Adhesive Subcategories of Functor Categories with Instantiation to Partial Triple Graphs
Extended Version

Jens Kosiol[1]Lars Fritsche[2]Andy Schürr[2]Gabriele Taentzer[1]

[1] Philipps-Universität Marburg
{kosiolje,taentzer}@mathematik.uni-marburg.de
[2] TU Darmstadt
{lars.fritsche,andy.schuerr}@es.tu-darmstadt.de

Abstract. Synchronization and integration processes of correlated models that are formally based on triple graph grammars often suffer from the fact that elements are unnecessarily deleted and recreated losing information in the process. It has been shown that this undesirable loss of information can be softened by allowing partial correspondence morphisms in triple graphs. We provide a formal framework for this new synchronization process by introducing the category **PTrG** of partial triple graphs and proving it to be adhesive. This allows for ordinary double pushout rewriting of partial triple graphs. To exhibit **PTrG** as an adhesive category, we present a fundamental construction of subcategories of functor categories and show that these are adhesive HLR if the base category already is. Secondly, we consider an instantiation of this framework by triple graphs to illustrate its practical relevance and to have a concrete example at hand.

Keywords: Adhesiveness · Functor Category · Double Pushout Rewriting · Triple Graphs

1 Introduction

Bidirectional transformation (bx) is a central concept in model-driven software development among others [3,1]. Bx provides the means to define and restore consistency between different kinds of artifacts or different views on a system. Triple Graph Grammars (TGGs) [27] are an established bx-formalism. A triple graph correlates two models (referred to as source and target) by defining a correspondence graph in between that contains elements relating elements of both sides. A TGG defines how correlated models co-evolve and can be used to, e.g., automatically synchronize source and target model after a user edited only one of them. Approaches that have been suggested for such synchronization processes are either informal and rather ad-hoc [9,12], quite inefficient and work under restricted circumstances only [16], or unnecessary deletions may be included leading to a loss of information [10,21]. In [7], we present a synchronization process based on triple graphs that allow correspondence morphisms to be

partial. This largely improved existing approaches with regard to information loss and runtime. The formal background in that work, however, was restricted and an elaborated theory of partial triple graphs was left to future work. Such a theory is one of the contributions of this paper. We show that the category of partial triple graphs is adhesive such that double pushout rewriting becomes possible [20,5].

When working with adhesive (HLR) categories, an often used technique is to exhibit a category $\mathcal{D}$ as (equivalent to) a functor category $[\mathcal{X}, \mathcal{C}]$ where $\mathcal{X}$ is small and $\mathcal{C}$ is known to be an adhesive (HLR) category. This ensures that $\mathcal{D}$ is an adhesive (HLR) category as well. Considering partial triple graphs and their morphisms, they can be formalized quite naturally as a subcategory of a functor category over an adhesive category. More precisely, they are formalized as those functors of the category $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graph}]$ that map both central morphisms to injective morphisms. This category can be seen as an instance of a more general principle. There are several categories of interest that form a proper subcategory of a functor category $[\mathcal{X}, \mathcal{C}]$ in a quite natural way. More precisely, we consider subcategories consisting of only those functors that map a designated subset S of the morphisms from $\mathcal{X}$ to monomorphisms (morphisms from $\mathcal{M}$) in $\mathcal{C}$. Besides the already mentioned partial triple graphs, examples include but are not limited to:[3]

1. Elements of $[\bullet \hookrightarrow \bullet, \mathcal{C}]$, where $\mathcal{C}$ is any adhesive (HLR) category, can be understood as objects that come with a marked ($\mathcal{M}$-)subobject. These are exactly the categories that Kastenberg and Rensink proved to be adhesive in [17], for the case that $\mathcal{C}$ is adhesive. They introduce a new concept of attribution for the case where $\mathcal{C}$ is the category **Graph**.
2. Elements of $[\bullet \hookleftarrow \bullet \hookrightarrow \bullet, \mathcal{C}]$, where $\mathcal{C}$ is any adhesive (HLR) category, are exactly the linear rules of $\mathcal{C}$.
3. Elements of $[\bullet \hookleftarrow \bullet \rightarrow \bullet, \mathcal{C}]$ are the partial morphisms of $\mathcal{C}$ (without the usual identification of equivalent ones [26]).
4. Let $\mathcal{S}_n$ be a star-like shape: there exists a central node with n outgoing spans (the shape $\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet$ being the special case for $n = 2$). Let $\bar{\mathcal{S}}_n$ denote the same shape with the arrows pointing to the central node designated to be mapped to monomorphisms. König et al. [19,29] use (slice categories of) $[\bar{\mathcal{S}}_n, \mathcal{C}]$ (where $\mathcal{C}$ is a suitable category of models) to formalize a correspondence relation between n different (meta-)models via n partial morphisms from a correspondence model.
5. If $\mathcal{T}$ is a finite tree and $\bar{\mathcal{T}}$ denotes the tree where every edge of $\mathcal{T}$ is marked as to be mapped to monomorphisms (morphisms from $\mathcal{M}$), after fixing an appropriate decoration of the nodes of $\mathcal{T}$ with quantifiers and logical connectives the elements of $[\bar{\mathcal{T}}, \mathcal{C}]$ are nested conditions [13] in $\mathcal{C}$ of a fixed structure.

