional rewriting theories of both DPO- and SqPO-type. Imposing the aforementioned structural constraint prohibiting parallel edges via the framework of conditions, we obtain an algorithmically versatile implementation of simple graphs. Following the rewriting theory paradigms, one may envision manipulating graphs via application of *graph rewriting rules*. Intuitively, specifying a rewriting rule amounts to providing the following data:

- An *Input pattern* I,
- an *Output pattern* O, and
- a *Kontext pattern* K together with embedding morphisms $i : K \hookrightarrow I$ and $o : K \hookrightarrow O$.

Then in order to apply a rule $r := (O \overset{o}{\hookleftarrow} K \overset{i}{\hookrightarrow} I)$ to a graph G, one has to

1. specify an embedding $m : I \hookrightarrow G$ of the input pattern I into G (i.e. one has to select a particular occurrence of the pattern I in G), referred to as a *match*,
2. in $m(I) \subset G$, replace $m(I)$ with $m \circ i(K)$, and
3. “glue” a copy of O onto $m \circ i(K)$ (according to the embedding morphism $o : K \hookrightarrow O$).

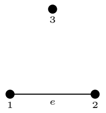
For example, one may consider the rewriting rules

$$e_+ := (\bullet\!\!-\!\!\bullet \hookleftarrow \bullet\ \ \bullet \hookrightarrow \bullet\ \ \bullet)\ , \quad e_- := (\bullet\ \ \bullet \hookleftarrow \bullet\ \ \bullet \hookrightarrow \bullet\!\!-\!\!\bullet)\ , \tag{1}$$

which implement the manipulations of *adding* a new edge between two vertices (e_+) and *deleting* an edge between two vertices (e_-). However, in the setting of rewriting of *simple* graphs, the

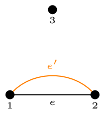
[1] Interestingly, the category **uGraph** constitutes one of the simplest non-trivial examples of a category that is $\mathcal{M}$-adhesive, but *not* adhesive, unlike the prototypical example of an adhesive category **Graph** [52].

above tentative definition for the rewriting of graphs is as of yet incomplete. For example, given the graph below,



,

if we were to apply our edge creation rule at a match comprising the vertices marked 1 and 2, we would produce the non-simple graph depicted below (with e' the newly produced edge):



Therefore, in order to ensure that our transformations via application of rules keep intact the constraint of graphs to be simple, we need to endow the rules with *application conditions*. While this approach is well known in the rewriting literature, it turns out that, via a careful re-implementation of the traditional framework, some interesting mathematical structures may be uncovered: rules with conditions are endowed with a structure of composition operation that allows one to synthesize sequences of causally sound rewriting steps *without* reference to a host object. Crucially, this operation can be shown to be *associative*. Contact with the traditional techniques of rewriting is then made in the form of a suitable adaption of the concurrency theorem, whereby a sequence of rule applications to a given object can always be equivalently described by a "one-shot" application of a sequential composite of the rules to that object (i.e. in a sense characterizing the effect of the sequential rule applications by the application of a single composite rule). The most subtle part of our results then consists in a certain compositional associativity property present in triple sequential compositions of rules.

To relate to the example at hand, in addition to using the standard application condition technique to express *constraints* on rules (such as the constraint that the edge creation rule should only be applied to a graph if its input is matched to two vertices that are not already linked), we will also be able to compute *causal information*, such as e.g. that applying the edge creation rule e_+ followed by applying the edge deletion rule e_- (in a fashion such as to delete the previously created edge) will lead to a rule with no effect, which can only be applied at two vertices that cannot be already linked (transforming the graph identically). Finally, we will present examples of the property of associativity in Examples 5 and 6 in the setting of DPO- and SqPO-type rewriting theories, respectively.

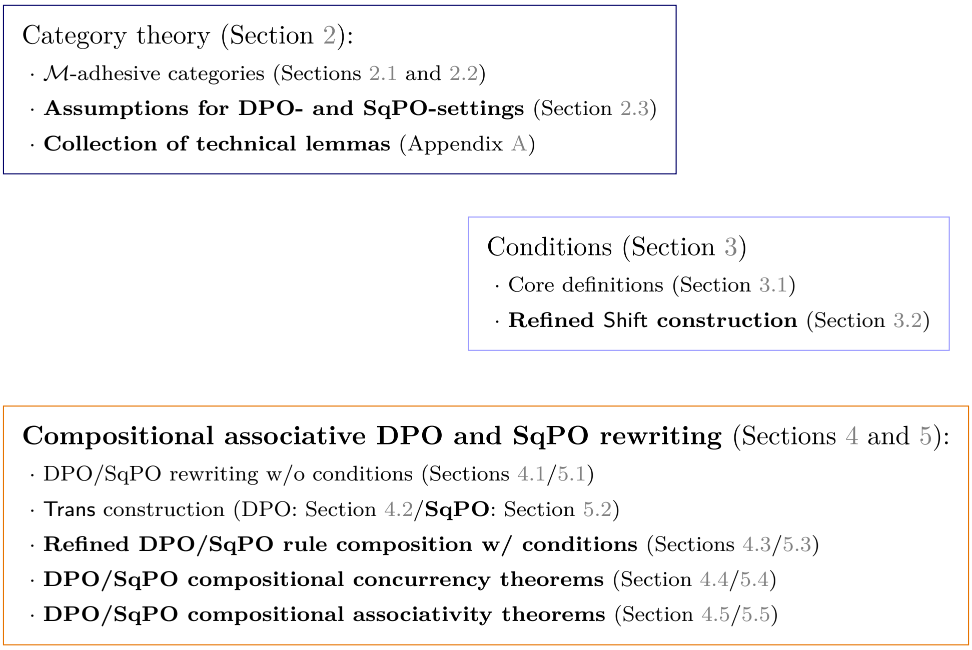
1.3 Overview of main results and structure of the paper

The main results of this paper are summarized as follows (cf. Figure 1):

- We provide a self-contained account of all category-theoretical prerequisites and specialized definitions extracted from the rich literature on categorical rewriting theories (Sections 2 and 3; Appendix 6), intended not least as an entry-point for the general applied category theory audience.

- For *Double-Pushout (DPO) rewriting* (Section 4), we identify a set of sufficient assumptions on the underlying categories such that the resulting rewriting theories have suitable *compositionality properties* (in the sense of a certain form of concurrency and associativity properties of sequential rule compositions). This part of our theory is thus to a large extent a careful "fine-tuning" of results from the traditional DPO-rewriting literature, combined with a number of original results.

- For *Sesqui-Pushout (SqPO) rewriting* (Section 5), we present the first-of-its-kind category-theoretical compositional SqPO-type rewriting theory for rules with conditions, profiting from fruitful analogies to results in the DPO-type theory as presented in the first part of the paper.



Category theory (Section 2):
- $\mathcal{M}$-adhesive categories (Sections 2.1 and 2.2)
- **Assumptions for DPO- and SqPO-settings** (Section 2.3)
- **Collection of technical lemmas** (Appendix A)

Conditions (Section 3)
- Core definitions (Section 3.1)
- **Refined Shift construction** (Section 3.2)

Compositional associative DPO and SqPO rewriting (Sections 4 and 5):
- DPO/SqPO rewriting w/o conditions (Sections 4.1/5.1)
- Trans construction (DPO: Section 4.2/**SqPO**: Section 5.2)
- **Refined DPO/SqPO rule composition w/ conditions** (Sections 4.3/5.3)
- **DPO/SqPO compositional concurrency theorems** (Section 4.4/5.4)
- **DPO/SqPO compositional associativity theorems** (Section 4.5/5.5)

Figure 1: Structure and **original contributions** of the paper.

2 Category-theoretical preliminaries

Rewriting in its modern formulations is a concept that heavily relies on specific types of categorical structures. In this section, we collect all the necessary prerequisites that allow one to formulate consistent frameworks of associative rewriting theories with conditions on objects and morphisms, in both the *Double-Pushout (DPO)* and the *Sesqui-Pushout (SqPO)* approaches. While many of the mathematical details of these setups are by now standard in the literature, we will emphasize the specific additional conditions that are required in order to guarantee *associativity* (in the sense of [11]) and *compositionality*, where both of these properties concern concurrent compositions of rules with conditions.

2.1 $\mathcal{M}$-adhesive categories

We begin by quoting a number of essential definitions and standard results from the literature, where our main references will be [22, 42, 46] (see also [62, 64] for some more recent works). Let us first recall the notion of $\mathcal{M}$-adhesive categories, which is the most general mathematical setting currently known that allows one to define DPO- and SqPO-type rewriting theories, and which generalizes adhesive categories [53].

Definition 1 ([22], Def. 2.1) *Let* $\mathbf{C}$ *be a category, and let* $\mathcal{M}$ *be a class of* monomorphisms. *Then the data* $(\mathbf{C}, \mathcal{M})$ *defines an* $\mathcal{M}$-adhesive category *if the following requirements are satisfied:*

(i) The class $\mathcal{M} \subset \mathsf{mor}(\mathbf{C})$ *contains all isomorphisms and is* closed under

(a) composition,

$$\forall g \circ f \in \mathsf{mor}(\mathbf{C}) : \; f \in \mathcal{M} \wedge g \in \mathcal{M} \Rightarrow g \circ f \in \mathcal{M}\,, \tag{2}$$

(b) decomposition,

$$\forall g \circ f \in \mathcal{M} : \; g \in \mathcal{M} \Rightarrow f \in \mathcal{M}\,. \tag{3}$$

(ii) $\mathbf{C}$ has pushouts and pullbacks along $\mathcal{M}$ morphisms, *i.e. pushouts of spans and pullbacks of cospans where at least one of the two morphisms of the (co-) span is in* $\mathcal{M}$ *exist.*

(iii) $\mathcal{M}$ morphisms are closed under pushouts and pullbacks: *if in the diagram below* (1) *is a pushout, then* $m \in \mathcal{M}$ *implies* $n \in \mathcal{M}$*, while if* (1) *is a pullback, then* $n \in \mathcal{M}$ *implies* $m \in \mathcal{M}$*:*

$$\begin{tikzcd}
A \arrow[r] \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] & C \arrow[d, "n"] \\
B \arrow[r] & D
\end{tikzcd} \tag{4}$$

(iv) Pushouts along $\mathcal{M}$-morphisms are $\mathcal{M}$-van Kampen ($\mathcal{M}$-VK) squares (also referred to as vertical weak VK squares in the literature): *given a commutative cube in* $\mathbf{C}$ *as shown below, where the bottom square is a pushout along an* $\mathcal{M}$*-morphism* $m \in \mathcal{M}$*, the back faces are both pullbacks, and if* $b, c, d \in \mathcal{M}$*, then the top face is a pushout if and only if the two front faces are pullbacks.*

$$\begin{tikzcd}
& A' \arrow[rr] \arrow[dl] \arrow[dd] & & C' \arrow[dl] \arrow[dd, "c"] \\
B' \arrow[rr, crossing over] \arrow[dd, "b"'] & & D' & \\
& A \arrow[rr, "f"] \arrow[dl, "m"'] & & C \arrow[dl] \\
B \arrow[rr] & & D \arrow[from=uu, crossing over, "d"] &
\end{tikzcd} . \tag{5}$$

For certain applications, it is also of interest to consider a variant of the definition called weak $\mathcal{M}$-adhesive categories*, which are categories in which all of the above axioms hold except for the* $\mathcal{M}$*-VK property; the latter is modified to the* weak $\mathcal{M}$-VK property*:*

(iv)' Pushouts along $\mathcal{M}$-morphisms are *weak* $\mathcal{M}$*-van Kampen (*$\mathcal{M}$*-VK) squares*: *given a commutative cube in* $\mathbf{C}$ *as shown in* (5)*, where the bottom square is a pushout along an* $\mathcal{M}$*-morphism* $m \in \mathcal{M}$*, the back faces are both pullbacks, and if*

$$(b, c, d \in \mathcal{M}) \quad \textit{or} \quad (f \in \mathcal{M})\,,$$

then the top face is a pushout if and only if the two front faces are pullbacks.

Strictly speaking, several points in the above definition are redundant, since the closure under isomorphisms (i) and the decomposition property (ib) are known to follow directly from closure under compositions and stability of $\mathcal{M}$-morphisms under pullbacks (see Appendix B for the former property). It is moreover worthwhile to note that in some of the earlier literature (cf. [41] for a review), several authors referred to categories satisfying properties ($i - iv$) as ***vertical*** *weak* $\mathcal{M}$*-adhesive categories*, and by analogy to categories satisfying ($i - iii$) and only the second alternative of (iv') (i.e. that the "horizontal" morphisms m and f should be in $\mathcal{M}$) as ***horizontal*** *weak* $\mathcal{M}$*-adhesive categories.* In practice (cf. e.g. [12] for a review and a list of examples), most categories considered in the rewriting literature in fact qualify as both horizontal and vertical weak $\mathcal{M}$-adhesive categories, yet since a few exceptions only satisfy the vertical variant, and since moreover in the relevant proofs only this variant is required, the standard convention has become to refer to the vertical variant simply as $\mathcal{M}$-adhesive categories [22]. $\mathcal{M}$-adhesive categories enjoy a number of special properties (some of which referred to in the literature as *high-level replacement (HLR) properties*) that will be important to our main constructions. We collect these properties in Appendix A.

As advocated in particular in the work of Ehrig [22], one of the optimal compromises for a general setting in which DPO (and, as we shall see, also SqPO) rewriting theories involving constraints on objects and morphisms can be formulated efficiently is provided by $\mathcal{M}$-adhesive categories with certain additional special properties. A central role in this setup is played by the following concepts:

Definition 2 **($\mathcal{M}$-initial object; [22], Def. 2.5)** *An object $\varnothing$ of an $\mathcal{M}$-adhesive category $\mathbf{C}$ is defined to be an* $\mathcal{M}$-initial object *if for each object $A \in \mathsf{obj}(\mathbf{C})$ there exists a unique monomorphism $i_A : \varnothing \hookrightarrow A$, which is moreover required to be in $\mathcal{M}$. An $\mathcal{M}$-initial object $\varnothing$ is said to be* strict *if for each object $X \in \mathsf{obj}(\mathbf{C})$, every morphism $X \to \varnothing$ must be an isomorphism.*

Lemma 1 **([22], Fact 2.6)** *If an $\mathcal{M}$-adhesive category $\mathbf{C}$ possesses an $\mathcal{M}$-initial object $\varnothing \in \mathsf{obj}(\mathbf{C})$, then the category has* finite coproducts, *and moreover the coproduct injections are in $\mathcal{M}$. In particular, the coproduct $A + B$ of two objects $A, B \in \mathsf{obj}(\mathbf{C})$ is then given as the pushout of the span $(A \xleftarrow{i_A} \varnothing \xrightarrow{i_B} B)$.*

For later convenience, we present a number of consequences of an $\mathcal{M}$-adhesive category possessing a (strict) $\mathcal{M}$-initial object in Appendix D.

Definition 3 **(Finite objects, finitary categories, finitary restrictions; [22], Def. 2.8 and Def. 4.1)** *Let $\mathbf{C}$ be an $\mathcal{M}$-adhesive category. An object $A \in \mathsf{obj}(\mathbf{C})$ is said to be* finite, *if there exist only finitely many isoclasses of $\mathcal{M}$-morphisms $B \hookrightarrow A$ into A (i.e. if "A has finitely many $\mathcal{M}$-subobjects"). $\mathbf{C}$ is* finitary *if all its objects are finite. Let $\mathbf{C}_{fin}$ denote the full subcatgory of $\mathbf{C}$ spanned by finite objects, and let $\mathcal{M}_{fin}$ denote the class of $\mathcal{M}$-morphisms between finite objects. Then we refer to $(\mathbf{C}_{fin}, \mathcal{M}_{fin})$ as the* finitary restriction *of $\mathbf{C}$.*

Theorem 1 **([22], Thm. 3.14)** *The finitary restriction $(\mathbf{C}_{fin}, \mathcal{M}_{fin})$ of an $\mathcal{M}$-adhesive category $\mathbf{C}$ is a finitary $\mathcal{M}$-adhesive category.*

Definition 4 **(Epi-$\mathcal{M}$-factorizations; cf. e.g. [46], Def. 3)** *An $\mathcal{M}$-adhesive category $\mathbf{C}$ is said to possess an* epi-$\mathcal{M}$-factorization *if every morphism f of $\mathbf{C}$ can be factorized into an epimorphism e and a monomorphism $m \in \mathcal{M}$ such that $f = m \circ e$, and such that this factorization is unique up to isomorphism.*

It is worthwhile to note that a large class of $\mathcal{M}$-adhesive categories of practical importance (see e.g. the examples listed below) indeed possess an epi-$\mathcal{M}$-factorization.

Example 1 *Referring to [22, Ex. 2.3ff] and [37, Section 4.2] for further details, well-known examples of $\mathcal{M}$-adhesive categories with $\mathcal{M}$-initial objects include:*

- $(\mathbf{Set}, \mathcal{M}_S)$, *the category of* sets and (total) set functions, *with $\mathcal{M}_S$ the class of all injective set morphisms, and with the $\mathcal{M}_S$-initial object the empty set $\varnothing_S \in \mathsf{obj}(\mathbf{Set})$.*
- $(\mathbf{uGraph}, \mathcal{M}_U)$, *the category of* undirected multigraphs *[12] and graph homomorphisms, with $\mathcal{M}_U$ the class of all injective homomorphisms of $\mathbf{uGraph}$, and with $\mathcal{M}_U$-initial object the empty graph $\varnothing_U \in \mathsf{obj}(\mathbf{uGraph})$.*
- $(\mathbf{Graph}, \mathcal{M}_G)$, *the category of* directed multigraphs, *with $\mathcal{M}_G$ the class of all injective graph homomorphisms, and with $\mathcal{M}_G$-initial object the empty graph $\varnothing_G \in \mathsf{obj}(\mathbf{Graph})$.*
- $(\mathbf{Graph}_{TG}, \mathcal{M}_{TG})$, *the category of* typed graphs *and morphisms thereof (constructed as the slice category $\mathbf{Graph}_{TG} := \mathbf{Graph}/TG$ for some fixed type graph $TG \in \mathbf{Graph}$), with $\mathcal{M}_{TG}$ the class of all injective typed graph homomorphisms.*
- *The categories of* Petri nets *and of* elementary Petri nets *[22, Ex. 2.3ff] are $\mathcal{M}$-adhesive and $\mathcal{M}$-initial for certain classes of $\mathcal{M}$.*

*All categories in the above list possess an epi-$\mathcal{M}$-factorization, as do their finitary restrictions. For example, in $\mathbf{Graph}$, every graph homomorphism can be factored into the a surjective composed with an injective graph homomorphism. Interestingly, the well-known **non-examples**, which are certain categories of (typed or untyped) attributed graphs, fail to possess an $\mathcal{M}$-initial object and an epi-$\mathcal{M}$-factorization. There exist a number of* functorial constructions *that allow one to construct finitary $\mathcal{M}$-adhesive categories with the desired properties from known such categories. We refer the reader to [22, Sec. 5] for the details.*

One of the most important additional properties required in view of compositionality of rewriting rules is the following one:

Definition 5 ($\mathcal{M}$-effective unions) *Let $\mathbf{C}$ be an $\mathcal{M}$-adhesive category, for $\mathcal{M}$ a class of monomorphisms. Then $\mathbf{C}$ is said to possess* $\mathcal{M}$-effective unions *if for every commutative diagram as below where all morphisms except d are in $\mathcal{M}$, where (1) is a pushout and where the exterior square is a pullback,*

$$\begin{tikzcd}
 & A \arrow[dl, "b'"'] \arrow[dr, "c'"] & \\
B \arrow[dr, "e"] \arrow[ddr, bend right, "b"'] & (1) & C \arrow[dl, "f"'] \arrow[ddl, bend left, "c"] \\
 & D \arrow[d, dotted, "\exists!\, d"] & \\
 & E &
\end{tikzcd}\quad , \tag{6}$$

the morphism d is in $\mathcal{M}$.

While *adhesive categories* (which constitute a special case of $\mathcal{M}$-adhesive categories where $\mathcal{M} = \mathsf{mono}(\mathbf{C})$) are well known to posses ($\mathsf{mono}(\mathbf{C})$-) effective unions [53], we are not aware of a set of sufficient conditions to ensure the property of $\mathcal{M}$-effective unions in the general $\mathcal{M}$-adhesive case. Referring to [12] for an extended discussion, for some of these more general cases such as for the category **uGraph** [10] the following technical result allows one to verify the property:

Theorem 2 ([12], Thm. 1.15) *Let $\mathbf{C}$ be a* horizontal weak *$\mathcal{M}$-adhesive category. Then in a diagram of the form (6), the morphism d is a monomorphism (however not necessarily in the class $\mathcal{M}$).*

Finally, the property of balancedness defined below will be an essential ingredient for our rewriting framework (cf. Theorem 3).

Definition 6 (compare [53], Lem. 4.9) *A category is said to be* balanced *if every morphism that is both a mono- and an epimorphism is an isomorphism.*

2.2 Additional prerequisites for the Sesqui-Pushout (SqPO) framework

Referring to [8] for a more extensive presentation, we focus here on quoting some necessary background materials, and on discussing the general $\mathcal{M}$-adhesive setting.

Definition 7 (Final Pullback Complement (FPC); [26, 55]) *Given a commutative diagram of the form*

$$\begin{tikzcd}
 & & P \arrow[dll, bend right, "x"'] \arrow[dl, "w"] \arrow[dd, "y"] \\
B \arrow[d, "c"'] & A \arrow[l, "a"'] \arrow[d, "b"] & \\
C & D \arrow[l, "d"] & Q \arrow[ll, bend left, "z"] \arrow[ul, dotted, "\exists!\, w^{*}"']
\end{tikzcd}\quad , \tag{7}$$

a pair of morphisms (d, b) is a final pullback complement (FPC) *of a pair (c, a) if*

(i) (a, b) is a pullback of (c, d), and

(ii) for each collection of morphisms (x, y, z, w) as in (7), where (x, y) is pullback of (c, z) and where $a \circ w = x$, there exists a morphism w^{*} with $d \circ w^{*} = z$ and $w^{*} \circ y = b \circ w$ that is unique (up to isomorphisms).

Lemma 2 (cf [55], Fact 2, and [26], Lemma 2ff) *For an arbitrary morphism $f : A \to B$, (id_B, f) is an FPC of (f, id_A) and vice versa. Moreover, every pushout square is also an FPC square. FPCs are unique up to isomorphism and preserve monomorphisms.*

If we are working over an *adhesive category* (i.e. an $\mathcal{M}$-adhesive category where $\mathcal{M}$ coincides with the class of all monomorphisms [22]), the stability of monomorphisms under FPCs as guaranteed by Lemma 2 will be sufficient for our purposes. However, in the more general $\mathcal{M}$-adhesive setting, we will have to require the following stronger property:

Definition 8 (Stability of $\mathcal{M}$-morphisms under FPCs) *Let* $\mathbf{C}$ *be an $\mathcal{M}$-adhesive category (for $\mathcal{M}$ a class of monomorphisms). Then $\mathcal{M}$-morphisms in* $\mathbf{C}$ *are said to be* stable under FPCs *if whenever for a pair of morphisms (a, b) with $a : A \hookrightarrow B$ in $\mathcal{M}$ and $b : C \to B$ arbitrary, if (b', a') such that $a \circ b = b' \circ a'$ is the FPC of (a, b), then $b' \in \mathcal{M}$.*

2.3 Summary: full set of assumptions for DPO and SqPO rewriting

Combining all findings of the previous two sections (together with some insights from the constructions presented in the following sections), we present here sets of requirements for associative Double-Pushout (DPO) and Sesqui-Pushout (SqPO) rewriting that admit conditions on both objects and morphisms. As we were primarily motivated by deriving a *sufficient* set of assumptions in order to study rewriting theories in the life-sciences (as in [10]), we have not attempted to prove that the assumptions provided are strictly *necessary*, albeit the latter point might in itself provide an interesting direction for future work.

Assumption 1 (Associative DPO rewriting with conditions) *We assume that* $\mathbf{C}$ *is an* $\mathcal{M}$-adhesive category *with* epi-$\mathcal{M}$-factorization. *We also assume that* $\mathbf{C}$ *is* balanced, *possesses a* strict $\mathcal{M}$-initial object $\varnothing \in \mathsf{obj}(\mathbf{C})$ *and* $\mathcal{M}$-effective unions.

Note that according to [53, Lem. 4.9], all adhesive categories are balanced, while for the general case the requirement is more non-trivial to demonstrate. An example of the former class which also satisfies all additional assumptions stated above is given by the category **Graph** of directed multigraphs [53], while a more general example of an $\mathcal{M}$-adhesive category satisfying Assumption 1 is given by the category **uGraph** [10, 12] of undirected multigraphs.

Assumption 2 (Associative SqPO rewriting with conditions) *We assume that* $\mathbf{C}$ *is an* $\mathcal{M}$-adhesive category *satisfying Assumption 1, and we assume in addition that for all pairs of composable $\mathcal{M}$-morphisms $A \overset{m}{\hookrightarrow} B \overset{n}{\hookrightarrow} C$, the final pullback complement exists, and moreover that $\mathcal{M}$-morphisms are stable under FPCs.*

According to [27], examples of $\mathcal{M}$-adhesive categories for which the existence of FPCs as required in Assumption 2 is guaranteed are those categories that possess an $\mathcal{M}$-partial map classifier (cf. [27, Thm. 1 and Sec. 2–5]; compare [24]). While we refer the interested readers to these references for the full technical details, we mention here that examples of categories possessing an $\mathcal{M}$-partial map classifier include the $\mathcal{M}$-adhesive categories **Set** and **Graph**, all presheaf categories, numerous variants of *typed* or *polarized* graphs, and more generally all slice categories $\mathbf{C} \downarrow X$ with $\mathbf{C}$ a topos and $X \in \mathsf{obj}(\mathbf{C})$. To the best of our knowledge there are no known sufficient conditions to guarantee this stability property, other than a result by J.R.B. Cockett and S. Lack [24, Prop. 4.16 and Example 4.17], which however in effect only reaffirms the case of $\mathbf{C}$ being an adhesive category. In the general $\mathcal{M}$-adhesive setting, $\mathcal{M}$-stability under FPCs will have to be verified at a case-by-case level. Note that the guaranteed existence of final pullback complements in the configurations encountered in SqPO rewriting will drastically simplify the framework, and is in fact necessary to guarantee associativity as discussed in [8].

3 Conditions on objects and morphisms

The central concepts of the framework of *conditions* are the following notions of *constraints* (i.e. conditions over objects), *application conditions* (i.e. conditions over *morphisms*) and the associated notions of satisfiability. We quote the precise definitions from [42], and also from [46], where some important clarifying details are given (on the notion of satisfiability on objects and morphisms).