Simple examples (as Example 4) show that, in general, the full subcategory generated by such a choice of functors will not be an adhesive (HLR) category. In

[3] We do not depict identities of $\mathcal{X}$ and mark the morphisms from the designated set S by a hooked arrow.

Sect. 4 we show that an adhesive (HLR) category is obtained by restricting the class of morphisms to those natural transformations where all squares induced by the designated morphisms are pullback squares (Theorem 11).

In the second part of this paper (Sect. 5), we instantiate our theory and consider an application to triple graphs with partial morphism between correspondence and source or target graphs. We discuss the expressiveness of double pushout rewriting in that category (Proposition 20) and provide a basic characterization of matches to partial triple graphs (Proposition 21). Moreover, we define the decomposition of a rule on partial triple graphs into a source and a forward rule (Theorem 23) in analogy to the procedure for TGGs [27].

We begin by presenting an introductory example in Sect. 2 and recall some preliminaries in Sect. 3. After the main contributions, we discuss related work in Sect. 6 and conclude in Sect. 7. All proofs that are skipped in the main part of this paper are presented in Appendix A, together with some additional technical preliminaries.

2 Introductory Example

We motivate our new construction of categories on an example of triple graph grammars (TGGs). TGGs [27] provide a means to define consistency between two correlated models in a declarative and rule-based way by incorporating a third model, a correspondence model, to connect elements from both sides via correspondence links. Elements connected in such a way are henceforth considered to be consistent. Figure 1 shows the rule set of our running example consisting of three TGG rules taken from [7]. They allow to simultaneously create correlated models of a Java abstract syntax tree and a custom documentation model. *Root-Rule* creates a root *Package* and a root *Folder* together with a correspondence link in between which is indicated by the annotation (++) and by green colouring. This rule can be applied arbitrarily often as it does not contain a precondition. *Sub-Rule* requires a *Package*, *Folder*, and a correspondence link between both as precondition and creates a *Package* and *Folder* hierarchy together with a *Doc-File*. Finally, *Leaf-Rule* creates a *Class* with a corresponding *Doc-File* under the same precondition as *Sub-Rule*. Given these rules, we can generate consistent triple graphs like the one depicted in Fig. 2 by iteratively applying the above rules. With *content* contained in the *Doc-Files* on the target side, we indicate that a user has edited them independently such that the model includes information that is private to the target side.

A common use case for TGGs is to synchronize changes between models in order to restore consistency after a user edit. Assume that changes are applied to the left side of Fig. 2 such that the element `p` is deleted and a new reference is created connecting `rootP` with `subP` as depicted in Fig. 3 (a). Note that this change also leads to a broken correspondence link, i.e., the result is not a triple graph any longer. There are several TGG-based approaches to synchronize source and target again. However, suggested incremental approaches [9,12] are rather ad-hoc and not proven to be correct. One provably correct approach

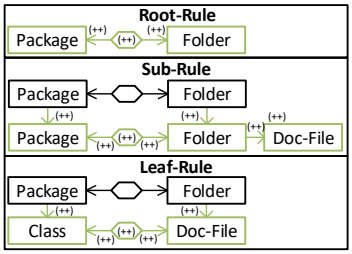


Fig. 1. Example: TGG Rules

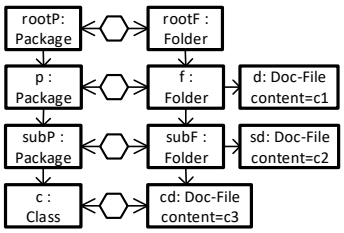


Fig. 2. Example: instance model

parses the whole instance for a maximal remaining valid submodel [16] and is thus quite inefficient and only applicable to a restricted class of TGGs. Other approaches [10,21] first derive the triple graph depicted in Fig. 3 (b) and then start a re-translation process using so-called forward rules, which are derived from the TGG, to obtain the consistent triple shown in Fig. 3 (c). But the *Doc-Files*' contents have been lost in the process.

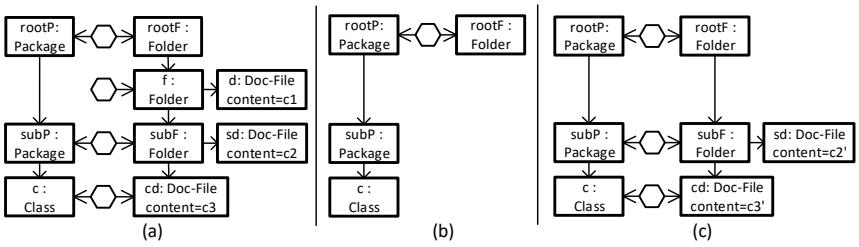


Fig. 3. A synchronization scenario

If it was possible to apply rules to partial triple graphs directly, the one depicted in Fig. 3 (a) can be synchronized by applying *Delta-Forward-Rule* as depicted in Fig. 5 where the red elements (additionally annotated with $(--)$) are to be deleted. The qualitative difference of the result is that the contents of both *Doc-Files* are preserved as both elements are not recreated in the process and furthermore, less rule applications are necessary. *Delta-Forward-Rule* can be obtained by splitting *Delta-Rule* (see Fig. 4) into two rules (which is also called *operationalization*): *Delta-Source-Rule* which is a projection to the source component and *Delta-Forward-Rule* which propagates the according changes to correspondence and target graphs. In [8], we show how to construct rules like *Delta-Rule* from given TGG rules. In [7], we operationalize these rules and use them for more efficient synchronization processes. However, we did not introduce partial triple graphs as a category. Moreover, we only defined rule applications to partial triple graphs were the rules arise by operationalizing rules for triple

graphs. Hence, an elaborated theory for applying and operationalizing rules for partial triple graphs is still needed which is one of the contributions of this paper.

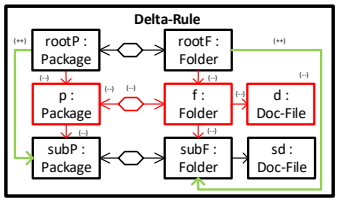


Fig. 4. Example: Delta-Rule

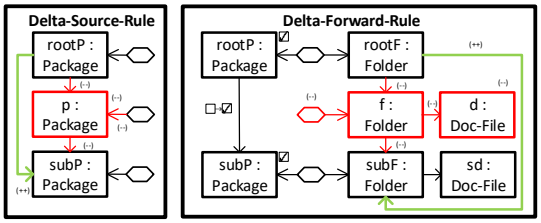


Fig. 5. Example: Operationalized Delta-Rule

3 Preliminaries

In this section, we introduce some preliminaries, namely adhesive (HLR) categories and double pushout rewriting. Adhesive categories can be understood as categories where pushouts along monomorphisms behave like pushouts along injective functions in the category of sets. They have been introduced by Lack and Sobociński [20] to offer a unifying formal framework for double pushout rewriting. Later, Ehrig et al. [5] introduced the more general notion of adhesive HLR categories that includes practically relevant examples which are not adhesive. We also introduce the notion of a partial van Kampen square [15] that we need later on.

Definition 1 (Adhesive and adhesive HLR categories). *A category $\mathcal{C}$ with a class of monomorphisms $\mathcal{M}$ is* adhesive HLR *if*

- *the class of monomorphisms $\mathcal{M}$ contains all isomorphisms and is* closed under composition and decomposition*, i.e., $f, g \in \mathcal{M}$ implies $g \circ f \in \mathcal{M}$ whenever the composition is defined and $g \circ f \in \mathcal{M}, g \in \mathcal{M}$ implies $f \in \mathcal{M}$.*
- *the category $\mathcal{C}$ has pushouts and pullbacks along $\mathcal{M}$-morphisms and $\mathcal{M}$-*morphisms are closed under pushouts and pullbacks *such that if Fig. 6 depicts a pushout square with $m \in \mathcal{M}$ then also $n \in \mathcal{M}$ and analogously if it depicts a pullback square with $n \in \mathcal{M}$ then also $m \in \mathcal{M}$.*
- *pushouts in $\mathcal{C}$ along $\mathcal{M}$-morphisms are* van Kampen squares*, i.e., for any commutative cube as depicted in Fig. 7 where the bottom square is a pushout along an $\mathcal{M}$-morphism m and the backfaces are pullbacks then the top square is a pushout if and only if both front faces are pullbacks.*

A category $\mathcal{C}$ is adhesive *if it has all pullbacks, and pushouts along monomorphisms exist and are van Kampen squares.*