3.1 Core definitions

Definition 9 (Conditions; cf. [42], Def. 3.3, and [46], Def. 4) *(Nested)* conditions *are recursively defined as follows:*

(i) Trivial condition: *for every object* $P \in \mathsf{obj}(\mathbf{C})$*,* true *is a condition over* P*.*

(ii) "Transported" conditions: *for every object* $P \in \mathsf{obj}(\mathbf{C})$*, for every*[2] $\mathcal{M}$*-morphism* $(a : P \to Q)$ *and for every condition* c_Q *over* Q*,* $\exists(a : P \to Q, \mathsf{c}_Q)$ *is a condition over* P*.*

(iii) Negation: *for every condition* c_P *over* $P \in \mathsf{obj}(\mathbf{C})$*,* $\neg\mathsf{c}_P$ *is a condition over* P*.*

(iv) Conjunction: *given a family of conditions* $\{\mathsf{c}_P^{(i)}\}_{i\in I}$ *(for some index set* I*) over an object* $P \in \mathsf{obj}(\mathbf{C})$*,* $\wedge_{i\in I}\mathsf{c}_P^{(i)}$ *is a condition over* P*.*

The following two **shorthand notations** *are customary:*

$$\exists a := \exists(a, \mathsf{true})\,, \quad \forall(a, \mathsf{c}) := \neg\exists(a, \neg\mathsf{c})\,. \tag{8}$$

The precise meaning of the above definitions is specified via the associated notions of satisfiability*, which are also defined inductively:*

S1 *Every* $\mathcal{M}$*-morphism* $p : P \to P'$ *satisfies the trivial condition* true*.*

S2 *Given* $\mathcal{M}$*-morphisms* $p : P \to P'$ *and* $a : P \to Q$ *as well as a condition* $\exists(a, \mathsf{c}_Q)$ *over* P*, the morphism* p *is defined to satisfy the condition* $\exists(a, \mathsf{c}_Q)$ *if and only if there exists an* $\mathcal{M}$*-morphism* $q : Q \to P'$ *such that* $q \circ a = p$ *and such that* q *satisfies the condition* c_Q*,*

$$\begin{tikzcd}
P \arrow[rr, "a"] \arrow[dr, "p"'] & & Q \arrow[dl, dashed, "q", "\models"'] & \mathsf{c}_Q \arrow[l, -{Triangle}] \\
& P' & &
\end{tikzcd} \,. \tag{9}$$

S3 *Given an* $\mathcal{M}$*-morphism* $p : P \to P'$ *and a condition* c_P *over* P*,* p *satisfies* $\neg\mathsf{c}_P$ *if it does not satisfy* c_P*. If* $\{\mathsf{c}_P^{(i)}\}_{i\in I}$ *(for some indexing set* I*) is a family of conditions over* P*,* p *satisfies* $\wedge_{i\in I}\mathsf{c}_P^{(i)}$ *if it satisfies each of the application conditions* c_{P_i}*.*

For an $\mathcal{M}$*-morphism* $p : P \to P'$*, we write*

$$p \models \mathsf{c}_P$$

to denote that p satisfies the condition c_P*. Two application conditions* $\mathsf{c}_P, \mathsf{c}'_P$ *over some object* $P \in \mathsf{obj}(\mathbf{C})$ *are* **equivalent***, denoted* $\mathsf{c}_P \equiv \mathsf{c}'_P$*, if and only if for all* $\mathcal{M}$*-morphisms* $p : P \to H$*, i.e. for arbitrary* $H \in \mathsf{obj}(\mathbf{C})$ *with* P *as an* $\mathcal{M}$*-subobject, we find that*

$$p \models \mathsf{c}_P \Leftrightarrow p \models \mathsf{c}'_P\,. \tag{10}$$

Finally, a (nested) condition or ***constraint*** *on an object* $P \in \mathsf{obj}(\mathbf{C})$ *is defined as a (nested) condition over the* $\mathcal{M}$*-initial object* $\varnothing$*, and the associated notion of satisfiability of constraints as satisfaction of the condition over* $\varnothing$ *by the* $\mathcal{M}$*-initial morphism* $(i_P : \varnothing \hookrightarrow P) \in \mathcal{M}$ *[46]:*

C1 *Every object* $P \in \mathsf{obj}(\mathbf{C})$ *satisfies* true*.*

C2 *An object* $P \in \mathsf{obj}(\mathbf{C})$ *satisfies the condition* $\exists(i_Q : \varnothing \to Q, \mathsf{c}_Q)$ *if there exists an* $\mathcal{M}$*-morphism* $q : Q \to P$ *such that* $q \circ i_Q = i_P$ *and* $q \models \mathsf{c}_Q$*.*

[2] It is here that our restriction to $\mathcal{M}$-morphisms in the formulation of conditions reflects our choice of framework, i.e. that of $\mathcal{M}$-satisfiability (for $\mathcal{M}$-morphisms). We refer the interested readers to [46] for the proof that this is in fact the most general framework available when working with $\mathcal{M}$-morphisms in matches and rewriting rules only, i.e. generalizing morphisms in conditions of arbitrary morphisms in this setting does not lead to more expressivity.

C3 *Given an object $P \in \mathsf{obj}(\mathbf{C})$ and a condition $\mathsf{c}_\varnothing$ over the $\mathcal{M}$-initial object $\varnothing$, P satisfies $\neg\mathsf{c}_\varnothing$ if it does not satisfy $\mathsf{c}_\varnothing$.*

C4 *If $\{\mathsf{c}_\varnothing^{(i)}\}_{i\in I}$ (for some indexing set I) is a family of conditions over $\varnothing$, P satisfies $\wedge_{i\in I}\mathsf{c}_\varnothing^{(i)}$ if it satisfies each of the conditions $\mathsf{c}_\varnothing^{(i)}$.*

It may be instructive to the readers to explicitly parse the potentially somewhat counter-intuitive definition of conditions on objects in a concrete example:

Example 2 *Given an object $P \in \mathsf{obj}(\mathbf{C})$, a condition of the form "P contains an $\mathcal{M}$-subobject Q" is expressed in the present framework as $P \vDash \exists(i_Q) = \exists(i_Q : \varnothing \to Q, \mathsf{true})$, since by virtue of the definition of satisfiability of conditions on objects, $P \vDash \exists(i_Q)$ iff there exists an $\mathcal{M}$-morphism $q : Q \to P$ such that $q \circ i_Q = i_P$:*

$$\begin{tikzcd}
\exists(i_Q) \arrow[r, phantom, "\blacktriangleright"] & \varnothing \arrow[rr, "i_Q"] \arrow[dr, "i_P"', "\models"] & & Q \arrow[dl, dashed, "q", "\models"'] & \mathsf{true} \arrow[l, phantom, "\blacktriangleleft"] \\
& & P & &
\end{tikzcd} \tag{11}$$

Next, consider the following example for illustration of the concept of nested conditions. We will, in practice, be interested exclusively in *finite* nested conditions, i.e. in sequences (or in general directed acyclic graphs) of conditions that ultimately end in an instance of a condition of the form $\exists(x, \mathsf{true})$. In this sense, the example below is sufficiently generic.

Example 3 *The condition below (on undirected multigraphs)*

$$\exists\left(a : \underset{1}{\bullet} \to \underset{1}{\bullet}\ \ \underset{2}{\bullet}, \exists\left(b : \underset{1}{\bullet}\ \ \underset{2}{\bullet} \to \underset{1}{\bullet}\!\!-\!\!\underset{2}{\bullet}, \mathsf{true}\right)\right) \tag{12}$$

parses more explicitly into the diagram

$$\begin{tikzcd}
& & \exists(b, \mathsf{true}) \arrow[d, phantom, "\blacktriangledown"] & & \\
\exists(a, \exists(b, \mathsf{true})) \arrow[r, phantom, "\blacktriangleright"] & \underset{1}{\bullet} \arrow[r, "a"] \arrow[dr, "p"'] & \underset{1}{\bullet}\ \ \underset{2}{\bullet} \arrow[r, "b"] \arrow[d, dashed, "q"] & \underset{1}{\bullet}\!\!-\!\!\underset{2}{\bullet} \arrow[dl, dotted, "r"] & \mathsf{true} \arrow[l, phantom, "\blacktriangleleft"] \\
& & G & &
\end{tikzcd} \tag{13}$$

expressing the condition that a morphism $p : \underset{1}{\bullet} \hookrightarrow G$ satisfies the condition if G contains at least one other vertex $\underset{2}{\bullet}$ (which is the information encoded in the first part of the condition), and such that $\underset{1}{\bullet}$ and $\underset{2}{\bullet}$ are linked by an edge. Moreover, the $\mathcal{M}$-morphism q in the above diagram automatically exists if the entire condition is satisfied. This is in fact a typical example of refinement *(or* $\mathcal{M}$-coverability *[46]), whereby the condition $\exists(a, \mathsf{true})$ is refined by the condition $\exists(a, \exists(b, \mathsf{true}))$.*

3.2 A refined notion of shift construction

One of the key concepts in the theory of rewriting with conditions is the notion of *shift construction*, which is, in essence, a category-theoretical characterization of the interplay between conditions and extensions of their domains. We introduce here an optimized version of the classical shift construction presented in [42]) (which is itself a variant of an earlier construction as reviewed in [46]). Our optimization hinges on the assumed properties of the underlying $\mathcal{M}$-adhesive categories according to Assumption 1. We believe this optimization will be of key importance in future developments of algorithms and software implementations of our framework. The following theorem is at the basis of our novel construction:

Theorem 3 *Given an $\mathcal{M}$-adhesive category $\mathbf{C}$ satisfying Assumption 1, consider a commutative diagram of the form below,*

$$\begin{tikzcd}
 & & & X \arrow[ll, bend right, "b_1"'] \arrow[dl, bend left, "b_2"] \\
 & B_1 \arrow[d, "d_1"'] \arrow[dl, bend right, "e_1"'] & A \arrow[l, hook', "a_1"'] \arrow[d, hook, "a_2"] \arrow[ur, dotted, hook, "x"] \arrow[dl, phantom, "(1)"] & \\
 & D \arrow[dl, dashed, "e"'] & B_2 \arrow[l, "d_2"] \arrow[dll, bend left, "e_2"] & \\
E & & &
\end{tikzcd} \tag{14}$$

where the square marked (1) *is a pushout, where* $a_1, a_2 \in \mathcal{M}$ *(and thus by stability of* $\mathcal{M}$*-morphisms under pushout also* $d_1, d_2 \in \mathcal{M}$*),* $e_1, e_2 \in \mathcal{M}$ *and* $X = \mathsf{PB}(B_1 \xrightarrow{e_1} E \xleftarrow{e_2} B_2)$ *(and thus by stability of* $\mathcal{M}$*-morphisms under pullbacks also* $b_1, b_2 \in \mathcal{M}$*, plus due to the decomposition property of* $\mathcal{M}$ *morphisms, also* $x \in \mathcal{M}$*). Then the following holds: the morphism* e *is an epimorphism if and only if the exterior square is a pushout.*

Proof: "⇒" direction: Suppose that the exterior square in (14) is a pushout. Let $e = m \circ f$ be the epi-$\mathcal{M}$-factorization of e (with $(f : D \to F) \in \mathsf{epi}(\mathbf{C})$ and $(m : F \to E) \in \mathcal{M}$). Then construct the commutative diagram presented in the left part of (15) as follows: form the two pullbacks $B_i' = \mathsf{PB}(F \to E \leftarrow B_i)$, which by the universal property of pullbacks also induces morphisms $B_i \to B_i'$ (for $i = 1, 2$); according to Lemma 13 of Appendix C, we conclude that $B_1' \cong B_1$ and $B_2' \cong B_2$. Next, let $X' = \mathsf{PB}(B_1' \to F \leftarrow B_2')$; by the universal property of pullbacks, this implies the existence of morphisms $A \to X'$ and $X' \to X$ (and by decomposition of $\mathcal{M}$-morphisms, the latter is in $\mathcal{M}$). Since the square just constructed is a pullback, the bottom square a pushout along $\mathcal{M}$-morphisms (and thus a pullback), and since, due to $B_i' \cong B_i$ (for $i = 1, 2$) and $m \in \mathcal{M}$, Lemma 12(i)c entails that the bottom left and bottom front vertical squares are pullbacks, we conclude via invoking pullback-pullback decomposition (Lemma 12(iii)a) that also the bottom back and bottom right vertical squares are pullbacks. Thus by virtue of stability of isomorphisms under pullbacks (Lemma 12(i)d), $X' \cong X$. Moreover, the $\mathcal{M}$-van Kampen property entails that the middle horizontal square is a pushout, whence by uniqueness of pushouts up to isomorphism, we have that $F \cong E$, which proves the claim that $e = m \circ f \in \mathsf{epi}(\mathbf{C})$.

$$\begin{tikzcd}
 & & B_1 \arrow[dl, hook'] \arrow[dd, dashed] & & A \arrow[ll, hook'] \arrow[dl, hook'] \arrow[dd, dotted] \\
 & D \arrow[dd, two heads, "f"] \arrow[dddd, bend right=40, "e"'] & & B_2 \arrow[ll, hook'] \arrow[dd, dashed] & \\
 & & B_1' \arrow[dl, hook'] \arrow[dd, hook] & & X' \arrow[ll, hook'] \arrow[dl, hook'] \arrow[dd, dotted] \\
 & F \arrow[dd, hook, "m"] & & B_2' \arrow[ll, hook'] \arrow[dd, hook] & \\
 & & B_1 \arrow[dl, hook'] & & X \arrow[ll, hook'] \arrow[dl, hook'] \\
 & E & & B_2 \arrow[ll, hook'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & & B_1 \arrow[dl, hook'] \arrow[dd, equal] & & A \arrow[ll, hook'] \arrow[dl, hook'] \arrow[dd, hook] \\
 & D \arrow[dd, dashed, "f'"] \arrow[dddd, bend right=40, "e"'] & & B_2 \arrow[ll, hook'] \arrow[dd, equal] & \\
 & & B_1 \arrow[dl, hook'] \arrow[dd, equal] & & X \arrow[ll, hook'] \arrow[dl, hook'] \arrow[dd, equal] \\
 & F' \arrow[dd, dashed, "m'"] & & B_2 \arrow[ll, hook'] \arrow[dd, equal] & \\
 & & B_1 \arrow[dl, hook'] & & X \arrow[ll, hook'] \arrow[dl, hook'] \\
 & E & & B_2 \arrow[ll, hook'] &
\end{tikzcd} \tag{15}$$

"$\Leftarrow$" direction: Suppose that $e \in \mathsf{epi}(\mathbf{C})$ and that the exterior square in (14) is a pullback. Construct the commutative diagram depicted in the right part of (15) as follows: start by forming the pushout $F' = \mathsf{PO}(B_1 \leftarrow X \rightarrow B_2)$, which by the universal property of pushouts (recalling that the top square is by assumption also a pushout) furnishes morphisms $f' : D \rightarrow F'$ and $m' : F' \rightarrow E$ such that $e = m' \circ f'$. By stability of $\mathcal{M}$-morphisms under pushouts, the morphisms $B_1 \rightarrow F'$ and $B_2 \rightarrow F'$ are in $\mathcal{M}$. The top commutative cube thus precisely satisfies the properties necessary to invoke the theorem in the "$\Rightarrow$" direction, in order to conclude that $f' \in \mathsf{epi}(\mathbf{C})$. Since by assumption $e \in \mathsf{epi}(\mathbf{C})$, and since $e = m' \circ f'$, invoking decomposition of epimorphisms yields that $m' \in \mathsf{epi}(\mathbf{C})$. On the other hand, since by assumption the bottom square is a pullback, invoking the property of $\mathcal{M}$-effective unions permits us to conclude that $m' \in \mathcal{M}$. As the underlying category is assumed to be balanced, and since $\mathcal{M} \subseteq \mathsf{mono}(\mathbf{C})$, we conclude that $m' \in \mathsf{iso}(\mathbf{C})$, i.e. $F' \cong E$, which proves that the bottom square is a pushout (by uniqueness of pushouts, and since pushouts along $\mathcal{M}$-morphisms are also pullbacks). □

An interesting consequence of the above theorem is the following result, which allows one to compare our technical framework more directly with the traditional literature:

Corollary 1 *In an $\mathcal{M}$-adhesive category $\mathbf{C}$ satisfying Assumption 1, given two $\mathcal{M}$-morphisms $(c_1 : C_1 \hookrightarrow D), (c_2 : C_2 \hookrightarrow D) \in \mathcal{M}$, let the $\mathcal{M}$-morphisms*

$$(c_1' : C_1 \hookrightarrow \bar{D})\,,\ (c_2' : C_2 \hookrightarrow \bar{D})\,,\ (d : \bar{D} \hookrightarrow D) \in \mathcal{M}$$

be defined (uniquely up to isomorphisms) as follows: denoting by $(C_1 \hookleftarrow X \hookrightarrow C_2)$ the pullback of the cospan (c_1, c_2), define (c_1', c_2') as the pushout of this span, and let d be defined as the induced morphism from $\bar{D}$ into D. Then the cospan (c_1', c_2') is **jointly epimorphic***.*

Proof: Given that $c_1, c_2, d \in \mathcal{M}$ (which follows from stability of $\mathcal{M}$-morphisms under pullback and pushout, and from the property of $\mathcal{M}$-effective unions, respectively), construct the following commutative diagram:

$$\begin{tikzcd}
 & X \arrow[dl, hook', bend right] \arrow[dr, hook, bend left] & \\
 & \varnothing \arrow[u, hook] \arrow[dl, hook', bend right=15] \arrow[dr, hook, bend left=15] & \\
C_1 \arrow[r, hook, "in_{C_1}"] \arrow[dr, hook, bend right=15, "c_1'"'] \arrow[ddr, hook, bend right=40, "c_1"'] & C_1 + C_2 \arrow[d, dashed, "e"] & C_2 \arrow[l, hook', "in_{C_2}"'] \arrow[dl, hook', bend left=15, "c_2'"] \arrow[ddl, hook', bend left=40, "c_2"] \\
 & \bar{D} \arrow[d, hook, "d"] & \\
 & D &
\end{tikzcd} \tag{16}$$

Here, given the unique $\mathcal{M}$-morphisms $(\varnothing \hookrightarrow C_1)$, $(\varnothing \hookrightarrow C_2)$ and $(\varnothing \hookrightarrow X)$ from the $\mathcal{M}$-initial object $\varnothing$, and with $(C_1 \hookrightarrow C_1 + C_2 \hookleftarrow C_2)$ the pushout of $(C_1 \hookleftarrow \varnothing \hookrightarrow C_2)$, the existence of the morphism $(e : C_1 + C_2 \rightarrow \bar{D})$ follows by the universal property of pushouts. Then it follows via invoking Theorem 3 that e is an epimorphism. □

We will take advantage of this corollary when discussing the results of Theorem 4 below as well as the notion of DPO-type rule compositions in its various forms in Section 4.1. Theorem 4 below constitutes another interesting consequence of Theorem 3, deriving an algorithmic refinement of the so-called Shift construction, which allows one to extend application conditions to larger contexts:

Theorem 4 (Shift construction; compare [46], Thm. 5 and Lem. 3, [42] Lem. 3.11) *Given an $\mathcal{M}$-adhesive category $\mathbf{C}$ satisfying Assumption 1, there exists a* shift construction*, denoted* Shift*, such that for every condition c_P over an object $P \in \mathsf{obj}(P)$ and for every $\mathcal{M}$-morphism $p : P \rightarrow Q$, an $\mathcal{M}$-morphism $n \circ p : P \rightarrow H$ (with $n \in \mathcal{M}$) satisfies the condition c_P iff $(n : Q \rightarrow H)$ satisfies $\mathsf{Shift}(p, \mathsf{c}_P)$, referred to as the* shift of c_P along p*:*

$$n \circ p \vDash \mathsf{c}_P \;\Leftrightarrow\; n \vDash \mathsf{Shift}(p, \mathsf{c}_P)\,, \tag{17a}$$

with

$$\begin{tikzcd}
\mathsf{c}_P \arrow[r, -triangle 90] & P \arrow[rr, "p"] \arrow[dr, "n\circ p"', "\vDash"] & & Q \arrow[dl, "n", "\vDash"'] & \mathsf{Shift}(p,\mathsf{c}_P) \arrow[l, -triangle 90] \\
& & H & &
\end{tikzcd} \; . \tag{17b}$$

Here, the application condition $\mathsf{Shift}(p,\mathsf{c}_P)$ *is constructed inductively as follows:*

(i) Case $\mathsf{c}_P = \mathsf{true}$:

$$\mathsf{Shift}(p,\mathsf{true}) := \mathsf{true}\,. \tag{18}$$

(ii) Case $\mathsf{c}_P = \exists(a,\mathsf{c}_A)$ (for some $(a : P \to A) \in \mathcal{M}$ and c_A an application condition over $A \in \mathsf{obj}(\mathbf{C})$): *construct the commutative diagram below, where the square marked* PO *is a pushout*[3]*:*

$$\begin{tikzcd}
& & & & X \arrow[dll, bend right, "p''"'] \arrow[dl, bend left, "a''"] \\
& & Q \arrow[d, "a'"'] \arrow[dll, bend right, "r"'] \arrow[dr, phantom, "\mathsf{PO}"] & P \arrow[l, "p"'] \arrow[d, "a"] \arrow[ur, dotted, "x"'] & \\
& & S \arrow[dl, dotted, "e"] & A \arrow[l, "p'"] \arrow[dll, bend left, "s"] & \\
\mathsf{Shift}(s,\mathsf{c}_A) \arrow[r, -triangle 90] & E & & & \mathsf{c}_A \arrow[ul, -triangle 90]
\end{tikzcd} \tag{19}$$

Here, each $\mathcal{M}$*-morphism* $x : P \to X$ *such that there exist* $\mathcal{M}$*-morphisms* $p'' : X \to Q$ *and* $a'' : X \to A$*, and with* $p'' \circ x = p$ *and* $a'' \circ x = a$*, induces an object* E *and* $\mathcal{M}$*-morphisms* $r : Q \to E$ *and* $s : A \to E$ *via taking the pushout of the span* $(Q \xleftarrow{p''} X \xrightarrow{a''} A)$*. Since objects in* $\mathbf{C}$ *are assumed to be finite, which entails in particular that there are only finitely many* $\mathcal{M}$*-subobjects of* Q *and* A*, up to isomorphisms of spans (induced by isomorphisms of* X*) there are only finitely many isomorphism classes of spans* $(Q \xleftarrow{p''} X \xrightarrow{a''} A)$*. Denoting the set of all isomorphism classes of* $\mathcal{M}$*-morphism pairs* (r,s) *thus obtained by* $\mathcal{E}$*, we define*

$$\mathsf{Shift}(p,\exists(a,\mathsf{c}_A)) := \bigvee_{(r,s)\in\mathcal{E}} \exists(r,\mathsf{Shift}(s,\mathsf{c}_A))\,. \tag{20}$$

(iii) Case $\mathsf{c}_P = \neg\mathsf{c}'_P$:

$$\mathsf{Shift}(b,\neg\mathsf{c}'_P) := \neg\mathsf{Shift}(b,\mathsf{c}'_P)\,. \tag{21}$$

(iv) Case $\wedge_{i\in I}\mathsf{c}_i$:

$$\mathsf{Shift}(b,\wedge_{i\in I}\mathsf{c}_i) := \wedge_{i\in I}\mathsf{Shift}(b,\mathsf{c}_i) \tag{22}$$

Proof: We will closely follow the proof strategy of [42, Lem. 3.11], adapted to our variant of the Shift construction (and proving en passent the equivalence of our construction to the ones of [46, Thm. 5 and Lem. 3] and [42, Lem. 3.11], see Remark 1 below). The statement is trivially true for the case $\mathsf{c}_P = \mathsf{true}$. For the case $\mathsf{c}_P = \exists(a : P \hookrightarrow A, \mathsf{c}_A)$ of a nested application condition, we will prove the claim by induction over the levels of nesting.

[3] Besides a more concrete characterization of Shift, we also profit from restricting all morphisms involved in this construction (except for the epimorphisms e) to be $\mathcal{M}$-morphisms, which in particular guarantees due to the assumed $\mathcal{M}$-adhesivity of the underlying category $\mathbf{C}$ that the pushout to form the object S always exists.

$$\begin{tikzcd}
& & & & X \arrow[dll, bend right, "p''"'] \arrow[ddl, bend left, "a''"] \\
& & Q \arrow[d, "a'"'] \arrow[ddl, bend right, "r"'] \arrow[dddll, bend right=40, "n"'] & P \arrow[l, "p"'] \arrow[d, "a"] \arrow[ur, dotted, "x"'] \arrow[dl, phantom, "\mathsf{PO}"] & \\
& & S \arrow[dl, dotted, "e"] & A \arrow[l, "p'"] \arrow[dll, bend left, "s"] \arrow[ddlll, bend left=40, "m"] & \\
& E \arrow[dl, dashed, "h"'] & & & \mathsf{c}_A \arrow[uul, -] \\
H & \mathsf{Shift}(s, \mathsf{c}_A) \arrow[u, -] & & &
\end{tikzcd} \tag{23}$$

"⇒" direction: Suppose that $n \circ p \vDash \exists(a : P \hookrightarrow A, \mathsf{c}_A)$, and take (20) as the *induction hypothesis* in case that c_A is itself a nested condition. By definition of satisfiability, this entails that there exists $(m : A \hookrightarrow H) \in \mathcal{M}$ such that $m \circ a = n \circ p$. Construct the span (p'', a'') by taking the pullback of the cospan (n, m), which by universal property of pullbacks furnishes a morphism $x : P \to X$. By stability of $\mathcal{M}$-morphisms under pullbacks, we find that $p'', a'' \in \mathcal{M}$, and thus by the decomposition property of $\mathcal{M}$-morphisms also that $x \in \mathcal{M}$. Next, construct the cospan (r, s) via taking the pushout of the span (p'', a''), which by stability of $\mathcal{M}$-morphisms under pushout entails that $r, s \in \mathcal{M}$. By the universal property of pushouts, there exist morphisms $e : S \to E$ and $h : E \to H$ (see (23)), with $e \in \mathsf{epi}(\mathbf{C})$ (via Theorem 3) and $h \in \mathcal{M}$ (via $\mathcal{M}$-effective unions). By definition of satisfiability, $m \vDash \mathsf{c}_A$ and $m = h \circ s$ together with the induction hypothesis for c_A entail that $h \vDash \mathsf{Shift}(s, \mathsf{c}_A)$. Again invoking the definition of satisfiability, since $h \circ r = n$ and $h \vDash \mathsf{Shift}(s, \mathsf{c}_A)$, we have thus verified that indeed $n \vDash \exists(r, \mathsf{Shift}(s, \mathsf{c}_A))$.

"⇐" direction: Let us assume that $n \vDash \mathsf{Shift}(p, \exists(a, \mathsf{c}_A))$, with the shifted condition constructed according to (20). We have to verify that this entails $n \circ p \vDash \exists(a, \mathsf{c}_A)$. Combining the assumption with the definition of satisfiability of conditions, there must exist $r, s, h \in \mathcal{M}$ such that $h \circ r = n$ and $h \vDash \mathsf{Shift}(s, \mathsf{c}_A)$. By the induction assumption, $h \vDash \mathsf{Shift}(s, \mathsf{c}_A)$ entails that the morphism $m \in \mathcal{M}$ given by $m := h \circ s$ (see (23)) satisfies $m \vDash \mathsf{c}_A$. Since by construction $m \circ a = n \circ p$, by definition of satisfiability we have thus demonstrated that indeed $n \circ p \vDash \exists(a, \mathsf{c}_A)$.

Finally, the proof of statements (iii) and (iv) is obtained in an analogous fashion. □

Remark 1 *Since an algorithmically tractable* Shift *construction is quintessential for developing a calculus of compositional rewriting, let us briefly discuss the precise relationship of our refined construction to the pre-existing constructions in the literature, highlighting in particular the nature of the refinement:*

- *In the variant of the* Shift *construction and corresponding proof strategy presented in [42, Lem. 3.11], the authors base their construction upon assuming* a priori *that a set of cospans*

  $$\mathcal{F} = \{(a', p') \in \mathcal{E}' \mid p' \in \mathcal{M} \ \wedge \ \square(P, Q, E, A) \ \textit{commutes}\}$$

  is provided, where no additional properties for the commutative squares $\square(P, Q, E, A)$ are required. As for the class of cospans $\mathcal{E}'$, the origin of this class in [42] is an $\mathcal{E}'$-$\mathcal{M}$-pair factorization *property that is assumed to hold for the category* $\mathbf{C}$*, whereby any cospan (f_1, f_2) of arbitrary morphisms $(f_i : C_i \to D) \in \mathsf{mor}(\mathbf{C})$ $(i = 1, 2)$ factors uniquely up to isomorphisms as*

  $$f_1 = m \circ e_1\,,\ f_2 = m \circ e_2\,,\ m \in \mathcal{M}\,,\ (e_1, e_2) \in \mathcal{E}'\,.$$

  On the one hand, this construction permits to consider the seemingly more general case of conditions formulated in terms of non-$\mathcal{M}$-morphisms, which however according to highly *technical results presented in [46] in fact does* not *increase the expressivity of the calculus of conditions. On the other hand, the construction of Ehrig et al. is strictly more general than ours in that it permits to consider* non-monic matches *for applications of linear rules. Yet for an $\mathcal{M}$-adhesive category* $\mathbf{C}$ *satisfying Assumption 1 and for the case of $\mathcal{M}$-morphisms as admissible matches, Corollary 1 entails that our construction is mathematically equivalent to the one of [42, Lem. 3.11], since in this case the relevant $\mathcal{E}'$-$\mathcal{M}$-pair-factorizations of cospans of $\mathcal{M}$-morphisms are precisely of the form presented in the corollary. From an algorithmic standpoint, the class $\mathcal{E}'$ is typically non-trivial to construct, whereas our construction based upon the results of Theorem 3 only requires algorithmically rather concrete notions of "forming all larger overlaps" ($Q \hookleftarrow X \hookrightarrow A$) starting from a given "overlap" ($Q \hookleftarrow P \hookrightarrow A$). In summary, for categories satisfying Assumption 1, our construction thus constitutes a certain algorithmic refinement of the "traditional" Ehrig et al. construction.*

- *The construction of [46, Thm. 5 and Lem. 3] was developed for the setting of $\mathcal{M}$-morphisms as admissible matches as in our case. This variant of a* Shift *construction is based upon a shape of commutative diagram as in (23) without the subdiagrams involving the object X and morphisms incident to it. Instead, the authors devise an algorithm whereby one iterates over the triples of morphisms (r, s, e) such that $r, s \in \mathcal{M}$, $e \in \mathsf{epi}(\mathbf{C})$ and such that the diagram commutes. Inspecting the proof of Theorem 4, our refined construction precisely mirrors this algorithmic idea, yet it provides by virtue of Theorem 3 also a concrete construction principle for such triples of morphisms (which was not available in [46]).*

- *Finally, it is worthwhile noting that the complexity of the* Shift *construction may be significantly reduced*[4] *in an alternative setting within which conditions as well as satisfiability are defined not in terms of $\mathcal{M}$-morphisms, but in terms of arbitrary morphisms [57, Lem .1]. However, this generalized setting is unfortunately not the setting relevant to the type of compositional rewriting theories considered in the present paper (i.e. those that satisfy both concurrency and associativity properties), which is why we indeed must necessarily rely upon the $\mathcal{M}$-morphism variant of the* Shift *construction as presented in Theorem 4.*

As a first application of our refined Shift construction, let us consider a special situation for shifts that plays a role later on in the theory of compositions of rewriting rules:

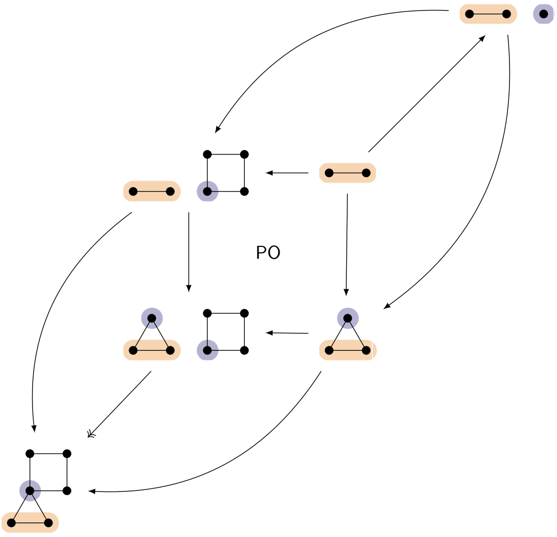
Lemma 3 **(**Shift **along coproduct injections)** *Let* $\mathbf{C}$ *be an $\mathcal{M}$-adhesive category satisfying Assumption 1. Let $P, Q \in \mathsf{obj}(\mathbf{C})$ be objects, and let $\exists(a : P \to A, \mathsf{c}_A)$ be a condition over P. Then* $\mathsf{Shift}(P \to P + Q, \exists(a : P \to A, \mathsf{c}_A))$ *is computed via the following type of diagram:*

$$\begin{tikzcd}
 & & & P+W \arrow[dll, bend right, "{[id_P,w_1]}"'] \arrow[ddl, bend left, "{[a,w_2]}"] \\
 & P+Q \arrow[d] \arrow[dd, bend right, "r"'] \arrow[r, phantom, "\mathsf{PO}" near end, shift right=4ex] & P \arrow[l, "in_P"'] \arrow[d, "a"] \arrow[ur, dotted, "in_P"] & \\
 & A+Q \arrow[d, dashed, "e"'] & A \arrow[l, "in_A"] \arrow[dl, bend left, "s"] & \\
\mathsf{Shift}(s,\mathsf{c}_A) \arrow[r, -{Triangle}] & E & & \mathsf{c}_A \arrow[ul, -{Triangle}]
\end{tikzcd} \tag{24}$$

Proof: Consider the following specialization of Lemma 16: starting from the diagram depicted in (14), any $\mathcal{M}$-morphism $f : X \to P + Q$ decomposes into the form $[f_p, f_q] : X_P + X_Q \to P + Q$ (where $X_P + X_Q \cong X$ and $f_p, f_q \in \mathcal{M}$), and analogously the morphism $x : P \to X$ decomposes into the pair of $\mathcal{M}$-morphisms $[g, h] : P' + P'' \to X_P + X_Q$ (with $P' + P'' \cong P$). But since by commutativity of the diagram in (24) $[f_p, f_q] \circ [g, h] = in_P$, it follows that $P' \cong P$ and $P'' \cong \varnothing$, which upon invoking Theorem 4 proves the claim. □

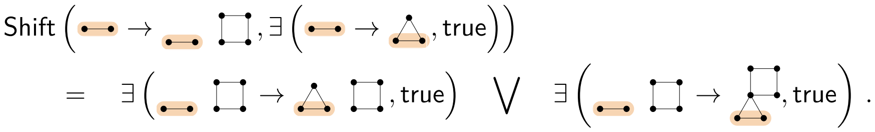
[4] We would like to thank one of the anonymous reviewers for drawing our attention to this result.

Example 4 *As an illustrative application of Lemma 3, consider the following explicit computation in the category* **uGraph** *of undirected multigraphs (which satisfies Assumption 1, with $\varnothing$ the empty graph):*



(25)

The image demonstrates how the shift along the embedding of the two-vertex graph with an edge (highlighted in orange) into a disjoint union with a "square" graph yields a condition over this disjoint union of graphs that tests for a disjoint pattern, but also (via the only other possible non-trivial overlap up to isomorphisms, along an additional disjoint vertex marked in blue) an alternative condition that tests for a non-disjoint pattern:



(26)

Finally, we will require two additional technical lemmas of key importance to our framework of compositional rewriting. Both results rely on the notion of *equivalences* of conditions (see Definition 9 and equation (10)).

Lemma 4 (Units for Shift) *For every object $P \in \mathsf{obj}(\mathbf{C})$ and for every condition c_P over P, we have that*

$$\mathsf{Shift}(id_P : P \to P, \mathsf{c}_P) \equiv \mathsf{c}_P\,. \tag{27}$$

Proof: This follows directly from the definition of Shift according to Theorem 4, by specializing (17) in the form

$$n \circ id_P = n \vDash \mathsf{c}_P \;\Leftrightarrow\; n \vDash \mathsf{Shift}(id_P, \mathsf{c}_P)\,, \tag{28a}$$

with

$$\begin{tikzcd} \mathsf{c}_P \arrow[r, phantom, "\blacktriangleright"] & P \arrow[rr, "id_P"] \arrow[dr, "\vDash"', "n\circ id_P = n"'] & & P \arrow[dl, "n", "\vDash"'] & \mathsf{Shift}(id_P, \mathsf{c}_P) \arrow[l, phantom, "\blacktriangleleft"] \\ & & H & & \end{tikzcd} \quad . \tag{28b}$$

□

Lemma 5 (Compositionality of Shift; compare [45], Fact 3.14) *Let $X \in \mathsf{obj}(\mathbf{C})$ be an object, c_X an application condition over X, and let $f : X \to Y$ and $g : Y \to Z$ be two morphisms of $\mathbf{C}$. Then the following equivalence of conditions holds:*

$$\mathsf{Shift}(g, \mathsf{Shift}(f, \mathsf{c}_X)) \equiv \mathsf{Shift}(g \circ f, \mathsf{c}_X) \tag{29}$$

Proof: The proof follows by a repeated application of Theorem 4. The equivalence holds if for all morphisms $c : Z \to H$, we find that

$$c \vDash \mathsf{Shift}(g, \mathsf{Shift}(f, \mathsf{c}_X)) \;\Leftrightarrow\; c \vDash \mathsf{Shift}(g \circ f, \mathsf{c}_X)\,. \tag{30}$$

Starting from the diagram below,

$$\begin{tikzcd}
 & & & \mathsf{Shift}(f, \mathsf{c}_X) \arrow[d, phantom, "\blacktriangledown"] & & \\
\mathsf{c}_X \arrow[r, phantom, "\blacktriangleright"] & X \arrow[rr, "f"] \arrow[dr, "c\circ g\circ f"', "\vDash"] & & Y \arrow[r, "g"] \arrow[dl, "c\circ g"', "\vDash"] & Z \arrow[dll, bend left, "c", "\vDash"'] & \mathsf{Shift}(g, \mathsf{c}_X) \arrow[l, phantom, "\blacktriangleleft"] \;, \\
 & & H & & &
\end{tikzcd} \tag{31}$$

we may calculate:

$$\begin{aligned}
&& c &\vDash \mathsf{Shift}(g, \mathsf{Shift}(f, \mathsf{c}_X)) \\
&\overset{(17)}{\Leftrightarrow}& c \circ g &\vDash \mathsf{Shift}(f, \mathsf{c}_X) \\
&\overset{(17)}{\Leftrightarrow}& c \circ g \circ f &\vDash \mathsf{c}_X \\
&\overset{(17)}{\Leftrightarrow}& c &\vDash \mathsf{Shift}(g \circ f, \mathsf{c}_X)\,.
\end{aligned} \tag{32}$$

□

4 Compositional associative Double-Pushout rewriting with conditions

In this section, we will develop a variant of *Double Pushout (DPO) rewriting* for *linear rules*, which may be obtained from the well-known “traditional” DPO-rewriting framework of Ehrig et al. [36] with its nearly 50 years of developments [22, 25, 37–39, 41, 42, 47, 53, 63] via requiring the underlying $\mathcal{M}$-adhesive categories to satisfy Assumption 1. As we will demonstrate, this particular specialization of the original theory yields a semantics for rewriting with certain additional properties, which we refer to as *compositionality* and *associativity*. The former property is a statement on the existence of a certain form of *sequential rule composition* (cf. Definition 16), which via a variant of the classical *concurrency theorem* (cf. Theorem 5) opens the possibility to develop algorithms for the static analysis of rule application sequences. The *associativity* phenomenon for sequential compositions of linear rules with conditions (cf. Theorem 6) is an original result of the present paper, adding to the aforementioned tool-set of static analysis the possibility to analyze sequences of more than two rule applications (ultimately leading to the notions of *tracelet analysis* [9] and *rule algebras* [8, 10–13]).

Notational convention 1 *Since many of the constructions and results presented in the following constitute variants of well-known results from the rich DPO-rewriting literature, by* ***convention*** *a mention of “**compare**” or “**cf.**” without additional comments indicates that the results coincide with those in the literature (modulo possibly some notational adjustments and minor modifications). On several occasions, we nevertheless provide full proofs of some of the essential statements, since these will provide the “blueprint” for the corresponding constructions in the SqPO-setting presented in Section 5. Our choices of notations follows mostly those taken in [11].*

4.1 DPO-rewriting in $\mathcal{M}$-adhesive categories

Contrary to the traditional graph rewriting literature, we prefer to interpret spans as encoding a partial injective map going from the *right* leg to the left leg, rather than the other way around. This notational convention as well as the focus on *linear rules* (i.e. rules based upon spans of $\mathcal{M}$-morphisms as opposed to *non-linear* rules based upon generic morphisms) was motivated by our work on the rule algebra and related tracelet constructions [8–13]. Furthermore, all constructions presented in the following are naturally defined only up to isomorphisms, whence the choice to consider isomorphism classes of linear rules.

Definition 10 (Linear rules) *Let* **C** *be an* $\mathcal{M}$*-adhesive category. We denote by* $\mathsf{Lin}(\mathbf{C})$ *the* set of linear rules*, defined as the set of isomorphism classes*

$$\mathsf{Lin}(\mathbf{C}) := \left\{ r = \left(O \xleftarrow{o} K \xrightarrow{i} I \right) \middle| \, O, K, I \in \mathsf{obj}(\mathbf{C}),\ o, i \in \mathcal{M} \right\} \diagup_{\cong} . \tag{33}$$

Here, we define $r = (O \leftarrow K \rightarrow I)$ *and* $r' = (O' \leftarrow K' \rightarrow I')$ *as isomorphic when there exist isomorphisms* $\omega : O \rightarrow O'$*,* $\kappa : K \rightarrow K'$ *and* $\iota : I \rightarrow I'$ *that make the evident diagram commute. Thus a (representative of a) linear rule* $r \in \mathsf{Lin}(\mathbf{C})$ *is a span of* $\mathcal{M}$*-morphisms* $o, i \in \mathcal{M}$ *with* Output object O*,* Kontext object K *and* Input object I*.*

The precise interpretation of the concept of linear rules is provided in the form of the following main definition of DPO rewriting:

Definition 11 (DPO rewriting) *Let* $r := \left(O \xleftarrow{o} K \xrightarrow{i} I \right) \in \mathsf{Lin}(\mathbf{C})$ *be a linear rule, let* $X \in \mathsf{obj}(\mathbf{C})$ *be an object, and let* $m : I \rightarrow X \in \mathcal{M}$ *be an* $\mathcal{M}$*-morphism*[5]*. Then* m *is defined to be an* admissible match *for the application of* r *to* X*, if and only if the diagram below is constructible:*

$$\begin{tikzcd}
O \arrow[d, dotted, "m^*"'] \arrow[dr, phantom, "\mathsf{PO}"] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, dashed] \arrow[dr, phantom, "\mathsf{POC}"] & I \arrow[d, "m"] \\
r_m(X) & K' \arrow[l, dotted] \arrow[r, dashed] & X
\end{tikzcd} \tag{34}$$

Here, the square marked POC *must be constructible as a* pushout complement*, while if this square exists the square marked* PO *is always constructible as a* pushout *(cf. Assumption 1), whence the moniker* Double-Pushout (DPO) rewriting *is derived. In this case, we refer to* $r_m(X) \in \mathsf{obj}(\mathbf{C})$ *as the* rewrite of X via the rule r along the (admissible) match m*. We introduce the notation* $\mathsf{M}_r(X)$ *for the* set of admissible matches *for the application of the rule* r *to the object* X*:*

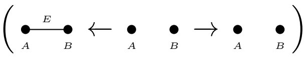
$$\mathsf{M}_r(X) := \{(m : I \rightarrow X) \in \mathcal{M} \mid POC\ (1)\ in\ (34)\ exists\} . \tag{35}$$

For compatibility with the standard DPO rewriting literature, we will sometimes use the notation

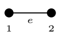
$$r_m(X) \underset{r,m}{\Longleftarrow} X$$

in order to explicitly reference the information contained in (34). Moreover, the morphism m^* *is referred to as the* comatch of m *(under the application of linear rule* r *to the object* X*).*

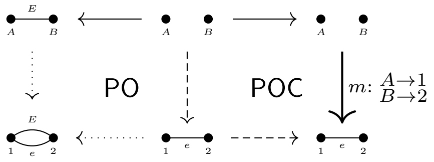
In order to provide a quick intuitive illustration of the DPO rewriting concept, consider the edge creation rule described in the introduction:

$e_+ :=$  (36)

Here, as customary in the graph rewriting literature, the structure of the linear rule (in this case a span of undirected multigraphs) is indicated via the labeling indices, i.e. in the present case reflecting that the vertices marked A and B are preserved in applications of this rule, and that the edge marked E is created. An example for an admissible match and the respective rule application to the example graph

$X :=$ 

along the example match m (which sends the vertices A and B of I to the vertices 1 and 2 of X, respectively) is depicted below:



[5] For our construction of a DPO-type rule algebra framework, we will only be interested in admissible matches that are in the class $\mathcal{M}$, even though DPO rewriting in its most general form as e.g. discussed in [42] would permit also non-monic matches.

4.2 From conditions to application conditions for rewriting rules

Conditions formulated for objects and for morphisms interact in a straightforward manner with the concept of rewriting rules, which requires two key constructions: the *shift construction*, as introduced in Theorem 4, and the so-called *transport construction*. We will follow the standard literature on $\mathcal{M}$-adhesive categories (cf. e.g. [42]) in defining the latter construction[6].

Definition 12 (Transport of conditions over rules; cf. [42], Construction 3.15) *Let*

$$r := \left(O \xleftarrow{o} K \xrightarrow{i} I\right) \in \mathsf{Lin}(\mathbf{C})$$

be a linear rule, and let c_O *be an application condition over* O*. Then we define a* transport construction Trans *such that* $\mathsf{Trans}(r, \mathsf{c}_O)$ *is an application condition over* I*, and which is constructed inductively as follows:*

(i) Case $\mathsf{c}_O = \mathsf{true}$*:*

$$\mathsf{Trans}(r, \mathsf{true}) := \mathsf{true}\,. \tag{37}$$

(ii) Case $\mathsf{c}_O = \exists(a, \mathsf{c}'_O)$ *with* $a : O \to O'$*: if the diagram below*

$$\begin{tikzcd}
 & O \arrow[d, "a"'] \arrow[dr, phantom, "\mathsf{POC}"] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d] \arrow[dr, phantom, "\mathsf{PO}"] & I \arrow[d, "a^{*}"] & \\
\mathsf{c}'_O \arrow[r, phantom, "\blacktriangleright"] & O' & K' \arrow[l, "o'"] \arrow[r, "i'"'] & I' & \mathsf{Trans}(r', \mathsf{c}'_O) \arrow[l, phantom, "\blacktriangleleft"]
\end{tikzcd} \tag{38}$$

is constructible, i.e. if the pushout complement marked POC *exists, we define*

$$\mathsf{Trans}(r, \exists(a, \mathsf{c}'_O)) := \exists(a^{*}, \mathsf{Trans}(r', \mathsf{c}'_O))\,, \tag{39}$$

with $r' := \left(O' \xleftarrow{o'} K' \xrightarrow{i'} I'\right)$*. Otherwise, we define*

$$\mathsf{Trans}(r, \exists(a, \mathsf{c}'_O)) := \mathsf{false}\,. \tag{40}$$

(iii) Case $\mathsf{c}_O = \neg\mathsf{c}'_O$*:*

$$\mathsf{Trans}(\neg\mathsf{c}'_O) := \neg\mathsf{Trans}(\mathsf{c}'_O)\,. \tag{41}$$

(iv) Case $\mathsf{c}_O = \wedge_{j\in J}\, \mathsf{c}_O^{(j)}$*:*

$$\mathsf{Trans}(r, \wedge_{j\in J}\, \mathsf{c}_O^{(j)}) := \wedge_{j\in J}\mathsf{Trans}(r, \mathsf{c}_O^{(j)})\,. \tag{42}$$

It is straightforward to verify that the transport construction is invariant under the various possible isomorphisms involved in the relevant constructions of pushouts and pushout complements, for precisely the same reasons as those ensuring the invariance of the shift construction under isomorphisms as detailed in the proof of Theorem 4.

Lemma 6 (Property of transport along DPO-type rules; cf. [42], Lemma 3.14) *In an* $\mathcal{M}$*-adhesive category* $\mathbf{C}$ *satisfying Assumption 1, let* $r = (O \leftarrow K \rightarrow I) \in \mathsf{Lin}(\mathbf{C})$ *be a linear rule, and let* c_O *be an application condition over* O*. Then for any DPO-admissible match* $(m : I \to X) \in \mathsf{M}_r(X)$ *of the rule* r *into an object* $X \in \mathsf{obj}(\mathbf{C})$*, and if* m^{*} *denotes the comatch of* m*, one finds that*

$$m^{*} \vDash \mathsf{c}_O \Leftrightarrow m \vDash \mathsf{Trans}(r, \mathsf{c}_O)\,, \tag{43a}$$

with

$$\begin{tikzcd}
\mathsf{c}_O \arrow[r, phantom, "\blacktriangleright"] & O \arrow[d, "\models" near start, "m^{*}"'] \arrow[dr, phantom, "\mathsf{PO}"] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d] \arrow[dr, phantom, "\mathsf{POC}"] & I \arrow[d, "m"] & \mathsf{Trans}(r, \mathsf{c}_O) \arrow[l, phantom, "\blacktriangleleft"] \\
 & r_m(X) & K' \arrow[l] \arrow[r] & X &
\end{tikzcd}\,. \tag{43b}$$

[6] In this definition, there is, *a priori*, a choice to be made about the "direction" of the transport; our chosen convention agrees with the one in the literature and will prove convenient in our later applications to notions of compositionality of rules with application conditions.

The transport construction allows us to choose, without loss of generality, a "standard position" for the application conditions in a linear rule, where we fix the following conventions:

Definition 13 (Standard form for DPO-type rules with conditions and admissible matches) *Let* $\overline{\mathsf{Lin}}(\mathbf{C})$ *denote the* set of linear rules with application conditions in standard form*, whose elements* $R \in \overline{\mathsf{Lin}}(\mathbf{C})$ *are defined to be of the form*

$$R = (r, \mathsf{c}_I)\,, \quad r = \left(O \xleftarrow{o} K \xrightarrow{i} I\right) \in \mathsf{Lin}(\mathbf{C})\,. \tag{44}$$

Consequently, we introduce the notion of admissible matches *for applications of rules with application conditions to objects as follows; let* $X \in \mathsf{obj}(\mathbf{C})$ *be an object,* $R \in \overline{\mathsf{Lin}}(\mathbf{C})$ *as above a rule with application conditions, and* $m : I \to X$ *an element of* $\mathcal{M}$*. Then we refer to* m *as an* admissible match *if and only if* m *satisfies the application condition,*

$$m \vDash \mathsf{c}_I\,,$$

and if the diagram below is constructible:

$$\begin{tikzcd}
O \arrow[d, "m^*"'] \arrow[dr, phantom, "\mathsf{PO}"] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d] \arrow[dr, phantom, "\mathsf{POC}"] & I \arrow[d, "m"] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \arrow[dl, phantom, "\vDash"] \\
R_m(X) & K' \arrow[l] \arrow[r] & X &
\end{tikzcd}\,. \tag{45}$$

Equivalently, admissibility of m *thus amounts to admissibility with respect to the linear rule without application conditions (i.e. in the sense of (35)) combined with satisfaction of the application condition, resulting in the following compact formula for the* set of admissible matches $\mathsf{M}_R(X)$*:*

$$\mathsf{M}_R(X) := \{m \in M_r(X) \mid m \vDash \mathsf{c}_I\}\,. \tag{46}$$

For later convenience, we will employ the shorthand notation $\dot{\equiv}$ to signify *"equivalence under the constraint of admissibility"* [7]:

Definition 14 *Let* $r = \left(O \xleftarrow{o} K \xrightarrow{i} I\right) \in \mathsf{Lin}(\mathbf{C})$ *be a linear rule and* $\mathsf{c}_I, \tilde{\mathsf{c}}_I$ *conditions over* I*. Then we define*

$$(\mathsf{c}_I \dot{\equiv} \tilde{\mathsf{c}}_I) :\Leftrightarrow (\forall X \in \mathsf{obj}(\mathbf{C}) : \forall m \in \mathsf{M}_r(X) : \quad m \vDash \mathsf{c}_I \Leftrightarrow m \vDash \tilde{\mathsf{c}}_I)\,. \tag{47}$$

As a further refinement, the Trans construction enables a certain form of compression of application conditions for linear rules.

Definition 15 (Compressed standard form for conditions) *Let* $R := (r = (O \leftarrow K \to I), \mathsf{c}_I) \in \overline{\mathsf{Lin}}(\mathbf{C})$ *be a linear rule with application conditions. Then we define the* compressed standard form *for* c_I *as*

$$\dot{\mathsf{c}}_I := \mathsf{Trans}(r, \mathsf{Trans}(\bar{r}, \mathsf{c}_I))\,, \tag{48}$$

where $\bar{r} := (I \leftarrow K \to O)$*.*

The intuition behind the above definition of *"equivalence up to non-transportable subconditions"* is that while it is perfectly possible to define arbitrary conditions of the form $\exists(a : I \to A, \mathsf{c}_A)$ over the input I of a linear rule, only those conditions will contribute in applications of the linear rule via matches that are transportable via Trans, since by definition of Trans the operation $\mathsf{Trans}(r, \mathsf{Trans}(\bar{r}, \exists(a : I \to A, \mathsf{c}_A))$ in effect tests whether or not the relevant pushout complement exists such that an admissible match of the rule could satisfy $\exists(a : I \to A, \mathsf{c}_A)$. This also implies that

$$\mathsf{c}_I \dot{\equiv} \dot{\mathsf{c}}_I\,, \tag{49}$$

[7] Following the standard notational convention in the rewriting literature, we choose to not make the linear rule with respect to which admissibility is required explicit in our notation $\dot{\equiv}$, since we will only utilize this form of equivalence in situations where the nature of this rule will be clear from the given context.

thus motivating the notation $\dot{\mathsf{c}}_I$. A further illustration of this phenomenon is provided in Example 5.