Pushouts along $\mathcal{M}$-morphisms are partial van Kampen squares *if for any commutative cube as depicted in Fig. 7 where the bottom square is a pushout along an $\mathcal{M}$-morphism m, the backfaces are pullbacks, and b and c are $\mathcal{M}$-morphisms, then the top square is a pushout if and only if both front faces are pullbacks and d is an $\mathcal{M}$-morphism.*

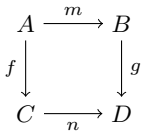


Fig. 6. A pushout square

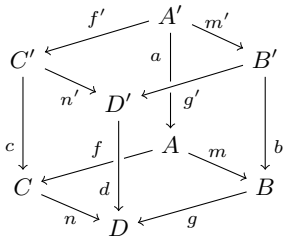
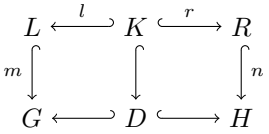


Fig. 7. Commutative cube over pushout square

Remark 2. Every adhesive category is adhesive HLR for $\mathcal{M}$ being the class of all monomorphisms [5]. Moreover, pushouts along monomorphisms are partial van Kampen squares in adhesive categories [15].

Important examples of adhesive categories include the categories of sets, of (typed) graphs, and of (typed) triple graphs [20,5]. Examples of categories that are not adhesive but adhesive HLR (for an appropriate choice of $\mathcal{M}$) include typed attributed [5] and symbolic attributed graphs [25]. Adhesive HLR categories are a suitable formal framework for rule-based rewriting as defined in the double pushout approach: Rules are a declarative way to define transformations of objects. They consist of a left-hand side (LHS) L, a right-hand side (RHS) R, and a common subobject K, the interface of the rule. In case of (typed) graphs, application of a rule p to a graph G amounts to choosing an image of the rule's LHS L in G, deleting the image of $L \setminus K$ and adding a copy of $R \setminus K$. This procedure can be formalized by two pushouts. Rules and their application semantics are defined as follows.

Definition 3 (Rules and transformations). *Given an adhesive HLR category $\boldsymbol{C}$, a* rule *p consists of three objects L, K, and R, called* left-hand side*,* interface*, and* right-hand side*, and two monomorphisms $l : K \hookrightarrow L, r : K \hookrightarrow R$. Given a rule p, an object G, and a monomorphism $m : L \hookrightarrow G$, called* match*, a* (direct) transformation *$G \Rightarrow_{p,m} H$ from G to H via p at match m is given by the diagram to the right where both squares are pushouts.*

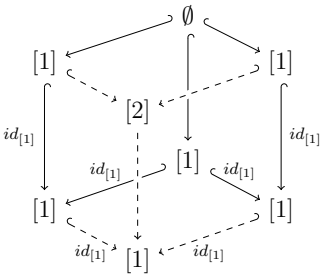


4 Adhesive Subcategories of Functor Categories

We are interested in investigating subcategories of functor categories $[\mathcal{X}, \mathcal{C}]$ over an adhesive HLR category $\mathcal{C}$ with a set of monomorphisms $\mathcal{M}$. In particular, these subcategories arise by restricting to those functors that map all morphisms from a designated set S of morphisms from $\mathcal{X}$ to $\mathcal{M}$-morphisms in $\mathcal{C}$. As the next example shows, the induced full subcategory fails to be adhesive already

for basic examples. The reason for this is that—in the category of morphisms—the (componentwise computed) pushout of monomorphisms does not need to result in a monomorphism again, even if additionally the morphisms between the monomorphisms are monomorphisms as well. The counterexample below has already been presented in [23].

Example 4. Let $\mathcal{C}$ be the adhesive category **Set** and $\mathcal{X}$ the category $\bullet \to \bullet$. Consider the full subcategory of the functor category $[\bullet \to \bullet, \mathbf{Set}]$ induced by those functors that map the only non-identity morphism to an injective function in **Set**. This is just the category with injective functions as objects and commuting squares as morphisms. Let $[n]$ denote the set with n elements and consider the commuting cube depicted to the right: The two squares in the back are a span of monomorphisms in that category. Computing the top and the bottom square as pushouts, i.e., computing the pushout of the two squares in the back in the category $[\bullet \to \bullet, \mathbf{Set}]$, results in a function that is not injective. It is not difficult to check that the category with injective functions as objects and commuting squares as morphisms does not have pushouts, even not along monomorphisms: There is no way to replace the vertical morphism in the front by an injective function and obtain a cube that is a pushout of the two squares in the back.



To resolve this problem, we introduce our categories of interest not as full subcategories of functor categories but restrict the class of allowed morphisms between them.

Definition 5 (S-functor. S-cartesian natural transformation). *Given a small category $\mathcal{X}$, a subset S of the morphisms of $\mathcal{X}$, and an arbitrary category $\mathcal{C}$ with designated class of monomorphisms $\mathcal{M}$, an* S-functor *is a functor $F : \mathcal{X} \to \mathcal{C}$ such that for every morphism $m \in S$ the morphism Fm is an $\mathcal{M}$-morphism.*

A natural transformation $\sigma : F \to G$ between two S-functors is S-cartesian *if for every morphism $S \ni m : x \to y$ the corresponding naturality square $\sigma_y \circ Fm = Gm \circ \sigma_x$ is a pullback square.*

Example 6. The partial triple graph that is depicted in Fig. 3 (a) is an S-functor from the category $\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet$ to the category of graphs where S consists of the two morphisms to the central object: The left object is mapped to the source graph depicted to the left in Fig. 3 (a), the right object to the target graph depicted to the right, and the central object to the correspondence graph consisting of the four hexagons. The second and the fourth object are mapped to the respective domains of the correspondence morphisms to source and target graph. While the domain of the correspondence morphism to the target graph is the whole correspondence graph, the domain of the correspondence morphism to the source graph just consists of three of the hexagons. The outer morphisms

of $\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet$ are mapped to the correspondence morphisms while the central morphisms are mapped to the inclusion of the domains of the correspondence morphisms into the correspondence graph, which are both injective.

Since the composition of two pullbacks is a pullback and the identity natural transformation is trivially S-cartesian, S-functors with S-cartesian natural transformations as morphisms form a category (associativity of composition and neutrality of the composition with the identity natural transformation are just inherited from the full functor category). We call such categories *S-cartesian subcategories* and denote them by $[\mathcal{X}_S, \mathcal{C}]$.

Proposition 7. *Given a small category $\mathcal{X}$, a subset S of the morphisms of $\mathcal{X}$ and an arbitrary category $\mathcal{C}$ with designated class of monomorphisms $\mathcal{M}$, S-functors with S-cartesian natural transformations as morphisms form a generally non-full subcategory of the functor category $[\mathcal{X}, \mathcal{C}]$. In particular, there is an inclusion functor $I : [\mathcal{X}_S, \mathcal{C}] \hookrightarrow [\mathcal{X}, \mathcal{C}]$.*

Section 5 is devoted to develop the category of partial triple graphs as an instantiation of this very general framework. In particular, it presents concrete examples illustrating the abstract notions. In this section, we prove that if a category $\mathcal{C}$ is adhesive HLR, categories $[\mathcal{X}_S, \mathcal{C}]$ are adhesive HLR again (Theorem 11), assuming pushouts along $\mathcal{M}$-morphisms to be partial van Kampen squares. We first collect results that contribute to the proof of our main theorem. They are of independent interest as they determine that pushouts and pullbacks along $\mathcal{M}$-morphisms are computed componentwise in categories of S-functors. Recall that a functor $F : \mathcal{C} \to \mathcal{D}$ is said to *create (co-)limits* of a certain type $\mathcal{J}$ if for every diagram $D : \mathcal{J} \to \mathcal{C}$ and every (co-)limit for $F \circ D$ in $\mathcal{D}$ there exists a unique preimage under F that is a (co-)limit of D in $\mathcal{C}$ [2].

In the following, let $\mathcal{X}$ always be a small category and S a subset of its morphisms; moreover, $\mathcal{C}$ is an arbitrary category with designated class of monomorphisms $\mathcal{M}$ (in particular, if $\mathcal{C}$ is adhesive HLR, $\mathcal{M}$ is understood to be the corresponding class of monomorphisms).

Proposition 8. *Let $[\mathcal{X}_S, \mathcal{C}]$ be an S-cartesian subcategory of $[\mathcal{X}, \mathcal{C}]$. If pullbacks along $\mathcal{M}$-morphisms exist, $[\mathcal{X}, \mathcal{C}]$ and $[\mathcal{X}_S, \mathcal{C}]$ have pullbacks along natural transformations where every component is an $\mathcal{M}$-morphism and the inclusion functor $I : [\mathcal{X}_S, \mathcal{C}] \hookrightarrow [\mathcal{X}, \mathcal{C}]$ creates these.*

The next lemma characterizes the monomorphisms of $[\mathcal{X}_S, \mathcal{C}]$.