We conclude this subsection by stating a number of technical lemmas that are necessary in order to derive our novel *associative compositional DPO rewriting* framework as presented in the following subsection, which concern certain important properties of the Trans construction and of the compatibility of the Shift and Trans constructions:

Lemma 7 (Units for Trans) *Let $X \in \mathsf{obj}(\mathbf{C})$ be an arbitrary object and c_X a condition over X. Then with $r_{id_X} := \left(X \xleftarrow{id_X} X \xrightarrow{id_X} X\right) \in \mathsf{Lin}(\mathbf{C})$ the "identity rule on X", we find that*

$$\mathsf{Trans}(r_{id_X}, \mathsf{c}_X) \equiv \mathsf{c}_X\,. \tag{50}$$

Proof: The proof follows directly from the property of the Trans construction stated in Lemma 6, whereby one finds for arbitrary admissible matches $(m : X \to Y) \in \mathsf{M}_{r_{id_X}}(X)$ that

$$m^* = m \vDash \mathsf{c}_X \;\Leftrightarrow\; m \vDash \mathsf{Trans}(r_{id_X}, \mathsf{c}_X)\,, \tag{51a}$$

with

$$\begin{tikzcd}
\mathsf{c}_X \blacktriangleright X \arrow[r, equal, "id_X"] \arrow[d, "\overset{\vDash}{m^*=m}"'] \arrow[dr, phantom, "\mathsf{PO}"] & X \arrow[r, equal, "id_X"] \arrow[d] \arrow[dr, phantom, "\mathsf{POC}"] & X \blacktriangleleft \mathsf{Trans}(r_{id_X}, \mathsf{c}_X) \arrow[d, "\overset{\vDash}{m}"] \\
r_m(X) = Y \arrow[r, equal, "id_Y"'] & Y \arrow[r, equal, "id_Y"'] & Y
\end{tikzcd}\;. \tag{51b}$$

Here, the pushout complement in the squares marked POC always exists by virtue of Lemma 12(i)a, and $m^* = m$ as well as $r_m(X) = Y$ follow due to stability of isomorphisms under pushouts. □

Lemma 8 (Compositionality of Trans) *Given two composable spans of $\mathcal{M}$-morphisms*

$$r \equiv \left(C \xleftarrow{b} B \xrightarrow{a} A\right) \qquad \textit{and} \qquad s \equiv \left(E \xleftarrow{d} D \xrightarrow{c} C\right)\,,$$

define their composition via pullback as

$$\begin{tikzcd}
 & & F \arrow[dl, "f"'] \arrow[dr, "e"] & & \\
 & D \arrow[dl, "d"'] \arrow[dr, "c"] \arrow[rr, phantom, "\mathsf{PB}"] & & B \arrow[dl, "b"'] \arrow[dr, "a"] & \\
E & & C & & A
\end{tikzcd}
\qquad s \circ r := (E \xleftarrow{d \circ f} F \xrightarrow{a \circ e} A)\,, \tag{52}$$

which is again a span of $\mathcal{M}$-morphisms (by stability of $\mathcal{M}$-morphisms under pullbacks and compositions), and thus $r, s, s \circ r \in \mathsf{Lin}(\mathbf{C})$. Let c_E be a condition over E. Then we find that

$$\mathsf{Trans}(r, \mathsf{Trans}(s, \mathsf{c}_E)) \doteq \mathsf{Trans}(s \circ r, \mathsf{c}_E)\,. \tag{53}$$

Proof: The proof relies upon the property of the transport construction stated in Lemma 6 as well as on the $\mathcal{M}$-adhesivity of the underlying category $\mathbf{C}$. We proceed by constructing the following commutative diagram in two different ways for the two directions of the proof:

$$\begin{tikzcd}
 & & & \mathsf{Trans}(s,\mathsf{c}_E) \arrow[d] & & \mathsf{Trans}(r,\mathsf{Trans}(s,\mathsf{c}_E)) \arrow[d] & \\
\mathsf{c}_E \arrow[r] & E \arrow[d, "m^{**}"'] & D \arrow[l, "d"'] \arrow[r, "c"] \arrow[d] & C \arrow[d, "m^{*}"'] & B \arrow[l, "b"'] \arrow[r, "a"] \arrow[d] & A \arrow[d, "m"] & \mathsf{Trans}(s\circ r,\mathsf{c}_E) \arrow[l] \\
 & X'' & D' \arrow[l] \arrow[r] & X' & B' \arrow[l] \arrow[r] & X & \\
 & & & F \arrow[uull, bend left] \arrow[uul] \arrow[uur] \arrow[uurr, bend right] \arrow[d] & & & \\
 & & & F' \arrow[uul] \arrow[uur] \arrow[uuull, bend left] \arrow[uuurr, bend right] & & &
\end{tikzcd} \tag{54}$$

“⇒” direction: Suppose that $m \in \mathsf{M}_r(X)$ and that the comatch m^* of m satisfies $m^* \in \mathsf{M}_s(X')$ (with $X' = r_m(X)$). Then according to Lemma 6, this implies that

$$m^{**} \vDash \mathsf{c}_E \quad \Leftrightarrow \quad m^* \vDash \mathsf{Trans}(s, \mathsf{c}_E) \Leftrightarrow \quad m \vDash \mathsf{Trans}(r, \mathsf{Trans}(s, \mathsf{c}_E))\,.$$

We have to demonstrate that $m \in \mathsf{M}_{s\circ r}(X)$ as well as

$$m \vDash \mathsf{Trans}(r, \mathsf{Trans}(s, \mathsf{c}_E)) \quad \Rightarrow \quad m \vDash \mathsf{Trans}(s \circ r, \mathsf{c}_E)\,. \tag{55}$$

Admissibility of m and m^* entails that the squares formed in the back row of (54) (the ones drawn in black and blue) are constructible as pushouts and pushout complements, respectively. Construct the objects F and F' as pullbacks,

$$F = \mathsf{PB}(D \xrightarrow{c} C \xleftarrow{b} B) \quad \text{and} \quad F' = \mathsf{PB}(D' \to X' \leftarrow B')\,,$$

which by the universal property of pullbacks induces a unique arrow $F \to F'$. By stability of $\mathcal{M}$-morphisms under pullbacks and by the $\mathcal{M}$-morphism decomposition property, respectively, all morphisms thus constructed are found to be in $\mathcal{M}$. Invoking pullback-pullback decomposition (Lemma 12(iii)a) and the $\mathcal{M}$-van Kampen property twice (cf. Definition 1), we conclude that the induced squares $\square(F, F', D', D)$ and $\square(F, F', B', B)$ are in fact pushouts. Thus by pushout composition, the front left and right “curvy” faces (in orange) are pushouts. This entails that $m \in \mathsf{M}_{s\circ r}(X)$, and thus by definition of Trans that indeed

$$m \vDash \mathsf{Trans}(s \circ r, \mathsf{c}_E)\,.$$

“⇐” direction: Suppose that $m \in \mathsf{M}_{s\circ r}(X)$, which by Lemma 6 implies that if m^{**} is the comatch of m under the application of the rule $s \circ r$, then

$$m^{**} \vDash \mathsf{c}_E \quad \Leftrightarrow \quad m \vDash \mathsf{Trans}(s \circ r, \mathsf{c}_E)\,.$$

Admissibility of m for the rule $s \circ r$ applied to the object X entails that the “curvy” front right and front left squares of the diagram (54) drawn in black and orange are constructible as pushout complement and pushout, respectively. We may then construct the remaining parts of the diagram via forming the pushouts

$$D' = \mathsf{PO}(D \leftarrow F \to F')\,,\ B' = \mathsf{PO}(B \leftarrow F \to F')\,,\ X' = \mathsf{PO}(C \leftarrow B \to B')\,,$$

which uniquely induces the remaining arrows drawn in blue (and where we could have equivalently defined X' as $X' = \mathsf{PO}(C \leftarrow D \to D')$). By virtue of three applications of pushout-pushout decomposition (Lemma 12(iii)b), we conclude that all squares in the back of diagram (54) thus constructed are pushouts. Furthermore, stability of $\mathcal{M}$-morphisms under pushouts implies that all newly constructed morphisms are in $\mathcal{M}$. Since thus the back part of the diagram encodes two DPO rewrite steps with $m \in \mathsf{M}_r(X)$, m^* the comatch of m under application of the rule r, $m^* \in \mathsf{M}_s(r_m(X))$, and since m^{**} is also the comatch of m^* under application of s, we find by Lemma 6 that

$$m^{**} \vDash \mathsf{c}_E \quad \Leftrightarrow \quad m^* \vDash \mathsf{Trans}(s, \mathsf{c}_E) \quad \Leftrightarrow \quad m \vDash \mathsf{Trans}(r, \mathsf{Trans}(s, \mathsf{c}_E))\,,$$

which concludes the proof. □

The compositionality of the Trans construction in particular permits an efficient encoding of the reduced standard form of application conditions:

Corollary 2 *Let $R := ((O \xleftarrow{o} K \xrightarrow{i} I), \mathsf{c}_I) \in \overline{\mathsf{Lin}}(\mathbf{C})$ be a linear rule with application conditions. Then the compressed standard form $\dot{\mathsf{c}}_I$ for c_I according to Definition 15 evaluates to*

$$\dot{\mathsf{c}}_I \doteq \mathsf{Trans}(I \xleftarrow{i} K \xrightarrow{i} I, \mathsf{c}_I)\,. \tag{56}$$

Lemma 9 (Compatibility of Shift and Trans; compare [45], Fact 3.14) *Given the data as in the commutative diagram below,*

$$\begin{tikzcd}
\mathsf{c}_O \arrow[d, phantom, "\blacktriangledown"] & & \\
O \arrow[d, "p^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, "\overline{p}"] \arrow[ld, phantom, "\mathsf{PO}"] \arrow[rd, phantom, "\mathsf{PO}"] & I \arrow[d, "p"] \\
O' & K' \arrow[l, "o'"] \arrow[r, "i'"'] & I'
\end{tikzcd}, \tag{57}$$

letting $r = \left(O \xleftarrow{o} K \xrightarrow{i} I\right)$ and $r' = \left(O' \xleftarrow{o'} K' \xrightarrow{i'} I'\right)$, we have that for all objects X and for all admissible matches $n \in \mathsf{M}_{r'}(X)$ of r' into X,

$$n \vDash \mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O)) \Leftrightarrow n \vDash \mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O))\,, \tag{58}$$

which we can write more compactly as

$$\mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O)) \doteq \mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O))\,. \tag{59}$$

Proof: Let us fix an object X and some admissible match $n \in \mathsf{M}_{r'}(X)$.

"⇒" direction: Suppose that $(n : I' \to X) \in \mathsf{M}_{r'}(X)$ satisfies $n \vDash \mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O))$ (which by definition of satisfaction of conditions entails in particular that $\mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O)) \not\equiv \mathsf{false}$). Since n is by assumption an admissible match of r', we can rewrite X by applying r' along n, resulting in the diagram below (where the top part is inserted from the assumption of the lemma):

$$\begin{tikzcd}
\mathsf{c}_O \arrow[d, phantom, "\blacktriangledown"] & & \mathsf{Trans}(r, \mathsf{c}_O) \arrow[d, phantom, "\blacktriangledown"] & \\
O \arrow[d, "p^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, "\overline{p}"'] \arrow[ld, phantom, "\mathsf{PO}"] \arrow[rd, phantom, "\mathsf{PO}"] & I \arrow[d, "p"] & \\
O' \arrow[d, "n^*"'] & K' \arrow[l, "o'"'] \arrow[r, "i'"] \arrow[d, "\overline{n}"'] \arrow[ld, phantom, "\mathsf{PO}"] \arrow[rd, phantom, "\mathsf{POC}"] & I' \arrow[d, "n"] & \mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O)) \arrow[l, phantom, "\blacktriangleleft"] \\
r'_n(X) & K' \arrow[l] \arrow[r] & X &
\end{tikzcd} \tag{60}$$

By composition of pushout squares, we conclude that the $\mathcal{M}$-morphism $m = n \circ p$ is an admissible match for r, which entails that $r'_n(X) \cong r_m(X)$. By definition of Shift, $n \vDash \mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O))$ implies that $m = n \circ p$ satisfies $m \vDash \mathsf{Trans}(r', \mathsf{c}_O)$, and moreover that $\mathsf{Trans}(r', \mathsf{c}_O) \not\equiv \mathsf{false}$. Since we found that $m \in \mathsf{M}_r(X)$, by definition of Trans we have that the comatch $m^* : O \to r_m(X)$ of m (which is by construction of DPO rule applications in $\mathcal{M}$) has the property $m^* = n^* \circ p^* \vDash \mathsf{c}_O$. The latter implies that $n^* \vDash \mathsf{Shift}(p^*, \mathsf{c}_O)$. Since by assumption $n \in \mathsf{M}_{r'}(X)$, and since n^* is the comatch of n, we finally conclude that indeed $n \vDash \mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O))$.

“$\Leftarrow$” direction: The proof is entirely analogous to the previous case (starting from the observation that $n \in \mathsf{M}_{r'}(X)$ together with the data provided in (57) entails that $m \in \mathsf{M}_r(X)$). □

4.3 A refined notion of sequential compositions of DPO-type rules with conditions

In this subsection, we will present a refinement of the notion of sequential rule compositions known from the traditional rewriting literature [42] obtained via requiring the underlying $\mathcal{M}$-adhesive category $\mathbf{C}$ to satisfy Assumption 1, which as we will demonstrate guarantees a number of additional technical properties of this type of composition. Referring to Remark 2 for the precise technical disambiguation, the motivation for our construction consisted in the idea that the operation of *sequential composition* of two linear rules should lift to an operation permitting to compose arbitrary numbers of linear rules, in the sense that the composition operation should satisfy a certain abstract *associativity* property (cf. Section 4.5). From a technical standpoint, ensuring this notion of *compositionality* permits to develop novel static analysis techniques for rule-based systems, such as the *DPO-type rule algebra framework* [11–13] for the study of stochastic rewriting systems, which has recently been extended to the setting of linear rules with conditions based upon the present article in [10]. The second main application of compositional rewriting has been the development of the theory of *tracelets* [9].

Definition 16 *Let* $\mathbf{C}$ *be a category satisfying Assumption 1. Let* $R_j \equiv (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ *be two linear rules with application conditions* ($j = 1, 2$)*, and let*

$$\mu_{21} \equiv \left(I_2 \xleftarrow{m_2} M_{21} \xrightarrow{m_1} O_1\right)$$

be a span of monomorphisms (i.e. $m_1, m_2 \in \mathcal{M}$*). If the diagram below is constructible,*

$$\begin{tikzcd}
 & & \mathsf{c}_{I_2} \arrow[d, -{Triangle}] & & & & \mathsf{c}_{I_1} \arrow[d, -{Triangle}] & \\
O_2 \arrow[d, dotted] \arrow[dr, phantom, "\mathsf{PO}"] & K_2 \arrow[l, "o_2"'] \arrow[r, "i_2"] \arrow[d, dashed] \arrow[dr, phantom, "\mathsf{POC}"] & I_2 \arrow[dr, bend right, "m'_2"] & M_{21} \arrow[l, "m_2"'] \arrow[r, "m_1"] \arrow[d, phantom, "\mathsf{PO}"] & O_1 \arrow[dl, bend left, "m'_1"'] \arrow[dr, phantom, "\mathsf{POC}"] & K_1 \arrow[l, "o_1"'] \arrow[r, "i_1"] \arrow[d, dashed] \arrow[dr, phantom, "\mathsf{PO}"] & I_1 \arrow[d, dotted, "p_1"] & \\
O_{21} & K'_2 \arrow[l, dotted] \arrow[rr, dashed] \arrow[d, phantom, "=", pos=0.2] & & N_{21} \arrow[rr, dashed] \arrow[d, phantom, "\mathsf{PB}"] & & K'_1 \arrow[r, dotted] \arrow[d, phantom, "=", pos=0.2] & I_{21} & \mathsf{c}_{I_{21}} \arrow[l, -{Triangle}] \\
 & & & K_{21} \arrow[llu, dashed] \arrow[rru, dashed] \arrow[lllu, bend left, "o_{21}"] \arrow[rrru, bend right, "i_{21}"'] & & & &
\end{tikzcd} \tag{61}$$

where

$$\mathsf{c}_{I_{21}} := \mathsf{Shift}(p_1, \mathsf{c}_{I_1}) \bigwedge \mathsf{Trans}\left(N_{21} \leftarrow K'_1 \rightarrow I_{21}, \mathsf{Shift}(m'_2, \mathsf{c}_{I_2})\right)\,, \tag{62}$$

and if $\mathsf{c}_{I_{21}} \not\equiv \mathsf{false}$*, then we call* μ_{21} *an* admissible match *for the rules with conditions* R_2 *and* R_1*, denoted*

$$\mu_{21} \in M_{R_2}(R_1)\,.$$

In this case, we introduce the notation $R_2 \overset{\mu_{21}}{\blacktriangleleft} R_1$ *to denote the composite,*

$$R_2 \overset{\mu_{21}}{\blacktriangleleft} R_1 := \left(O_{21} \xleftarrow{o_{21}} K_{21} \xrightarrow{i_{21}} I_{21}, \mathsf{c}_{I_{21}}\right). \tag{63}$$

Remark 2 *The variant of rule composition provided in Definition 16 (i.e. starting from an “overlap” of two rules encoded as an* $\mathcal{M}$*-monic span, followed by taking a pushout of this span) follows the philosophy put forward in [19, 53] (sometimes referred to as* D-concurrent *composition). However, for the setting of DPO-rewriting over* $\mathcal{M}$*-adhesive categories for rules with conditions, an alternative construction, referred to as* E-concurrent *composition, had been the de facto standard in the rewriting literature, developed by Ehrig et al. (cf. e.g. [42], Definition 4.13). More precisely, based upon the assumption that the underlying* $\mathcal{M}$*-adhesive category* $\mathbf{C}$ *possesses an* $\mathcal{E}'$*-*$\mathcal{M}$*-pair-*factorization *(cf. Remark 1), the starting point of constructing an* E-concurrent *rule composition according to [42] consists in picking a* cospan $(m'_2, m'_1) \in \mathcal{E}'$*, as opposed to picking a* span *of*

$\mathcal{M}$-morphisms (m_2, m_1) and taking the pushout to obtain the cospan (m_2', m_1') as in (61) for the D-concurrent *approach. While the* E-concurrent *approach possesses the technical advantage that it does not require the category $\mathbf{C}$ to possess $\mathcal{M}$-effective unions, nor that rules must be linear nor matches necessarily in $\mathcal{M}$ (and thus applying to a broader class of rewriting systems), the reliance upon the $\mathcal{E}'$-$\mathcal{M}$-factorizations is also responsible for certain algorithmic disadvantages. In close analogy to the discussion provided in Remark 1, exhaustively enumerating all possible cospans in $\mathcal{E}'$ suitable for the composition of two rewriting rules is algorithmically quite non-trivial in the general case, while enumerating the (isomorphism classes of $\mathcal{M}$-) monic spans (i.e. the "overlaps") of two linear rules is comparatively straightforward (c.f. e.g. [16] for a recent implementation via the* Microsoft Z3 *SMT-solver). On the other hand, utilizing yet again the results of Corollary 1, our refinement of* D-concurrent *compositions as well as the respective variant of the* concurrency theorem *(see Theorem 5) can be demonstrated to constitute a special case of the* E-concurrent *constructions of Ehrig et al. [42]: for an $\mathcal{M}$-adhesive category satisfying Assumption 1, a cospan (m_2', m_1') of $\mathcal{M}$-morphisms constructed via taking the pushout of a span of $\mathcal{M}$-morphisms (m_2, m_1) is in particular jointly epimorphic, and thus constitutes a special case of a cospan in $\mathcal{E}'$ (cf. [22, Fact 3.7], applied under the assumption of $\mathcal{M}$-effective unions).*

The definition of the composition operation $.\overset{\cdot}{\blacktriangleleft}.$ entails a number of highly non-trivial effects in practical computations, which originate from the interplay of admissibility of matches for rules without application conditions and the requirements on the induced composite application conditions. One of the most striking such results, well known also from the traditional graph rewriting literature [42], is the following:

Lemma 10 (Trivial matches) *By definition of the notion of admissible matches, for any two linear rules with application conditions $R_j \equiv (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ $(j = 1, 2)$, the* trivial match

$$\mu_{\varnothing} := (I_2 \hookleftarrow \varnothing \hookrightarrow O_1)$$

is an admissible match $\mu_{\varnothing} \in M_{R_2}(R_1)$ if and only if the composite condition $\mathsf{c}_{I_{21}}$ does not evaluate to false.

Proof: The proof follows directly from the definition of the composition operation $.\overset{\cdot}{\blacktriangleleft}.$, namely by construction of the following diagram:

$$\begin{tikzcd}
 & & \mathsf{c}_{I_2} \arrow[d] & & & & \mathsf{c}_{I_1} \arrow[d] & \\
O_2 \arrow[d, dotted] \arrow[dr, phantom, "\mathsf{PO}"] & K_2 \arrow[l, "o_2"'] \arrow[r, "i_2"] \arrow[d, dashed] \arrow[dr, phantom, "\mathsf{POC}"] & I_2 \arrow[dr, bend right, "m_2'"] & \varnothing \arrow[l, "m_2"'] \arrow[r, "m_1"] \arrow[d, phantom, "\mathsf{PO}"] & O_1 \arrow[dl, bend left, "m_1'"'] \arrow[dr, phantom, "\mathsf{POC}"] & K_1 \arrow[l, "o_1"'] \arrow[r, "i_1"] \arrow[d, dashed] \arrow[dr, phantom, "\mathsf{PO}"] & I_1 \arrow[d, dotted, "p_1"] & \\
O_2+O_1 & K_2+O_1 \arrow[l, dotted] \arrow[rr, dashed] \arrow[d, phantom, "=" near start] & & I_2+O_1 \arrow[rr, dashed] \arrow[d, phantom, "\mathsf{PB}"] & & I_2+K_1 \arrow[r, dotted] \arrow[d, phantom, "=" near start] & I_2+I_1 & \mathsf{c}_{I_{21}} \arrow[l] \\
 & & & K_2+K_1 \arrow[ulll, bend left, "o_2+O_1"] \arrow[ull, dashed] \arrow[urr, dashed] \arrow[urrr, bend right, "i_2+i_1"'] & & & &
\end{tikzcd} \tag{64}$$

By virtue of Lemma 1 and Lemma 14, the pushout complements marked POC in the diagram above always exist. To determine whether the trivial match $\mu_{\varnothing}$ is an admissible match, it then remains to evaluate the composite condition $\mathsf{c}_{I_{21}}$, which according to (62) of Definition 16 reads

$$\begin{aligned}\mathsf{c}_{I_{21}} := & \,\mathsf{Shift}(p_1 : I_1 \to I_2 + I_1, \mathsf{c}_{I_1}) \\ & \bigwedge \mathsf{Trans}\left(I_2 + O_1 \leftarrow I_2 + K_1 \to I_2 + I_1, \mathsf{Shift}(m_2' : I_2 \to I_2 + O_1, \mathsf{c}_{I_2})\right) .\end{aligned} \tag{65}$$

Thus the claim follows, since the above condition may evaluate to false in general, such as in the case where $\mathsf{c}_{I_2} = \neg\exists(I_2 \to I_2 + O_1, \mathsf{true})$. □

Nevertheless, it is possible to exhibit one special rule for which $\mu_{\varnothing}$ is always an admissible match:

Lemma 11 (Neutral element for $.\overset{\cdot}{\blacktriangleleft}.$) *By definition of $.\overset{\cdot}{\blacktriangleleft}.$, the* trivial rule

$$R_{\varnothing} := (\varnothing \leftarrow \varnothing \to \varnothing, \mathsf{true})$$

is the (left- and right-) neutral element *for $.\overset{\cdot}{\blacktriangleleft}.$.*

Proof: The proof follows from a specialization of (64) in the proof of Lemma 10, by specializing either of the two linear rules involved to the trivial rule. Note first that on the level of rules without application conditions, the only admissible match between the trivial rule and another linear rule is the trivial match $\mu_\varnothing$. Let us then compute the condition $\mathsf{c}_{I_{21}}$ of the composite for the case $r_1 = r_\varnothing$ and for generic $r_2 = (O_2 \leftarrow K_2 \rightarrow I_2)$, which reads according to (62) of Definition 16 reads

$$\begin{aligned}\mathsf{c}_{I_{21}} &= \mathsf{Shift}(p_1 : \varnothing \rightarrow I_2, \mathsf{true})\\ &\qquad \bigwedge \mathsf{Trans}\left(I_2 \leftarrow I_2 \rightarrow I_2, \mathsf{Shift}(q_2 : I_2 \rightarrow I_2, \mathsf{c}_{I_2})\right)\\ &\equiv \mathsf{c}_{I_2}\,.\end{aligned} \tag{66}$$

Thus for every linear rule r_2 with application condition $\mathsf{c}_{I_2} \not\equiv \mathsf{false}$, we have $\mu_\varnothing \in \mathsf{M}_{R_2}(R_\emptyset)$. For the remaining case, consider that $r_1 = (O_1 \leftarrow K_1 \rightarrow I_1)$ is an arbitrary linear rule with application condition $\mathsf{c}_{I_1} \not\equiv \mathsf{false}$. Again, admissibility of $\mu_\varnothing$ as a match of the rules without conditions follows by a specialization of (64), so it remains to compute the composite condition $\mathsf{c}_{I_{21}}$:

$$\begin{aligned}\mathsf{c}_{I_{21}} &= \mathsf{Shift}(p_1 : I_1 \rightarrow I_1, \mathsf{c}_{I_1})\\ &\qquad \bigwedge \mathsf{Trans}\left(O_1 \leftarrow K_1 \rightarrow I_1, \mathsf{Shift}(q_2 : \varnothing \rightarrow O_1, \mathsf{true})\right)\\ &\equiv \mathsf{c}_{I_1}\,.\end{aligned} \tag{67}$$

□

4.4 Concurrency theorem for DPO-type rules with conditions

We will need the following *concurrency theorem*, which is a variant of a result of [47] adapted to our refined notion of rule compositions, and which for the case of rules without conditions was introduced in [12]:

Theorem 5 (Concurrency theorem, compare Thm. 4 of [48] and Thm. 2.7 of [12]) *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category satisfying Assumption 1,* $X_0 \in \mathsf{obj}(\mathbf{C})$ *an object, and* $R_j \equiv (r_j, \mathsf{c}_{I_j})$ *be two linear rules with application conditions* $(j = 1, 2)$*. Then there exists the following* bijection*:*

(i) "Synthesis": *For every sequence of rule applications*

$$X_2 \xLeftarrow[R_2, m_2]{} X_1 \xLeftarrow[R_1, m_2]{} X_0 \tag{68}$$

along admissible matches $m_1 \in M_{R_1}(X_0)$ *and* $m_2 \in M_{R_2}(X_1)$ *with* $X_1 = r_{1_{m_1}}(X_0)$*, there exist admissible matches* $\mu_{21} \in M_{R_2}(R_1)$ *of the linear rule* R_2 *into* R_1 *and* $m_{21} \in M_{R_{21}}(X_0)$*, with* $R_{21} \equiv (r_{21}, \mathsf{c}_{I_{21}})$*,* $r_{21} = R_2 \overset{\mu_{21}}{\blacktriangleleft} R_1$*, and an application condition* $\mathsf{c}_{I_{21}}$ *computed as*

$$\mathsf{c}_{I_{21}} = \mathsf{Shift}(I_1 \rightarrow I_{21}, \mathsf{c}_{I_1}) \bigwedge \mathsf{Trans}\left((N_{21} \hookleftarrow K_1' \hookrightarrow I_{21}), \mathsf{Shift}(I_2 \rightarrow N_{21}, \mathsf{c}_{I_1})\right)\,, \tag{69}$$

where the morphisms and objects in this formula depend (uniquely up to isomorphism) on the input data, such that $X_2 \cong R_{21_{m_{21}}}(X_0)$*.*

(ii) "Analysis": *For every admissible match* $\mu \in M_{R_2}(R_1)$ *and for every rule application*

$$X_2 \xLeftarrow[R_{21}, m_{21}]{} X_0 \tag{70}$$

with $m_{21} \in M_{R_{21}}(X)$ *and* $R_{21} = R_2 \overset{\mu_{21}}{\blacktriangleleft} R_1$*, there exists a pair of admissible matches such as in* (68) *which transform* X_0 *via* X_1 *into the same (up to isomorphism) object* X_2*.*

Proof: Referring the interested readers to [12] for the precise details, note first that at the level of linear rules without application conditions, the concurrency theorem of [12] entails the parts of the above statements pertaining to the existence of the admissible matches of "plain" rules. The concrete technical construction of the proof provided in [12] is summarized by the diagram below, where all vertical squares are pushouts (and where we have marked the relevant conditions for later convenience). The aforementioned proof consisted in verifying that the parts of the diagram marked in blue can be uniquely constructed from the parts of the diagram colored in orange and vice versa:

$$\begin{tikzcd}
 & & \mathsf{c}_{I_2} \arrow[d, phantom, "\blacktriangledown"] & & & & \mathsf{c}_{I_1} \arrow[d, phantom, "\blacktriangledown"] & \\
O_2 \arrow[dd] \arrow[dddd, bend right] & K_2 \arrow[l] \arrow[r] \arrow[dd] \arrow[dddd, bend right] & I_2 \arrow[ddr, bend right] \arrow[ddddr, bend right] & M_{21} \arrow[l] \arrow[r] & O_1 \arrow[ddl, bend left] \arrow[ddddl, bend left] & K_1 \arrow[l] \arrow[r] \arrow[dd] \arrow[dddd, bend left] & I_1 \arrow[dd] \arrow[dddd, bend left] & \\
 & & & \mathsf{Shift}(I_2 \to N_{21}, \mathsf{c}_{I_2}) \arrow[d, phantom, "\blacktriangledown"] & & & & \\
O_{21} \arrow[dd] & K_2' \arrow[l] \arrow[rr] \arrow[dd, dashed] & & N_{21} \arrow[dd, dashed] & & K_1' \arrow[ll] \arrow[r] \arrow[dd, dashed] & I_{21} \arrow[dd] & \mathsf{c}_{I_{21}} \arrow[l, phantom, "\blacktriangleleft"] \\
 & & K_{21} \arrow[ull] \arrow[ul] \arrow[urrr] \arrow[urrrr] \arrow[dd] & & & & & \\
X_2 & \overline{K}_2 \arrow[l] \arrow[rr] & & X_1 & & \overline{K}_1 \arrow[ll] \arrow[r] & X_0 & \\
 & & \overline{K}_{21} \arrow[ull] \arrow[ul, dashed] \arrow[urrr, dashed] \arrow[urrrr] & & & & &
\end{tikzcd} \tag{71}$$

It thus remains to verify the part of the claim pertaining to the relevant conditions of the rules.

“Analysis” part of the proof: Suppose that we are given admissible matches $(m_1 : I_1 \to X_0) \in M_{R_1}(X)$ and $(m_2 : I_2 \to X_1) \in M_{R_2}(X_1)$ with $X_1 = r_{1_{m_1}}(X_0)$. Admissibility entails in particular that $m_1 \vDash \mathsf{c}_{I_1}$ and $m_2 \vDash \mathsf{c}_{I_2}$. By construction of the diagram in (71), we have that m_1 and m_2 factor as

$$\begin{aligned}
m_1 &= (X_0 \leftarrow I_1) = (X_0 \leftarrow I_{21}) \circ (I_{21} \leftarrow I_1) \\
m_2 &= (X_1 \leftarrow I_2) = (X_1 \leftarrow N_{21}) \circ (N_{21} \leftarrow I_2)\,,
\end{aligned}$$

which entails by definition of the Shift construction that $m_{21} := (I_{21} \to X_0)$ and $\bar{m}_{21} = (N_{21} \to X_1)$ satisfy $m_{21} \vDash \mathsf{Shift}(I_1 \to I_{21}, \mathsf{c}_{I_1})$ and $\bar{m}_{21} \vDash \mathsf{Shift}(I_2 \to N_{21}, \mathsf{c}_{I_2})$, respectively. Noting that the rightmost two bottom squares in the back of (71) are pushouts, we find in addition that

$$m_{21} \vDash \mathsf{Trans}\left(N_{21} \leftarrow K_1' \to I_{21}, \mathsf{Shift}(I_2 \to N_{21}, \mathsf{c}_{I_1})\right)\,.$$

Since according to Definition 16, $R_{21} := R_2 \overset{\mu_{21}}{\blacktriangleleft} R_1 = (r_{21}, \mathsf{c}_{I_{21}})$, with $\mathsf{c}_{I_{21}}$ as defined in (69), we confirm that $m_{21} \vDash \mathsf{c}_{I_{21}}$, which concludes the proof of the *“analysis”* part of the theorem.

“Synthesis” part of the proof: Supposing that we are given an admissible match m_{21} of the composite R_{21} of the rules with application conditions R_2 with R_1 along the admissible match $\mu_{21} \in \mathsf{M}_{R_1}(R_2)$, the construction of the diagram in (71) provides two admissible matches $(m_1 : I_1 \to X_0) \in \mathsf{M}_{r_1}(X_0)$ and $(m_2 : I_2 \to X_1) \in \mathsf{M}_{r_2}(X_1)$ with $X_1 = r_{1_{m_1}}(X_0)$. It thus remains to verify the claim that these matches satisfy the conditions c_{I_1} and c_{I_2}, respectively, which is demonstrated by running the corresponding arguments of the *“analysis”* part of the proof “in reverse”. □

4.5 Associativity of DPO-type composition of rules with conditions

We will now state one of the main results of this paper, in the form of an associativity property afforded by the DPO-type composition operation on rules with conditions. The case of DPO-type compositions of rules without conditions was studied in [11–13], and the following result is an extension to the setting of rules with conditions afforded by our refined framework for conditions as introduced in Section 3 and the current section. In contrast to the DPO-type concurrency theorem (which, in a slightly different formulation, had been previously known in the literature), the associativity result presented below is, to the best of our knowledge, the first of its kind.

Theorem 6 (DPO-type Associativity Theorem) *Let $R_j \equiv (r_j, \mathsf{c}_{I_j})$ $(j = 1, 2, 3)$ be three linear rules with application conditions. Then there exists a* bijection *between the pairs of admissible matches M_A and M_B defined as*

$$
\begin{aligned}
M_A &:= \{(\mu_{21}, \mu_{3(21)}) | \mu_{21} \in \mathsf{M}_{R_2}(R_1)\,,\ \mu_{3(21)} \in \mathsf{M}_{R_3}(R_{21})\} \\
M_B &:= \{(\mu_{32}, \mu_{(32)1}) | \mu_{32} \in \mathsf{M}_{R_3}(R_2)\,,\ \mu_{(32)1} \in \mathsf{M}_{R_{32}}(R_1)\}
\end{aligned}
\tag{72}
$$

with $R_{i,j} := (r_i \overset{\mu_{ij}}{\blacktriangleleft} r_j, \mathsf{c}_{I_{ij}})$ (and $\mathsf{c}_{I_{ij}}$ defined as in (62)) such that

$$
\begin{aligned}
&\forall (\mu_{21}, \mu_{3(21)}) \in M_A : \exists! (\mu_{32}, \mu_{(32)1}) \in M_B : \\
&\qquad r_3 \overset{\mu_{3(21)}}{\blacktriangleleft} \left(r_2 \overset{\mu_{21}}{\blacktriangleleft} r_1 \right) \cong \left(r_3 \overset{\mu_{32}}{\blacktriangleleft} r_2 \right) \overset{\mu_{(32)1}}{\blacktriangleleft} r_1 && \text{(73a)} \\
&\qquad \wedge \quad \mathsf{c}_{I_{3(21)}} \doteq \mathsf{c}_{I_{(32)1}} && \text{(73b)}
\end{aligned}
$$

and vice versa. In this particular sense, the operation $.\overset{\cdot}{\blacktriangleleft}.$ is **associative***.*

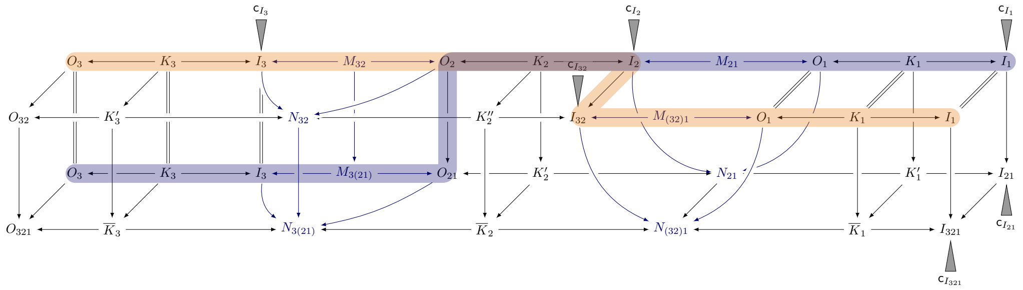
Proof: Considering first the case of compositions of "plain" rules $r_1, r_2, r_3 \in \mathsf{Lin}(\mathbf{C})$, i.e. of rules without application conditions, we quote from [12] (Theorem 2.9) an isomorphism of pairs of admissible matches of the form

$$
\begin{aligned}
&M'_A := \{(\mu_{21}, \mu_{3(21)}) | \mu_{21} \in \mathsf{M}_{r_2}(r_1)\,,\ \mu_{3(21)} \in \mathsf{M}_{r_3}(r_{21})\} \\
\cong\quad &M'_B := \{(\mu_{32}, \mu_{(32)1}) | \mu_{32} \in \mathsf{M}_{r_3}(r_2)\,,\ \mu_{(32)1} \in \mathsf{M}_{r_{32}}(r_1)\}\,.
\end{aligned}
\tag{74}
$$

The isomorphism entails that for each corresponding pair, equation (73a) is verified. Consider then two such isomorphic pairs

$$
(\mu_{21}, \mu_{3(21)}) \in M'_A \quad \text{and} \quad (\mu_{32}, \mu_{(32)1}) \in M'_B
$$

of admissible matches of "plain" rules. The isomorphism entails in particular that the following commutative diagram is uniquely constructible (where we also draw the positions of the various application conditions at play for later convenience):



(75)

In order to verify the validity of (73b), it is sufficient to utilize our various technical lemmas pertaining to the Shift and Trans constructions, and to follow the "paths" in the diagram depicted in (75) along which the three conditions c_{I_j} for $j = 1, 2, 3$ have to be shifted and transported in order to form the conditions $\mathsf{c}_{I_{3(21)}}$ and $\mathsf{c}_{I_{(32)1}}$, respectively.

(i) contribution of c_{I_1}:

$$
\mathsf{Shift}\left(I_{321} \leftarrow I_{21}, \mathsf{Shift}\left(I_{21} \leftarrow I_1, \mathsf{c}_{I_1}\right)\right) \overset{Lem.\ 5}{\equiv} \mathsf{Shift}\left(I_{321} \leftarrow I_1, \mathsf{c}_{I_1}\right)
\tag{76}
$$

(ii) contribution of c_{I_2}:

$$
\begin{aligned}
&\mathsf{Shift}\left(I_{321} \leftarrow I_{21}, \mathsf{Trans}\left(N_{21} \leftarrow K'_1 \rightarrow I_{21}, \mathsf{Shift}\left(N_{21} \leftarrow I_2, \mathsf{c}_{I_2}\right)\right)\right) \\
&\overset{Lem.\ 9}{\doteq} \mathsf{Trans}\left(N_{(32)1} \leftarrow \overline{K}_1 \rightarrow I_{321}, \mathsf{Shift}\left(N_{(32)1} \leftarrow N_{21}, \mathsf{Shift}\left(N_{21} \leftarrow I_2, \mathsf{c}_{I_2}\right)\right)\right) \\
&\overset{Lem.\ 5}{\equiv} \mathsf{Trans}\left(N_{(32)1} \leftarrow \overline{K}_1 \rightarrow I_{321}, \mathsf{Shift}\left(N_{(32)1} \leftarrow I_{32}, \mathsf{Shift}\left(I_{32} \leftarrow I_2, \mathsf{c}_{I_2}\right)\right)\right)\,.
\end{aligned}
\tag{77}
$$

Here, in the last step, we have made use of the commutativity of the diagram (75), whereby

$$
\left(N_{(32)1} \leftarrow N_{21}\right) \circ \left(N_{21} \leftarrow I_2\right) = \left(N_{(32)1} \leftarrow I_{32}\right) \circ \left(I_{32} \leftarrow I_2\right)\,.
\tag{78}
$$

(iii) contribution of c_{I_3}:

$$\begin{aligned}
&\mathsf{Trans}\Big(N_{(32)1} \leftarrow \overline{K}_1 \rightarrow I_{321}, \mathsf{Trans}\big(N_{3(21)} \leftarrow \overline{K}_2 \rightarrow N_{(32)1},\\
&\qquad \mathsf{Shift}\big(N_{3(21)} \leftarrow I_3, \mathsf{c}_{I_3}\big)\big)\Big)\\
\overset{Lem.~5}{\equiv}\ &\mathsf{Trans}\Big(N_{(32)1} \leftarrow \overline{K}_1 \rightarrow I_{321}, \mathsf{Trans}\big(N_{3(21)} \leftarrow \overline{K}_2 \rightarrow N_{(32)1},\\
&\qquad \mathsf{Shift}\big(N_{3(21)} \leftarrow N_{32}, \mathsf{Shift}\big(N_{32} \leftarrow I_3, \mathsf{c}_{I_3}\big)\big)\big)\Big)\\
\overset{Lem.~9}{\doteq}\ &\mathsf{Trans}\Big(N_{(32)1} \leftarrow \overline{K}_1 \rightarrow I_{321}, \mathsf{Shift}\big(N_{(32)1} \leftarrow I_{32},\\
&\qquad \mathsf{Trans}\big(N_{32} \leftarrow K''_2 \rightarrow I_{32}, \mathsf{Shift}\big(N_{32} \leftarrow I_3, \mathsf{c}_{I_3}\big)\big)\big)\Big)
\end{aligned} \tag{79}$$

It is then easy to verify (using the definition of concurrent compositions of rules with application conditions with rules according to Definition 16 and eq. (62)) that since

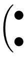
$$\mathsf{c}_{I_{3(21)}} = lhs(76) \bigwedge lhs(77) \bigwedge lhs(79)\,, \qquad \mathsf{c}_{I_{(32)1}} = rhs(76) \bigwedge rhs(77) \bigwedge rhs(79)\,, \tag{80}$$

where by "rhs" we mean the very last equality in each set of equations, we find indeed that

$$\mathsf{c}_{I_{3(21)}} \doteq \mathsf{c}_{I_{(32)1}}\,. \tag{81}$$

□

Example 5 *In order to illustrate the notion of associativity more intuitively, we provide a concrete example of a triple rule composition as depicted in Figure 2, with the application conditions of the three rules given as*

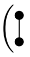
$\mathsf{c}_{I_1} := \neg\exists\Big($  $\hookrightarrow$ $, \mathsf{true}\Big)\ ,\ \mathsf{c}_{I_2} := \mathsf{true}\,,\ \mathsf{c}_{I_3} := \mathsf{c}_{I_1}\,.$

Note that the application conditions as specified guarantee that applying the rules to simple undirected graphs *preserves the constraint of no two vertices being linked by more than one edge. Rather than depicting the relevant data for this computation in the form of a "commutative tube" as in (75), we opt here to "unfold" the two pair-wise sequential rule composition steps for each branch of the equivalence*

$$r_{3(21)} \cong r_{(32)1} \quad \wedge \quad \mathsf{c}_{I_{3(21)}} \doteq \mathsf{c}_{I_{(32)1}}\,,$$

with the two sides of the equivalences depicted on the top and bottom parts of Figure 2, respectively. The example allows us to highlight a number of interesting phenomena:

- *Upon closer inspection, one may verify that the application condition c_{I_1} of the rule R_1 is not specified in "compressed standard form" in the sense of Definition 15, since in fact $\mathsf{c}_{I_1} \doteq \mathsf{true}$. Intuitively, under DPO-semantics the application condition c_{I_1} does not constrain the admissible matches, since in cases where the two vertices in I_1 would be matched against two vertices in a host graph linked by an edge, the rule R_1 would not be applicable (since edges cannot be deleted implicitly in DPO-rewriting).*
- *Utilizing the* Shift *and* Trans *constructions, followed by applying the "compression" according to Definition 15, the application condition of the composite rule can be computed as follows:*

  $\mathsf{c}_{I_{21}} \doteq \mathsf{true}\,,\ \mathsf{c}_{I_{32}} \doteq \neg\exists\Big($  $\hookrightarrow$ $, \mathsf{true}\Big)\ ,\ \mathsf{c}_{I_{3(21)}} \doteq \mathsf{c}_{I_{(32)1}} \doteq \mathsf{true}\,.$

  This is a typical illustration of the way associativity in sequential compositions gives rise to causal relationships*: while the rule R_3 can only be applied to vertices not linked by an edge, the fact that in the triple composition depicted the rule R_1 is in a sense "providing" the two vertices on its output that are later acted upon by R_3, and since by the application of R_2 as shown the edge between the two vertices on the output of R_1 is removed, it is indeed the case that we find a graph pattern that R_3 can be applied to.*

- *Finally, associativity also entails that one may equivalently lead the preceding causality argument from the viewpoint depicted in the bottom part of Figure 2: composing rule R_3 with rule R_2 as demonstrated, the composite rule has an application condition $\mathsf{c}_{I_{32}}$ that demands that the two vertices on the input of the composite rule are not linked to each other by two edges. But then if the input pattern of the composite rule (two vertices linked by an edge) is "provided" by the output of R_1 as depicted (i.e. via computing the composition $r_3 \overset{\mu_{32}}{\blacktriangleleft} r_2$ and the composite rule's application condition $\mathsf{c}_{I_{32}}$), and since R_1 can only be applied at two vertices not linked to each other, one finds again that the overall triple composite possesses an application condition that is equivalent to a trivial condition* true.

We refer the interested readers to [9] for further examples of such types of sequential rule compositions and associativity properties, which play a quintessential role in the theory of *tracelets*, where the latter are a form of invariant encoding of length n sequences of rule compositions for general $n \geq 1$. Associativity is moreover at the core of the construction of *rule algebra theories* [8, 11–13], and with the case of rule algebras for linear rules with conditions recently covered in [10], yet a full discussion of this novel mathematical concept is out of the scope of the present paper.

5 Compositional associative Sesqui-Pushout rewriting with conditions

While the previously discussed notion of compositional DPO-type rewriting may be seen as a refinement of pre-existing notions of DPO-rewriting from the literature (apart from the associativity theorem), the corresponding construction for a framework of SqPO-rewriting is almost entirely new. The first framework for a compositional SqPO-rewriting framework for rules without conditions was introduced in [8]. The essential technical step in order to extend our framework from DPO- to SqPO-type semantics consists in analyzing the interplay of final pullback complements with application conditions and transformations thereof. We will first provide a brief introduction to this type of rewriting, and then develop our new framework.

5.1 Definition of SqPO rule applications and rule compositions

Definition 17 (compare [26], Def. 4) *Given an object $X \in \mathsf{obj}(\mathbf{C})$ and a linear rule $r \in \mathsf{Lin}(\mathbf{C})$, we denote the* set of SqPO-admissible matches $\mathsf{M}^{sq}_r(X)$ *as*

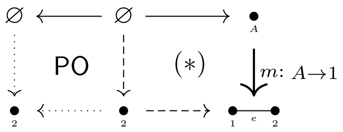
$$\mathsf{M}^{sq}_r(X) := \{(m : I \to X) \in \mathcal{M}\}\,. \tag{82}$$

Let $m \in \mathsf{M}^{sq}_r(X)$. Then the diagram below is constructed by taking the final pullback complement marked FPC *followed by taking the pushout marked* PO*:*

$$\begin{tikzcd}
O \arrow[d, "m^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, "k"'] \arrow[dl, phantom, "\mathsf{PO}"] \arrow[dr, phantom, "\mathsf{FPC}"] & I \arrow[d, "m"] \\
X' & \overline{K} \arrow[l, "o'"] \arrow[r, "i'"'] & X
\end{tikzcd} \tag{83}$$

We write $r_m(X) := X'$ for the object "produced" by the above diagram. The process is called (SqPO)-derivation *of X along rule r and admissible match m, and denoted $r_m(X) \xLeftarrow[r,m]{SqPO} X$.*

Notably, SqPO-type rewriting thus differs from DPO-type rewriting in the important aspect that final pullback complements as well as pushouts are guaranteed to exist in our base category (cf. Assumption 2), whereas pushout complements may fail to exist in general. A typical example already mentioned in the introduction concerns the application of a vertex-deletion rule to a graph that consists of two vertices linked by an edge:



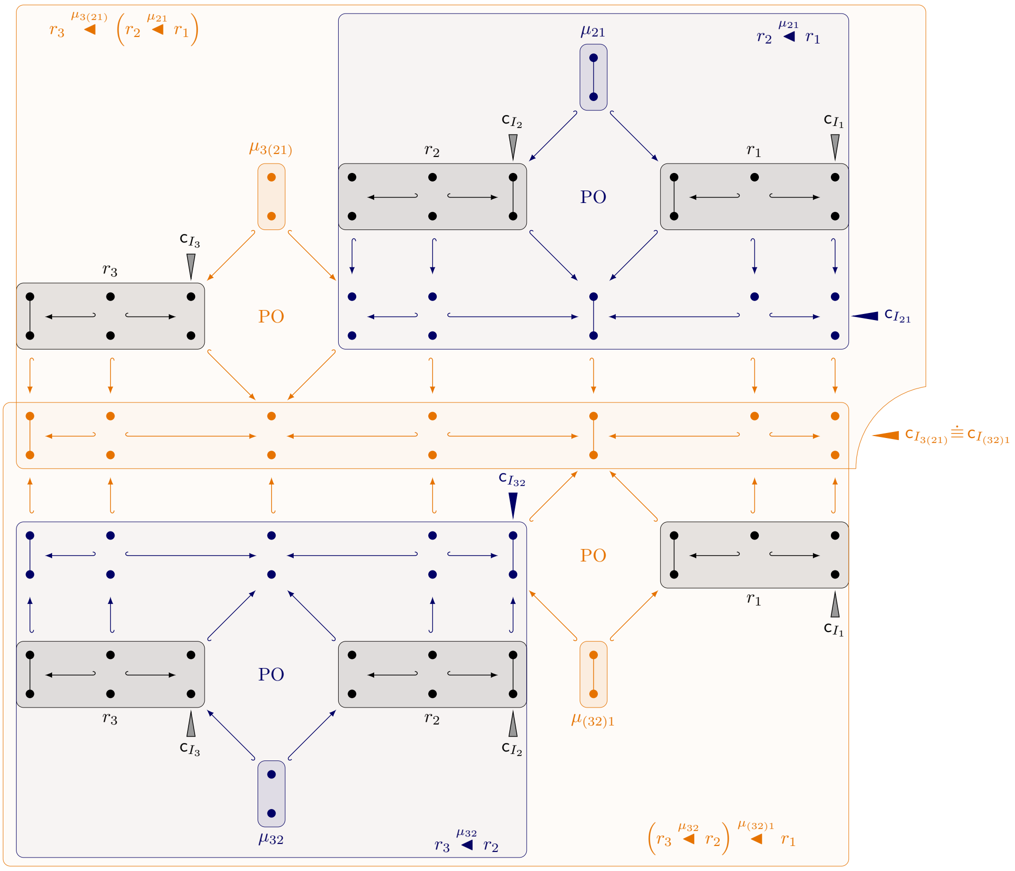

$r_3 \overset{\mu_{3(21)}}{\blacktriangleleft} \left(r_2 \overset{\mu_{21}}{\blacktriangleleft} r_1\right)$
μ_{21}
$r_2 \overset{\mu_{21}}{\blacktriangleleft} r_1$
c_{I_2}
c_{I_1}
$\mu_{3(21)}$
r_2
PO
r_1
c_{I_3}
r_3
PO
$\mathsf{c}_{I_{21}}$
$\mathsf{c}_{I_{3(21)}} \doteqdot \mathsf{c}_{I_{(32)1}}$
$\mathsf{c}_{I_{32}}$
PO
r_1
c_{I_1}
PO
r_3
r_2
$\mu_{(32)1}$
c_{I_3}
c_{I_2}
μ_{32}
$r_3 \overset{\mu_{32}}{\blacktriangleleft} r_2$
$\left(r_3 \overset{\mu_{32}}{\blacktriangleleft} r_2\right) \overset{\mu_{(32)1}}{\blacktriangleleft} r_1$

Figure 2: Illustration of the notion of associativity in a sequential composition of three rules with conditions. The squares explicitly marked with PO are pushouts of the admissible matches of rules; all other squares are obtained according to the semantics of DPO rule compositions (and are thus also pushout squares). The top portion of the figure illustrates the order of composition $r_2 \overset{\mu_{21}}{\blacktriangleleft} r_1$ followed by a composition with r_3 along the admissible match $\mu_{3(21)}$. The bottom portion depicts the composition $r_3 \overset{\mu_{32}}{\blacktriangleleft} r_2$ (with μ_{32} computed via taking pullback as described in detail in Theorem 6), precomposed with r_1 along the induced match $\mu_{(32)1}$. Conversely, one could start from providing explicitly the pair of admissible matches $(\mu_{32}, \mu_{(32)1})$ and compute from this data the pair of admissible matches $(\mu_{21}, \mu_{3(21)})$. Here, the composite rules' application conditions are indicated at the input interfaces of the composite rules. The second part of the statement of associativity of rule compositions entails that in both orders of pair-wise sequential compositions along the matches as specified, the resulting "triple composites" are isomorphic on their plain rule parts and have equivalent application conditions (up to satisfiability of admissible matches, i.e. in the sense of $\doteqdot$).