Lemma 9. *Let $[\mathcal{X}_S, \mathcal{C}]$ be an S-cartesian subcategory of $[\mathcal{X}, \mathcal{C}]$. Then every morphism in $[\mathcal{X}, \mathcal{C}]$ or in $[\mathcal{X}_S, \mathcal{C}]$ where every component is a monomorphism is a monomorphism in the respective category. If $\mathcal{C}$ has pullbacks, then the converse is true.*

The next proposition states that also pushouts along $\mathcal{M}$-morphisms are calculated componentwise in a category $[\mathcal{X}_S, \mathcal{C}]$. In contrast to the case of pullbacks, the proof requires that $\mathcal{C}$ is adhesive HLR and that pushouts along $\mathcal{M}$-morphisms are partial van Kampen squares.

Proposition 10. *Let $\mathcal{C}$ be an adhesive HLR category such that pushouts along $\mathcal{M}$-morphisms are partial van Kampen squares. Let $[\mathcal{X}_S, \mathcal{C}]$ be an S-cartesian subcategory of $[\mathcal{X}, \mathcal{C}]$. Then $[\mathcal{X}, \mathcal{C}]$ and $[\mathcal{X}_S, \mathcal{C}]$ have pushouts along morphisms where every component is an $\mathcal{M}$-morphism and the inclusion functor $I : [\mathcal{X}_S, \mathcal{C}] \hookrightarrow [\mathcal{X}, \mathcal{C}]$ creates these.*

Together, the obtained results guarantee that our construction leads to categories that are adhesive HLR again:

Theorem 11 (Adhesive HLR). *Let $\mathcal{C}$ be an adhesive HLR category such that pushouts along $\mathcal{M}$-morphisms are partial van Kampen squares. Let $[\mathcal{X}_S, \mathcal{C}]$ be an S-cartesian subcategory of $[\mathcal{X}, \mathcal{C}]$. Then $[\mathcal{X}_S, \mathcal{C}]$ is adhesive HLR for the class $\mathcal{M}'$ of natural transformations where every component is an $\mathcal{M}$-morphism.*

Proof. By Lemma 9, $\mathcal{M}'$ consists of monomorphisms if $\mathcal{C}$ is adhesive HLR and is the class of all monomorphisms if $\mathcal{C}$ is adhesive. By definition of $\mathcal{M}'$, the necessary composition and decomposition properties are inherited from $\mathcal{M}$. Since the inclusion functor from $[\mathcal{X}_S, \mathcal{C}]$ to $[\mathcal{X}, \mathcal{C}]$ creates pullbacks and pushouts along $\mathcal{M}'$-morphisms (Propositions 8 and 10) and $[\mathcal{X}, \mathcal{C}]$ is adhesive HLR with respect to natural transformations that are $\mathcal{M}$-morphisms in every component, $[\mathcal{X}_S, \mathcal{C}]$ is adhesive HLR as well. □

The next proposition states that—whenever the involved objects and morphisms belong to $[\mathcal{X}_S, \mathcal{C}]$—applying a rule in $[\mathcal{X}_S, \mathcal{C}]$ or in $[\mathcal{X}, \mathcal{C}]$ yields the same result. For simplicity, we suppress the inclusion functor I in its formulation.

Proposition 12 (Functoriality of rule application). *Let $\mathcal{C}$ be an adhesive HLR category such that pushouts along $\mathcal{M}$-morphisms are partial van Kampen squares. Let $[\mathcal{X}_S, \mathcal{C}]$ be an S-cartesian subcategory of $[\mathcal{X}, \mathcal{C}]$. Let $p = (L \overset{\lambda}{\hookleftarrow} K \overset{\rho}{\hookrightarrow} R)$ be a rule and $\mu : L \to G$ a match such that $\lambda, \rho, \mu, L, K, R, G$ are morphisms and objects of $[\mathcal{X}_S, \mathcal{C}]$. Then p is applicable to G with match μ in $[\mathcal{X}_S, \mathcal{C}]$ if and only if it is in $[\mathcal{X}, \mathcal{C}]$. Moreover, the resulting object H coincides (up to isomorphism).*

5 The Category of Partial Triple Graphs

In this section, we apply the theory developed in the section above to the category of (typed) partial triple graphs. Our definition of these rests upon the following (simple) definition of partial morphisms in arbitrary categories. We refrain from identifying equivalent partial morphisms as usually done [26].

Definition 13 (Partial morphism). *A* partial morphism $a : A \dashrightarrow B$ *is a span $A \overset{\iota_A}{\hookleftarrow} A' \overset{a}{\to} B$ where ι_A is a monomorphism; A' is called the* domain *of a.*

In the section above, we address the framework of adhesive HLR categories since we generally want to be able to support attribution concepts for partial triple graphs. Two influential such concepts, namely attributed graphs and

symbolic attributed graphs, have been shown to constitute adhesive HLR categories [5,25]. Moreover, it is not difficult to check that in both cases pushouts along the respective $\mathcal{M}$-morphisms are partial van Kampen squares. Hence, they can be used as base categories $\mathcal{C}$ when instantiating the framework from above. But for simplicity, we here just present (typed) partial triple graphs without attributes.