In DPO-type rewriting, the deletion rule is not applicable along the match presented, since the square $(*)$ is not constructible as a pushout complement. In SqPO-type rewriting however, since the square $(*)$ is constructible as a final pullback complement as presented, the deletion rule is applicable, resulting in a graph with just a single vertex. This example demonstrates the distinguishing feature of SqPO-type rewriting over DPO-rewriting, in that the former admits "deletion in unknown context" (here the implicit deletion of the edge via application of the vertex deletion rule).

It should be noted that one of the additional distinctive features of SqPO-rewriting (see [26]) consists in the possibility to faithfully model *cloning* of structures via considering *non-linear rules* (i.e. rules based upon non-monic spans). While an interesting topic in its own right, we found that our proofs for key properties such as the concurrency and associativity theorems in the forms presented here would not easily carry over to this more general setting, which is why we did not consider the case of non-linear rules in any further detail.

5.2 The transport construction in the SqPO-type setting

The following theorem demonstrates that the construction Trans as introduced in Definition 12 is precisely the construction needed in order to implement "transporting" conditions over linear rules also in the SqPO-rewriting setting. This quintessential result appears to be new.

Theorem 7 (Transport construction in SqPO rewriting) *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category satisfying Assumption 2. Then the* transport construction Trans *satisfies the following property: for every linear rule* $r \equiv (O \overset{o}{\hookleftarrow} K \overset{i}{\hookrightarrow} I) \in \mathsf{Lin}(\mathbf{C})$*, for every SqPO-admissible match* $(m : I \to X) \in \mathsf{M}^{sq}_{r}(X)$ *of* r *into an object* $X \in \mathsf{obj}(C)$ *and for every application condition* c_O *over* O*, in the commutative diagram below*

$$\begin{tikzcd}
\mathsf{c}_O \arrow[r, phantom, "\blacktriangleright"] & O \arrow[d, "m^*"'] \arrow[rd, phantom, "\mathsf{PO}"] & K \arrow[l, hook', "o"'] \arrow[r, hook, "i"] \arrow[d] \arrow[rd, phantom, "\mathsf{FPC}"] & I \arrow[d, "m"] & \mathsf{Trans}(r, \mathsf{c}_O) \arrow[l, phantom, "\blacktriangleleft"] \\
& r_m(X) & K' \arrow[l] \arrow[r] & X &
\end{tikzcd}, \tag{84}$$

it holds that

$$m^* \vDash \mathsf{c}_O \quad \Leftrightarrow \quad m \vDash \mathsf{Trans}(r, \mathsf{c}_O)\,. \tag{85}$$

Proof: Due to the recursive nature of the definition of Trans (Definition 12), it suffices to verify the claim on conditions of the form $\mathsf{c}_O = \exists(b^* : O \hookrightarrow B, \mathsf{c}_B)$ (for $b^* \in \mathcal{M}$).

"⇒" direction: Suppose that we are given an SqPO-type rewriting step and a condition of the form $\mathsf{c}_O = \exists(b^* : O \hookrightarrow B, \mathsf{c}_B)$, which by definition of satisfiability of conditions according to Definition 9 furnishes a morphism $(n^* : B \hookrightarrow r_m(X)) \in \mathcal{M}$ such that $n^* \circ b^* = m^*$:

$$\begin{tikzcd}
& & O \arrow[ld, "b^*"'] \arrow[dd, "m^*"] \arrow[rdd, phantom, "\mathsf{PO}"] & K \arrow[l] \arrow[r] \arrow[dd] \arrow[rdd, phantom, "\mathsf{FPC}"] & I \arrow[dd, "m"] \\
\mathsf{c}_B \arrow[r, phantom, "\blacktriangleright"] & B \arrow[rd, "n^*"'] & & & \\
& & r_m(X) & \overline{K} \arrow[l] \arrow[r] & X
\end{tikzcd} \tag{86}$$

We then construct the commutative diagram below,

$$\begin{tikzcd}
 & & O \arrow[dl, "b^*"'] \arrow[dd, "m^*"'] \arrow[dr, phantom, "(2)"] & & K \arrow[ll] \arrow[rr] \arrow[dl] \arrow[dd] \arrow[dr, phantom, "(3)"] & & I \arrow[dl, "b"'] \arrow[dd, "m"] & \\
\mathsf{c}_B \arrow[r, phantom, "\blacktriangleright"] & B \arrow[dr, "n^*"'] & & K' \arrow[ll] \arrow[rr] \arrow[dr] & & C \arrow[dr, "n"'] & & \mathsf{Trans}(r',\mathsf{c}_B) \arrow[ll, phantom, "\blacktriangleleft"] \\
 & & r_m(X) \arrow[ur, phantom, "(1)"] & & \overline{K} \arrow[ll] \arrow[rr] \arrow[ur, phantom, "(4)"] & & X &
\end{tikzcd}\;, \tag{87}$$

with the following individual steps taken:

(i) The square (1) is formed by taking pullback, which by the universal property of pullbacks furnishes a morphism $K \to K'$. By stability of $\mathcal{M}$-morphisms under pullback, the morphisms of the induced span are in $\mathcal{M}$. Since thus in particular $K' \to \overline{K}$ is in $\mathcal{M}$, and since by stability of $\mathcal{M}$-morphisms under pullback (and thus under FPC) also $(K \to \overline{K}) \in \mathcal{M}$, we conclude by $\mathcal{M}$-morphism decomposition that $K \to K'$ is in $\mathcal{M}$. Moreover, by virtue of pushout-pullback decomposition (Lemma 12(iii)c), the squares (1) and (2) are both pushouts.

(ii) The square (3) is formed by taking pushout. The universal property of pushouts provides a morphism $n : C \to X$ such that $n \circ b = m$. Then invoking vertical FPC-pushout decomposition (Lemma 12(iii)h), the resulting square (4) is an FPC and $n \in \mathcal{M}$.

If $\mathsf{c}_B \equiv \mathsf{true}$, we have thus demonstrated by virtue of the definition of satisfiability of conditions that the $\mathcal{M}$-morphism n satisfies the condition $\exists(b : I \hookrightarrow C, \mathsf{Trans}(r', \mathsf{true}))$, since by definition of Trans we have that $\mathsf{Trans}(r', \mathsf{true}) \equiv \mathsf{true}$. If c_B is itself nested, we proceed by induction in the evident fashion, i.e. repeat the preceding argument for c_B and $r' := (B \leftarrow K' \to C)$, thus the claim of the "⇒" part of the theorem statement follows.

"⇐" direction Suppose that we are given the following part of data presented in diagram (87):

(i) The squares $(1) + (2)$ (a pushout), (2) (a pushout), $(3) + (4)$ (an FPC) and (3) (a pushout).

(ii) An arrow $n \in \mathcal{M}$ that satisfies the condition $\mathsf{Trans}(r', \mathsf{c}_B)$, and with $n \circ b = m$.

What is somewhat hidden in this set of data is the fact that the square $\square(K, K', X, I)$ is a pullback (where the $\mathcal{M}$-morphism $K' \to X$ is provided as the composition of the $\mathcal{M}$-morphisms $K' \to C$ and $n : C \to X$, which are by assumption part of the above data). This statement can be verified by constructing the commutative cube below left:

$$\begin{tikzcd}
 & K \arrow[rr, hook] \arrow[dl, hook] \arrow[dd, equal] & & I \arrow[dl, hook, "b"] \arrow[dd, equal] \\
K' \arrow[rr, hook] \arrow[dd, equal] & & C \arrow[dd, "n"] & \\
 & K \arrow[rr, hook] \arrow[dl, hook] & & I \arrow[dl, hook, "m"] \\
K' \arrow[rr, hook] & & X &
\end{tikzcd}
\qquad
\begin{tikzcd}
K \arrow[drr, equal] \arrow[drrr] \arrow[ddd] & & & \\
 & & K \arrow[r] \arrow[d] \arrow[dr, phantom, "\mathsf{FPC}"] & I \arrow[d, "m"] \\
 & & \overline{K} \arrow[r] & X \\
K' \arrow[urr, dotted, "\exists!"] \arrow[urrr] & & &
\end{tikzcd} \tag{88}$$

By virtue of Lemma 12(i)c, the right square is a pullback. Since the top square is a pushout and thus also a pullback, by pullback composition the square $\square(K, K', X, I)$ (i.e. the composite of the top and right squares) is indeed a pullback.

We then invoke the universal property of FPCs (compare Definition 7) in the form presented in the right part of (88), which entails that there exists a morphism $K' \to \overline{K}$. Then by $\mathcal{M}$-morphism decomposition, as $K' \rightarrow X$ and $\overline{K} \rightarrow X$ are in $\mathcal{M}$, so is the morphism $K' \rightarrow \overline{K}$. Consequently, vertical FPC-pushout decomposition (Lemma 12(iii)h) implies that the square (4) in (87) is an FPC, while due to pushout-pushout decomposition (Lemma 12(iii)b) the square (1) is verified to be a pushout. The latter result entails in particular, by virtue of stability of $\mathcal{M}$-morphisms under pushout, that the induced morphism n^* is in $\mathcal{M}$.

In summary, if $\mathsf{c}_B \equiv \mathsf{true}$, we have verified that $n^* \vDash \mathsf{c}_B$, and thus that $m^* \vDash \exists(b^* : O \to B, \mathsf{c}_B)$. If c_B is itself nested, we can prove the statement inductively in the evident fashion. □

The detailed structure of the above proof allows us to clarify that, while the Trans construction from the DPO rewriting setting carries over to the SqPO setting seemingly verbatim, the precise reasons for why it indeed furnishes an operation of transport of conditions in the desired sense in the two settings are dependent on the semantics (i.e. in particular why satisfaction of conditions is "transported" against the direction of SqPO linear rules as detailed above). Moreover, since there does not exist a construction that would permit to transport conditions "with" the direction of the linear rules (i.e. in the direction of rule applications) in SqPO rewriting[8], the only degree of freedom in describing linear rules with application conditions in SqPO semantics is to transport any conditions a rule might carry on its output to its input, motivating the following definition:

Definition 18 (Standard form for SqPO-type linear rules with application conditions and for admissble matches) . *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category satisfying Assumption 2. Let* $\overline{\mathsf{Lin}}(\mathbf{C})$ *denote the* set of linear rules with application conditions in standard form *as introduced in Definition 13, whence elements of* $R \in \overline{\mathsf{Lin}}(\mathbf{C})$ *are of the form*

$$R = (r, \mathsf{c}_I)\,, \quad r = \left(O \xleftarrow{o} K \xrightarrow{i} I\right) \in \mathsf{Lin}(\mathbf{C})\,. \tag{89}$$

We then introduce the notion of SqPO-admissible matches *for applications of rules with application conditions to objects under SqPO-type semantics as follows: let* $X \in \mathsf{obj}(\mathbf{C})$ *be an object,* $R \in \overline{\mathsf{Lin}}(\mathbf{C})$ *as above a rule with application conditions, and* $(m : I \hookrightarrow X)$ *an element of* $\mathcal{M}$*. Since according to Assumption 2 FPCs of arbitrary pairs of composable* $\mathcal{M}$*-morphisms exist, the diagram below is always constructible,*

$$\begin{tikzcd}
O \arrow[d, "m^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d] \arrow[l, phantom, shift right=20, "\mathsf{PO}"] \arrow[r, phantom, shift right=20, "\mathsf{FPC}"] & I \arrow[d, "m"] \arrow[r, phantom, "\blacktriangleleft"] & \mathsf{c}_I \\
R_m(X) & K' \arrow[l] \arrow[r] & X \arrow[ur, phantom, "\vDash"] &
\end{tikzcd}\quad, \tag{90}$$

the SqPO-admissibility *of* m *hinges solely on whether the match satisfies the condition* c_I *of the rule* R*. We thus define the* set of SqPO-admissible matches *for the application of the rule* R *to the object* X *as*

$$\mathsf{M}^{sq}_R(X) := \{(m : I \to X) \in \mathcal{M} \mid m \vDash \mathsf{c}_I\}\,. \tag{91}$$

A beneficial side-effect of the former observation is the following result, most statements of which carry over from the DPO-rewriting setting for the Trans construction:

Theorem 8 (Properties of Trans in SqPO-rewriting) *Given an* $\mathcal{M}$*-adhesive category* $\mathbf{C}$ *satisfying Assumption 2, the transport construction* Trans *has the following properties:*

(i) Units for Trans: *Let* $X \in \mathsf{obj}(\mathbf{C})$ *be an arbitrary object and* c_X *a condition over* X*. Then with* $r_{id_X} \equiv \left(X \xleftarrow{id_X} X \xrightarrow{id_X} X\right) \in \mathsf{Lin}(\mathbf{C})$ *the "identity rule on* X*", we find that*

$$\mathsf{Trans}(r_{id_X}, \mathsf{c}_X) \equiv \mathsf{c}_X\,. \tag{92}$$

[8] Consider for example the case of a rule $R = (r, \mathsf{c}_I)$ with "plain" rule $r = (\bullet_1 \leftarrow \bullet_1 \to \bullet_1\ \bullet_2)$ and application condition $\mathsf{c}_I = \nexists(\bullet_1\ \bullet_2 \hookrightarrow \bullet_1{-}\bullet_2)$. Applying the "plain" rule to a graph with two vertices linked by an arbitrary number of edges would result in a single-vertex graph under SqPO-semantics regardless of the number of edges (which are all implicitly deleted), yet only the case without edges permits admissible matches of R, which provides a counter-example to the hypothesis that one could formulate a post-condition "transport-equivalent" to c_I.

(ii) Compositionality of Trans: *Given two composable spans of $\mathcal{M}$-morphisms*

$$r \equiv \left(C \xleftarrow{b} B \xrightarrow{a} A\right) \quad \textit{and} \quad s \equiv \left(E \xleftarrow{d} D \xrightarrow{c} C\right)\,,$$

we find that

$$\mathsf{Trans}(r, \mathsf{Trans}(s, \mathsf{c}_E)) \equiv \mathsf{Trans}(s \circ r, \mathsf{c}_E)\,. \tag{93}$$

(iii) Compatibility of Shift and Trans:

$$\begin{tikzcd}
\mathsf{c}_O \arrow[d, phantom, "\blacktriangledown"] & & \\
O \arrow[d, "p^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, "\overline{p}"] \arrow[l, phantom, shift right=6, "\mathsf{PO}"] & I \arrow[d, "p"] \arrow[l, phantom, shift right=6, "\mathsf{FPC}"] \\
O' & K' \arrow[l, "o'"] \arrow[r, "i'"'] & I'
\end{tikzcd}\,, \tag{94}$$

letting $r = \left(O \xleftarrow{o} K \xrightarrow{i} I\right)$ and $r' = \left(O' \xleftarrow{o'} K' \xrightarrow{i'} I'\right)$, we have that for all objects X and for all admissible matches $n \in \mathsf{M}^{sq}_{r'}(X)$ of r' into X,

$$n \vDash \mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O)) \Leftrightarrow n \vDash \mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O))\,, \tag{95}$$

which we write more compactly as[9]

$$\mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O)) \;\equiv\; \mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O))\,. \tag{96}$$

Proof: The proofs of the first two statements take precisely the same shape as the corresponding proofs of Lemma 7 (units for Trans) and Lemma 8 (compositionality of Trans) for the DPO-rewriting setting, as they are independent of the underlying type of rewriting. It remains to prove the third statement, which follows by suitably adapting the proof strategy of Lemma 9 (compatibility of Shift and Trans in DPO-rewriting). More precisely, construct the commutative diagram below, where the top part is inserted from the assumption of the lemma, and where the bottom part constitutes an SqPO-type rewrite step of applying the rule r' to X along the match n:

$$\begin{tikzcd}
\mathsf{c}_O \arrow[d, phantom, "\blacktriangledown"] & & \mathsf{Trans}(r, \mathsf{c}_O) \arrow[d, phantom, "\blacktriangledown"] & \\
O \arrow[d, "p^*"'] & K \arrow[l, "o"'] \arrow[r, "i"] \arrow[d, "\overline{p}"'] \arrow[l, phantom, shift right=6, "\mathsf{PO}"] & I \arrow[d, "p"] \arrow[l, phantom, shift right=6, "\mathsf{FPC}"] & \\
O' \arrow[d, "n^*"'] & K' \arrow[l, "o'"] \arrow[r, "i'"] \arrow[d, "\overline{n}"'] \arrow[l, phantom, shift right=6, "\mathsf{PO}"] & I' \arrow[d, "n"] \arrow[l, phantom, shift right=6, "\mathsf{FPC}"] & \mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O)) \arrow[l, phantom, "\blacktriangleleft"] \\
r'_n(X) & K' \arrow[l] \arrow[r] & X &
\end{tikzcd} \tag{97}$$

From hereon, the proof structure is fully analogous to the DPO rewriting case: since

$$n \vDash (\mathsf{Shift}(p, \mathsf{Trans}(r, \mathsf{c}_O))\,,$$

we find that $m = n \circ p \vDash \mathsf{Trans}(r, \mathsf{c}_O)$. Since the top and bottom left squares compose into a pushout and the top and bottom right squares into an FPC, $m \vDash \mathsf{Trans}(r, \mathsf{c}_O)$ implies that $m^* = n^* \circ p^* \vDash \mathsf{c}_O$. Since $m^* = n^* \circ p^*$, $m^* \vDash \mathsf{c}_O$ implies that $n^* \vDash \mathsf{Shift}(p^*, \mathsf{c}_O)$, and since the bottom left and right squares are of the form of an SqPO-type rewriting step, we find that indeed $n \vDash \mathsf{Trans}(r', \mathsf{Shift}(p^*, \mathsf{c}_O)$. The proof of the converse direction follows by reversing the order of the preceding steps of the proof. □

[9] Note that unlike in the DPO rewriting case, since by Assumption 2 FPCs of arbitrary composable pairs of $\mathcal{M}$-morphisms exist, it is indeed the case that *any* $\mathcal{M}$-morphism $(n : I' \to X) \in \mathcal{M}$ is an admissible match for the linear rule without application conditions r'; consequently, the equivalence takes precisely the form of standard equivalence of application conditions without a constraint of admissibility.

5.3 SqPO-type concurrent composition of rules with conditions

The second central definition for our rewriting framework is the following notion of concurrent composition, which is an extension of a construction first introduced in [8] to the setting of rules with conditions.

Definition 19 (SqPO-type concurrent composition) *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category satisfying Assumption 2. Let* $R_j \equiv (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ *be two linear rules with application conditions* ($j = 1, 2$)*, and let*

$$\mu_{21} \equiv \left(I_2 \xleftarrow{m_2} M_{21} \xrightarrow{m_1} O_1\right)$$

be a span of $\mathcal{M}$*-morphisms (i.e.* $m_1, m_2 \in \mathcal{M}$*). If the diagram below is constructible (if the pushout complement marked* POC *exists),*

$$\begin{tikzcd}
 & & \mathsf{c}_{I_2} \arrow[d] & & & & \mathsf{c}_{I_1} \arrow[d] & \\
O_2 \arrow[d, dotted] \arrow[dr, phantom, "\mathsf{PO}"] & K_2 \arrow[l, "o_2"'] \arrow[r, "i_2"] \arrow[d, dotted] \arrow[dr, phantom, "\mathsf{FPC}"] & I_2 \arrow[dr, bend right, "m_2'"'] & M_{21} \arrow[l, "m_2"'] \arrow[r, "m_1"] \arrow[d, phantom, "\mathsf{PO}"] & O_1 \arrow[dl, bend left, "m_1'"] \arrow[dr, phantom, "\mathsf{POC}"] & K_1 \arrow[l, "o_1"'] \arrow[r, "i_1"] \arrow[d, dotted] \arrow[dr, phantom, "\mathsf{PO}"] & I_1 \arrow[d, dotted, "p_1"] & \\
O_{21} & K_2' \arrow[l, dotted] \arrow[rr, dashed] \arrow[d, phantom, "="] & & N_{21} \arrow[d, phantom, "\mathsf{PB}"] & & K_1' \arrow[ll, dashed] \arrow[r, dotted] \arrow[d, phantom, "="] & I_{21} & \mathsf{c}_{I_{21}} \arrow[l] \\
 & {} & & K_{21} \arrow[ull, dashed] \arrow[urr, dashed] \arrow[ulll, bend left, "o_{21}"] \arrow[urrr, bend right, "i_{21}"'] & & {} & &
\end{tikzcd} \tag{98}$$

where (in close analogy to the DPO-type rewriting setting)

$$\mathsf{c}_{I_{21}} := \mathsf{Shift}(p_1, \mathsf{c}_{I_1}) \bigwedge \mathsf{Trans}\left(N_{21} \leftarrow K_1' \rightarrow I_{21}, \mathsf{Shift}(m_2', \mathsf{c}_{I_2})\right) \,, \tag{99}$$

and if $\mathsf{c}_{I_{21}} \neq \mathsf{false}$*, then we call* μ_{21} *an* SqPO-admissible match *for the rules with conditions* R_2 *into* R_1*, denoted*

$$\mu_{21} \in \mathsf{M}^{sq}_{R_2}(R_1)\,.$$

In this case, we introduce the notation $R_2 \overset{\mu_{21}}{\triangleleft} R_1$ *to denote the composite,*

$$R_2 \overset{\mu_{21}}{\triangleleft} R_1 := \left(\left(O_{21} \xleftarrow{o_{21}} K_{21} \xrightarrow{i_{21}} I_{21}\right), \mathsf{c}_{I_{21}}\right)\,. \tag{100}$$

As already noted in [8] for the case of SqPO-type rewriting for rules without conditions, it might appear surprising at first sight that there is an asymmetry in the above definition (in that the left part of the diagram consists of an FPC and a pushout, while the right part is formed by a pushout complement and a pushout, respectively), but the precise reason for this definition will become apparent when considering the concurrency theorem for SqPO-type rules with conditions in the following subsection.

5.4 SqPO-type concurrency theorem for rules with conditions

To the best of our knowledge, the following key result is the first of its kind in the SqPO-rewriting setting:

Theorem 9 (SqPO-type Concurrency Theorem, extended from [8], Thm. 2.9) *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category satisfying Assumption 2, let* $R_j \equiv (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ ($j = 1, 2$) *be two linear rules with application conditions, and let* $X_0 \in ob(\mathbf{C})$ *be an object.*

- ***Synthesis:*** *Given a two-step sequence of SqPO derivations*

$$X_2 \xLeftarrow[R_2, M_2]{SqPO} X_1 \xLeftarrow[R_1, M_1]{SqPO} X_0\,,$$

with $X_1 := r_{1_{M_1}}(X_0)$ *and* $X_2 := r_{2_{M_2}}(X_1)$*, there exists a SqPO-composite rule* $R_{21} = R_2 \overset{\mu_{21}}{\triangleleft} R_1$ *for a unique* $\mu_{21} \in \mathbf{M}^{sq}_{R_2}(R_1)$ *as well as an SqPO-admissible match* $n \in \mathsf{M}^{sq}_{R}(X)$ *(that is unique up to isomorphisms) such that*

$$R_{21_n}(X_0) \xLeftarrow[R_{21}, n]{SqPO} X_0 \qquad \text{and} \qquad R_{21_n}(X_0) \cong X_2\,.$$

- ***Analysis:*** *Given an SqPO-admissible match* $\mu_{21} \in \mathbf{M}^{sq}_{R_2}(R_1)$ *of* R_2 *into* R_1 *and an SqPO-admissible match* $n \in \mathsf{M}^{sq}_{R_{21}}(X)$ *of the SqPO-composite* $R_{21} = R_2 \overset{\mu_{21}}{\triangleleft} R_1$ *into* X_0*, there exists a pair of SqPO-admissible matches* $m_1 \in \mathsf{M}^{sq}_{R_1}(X_0)$ *and* $m_2 \in \mathsf{M}^{sq}_{R_2}(X_1)$ *(with* $X_1 := R_{1_{m_1}}(X_0)$*, and unique up to isomorphisms) such that*

$$X_2 \xLeftarrow[R_2,m_2]{SqPO} X_1 \xLeftarrow[R_1,m_1]{SqPO} X_0 \qquad \textit{and} \qquad X_2 \cong R_{21_n}(X_0)\,.$$

Proof: For the part of the proof pertaining to the concurrency of SqPO-type rules without application conditions, we will follow the strategy presented in [8] (where slightly stronger conditions than the ones required according to Assumption 2 were made, i.e. in [8] $\mathbf{C}$ was assumed to be adhesive, whence the re-derivation here in the $\mathcal{M}$-adhesive setting).

"Synthesis" part of the proof: Suppose we are given rules with application conditions $R_1, R_2 \in \overline{\mathsf{Lin}}(\mathbf{C})$ and SqPO-admissible matches $m_1 \in \mathsf{M}^{sq}_{R_1}(X_0)$ and $m_2 \in \mathsf{M}^{sq}_{R_2}(X_1)$, with $X_1 = R_{1_{m_1}}(X_0)$. This data is encoded in the blue part of the diagram in (71). Let us begin by constructing the orange parts of (71) as follows: take the pullback $M_{21} = \mathsf{PB}(I_2 \rightarrow X_1 \leftarrow O_1)$, and then the pushout $N_{21} = \mathsf{PO}(I_2 \leftarrow M_{21} \rightarrow O_1)$; by the universal property of pushouts, there exists a morphism $N_{21} \rightarrow X_1$, which is in $\mathcal{M}$ since $\mathbf{C}$ is assumed to possess $\mathcal{M}$-effective unions according to Assumption 2. Next, form the pullbacks $K'_i = \mathsf{PB}(\overline{K}_i \rightarrow X_1 \leftarrow N_{21})$ (for $i = 1, 2$), which furnishes morphisms $K_i \rightarrow K'_i$ (for $i = 1, 2$) that are in $\mathcal{M}$ due to the decomposition property of $\mathcal{M}$-morphism. By virtue of vertical FPC-pullback decomposition (Lemma 12(iii)g), the second from the left bottom and top squares in the back of (71) are FPCs, while via pushout-pullback decomposition (Lemma 12(iii)c) the second from the right top and bottom squares in the back of (71) are pushouts. Let $O_{21} = \mathsf{PO}(O_2 \leftarrow K_2 \rightarrow K'_2)$, which by the universal property of pushouts furnishes a morphism $O_{21} \rightarrow X_2$; by pushout-pushout decomposition (Lemma 12(iii)b), the leftmost bottom square in the back of (71) is a pushout, which also entails by stability of $\mathcal{M}$-morphisms under pushouts that $(O_{21} \rightarrow X_2) \in \mathcal{M}$. Analogously, let $I_{21} = \mathsf{PO}(K'_1 \leftarrow K_1 \rightarrow I_1)$, which furnishes a morphism $I_{21} \rightarrow X_0$, and via vertical FPC-pushout decomposition (Lemma 12(iii)h) that the rightmost bottom back square in (71) is an FPC and that $(I_{21} \rightarrow X_0) \in \mathcal{M}$.