Definition 14 (Partial triple graph). *The category of* triple graphs **TrG** *is the functor category* $[\bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$. *The category of* partial triple graphs **PTrG** *is the category* $[\bullet \leftarrow \bullet \hookrightarrow \bullet \hookleftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$.

Remark 15. By the definitions above, an object $G = (G_S \leftarrow \tilde{G}_S \hookrightarrow G_C \hookleftarrow \tilde{G}_T \rightarrow G_T)$ of **PTrG** might equivalently be considered to consist of a graph G_C with partial morphisms $\sigma : G_C \dashrightarrow G_L$ and $\tau : G_C \dashrightarrow G_T$ where $\tilde{G}_S$ and $\tilde{G}_T$ are the domains of σ and τ, respectively. A morphism $f : G \to H$ between partial triple graphs then is a triple $(f_S : G_S \to H_S, f_C : G_C \to H_C, f_T : G_T \to H_T)$ of graph morphisms such that both induced squares of partial morphisms commute. In our context, such a square with two opposed partial morphisms (as depicted as square (1) in Fig. 8) *commutes* if there exists a morphism $\tilde{f}_S : \tilde{G}_S \to \tilde{H}_S$ such that both arising squares (2) and (3) (compare Fig. 9) commute and, moreover, (2) is a pullback square. If $\tilde{f}_S$ exists, it is necessarily unique since ι_{H_S} is a monomorphism. This is stricter than, e.g., the weak commutativity used in [24] that does not require the square (2) to be a pullback square.

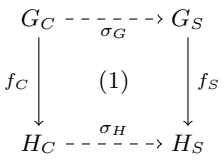


Fig. 8. Square of partial morphisms

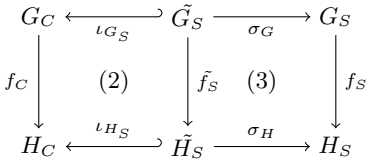


Fig. 9. Commuting square of partial morphisms

The next proposition states that the category of triple graphs is isomorphic to a full subcategory of the category of partial triple graphs.

Proposition 16. *The category* **TrG** *of triple graphs is isomorphic to a full subcategory of the category* **PTrG** *of partial triple graphs, i.e., there exists a full and faithful functor* $J : \mathbf{TrG} \to \mathbf{PTrG}$ *that is injective on objects. Moreover, rule application is functorial.*

In practical applications, the considered triple graphs are generally typed over a fixed triple graph. The next definition introduces typing of partial triple graphs over a fixed triple graph. For, e.g., synchronization scenarios as discussed in [7], it is convenient if the partial triple graphs are still typed over essentially the same triple graph as the original triple graph was.

Definition 17 (Typed partial triple graph). *The category of* triple graphs typed over a fixed triple graph $TG = (TG_S \leftarrow TG_C \rightarrow TG_T)$ *is the slice category* $\mathbf{TrG}/TG$*, denoted* $\mathbf{TrG_{TG}}$*. The category of* partial triple graphs typed over a fixed triple graph TG*, denoted* $\mathbf{PTrG_{TG}}$*, has as objects morphisms* $t_G : G \rightarrow J(TG)$ *from* $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ *where* J *is the inclusion functor and* G *is a partial triple graph. Morphisms are morphisms* $g : G \rightarrow H$ *from* $\mathbf{PTrG}$ *such that* $t_H \circ g = t_G$*.*

Example 18. Figure 3 (a) was already presented as a partial triple graph in Example 6. To consider it as still typed over the same type graph (not depicted) as the original triple graph from Fig. 2, one just restricts its typing morphism accordingly. Note that the resulting typing morphism is not S-cartesian but a morphism in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$. We therefore did not define typing of partial triple graphs just as slice category as well. We exemplify rules and morphisms in $\mathbf{PTrG_{TG}}$ using *Delta-Rule* depicted in Fig. 4: Its LHS consists of the elements depicted in black or in red (not annotated or annotated with $(--)$) while its interface only consists of the black elements. First, the mapping of the interface into the LHS is injective in every component. Moreover, all three correspondence nodes in the LHS of the rule are in the domain of the according correspondence morphism to the source side and the two correspondence nodes that are already part of the interface of the rule are in the domain of the according correspondence morphism as well. This means, the induced square of morphisms is a pullback square. The same holds for the target side. Hence, the morphism from the interface of *Delta-Rule* to its LHS is a morphism in $\mathbf{PTrG_{TG}}$.