At this point, note that the bottom row of squares in the back of (71) has the shape of two consecutive SqPO rewriting steps. Since $(m_1 : I_1 \rightarrow X_0) \vDash \mathsf{c}_{I_1}$ and $(m_2 : I_2 \rightarrow X_1) \vDash \mathsf{c}_{I_2}$ by assumption, we conclude that $m_{21} = (I_{21} \rightarrow X_0)$ satisfies $m_{21} \vDash \mathsf{c}_{21}$, with

$$\mathsf{c}_{21} = \mathsf{Shift}(I_1 \rightarrow I_{21}, \mathsf{c}_{I_1}) \bigwedge \mathsf{Trans}(N_{21} \leftarrow K'_1 \rightarrow I_{21}, \mathsf{Shift}(I_2 \rightarrow N_{21}, \mathsf{c}_{I_2})\,. \tag{101}$$

To complete this part of the proof, take the pullbacks

$$K_{21} = \mathsf{PB}(K'_2 \leftarrow N_{21} \rightarrow K'_1) \quad \text{and} \quad \overline{K}_{21} = \mathsf{PB}(\overline{K}_2 \leftarrow X_1 \rightarrow \overline{K}_1)\,,$$

which also induces a morphism $K_{21} \rightarrow \overline{K}_{21}$. By pullback-pullback decomposition (Lemma 12(iii)a), the induced square $\square(K_{21}, \overline{K}_{21}, \overline{K}_1, K'_1)$ (i.e. the inner right "curvy" square) is a pullback, which entails by stability of $\mathcal{M}$-morphisms under pullbacks that $K_{21} \rightarrow \overline{K}_{21}$ is in $\mathcal{M}$. Then by the $\mathcal{M}$-van Kampen property, the square $\square(K_{21}, \overline{K}_{21}, \overline{K}_2, K'_2)$ (i.e. the inner left "curvy" square) is a pushout. The composition of the latter pushout (and thus an FPC) square and of the second bottom square from the left in the back of (71) (an FPC) yields an FPC square, while the third from the left bottom back square is a pushout (and thus an FPC), whence, by horizontal FPC decomposition (Lemma 12(iii)e), we derive that the inner right "curvy" square is an FPC. Consequently, forming the front left "curvy" square as a composition of pushout squares and the front right "curvy" square as a composition of FPCs, we have in summary exhibited an SqPO rewriting step of X_0 along the rule $(O_{21} \leftarrow K_{21} \rightarrow I_{21})$ and admissible match m_{21}.

"Analysis" part of the proof: Suppose that we are given an SqPO-composite $R_{21} = R_2 \overset{\mu_{21}}{\triangleleft} R_1$ of linear rules with application conditions R_2 with R_1 along the SqPO-admissible match $\mu_{21} = (I_2 \leftarrow M_{21} \rightarrow O_1)$, and that moreover $m_{21} = (I_{21} \rightarrow X_0)$ is an SqPO-admissible match of R_{21} into X_0. Forming a SqPO-rewrite step by applying R_{21} along m_{21} to X_0 yields the orange parts of the diagram in (71). In order to prove the claim, we have to construct two SqPO-admissible matches $m_1 \in \mathsf{M}^{sq}_{R_1}(X_0)$ and $m_2 \in \mathsf{M}^{sq}_{R_1}(X_1)$, with $X_1 = R_{1_{m_1}}(X_0)$, and that $X_2 \cong R_{2_{m_2}}(X_1)$ under these assumptions.

We begin by forming the FPC $\overline{K}_1 = \mathsf{FPC}(K_1' \to I_{21} \to X_0)$, which according to Assumption 2 is guaranteed to exist and to yield two $\mathcal{M}$-morphisms $K_1' \to \overline{K}_1$ and $\overline{K}_1 \to X_0$. By the universal property of FPCs, there exists a morphism $\overline{K}_{21} \to \overline{K}_1$, which by the $\mathcal{M}$-morphism decomposition property is in $\mathcal{M}$. Horizontal FPC decomposition (Lemma 12(iii)e) entails that the square $\square(K_{21}, \overline{K}_{21}, \overline{K}_1, K_1')$ (inner right "curvy" square) is an FPC. Next, we take the pushout $X_1 = \mathsf{PO}(M_{21} \leftarrow K_1' \to \overline{K}_1)$, followed by forming the FPC

$$\overline{K}_2 = \mathsf{FPC}(K_2' \to N_{21} \to X_1)\,.$$

Since the inner right "curvy" square $\square(K_{21}, \overline{K}_{21}, \overline{K}_1, K_1')$ is an FPC and the second from the right bottom back square in (71) a pushout, the composition of these two squares is an FPC and thus a pullback. Then by the universal property of FPCs, there exists a morphism $\overline{K}_{21} \to \overline{K}_2$, which is by the $\mathcal{M}$-morphism decomposition property in $\mathcal{M}$, and by the horizontal FPC decomposition property, the left inner "curvy" square is an FPC. Noting that the square $\square(K_{21}, K_2', N_{21}, K_1')$ is by assumption a pullback, by pullback-pullback decomposition so is $\square(\overline{K}_{21}, \overline{K}_2, X_1, \overline{K}_1)$; thus by the $\mathcal{M}$-van Kampen property, the left inner "curvy" square is an FPC, noting that the square is in fact a pushout. The latter entails by the universal property of pushouts the existence of a morphism $\overline{K}_2 \to X_2$, and then by pushout-pushout decomposition that the leftmost bottom back square in (71) is a pushout (and thus by stability of $\mathcal{M}$-morphisms under pushouts that $(\overline{K}_2 \to X_2) \in \mathcal{M}$).

To complete the proof, note that the rightmost and second from right bottom back squares in (71) are an FPC and a pushout, respectively, thus $m_{21} \vDash \mathsf{c}_{I_{21}}$ (with $\mathsf{c}_{I_{21}}$ defined as in (101)) implies that the $\mathcal{M}$-morphisms

$$m_1 = (X_0 \leftarrow I_{21}) \circ (I_{21} \leftarrow I_1) \quad \text{and} \quad m_2 = (X_1 \leftarrow N_{21}) \circ (N_{21} \leftarrow I_2)$$

satisfy $m_1 \vDash \mathsf{c}_{I_1}$ and $m_2 \vDash \mathsf{c}_{I_2}$ according to the properties of the Shift and Trans constructions. It then remains to compose pushout squares and FPC squares in the top and bottom back of the diagram (71) in order to form the two SqPO rewriting steps claimed to exist by the statement of the theorem, which concludes the proof. □

5.5 Associativity of SqPO rewriting with conditions

Based upon the developments presented thus far and on the central result of [8] (the associativity theorem for SqPO rewriting without rules), we may now state another original contribution of this paper. Due to the structural similarities between the strategies of the associativity proof in the DPO- and SqPO-type cases for rules without application conditions, the following statement is almost verbatim equivalent to the corresponding DPO-type statement. More precisely, in both cases, the part of the proof needed on top of the case without conditions consists of a "diagram chase" involving the Shift and Trans constructions and their properties.

Theorem 10 (SqPO-type Associativity Theorem) *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category satisfying Assumption 2. Let* $R_j \equiv (r_j, \mathsf{c}_{I_j}) \in \overline{\mathsf{Lin}}(\mathbf{C})$ $(j = 1, 2, 3)$ *be three linear rules with application conditions. Then there exists a* bijection *between the sets of pairs of admissible matches* M_A *and* M_B *defined as*

$$\begin{aligned} M_A &:= \{(\mu_{21}, \mu_{3(21)}) | \mu_{21} \in \mathsf{M}^{sq}_{R_2}(R_1)\,,\ \mu_{3(21)} \in \mathsf{M}_{R_3}(R_{21})\} \\ M_B &:= \{(\mu_{32}, \mu_{(32)1}) | \mu_{32} \in \mathsf{M}^{sq}_{R_3}(R_2)\,,\ \mu_{(32)1} \in \mathsf{M}_{R_{32}}(R_1)\} \end{aligned} \tag{102}$$

with $R_{i,j} := (r_i \overset{\mu_{ij}}{\triangleleft} r_j, \mathsf{c}_{I_{ij}})$ *(and* $\mathsf{c}_{I_{ij}}$ *defined as in* (62)*) such that*

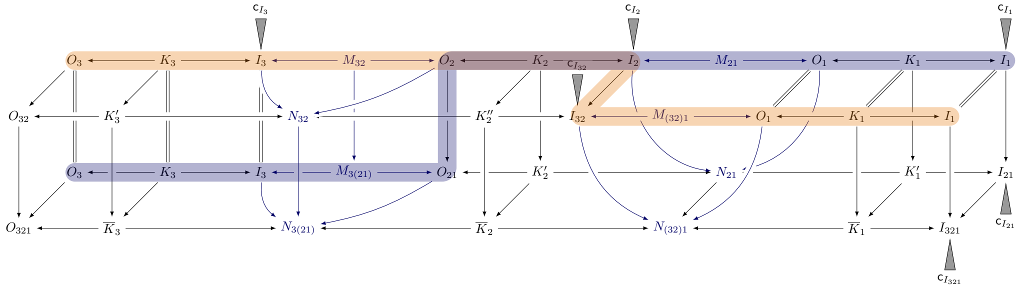
$$\forall (\mu_{21}, \mu_{3(21)}) \in M_A : \exists! (\mu_{32}, \mu_{(32)1}) \in M_B :$$

$$r_3 \overset{\mu_{3(21)}}{\triangleleft} \left(r_2 \overset{\mu_{21}}{\triangleleft} r_1\right) \cong \left(r_3 \overset{\mu_{32}}{\triangleleft} r_2\right) \overset{\mu_{(32)1}}{\triangleleft} r_1 \tag{103a}$$

$$\wedge \quad \mathsf{c}_{I_{3(21)}} \equiv \mathsf{c}_{I_{(32)1}} \tag{103b}$$

and vice versa. In this particular sense, the operation $. \dot{\triangleleft} .$ *is* **associative***.*

Proof: Referring the interested readers to [8] for the precise details of the proof for the part of the above statement pertaining to rules without application conditions, and to [12] for the generalization of the requisite technical lemmas from the adhesive to the $\mathcal{M}$-adhesive setting, suffice it here to quote the following result: the diagram (where we have also indicated the relevant conditions on rules for later convenience) is constructible starting from either of the sets of pairs of SqPO-admissible matches in (102), and thereby proves the bijection for the sets of matches of rules without conditions:



(104)

In contrast to the analogous diagram (75) of the proof of the DPO-type associativity theorem, the nature of the various squares in (104) differs at several positions. For the purpose of proving the part of the SqPO-type associativity theorem pertaining to the conditions on rules, we thus quote from [8] that

- the third and fourth squares in the front (counting from the left) are a pushout and an FPC, respectively
- the second to right and rightmost bottom squares are pushouts.

Consequently, it suffices to replace the various applications of the DPO-type compatibility of Shift and Trans (Lemma 9) in the proof of Theorem 6 with its SqPO-type variant (Theorem 8) in order to obtain a proof of the statements of the present Theorem 10 pertaining to the conditions on rules. □

Example 6 (Ex. 5 continued) *It is instructive to compare the associativity property in the SqPO-type variant directly with its counterpart in the DPO-type setting by considering once more the triple rule composition depicted in Figure 2. Note first that since pushout complements along* $\mathcal{M}$*-morphisms are also FPCs, the triple composition at the level of "plain" rules is a valid composition both in DPO- and in SqPO-semantics. However, the two semantics differ at the level of the application conditions. More precisely, since the SqPO-type* Trans *construction is in a sense "asymmetric" (in that conditions can be transported from the output to the input of a rule, but not vice versa), the "compression" as described in Definition 15 for the DPO-setting is not available in the SqPO-setting. Indeed, the application condition*

$$\mathsf{c}_{I_1} := \neg\exists\left(\begin{smallmatrix}\bullet\\ \\ \bullet\end{smallmatrix} \hookrightarrow \begin{smallmatrix}\bullet\\ |\\ \bullet\end{smallmatrix}, \mathsf{true}\right)$$

yields a non-trivial *test on candidate matches of rule* R_1 *in the SqPO-rewriting, since an application of* R_1 *at two vertices linked by an arbitrary number of vertices is possible for the "plain" rule* r_1 *in SqPO-rewriting (leading to the implicit deletion of all edges incident to the deleted vertex). Applying the SqPO-type* Trans *construction as well as* Shift*, one may compute that the application conditions* $\mathsf{c}_{I_{3(21)}}$ *and* $\mathsf{c}_{I_{(32)1}}$ *are again equivalent to each other, and are found to evaluate to* c_{I_1}*.*

Having scratched but the very surface of the intricate matter of *causality* in SqPO-type rewriting, and referring to [9, 10] for a more in-depth discussion, suffice it here to mention that it is precisely this type of causality that is of key importance to the analysis of biochemical reaction systems in the Kappa framework [20, 33, 56].

6 Conclusion and Outlook

This paper provides a self-contained account of a class of rewriting theories that possess the special property of *compositionality*. Based upon the rich theory of "traditional" Double-Pushout (DPO) [25, 36–38, 40–42, 47, 63] and Sesqui-Pushout (SqPO) [26, 55] rewriting in the setting of $\mathcal{M}$-adhesive categories [22, 39, 53], we lift our earlier results on *compositional concurrency* and *compositional associativity* as developed in [8, 11–13] to the realm of rewriting systems with conditions on objects and morphisms.

A key original contribution of the present paper is a derivation of the precise technical conditions under which compositionality in the aforementioned sense is attainable for a given rewriting theory. Concretely (cf. Section 2), it has proved essential to suitably adapt the requirements on the host categories for both DPO- and SqPO-type rewriting with application conditions on objects and morphisms [42, 45, 46, 51, 60], concluding that these categories should be $\mathcal{M}$-adhesive and possess certain additional properties such as the existence of epi-$\mathcal{M}$-factorizations and $\mathcal{M}$-effective unions as described in Assumptions 1 (DPO-case) and 2 (SqPO-case).

Our second original contribution consists of a refinement of the theory of conditions in $\mathcal{M}$-adhesive categories as presented in Section 3. For categories satisfying either of the assumptions mentioned earlier, it is possible to refine a central construction of the theory of conditions, the so-called *shift construction*, into a form that leads to new results on the interplay of rewriting rules and conditions. In particular, the *transport construction* is shown to possess a compatibility property in interaction with the *shift construction*, which is required to ensure compositionality of rewriting with conditions.

The main results of this paper are the *compositional concurrency* and *associativity* of rewriting with conditions in both the DPO (Section 4) and SqPO (Section 5) cases, demonstrated to hold under the aforementioned assumptions. Our proof strategy and techniques rely heavily on earlier developments in the setting of rewriting without conditions [8, 11] in conjunction with the aforementioned results on the properties of the refined shift and transport constructions. While admittedly a rather technical work, we believe that our results can serve as a starting point for a new generation of developments in the field of rewriting, in particular in view of static analysis tasks. Indeed, in most applications of practical interest, idealized data structures such as multigraphs must be restricted to more rigid structures (such as e.g. site graphs in the Kappa framework [20, 30]) in order to obtain tractable algorithms of sufficient predictive power. First results in the fields of stochastic mechanics of continuous-time Markov chains based upon stochastic rewriting systems [8, 11, 13, 15] hint at a great potential of the framework of compositional rewriting with conditions as presented here. We recently demonstrated that based upon the results of the present paper, it is possible to develop a faithful encoding of both bio- and organo-chemical reaction systems via typed undirected simple graphs with suitable sets of additional structure constraints [10]. Together with the respective SqPO- and DPO-type stochastic mechanics frameworks introduced in [10], a direct and in-detail comparison of the sophisticated rule-based modeling frameworks Kappa [20] and BioNetGen [50] (SqPO/biochemistry) as well as MØD [3] (DPO/organic chemistry) with the general theory of categorical rewriting over $\mathcal{M}$-adhesive categories has become possible. As a first hint at the potential of such an approach in view of the development of algorithmic implementations of rewriting-theoretical concepts, we have recently presented a first prototype of an implementation of SqPO-type compositional rewriting systems for rules with conditions based upon the present theory as well as on the Microsoft Z3 SMT-solver in [16]. Work in progress further includes the theory of *tracelets* as introduced in [9], whereby the classical concept of *derivation traces* (i.e. of sequences of applications of rewriting rules) is analyzed via exploiting the associativity theorems in order to characterize a derivation trace of length $n \geq 1$ in terms of a *minimal* derivation trace of the same length (a so-called *tracelet* of length n). As briefly touched upon in the discussion of Examples 5 and 6 (which consider the case of a "triple" composite of rules, i.e. the case $n = 3$), compositional associativity of the operations of DPO- and SqPO-type rule compositions (Theorems 6 and 10) opens the possibility to *statically generate* such minimal derivation traces *without* the need to first obtain generic derivation traces (either from simulations or from unfoldings), posing thus considerable potential in terms of analyzing *pathways* and their dynamics in chemical reaction systems.

Appendix

A Collection of technical lemmata for $\mathcal{M}$-adhesive categories

In many practical computations in the framework of $\mathcal{M}$-adhesive categories, one may take advantage of a number of technical results, some of which elementary, some of which rather specialized (such as in particular the lemmata pertaining to final pullback complements (FPCs)). For the readers' convenience, we provide here the full list of results used in the framework of this paper. The list is an adaptation of the list provided in [8] from the setting of adhesive to $\mathcal{M}$-adhesive categories. Note also that while the category-theoretical constructions of objects and morphisms via pullbacks, pushouts, pushout complements and FPCs are by definition unique only up to universal isomorphisms, we follow standard practice in simplifying our notations by employing a convention whereby e.g. the pushout of an isomorphism as in (105)(A) is denoted by "equality arrows" (rather than keeping a notation with generic labels and $\cong$ decorations on arrows).

Lemma 12 *Let* $\mathbf{C}$ *be a category.*

(i) "Single-square" lemmas (see e.g. [12], Lem. 1.7): *In any category, given commutative diagrams of the form*

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, equal] \arrow[dr, phantom, "(A)"] & B \arrow[d, equal] \\ A \arrow[r, "f"'] & B \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, equal] \arrow[d, equal] \arrow[dr, phantom, "(B)"] & A \arrow[d, "g"] \\ A \arrow[r, "g"'] & B \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "f"] \arrow[d, equal] \arrow[dr, phantom, "(C)"] & B \arrow[d, "g"] \\ A \arrow[r, "g\circ f"'] & C \end{tikzcd} \qquad \begin{tikzcd} A \arrow[r, "a"] \arrow[d, "b"'] \arrow[dr, phantom, "(D)"] & C \arrow[d, equal] \\ B \arrow[r, "c"'] & C \end{tikzcd}\,, \tag{105}$$

(a) *(A) is a pushout for arbitrary morphisms f,*

(b) *(B) is a pullback if and only if the morphism g is a monomorphism,*

(c) *(C) is a pullback for arbitrary morphisms f if g is a monomorphism,*

(d) *If $a, c \in \mathsf{mono}(\mathbf{C})$ and (D) is a pullback, then $A \cong B$.*

(ii) special $\mathcal{M}$-adhesivity corollaries *(cf. e.g. [42], Lemma 2.6): in any adhesive category,*

(a) *pushouts along $\mathcal{M}$-morphisms are also pullbacks,*

(b) *(*uniqueness of pushout complements*) given an $\mathcal{M}$-morphism $A \hookrightarrow C$ and a generic morphism $C \to D$, the respective pushout complement $A \to B \overset{b}{\hookrightarrow} D$ (if it exists) is unique up to isomorphism, and with $b \in \mathcal{M}$ (due to stability of $\mathcal{M}$-morphisms under pushouts).*

(iii) "Double-square lemmas"*: given commutative diagrams of the shapes*

$$\begin{tikzcd} A \arrow[d, "a"'] \arrow[dr, phantom, "(1)"] & B \arrow[l, "d"'] \arrow[d, dashed, "b"] \arrow[dr, phantom, "(2)"] & C \arrow[l, "e"'] \arrow[d, "c"] \\ A' & B' \arrow[l, "d'"] & C' \arrow[l, "e'"] \end{tikzcd} \qquad \begin{tikzcd} Z \arrow[d, "w"'] \arrow[dr, phantom, "(3)"] & Z' \arrow[l, "z"'] \arrow[d, "w'"] \\ Y \arrow[d, "v"'] \arrow[dr, phantom, "(4)"] & Y' \arrow[l, "y"'] \arrow[d, "v'"] \\ X & X' \arrow[l, "x"] \end{tikzcd} \tag{106}$$

then in any category $\mathbf{C}$ *(cf. e.g. [53]):*

(a) Pullback-pullback (de-)composition*: If* (1) *is a pullback, then* (1) + (2) *is a pullback if and only if* (2) *is a pullback.*

(b) Pushout-pushout (de-)composition*: If* (2) *is a pushout, then* (1) + (2) *is a pushout if and only if* (1) *is a pushout.*

If the category is $\mathcal{M}$-adhesive:

(c) pushout-pullback decomposition *([42], Lemma 2.6): If* (1) + (2) *is a pushout,* (1) *is a pullback, and if* $d' \in \mathcal{M}$ *and* ($c \in \mathcal{M}$ *or* $e \in \mathcal{M}$)*, then* (1) *and* (2) *are both pushouts (and thus also pullbacks).*

(d) pullback-pushout decomposition *([45], Lem. B.2): if* (1) + (2) *is a pullback,* (2) *a pushout,* (1) *commutes and* $a \in \mathcal{M}$*, then* (1) *is a pullback.*

(e) Horizontal FPC (de-)composition *(cf. [26], Lem. 2 and Lem. 3, compare [55], Prop. 36):*[10] *If* (1) *is an FPC (i.e. if* (d', b) *is FPC of* (a, d)*), then* (1) + (2) *is an FPC if and only if* (2) *is an FPC.*

(f) Vertical FPC (de-)composition *(ibid): if* (3) *is an FPC (i.e. if* $(v.w')$ *is FPC of* (w, z)*), then*

   i. *if* (4) *is an FPC (i.e. if* (x, v') *is FPC of* (v, y)*), then* (3) + (4) *is an FPC (i.e.* $(x, v' \circ w')$ *is FPC of* $(v \circ w, z)$*);*

   ii. *if* (3) + (4) *is an FPC (i.e. if* $(x, v' \circ w')$ *is FPC of* $(v \circ w, z)$*), and if* (4) *is a pullback, then* (4) *is an FPC (i.e.* (x, v') *is FPC of* (v, y)*).*

(g) Vertical FPC-pullback decomposition *(compare [55], Lem. 38): If* $v \in \mathcal{M}$*, if* (4) *is a pullback and if* (3) + (4) *is an FPC (i.e. if* $(x, v' \circ w')$ *is FPC of* $(v \circ w, z)$*), then* (3) *and* (4) *are FPCs.*

If the category is $\mathcal{M}$*-adhesive and in addition possesses an epi-*$\mathcal{M}$*-factorization and* $\mathcal{M}$*-effective unions:*

(h) Vertical FPC-pushout decomposition*: If all morphisms of the squares* (3) *and* (4) *except* v *are in* $\mathcal{M}$*, if* $v \circ w \in \mathcal{M}$*, if* (3) *is a pushout and if* (3) + (4) *is an FPC (i.e. if* $(x, v' \circ w')$ *is FPC of* $(v \circ w, z)$*), then* (4) *is an FPC and* $v \in \mathcal{M}$*.*

Proof: *Proofs of all but the very last statement are found in the references quoted in each statement. It thus remains to prove our novel vertical FPC-pushout decomposition result (in the setting of* $\mathcal{M}$*-adhesive categories). To this end, we first invoke pullback-pushout decomposition (Lemma 12((iii)d)): since* (3) + (4) *is an FPC and thus in particular a pullback, since* (3) *is a pushout,* (4) *commutes, and finally since* $x \in \mathcal{M}$*, we find that* (4) *is a pullback.*

In order to demonstrate that $v \in \mathcal{M}$*, construct the commutative cube below left:*

$$
\begin{tikzcd}
 & Z \arrow[dl, hook, "w"'] \arrow[dd, equal] & & Z' \arrow[ll, hook', "z"'] \arrow[dl, hook, "w'"'] \arrow[dd, equal] \\
Y \arrow[dd, "v"'] & & Y' \arrow[ll, hook', "y"'] \arrow[dd, hook, "v'"'] & \\
 & Z \arrow[dl, hook] & & Z' \arrow[ll, hook', "z"'] \arrow[dl, hook] \\
X & & X' \arrow[ll, hook', "x"'] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & Z \arrow[d, hook, "w"'] \arrow[dl, hook, bend right, "v \circ w"'] & Z' \arrow[l, hook', "z"'] \arrow[d, hook, "w'"] \arrow[dl, phantom, "(3)"] \\
 & Y \arrow[dl, "v"'] & Y' \arrow[l, hook', "y"] \arrow[dll, hook, bend left, "v \circ y"] \\
X & &
\end{tikzcd}
\tag{107}
$$

Since the bottom square is the FPC (and thus pullback) (3) + (4)*, and since the right square is a pullback via Lemma 12((i)c) (because* $v' \in \mathcal{M} \subset \mathsf{mono}(\mathbf{C})$*), by pullback composition the square* $\square(Z', Z, X, Y')$ *(the right plus the bottom square) is a pullback. Thus assembling the commutative diagram as shown above right, since by assumption* (3) *is a pushout and all arrows except* v *are in* $\mathcal{M}$*, invoking the property of* $\mathcal{M}$*-effective unions (Definition 5) allows us to prove that also* $v \in \mathcal{M}$*. Finally, by applying vertical FPC-pullback decomposition, we conclude that* (4) *is an FPC.* □

[10] It is worthwhile emphasizing that in these FPC-related lemmas, the "orientation" of the diagrams plays an important role. Moreover, the precise identity of the pair of morphisms that plays the role of the final pullback complement in a given square can be inferred from the "orientation" specified in the condition part of each statement.

B Closure of $\mathcal{M}$ under isomorphisms

As mentioned in the main text, the definition of $\mathcal{M}$-adhesive categories as given in Definition 1 is well known to be overcomplete. With the $\mathcal{M}$-decomposition property derivable directly via stability of $\mathcal{M}$-morphisms under pullbacks, we present below a derivation of the fact that $\mathcal{M}$ contains all isomorphisms.

(i) Given an arbitrary $\mathcal{M}$-morphism $(m : A \hookrightarrow B) \in \mathcal{M}$, since $m = m \circ id_A$, by the decomposition property of $\mathcal{M}$-morphisms (Definition $1(i)(a)$), we find that $id_A \in \mathcal{M}$, for every $A \in \mathsf{obj}(\mathbf{C})$.

(ii) Let $(\phi : A \to B) \in \mathsf{iso}(\mathbf{C})$ be an isomorphism, and take the pullback of the cospan (ϕ, id_A), resulting in a span (id_A, ϕ^{-1}). Then by stability of $\mathcal{M}$-morphisms under pullback (Definition $1(iii)$), we conclude that $\phi^{-1} \in \mathcal{M}$ (i.e. by $\mathcal{M}$-decomposition also $\phi \in \mathcal{M}$).

C Some useful consequences of epi-$\mathcal{M}$-factorizations

Lemma 13 *Let $\mathbf{C}$ be an $\mathcal{M}$-adhesive category satisfying Assumption 1. Then given a commutative diagram as below left where $a, b \in \mathcal{M}$ and c is an arbitrary morphism,*

$$\begin{tikzcd}
A \arrow[d, "c"'] & B \arrow[l, hook', "a"'] \arrow[d, equal] \\
C & B \arrow[l, hook', "b"]
\end{tikzcd}
\qquad
\begin{tikzcd}
A \arrow[d, two heads, "e"] \arrow[dd, bend right=50, "c"'] & B \arrow[l, hook', "a"'] \arrow[d, dotted, "e'"] \arrow[dd, bend left=50, equal] \\
E \arrow[d, hook, "m"] \arrow[dr, phantom, "\mathsf{PB}"] & D \arrow[l, "d"'] \arrow[d, "m'"] \\
C & B \arrow[l, hook', "b"]
\end{tikzcd}
\qquad
\begin{tikzcd}
& A \arrow[dl, equal] \arrow[dd, two heads, "e"] & & B \arrow[ll, hook', "a"'] \arrow[dl, equal] \arrow[dd, dotted, "e'"] \\
A \arrow[dd, "c"'] & & B \arrow[ll, hook', "a"'] \arrow[dd, equal] & \\
& E \arrow[dl, hook', "m"'] & & D \arrow[ll, "d"'] \arrow[dl, "m'"] \\
C & & B \arrow[ll, hook', "b"'] &
\end{tikzcd}
\tag{108}$$

then if the epi-$\mathcal{M}$-factorization of the morphism c reads $c = m \circ e$, with $(m : E \hookrightarrow C) \in \mathcal{M}$ and $(e : A \to E) \in \mathsf{epi}(\mathbf{C})$, and if we form the pullback $D = \mathsf{PB}(E \hookrightarrow C \hookleftarrow B)$ (thus inducing by the universal property a morphism $e' : B \to D$), then $B \cong D$.

Proof: Note first that by virtue of stability of $\mathcal{M}$-morphisms under pullbacks and by the decomposition property of $\mathcal{M}$-morphisms, respectively, one may conclude that $d, m' \in \mathcal{M}$ and $e' \in \mathcal{M}$ (since $(id_B : B \to B) \in \mathcal{M}$). Next, form the commutative cube diagram as depicted in the right part of (108). The left square is a pullback (Lemma 12(i)c), and so is the top square (Lemma 12(i)a). Since by construction also the bottom square is a pullback, by virtue of pullback-pullback decomposition (Lemma 12(iii)a), the right square is a pullback. Thus by stability of isomorphisms under pullbacks (Lemma 12(i)d), we obtain that $D \cong B$. □

D Some useful consequences of (strict) $\mathcal{M}$-initiality

In the case that an $\mathcal{M}$-adhesive category possesses a (strict) $\mathcal{M}$-initial object, the following useful properties can be derived.

Lemma 14 *Let $\mathbf{C}$ be an $\mathcal{M}$-adhesive category with $\mathcal{M}$-initial object $\varnothing$. Then the commutative diagram of $\mathcal{M}$-morphisms below is both a pushout and a pullback:*

$$\begin{tikzcd}
A \arrow[r, hook] \arrow[d, hook] & B \arrow[d, hook] \\
A + C \arrow[r, hook] & B + C
\end{tikzcd}
.
\tag{109}$$

Proof: *Consider the following commutative diagram:*

$$\begin{tikzcd}
\varnothing \arrow[r, hook] \arrow[d, hook] \arrow[rr, hook, bend left] & A \arrow[r, hook] \arrow[d, hook] & B \arrow[d, hook] \\
C \arrow[r, hook] \arrow[rr, hook, bend right] & A+C \arrow[r, hook] & B+C
\end{tikzcd} \tag{110}$$

Since the outer square and the left square are pushouts, according to the pushout-pushout decomposition property stated in Lemma 12(iii)b, the right inner square is also a pushout (and thus a pullback). □

Lemma 15 *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category with a strict* $\mathcal{M}$*-initial object* $\varnothing$*, and consider the commutative diagrams of* $\mathcal{M}$*-morphisms below,*

$$\begin{tikzcd}
A+\varnothing \arrow[r, equal] \arrow[d, hook, "{[a,\emptyset]}"'] \arrow[dr, phantom, "(1)"] & A+\varnothing \arrow[d, hook, "{[a',\emptyset]}"] & A+B \arrow[r, hook] \arrow[d, hook, "{[a,b]}"'] \arrow[dr, phantom, "(2)"] & A+C \arrow[d, hook, "{[a,c]}"] \\
B+C \arrow[r, hook] & A+D & E+F \arrow[r, hook] & D+C
\end{tikzcd} . \tag{111}$$

If (1) *is a pullback, then* $A \cong B$*, and if* (2) *is a pullback, then* $B \cong F$*.*

Proof: Consider the following auxiliary commutative diagrams:

$$\begin{tikzcd}[row sep=small, column sep=small]
 & A \arrow[rr, equal] \arrow[dd, hook] & & A \arrow[dd, hook] \\
A \arrow[ur, equal] \arrow[rr, hook, dotted] \arrow[dd, hook] & & A \arrow[ur, equal] \arrow[dd, equal] & \\
 & B+C \arrow[rr, hook] & & A+D \\
B \arrow[ur, hook] \arrow[rr, hook] & & A \arrow[ur, hook] &
\end{tikzcd}
\qquad
\begin{tikzcd}[row sep=small, column sep=small]
 & A+B \arrow[rr, hook] \arrow[dd, hook] & & A+C \arrow[dd, hook] \\
B \arrow[ur, hook] \arrow[rr, hook, dotted] \arrow[dd, hook] & & C \arrow[ur, hook] \arrow[dd, equal] & \\
 & E+F \arrow[rr, hook] & & D+C \\
F \arrow[ur, hook] \arrow[rr, hook] & & C \arrow[ur, hook] &
\end{tikzcd} . \tag{112}$$

The left diagram is formed by taking a pullback to obtain the bottom square, followed by taking the appropriate pullbacks to form the left and right squares (which induces the dotted arrow by the universal property of pullbacks). Since the left, back and right squares are pullbacks, by pullback-pullback decomposition (see Lemma 12(iii)a) so is the front square. Then by stability of isomorphisms under pullbacks (see Lemma 12(i)d), $A \cong B$. The right diagram is constructed in precisely the same fashion, thus proving that $B \cong F$. □

The notion of *strict* $\mathcal{M}$-initiality also plays an interesting role in our framework of conditions on objects and rewriting rules studied in the main part of this paper due to the following result:

Lemma 16 ($\mathcal{M}$**-morphisms into binary coproducts)** *Let* $\mathbf{C}$ *be an* $\mathcal{M}$*-adhesive category (for* $\mathcal{M}$ *a class of monomorphisms) that possesses a strict* $\mathcal{M}$*-initial object* $\varnothing \in \mathsf{obj}(\mathbf{C})$*. Then for all objects* $X, Y, Z \in \mathsf{obj}(\mathbf{C})$*, if there exists an* $\mathcal{M}$*-morphism* $X+Y \overset{f}{\hookleftarrow} Z$*, then* $Z \cong V+W$ *with*

$$V = \mathsf{PB}(X \hookrightarrow X+Y \overset{f}{\hookleftarrow} Z) \quad \text{and} \quad W = \mathsf{PB}(Z \overset{f}{\hookrightarrow} X+Y \hookleftarrow Y)\,, \tag{113}$$

and consequently (by the universal property of binary coproducts) $f = [v, w]$ *(with* $v : V \hookrightarrow X$ *and* $w : W \hookrightarrow Y$ *both in* $\mathcal{M}$*).*

Proof: Construct the following commutative cube, with the bottom square a pushout, and where the front and left faces are formed by taking pullbacks as described in (113) in order to obtain V and W, followed by forming $Z' = \mathsf{PB}(V \hookrightarrow Z \hookleftarrow W)$, which provides an arrow $Z' \to \varnothing$ by the universal property of pushouts (since the bottom square is a PO):

$$\begin{tikzcd}
& V \arrow[dl, hook'] \arrow[dd, hook, "v"] & & Z' \arrow[ll, hook'] \arrow[dl, hook] \arrow[dd, dotted] \\
Z \arrow[dd, hook, "f"'] & & W \arrow[ll, hook', crossing over] \arrow[dd, hook, "w"] & \\
& X \arrow[dl, hook'] & & \varnothing \arrow[ll, hook', crossing over] \arrow[dl, hook] \\
X+Y & & Y \arrow[ll, hook'] &
\end{tikzcd} \tag{114}$$

By virtue of strict $\mathcal{M}$-initiality of $\varnothing$, the existence of the arrow $Z' \to \varnothing$ entails that $Z' \cong \varnothing$. Since the bottom square is a pushout, all vertical squares are pullbacks and all vertical morphisms are in $\mathcal{M}$, by the $\mathcal{M}$-van Kampen property the top square is a pushout. Thus $Z \cong V + W$ by uniqueness of pushouts up to isomorphisms, and the claim follows. □

Acknowledgments. The work of NB was supported by a *Marie Skłodowska-Curie Individual fellowship* (Grant Agreement No. 753750 – RaSiR). We would like to thank P.A.M. Melliès, P. Sobociński and N. Zeilberger for fruitful discussions.

References

[1] Wassim Abou-Jaoudé, Denis Thieffry, and Jérôme Feret. Formal derivation of qualitative dynamical models from biochemical networks. *Biosystems*, 149:70–112, 2016. DOI: https://doi.org/10.1016/j.biosystems.2016.09.001.

[2] Jiří Adámek, Horst Herrlich, and George Strecker. Abstract and concrete categories, 1990.

[3] Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. A Software Package for Chemically Inspired Graph Transformation. In R. Echahed and M. Minas, editors, *Graph Transformation (ICGT 2016)*, volume 9761 of *Lecture Notes in Computer Science,*, pages 73–88, Cham, 2016. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-319-40530-8_5.

[4] Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. Chemical Transformation Motifs — Modelling Pathways as Integer Hyperflows. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 16(2):510–523, 2019. DOI: https://doi.org/10.1109/tcbb.2017.2781724.

[5] Jakob Lykke Andersen, Rolf Fagerberg, Christoph Flamm, Rojin Kianian, Daniel Merkle, and Peter F Stadler. Towards mechanistic prediction of mass spectra using graph transformation. *MATCH Commun. Math. Comput. Chem.*, 80:705–731, 2018.

[6] Jakob Lykke Andersen, Christoph Flamm, Daniel Merkle, and Peter F Stadler. Rule composition in graph transformation models of chemical reactions. *MATCH Commun. Math. Comput. Chem.*, 80(661-704):45, 2018.

[7] Wolfgang Banzhaf, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. Algorithmic Cheminformatics (Dagstuhl Seminar 14452). *Dagstuhl Reports*, 4(11):22–39, 2015. DOI: https://doi.org/10.4230/DagRep.4.11.22.

[8] Nicolas Behr. Sesqui-Pushout Rewriting: Concurrency, Associativity and Rule Algebra Framework. In Rachid Echahed and Detlef Plump, editors, *Proceedings of theTenth International*

Workshop on Graph Computation Models (GCM 2019) in Eindhoven, The Netherlands, volume 309 of *Electronic Proceedings in Theoretical Computer Science*, pages 23–52. Open Publishing Association, 2019. DOI: https://doi.org/10.4204/eptcs.309.2.

[9] Nicolas Behr. Tracelets and tracelet analysis of compositional rewriting systems. In John Baez and Bob Coecke, editors, Proceedings *Applied Category Theory 2019,* University of Oxford, UK, 15-19 July 2019, volume 323 of *Electronic Proceedings in Theoretical Computer Science*, pages 44–71. Open Publishing Association, 2020. DOI: https://doi.org/10.4204/EPTCS.323.4.

[10] Nicolas Behr and Jean Krivine. Rewriting theory for the life sciences: A unifying framework for CTMC semantics. In Fabio Gadducci and Timo Kehrer, editors, *Graph Transformation, 13th International Conference, ICGT 2020, Held as Part of STAF 2020, Bergen, Norway, June 25–26, 2020, Proceedings*, volume 12150 of *Theoretical Computer Science and General Issues*. Springer International Publishing, 2020. DOI: https://doi.org/10.1007/978-3-030-51372-6.

[11] Nicolas Behr and Pawel Sobocinski. Rule Algebras for Adhesive Categories. In Dan Ghica and Achim Jung, editors, *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:21, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. DOI: https://doi.org/10.4230/LIPIcs.CSL.2018.11.

[12] Nicolas Behr and Pawel Sobocinski. Rule Algebras for Adhesive Categories (extended journal version). *Logical Methods in Computer Science*, Volume 16, Issue 3, July 2020. URL `https://lmcs.episciences.org/6615`.

[13] Nicolas Behr, Vincent Danos, and Ilias Garnier. Stochastic mechanics of graph rewriting. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science - LICS '16*. ACM Press, 2016. DOI: https://doi.org/10.1145/2933575.2934537.

[14] Nicolas Behr, Vincent Danos, Ilias Garnier, and Tobias Heindel. The algebras of graph rewriting. *arXiv preprint arXiv:1612.06240*, 2016.

[15] Nicolas Behr, Vincent Danos, and Ilias Garnier. Combinatorial Conversion and Moment Bisimulation for Stochastic Rewriting Systems. *Logical Methods in Computer Science*, Volume 16, Issue 3, July 2020. URL `https://lmcs.episciences.org/6628`.

[16] Nicolas Behr, Reiko Heckel, and Maryam Ghaffari Saadat. Efficient Computation of Graph Overlaps for Rule Composition: Theory and Z3 Prototyping. In Berthold Hoffmann and Mark Minas, editors, Proceedings of the Eleventh International Workshop on *Graph Computation Models,* Online-Workshop, 24th June 2020, volume 330 of *Electronic Proceedings in Theoretical Computer Science*, pages 126–144. Open Publishing Association, 2020. DOI: https://doi.org/10.4204/EPTCS.330.8.

[17] Gil Benkö, Christoph Flamm, and Peter F. Stadler. A Graph-Based Toy Model of Chemistry. *Journal of Chemical Information and Computer Sciences*, 43(4):1085–1093, 2003. DOI: https://doi.org/10.1021/ci0200570.

[18] M. L. Blinov, J. R. Faeder, B. Goldstein, and W. S. Hlavacek. BioNetGen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics*, 20(17):3289–3291, 2004. DOI: https://doi.org/10.1093/bioinformatics/bth378.

[19] Paul Boehm, Harald-Reto Fonio, and Annegret Habel. Amalgamation of graph transformations: A synchronization mechanism. *Journal of Computer and System Sciences*, 34(2-3): 377–408, 1987. DOI: https://doi.org/10.1016/0022-0000(87)90030-4.

[20] Pierre Boutillier, Mutaamba Maasha, Xing Li, Héctor F Medina-Abarca, Jean Krivine, Jérôme Feret, Ioana Cristescu, Angus G Forbes, and Walter Fontana. The Kappa platform for rule-based modeling. *Bioinformatics*, 34(13):i583–i592, 2018. DOI: https://doi.org/10.1093/bioinformatics/bty272.

[21] Benjamin Braatz and Christoph Brandt. Graph transformations for the resource description framework. *Electronic Communications of the EASST*, 10, 2008. DOI: https://doi.org/10.14279/tuj.eceasst.10.158.

[22] Benjamin Braatz, Hartmut Ehrig, Karsten Gabriel, and Ulrike Golas. Finitary $\mathcal{M}$ -adhesive categories. *Mathematical Structures in Computer Science*, 24(4):240403–240443, 2014. DOI: https://doi.org/10.1017/S0960129512000321.

[23] Ferdinanda Camporesi, Jérôme Feret, and Kim Quyên Lý. KaDE: A Tool to Compile Kappa Rules into (Reduced) ODE Models. In *Computational Methods in Systems Biology*, pages

291–299. Springer International Publishing, 2017. DOI: https://doi.org/10.1007/978-3-319-67471-1_18.

[24] J.R.B. Cockett and Stephen Lack. Restriction categories II: partial map classification. *Theoretical Computer Science*, 294(1-2):61–102, 2003. DOI: https://doi.org/10.1016/s0304-3975(01)00245-6.

[25] 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 *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*, pages 163–246, 1997.

[26] Andrea Corradini, Tobias Heindel, Frank Hermann, and Barbara König. Sesqui-Pushout Rewriting. In A. Corradini, H. Ehrig, U. Montanari, L. Ribeiro, and G. Rozenberg, editors, *Graph Transformations (ICGT 2006)*, volume 4178 of *Lecture Notes in Computer Science*, pages 30–45. Springer Berlin Heidelberg, 2006. DOI: https://doi.org/10.1007/11841883_4.

[27] Andrea Corradini, Dominique Duval, Rachid Echahed, Frederic Prost, and Leila Ribeiro. AGREE – Algebraic Graph Rewriting with Controlled Embedding. In F. Parisi-Presicce and B. Westfechtel, editors, *Graph Transformation (ICGT 2015)*, volume 9151 of *Lecture Notes in Computer Science*, pages 35–51, Cham, 2015. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-319-21145-9_3.

[28] Vincent Danos and Cosimo Laneve. Graphs for Core Molecular Biology. In C. Priami, editor, *Computational Methods in Systems Biology (CMSB 2003)*, volume 2602 of *Lecture Notes in Computer Science*, pages 34–46. Springer Berlin Heidelberg, 2003. DOI: https://doi.org/10.1007/3-540-36481-1_4.

[29] Vincent Danos and Cosimo Laneve. Core Formal Molecular Biology. In P. Degano, editor, *Programming Languages and Systems (ESOP 2003)*, volume 2618 of *Lecture Notes in Computer Science*, pages 302–318. Springer Berlin Heidelberg, 2003. DOI: https://doi.org/10.1007/3-540-36575-3_21.

[30] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69 – 110, 2004. DOI: https://doi.org/10.1016/j.tcs.2004.03.065.

[31] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-Based Modelling of Cellular Signalling. In L. Caires and V.T. Vasconcelos, editors, *Concurrency Theory (CONCUR 2007)*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41. Springer Berlin Heidelberg, 2007. DOI: https://doi.org/10.1007/978-3-540-74407-8_3.

[32] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling, symmetries, refinements. In Jasmin Fisher, editor, *Formal Methods in Systems Biology (FMSB 2008)*, volume 5054 of *Lecture Notes in Computer Science*, pages 103–122. Springer Berlin Heidelberg, 2008. DOI: https://doi.org/10.1007/978-3-540-68413-8_8.

[33] Vincent Danos, Jerome Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Chris Thompson-Walsh, and Glynn Winskel. Graphs, Rewriting and Pathway Reconstruction for Rule-Based Models. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 276–288, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. DOI: https://doi.org/10.4230/LIPIcs.FSTTCS.2012.276.

[34] Vincent Danos, Tobias Heindel, Ricardo Honorato-Zimmer, and Sandro Stucki. Reversible Sesqui-Pushout Rewriting. In Holger Giese and Barbara König, editors, *Graph Transformation (ICGT 2014)*, volume 8571 of *Lecture Notes in Computer Science*, pages 161–176, Cham, 2014. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-319-09108-2_11.

[35] H. Ehrig and A. Habel. *Graph Grammars with Application Conditions*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986. DOI: https://doi.org/10.1007/978-3-642-95486-3_7.

[36] H. Ehrig, M. Pfender, and H. J. Schneider. Graph-grammars: An algebraic approach. In *14th Annual Symposium on Switching and Automata Theory (SWAT 1973)*, pages 167–180, 1973. DOI: https://doi.org/10.1109/SWAT.1973.11.

[37] H. Ehrig, K. Ehrig, U. Prange, and G. Taentzer. Fundamentals of algebraic graph transformation. *Monographs in Theoretical Computer Science. An EATCS Series*, 2006. DOI: https://doi.org/10.1007/3-540-31188-2.

[38] 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(03):361, 1991. DOI: https://doi.org/10.1017/s0960129500001353.

[39] Hartmut Ehrig, Julia Padberg, Ulrike Prange, and Annegret Habel. Adhesive High-Level Replacement Systems: A New Categorical Framework for Graph Transformation. *Fundamenta Informaticae*, 74(1):1–29, 2006. DOI: https://doi.org/10.5555/1231199.1231201.

[40] Hartmut Ehrig, Reiko Heckel, Grzegorz Rozenberg, and Gabriele Taentzer, editors. *Graph Transformations (ICGT 2008)*, volume 5214 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008. DOI: https://doi.org/10.1007/978-3-540-87405-8.

[41] Hartmut Ehrig, Ulrike Golas, Frank Hermann, et al. Categorical frameworks for graph transformation and HLR systems based on the DPO approach. *Bulletin of the EATCS*, (102): 111–121, 2010.

[42] Hartmut Ehrig, Ulrike Golas, Annegret Habel, Leen Lambers, and Fernando Orejas. $\mathcal{M}$-adhesive transformation systems with nested application conditions. Part 1: parallelism, concurrency and amalgamation. *Mathematical Structures in Computer Science*, 24(04), 2014. DOI: https://doi.org/10.1017/s0960129512000357.

[43] Rolf Fagerberg, Christoph Flamm, Rojin Kianian, Daniel Merkle, and Peter F. Stadler. Finding the K best synthesis plans. *Journal of Cheminformatics*, 10(1), 2018. DOI: https://doi.org/10.1186/s13321-018-0273-z.

[44] Jerome Feret, Thomas Henzinger, Heinz Koeppl, and Tatjana Petrov. Lumpability abstractions of rule-based systems. *Theoretical Computer Science*, 431(0):137 – 164, 2012. DOI: https://doi.org/10.1016/j.tcs.2011.12.059.

[45] Ulrike Golas, Annegret Habel, and Hartmut Ehrig. Multi-amalgamation of rules with application conditions in $\mathcal{M}$-adhesive categories. *Mathematical Structures in Computer Science*, 24(04), 2014. DOI: https://doi.org/10.1017/s0960129512000345.

[46] Annegret Habel and Karl-Heinz Pennemann. Correctness of high-level transformation systems relative to nested conditions. *Mathematical Structures in Computer Science*, 19(02):245, 2009. DOI: https://doi.org/10.1017/s0960129508007202.

[47] Annegret Habel and Detlef Plump. $\mathcal{M},\mathcal{N}$ -Adhesive Transformation Systems. In H. Ehrig, G. Engels, H.J. Kreowski, and G. Rozenberg, editors, *Graph Transformations (ICGT 2012)*, volume 7562 of *Lecture Notes in Computer Science*, pages 218–233. Springer Berlin Heidelberg, 2012. DOI: https://doi.org/10.1007/978-3-642-33654-6_15.

[48] Annegret Habel and Detlef Plump. $\mathcal{M},\mathcal{N}$ -Adhesive Transformation Systems. (Long Version), 2012. URL `http://formale-sprachen.informatik.uni-oldenburg.de/~skript/fs-pub/HaPl12b.pdf`.

[49] Annegret Habel, Reiko Heckel, and Gabriele Taentzer. Graph grammars with negative application conditions. *Fundam. Inf.*, 26(3,4):287–313, 1996. DOI: https://doi.org/10.3233/FI-1996-263404.

[50] Leonard A. Harris, Justin S. Hogg, José-Juan Tapia, John A. P. Sekar, Sanjana Gupta, Ilya Korsunsky, Arshi Arora, Dipak Barua, Robert P. Sheehan, and James R. Faeder. BioNetGen 2.2: advances in rule-based modeling. *Bioinformatics*, 32(21):3366–3368, 2016. DOI: https://doi.org/10.1093/bioinformatics/btw469.

[51] Frank Hermann, Andrea Corradini, and Hartmut Ehrig. Analysis of permutation equivalence in $\mathcal{M}$-adhesive transformation systems with negative application conditions. *Mathematical Structures in Computer Science*, 24(4), 2014.

[52] Stephen Lack and Paweł Sobociński. Adhesive Categories. In Igor Walukiewicz, editor, *Foundations of Software Science and Computation Structures (FoSSaCS 2004)*, volume 2987 of *Lecture Notes in Computer Science*, pages 273–288. Springer Berlin Heidelberg, 2004. DOI: https://doi.org/10.1007/978-3-540-24727-2_20.

[53] Stephen Lack and Paweł Sobociński. Adhesive and quasiadhesive categories. *RAIRO - Theoretical Informatics and Applications*, 39(3):511–545, 2005. DOI: https://doi.org/10.1051/ita:2005028.

[54] Michael Löwe. Algebraic Approach to Single-Pushout Graph Transformation. *Theor. Comput. Sci.*, 109(1–2):181–224, March 1993. ISSN 0304-3975. DOI: https://doi.org/10.1016/0304-3975(93)90068-5.

[55] Michael Löwe. Polymorphic Sesqui-Pushout Graph Rewriting. In F. Parisi-Presicce and B. Westfechtel, editors, *Graph Transformation*, volume 9151, pages 3–18, Cham, 2015. Springer International Publishing. DOI: https://doi.org/10.1007/978-3-319-21145-9_1.

[56] Elaine Murphy, Vincent Danos, Jérôme Féret, Jean Krivine, and Russell Harmer. Rule-based modeling and model refinement. In *Elements of Computational Systems Biology*, pages 83–114. John Wiley & Sons, Inc., 2010. DOI: https://doi.org/10.1002/9780470556757.ch4.

[57] Marisa Navarro Gomez, Fernando Orejas Valdés, Elvira Pino Blanco, and Leen Lambers. A logic of graph conditions extended with paths. In *Actas de las XVI Jornadas de Programación y Lenguajes (PROLE 2016): Salamanca, septiembre de 2016*, pages 1–15, 2016.

[58] James R. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[59] Julia Padberg. Towards M-Adhesive Categories based on Coalgebras and Comma Categories. *arXiv:1702.04650*, 2017.

[60] Karl-Heinz Pennemann. Resolution-Like Theorem Proving for High-Level Conditions. In H. Ehrig, R. Heckel, G. Rozenberg, and G. Taentzer, editors, *Graph Transformations (ICGT 2008)*, volume 5214 of *Lecture Notes in Computer Science*, pages 289–304. Springer Berlin Heidelberg, 2008. DOI: https://doi.org/10.1007/978-3-540-87405-8_20.

[61] Tatjana Petrov, Jerome Feret, and Heinz Koeppl. Reconstructing species-based dynamics from reduced stochastic rule-based models. In *Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)*. IEEE, 2012. DOI: https://doi.org/10.1109/wsc.2012.6465241.

[62] Hendrik Radke. *A Theory of HR* Graph Conditions and their Application to Meta-Modeling*. PhD thesis, Carl von Ossietzky Universität Oldenburg, Fakultät II, Department für Informatik, 2016.

[63] Grzegorz Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations*. World Scientific, 1997. DOI: https://doi.org/10.1142/3303.

[64] Sven Schneider, Leen Lambers, and Fernando Orejas. Symbolic Model Generation for Graph Properties. In M. Huisman and J. Rubin, editors, *Fundamental Approaches to Software Engineering (FASE 2017)*, volume 10202 of *Lecture Notes in Computer Science*, pages 226–243. Springer Berlin Heidelberg, 2017. DOI: https://doi.org/10.1007/978-3-662-54494-5_13.