In contrast, the morphism from the interface to the LHS of *Delta-Source-Rule* as depicted in Fig. 5 is not a morphism in $\mathbf{PTrG_{TG}}$: The according square $\tilde{K_S} \rightarrow K_C \rightarrow L_C \leftarrow \tilde{L_S} \leftarrow \tilde{K_S}$ is not a pullback square since the domain of the correspondence morphism to the source side in the interface graph only contains two elements and not three. Theorem 23 explains why this is not a hindrance to our desired application.

Proposition 19. *Typed partial triple graphs form an adhesive category. Moreover, the category* $\mathbf{TrG_{TG}}$ *of triple graphs typed over* TG *is isomorphic to a full subcategory of the category* $\mathbf{PTrG_{TG}}$ *of partial triple graphs typed over* TG *and rule application is functorial.*

We formulate the following results for the category $\mathbf{PTrG}$; they hold for categories $\mathbf{PTrG_{TG}}$ as well. The restriction to morphisms where certain squares are required to form pullback squares comes with some limitations. Namely, it is not possible to delete a reference (i.e., an element from the domain of a correspondence morphism) without deleting the referencing element (i.e., the according element in the correspondence graph), nor is it possible to create a reference from an already existing correspondence element. And for every matched correspondence element, a morphism also needs to match the according preimages in the domains of the two correspondence morphisms (if they exist) to become a valid match.

Proposition 20 (Characterizing valid rules and matches). *Let a rule* $p = (L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R)$ *and a morphism* $m : L \to G$ *in* $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ *be given where* $L, K, R,$ *and* G *are already objects from* **PTrG** *(i.e.,* $L = (L_S \overset{\sigma_L}{\hookleftarrow} \tilde{L_S} \overset{\iota_{L_S}}{\hookrightarrow} L_C \overset{\iota_{L_T}}{\hookleftarrow} \tilde{L_T} \overset{\tau_L}{\hookrightarrow} L_T)$ *and similar for* K, R, G*). Then* p *is a rule with match already in* **PTrG** *if and only if (compare Fig. 10 for notation)*

1. $\forall x \in L_C.((x \in \iota_{L_S}(\tilde{L_S}) \wedge x \in l_C(K_C)) \Rightarrow x \in l_C(\iota_{K_S}(\tilde{K_S})))$ *and analogously* $\forall x \in L_C.((x \in \iota_{L_T}(\tilde{L_T}) \wedge x \in l_C(K_C)) \Rightarrow x \in l_C(\iota_{K_T}(\tilde{K_T})))$,
2. $\forall x \in R_C.((x \in \iota_{R_S}(\tilde{R_S}) \wedge x \in r_C(K_C)) \Rightarrow x \in r_C(\iota_{K_S}(\tilde{K_S})))$ *and analogously* $\forall x \in R_C.((x \in \iota_{R_T}(\tilde{R_T}) \wedge x \in r_C(K_C)) \Rightarrow x \in r_C(\iota_{K_T}(\tilde{K_T})))$, *and*
3. $\forall x \in G_C.((\exists y_1 \in \tilde{G_S}.x = \iota_{G_S}(y_1) \wedge \exists y_2 \in L_C.x = m_C(y_2)) \Rightarrow \exists z \in \tilde{L_S}.(\tilde{m_S}(z) = y_1 \wedge \iota_{L_S}(z) = y_2))$ *and analogously* $\forall x \in G_C.((\exists y_1 \in \tilde{G_T}.x = \iota_{G_T}(y_1) \wedge \exists y_2 \in L_C.x = m_C(y_2)) \Rightarrow \exists z \in \tilde{L_T}.(\tilde{m_T}(z) = y_1 \wedge \iota_{L_T}(z) = y_2))$.

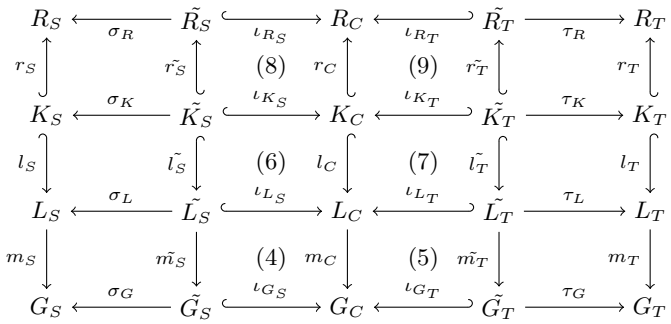


Fig. 10. Rule with match in **PTrG**

We now elementary characterize matches at which a rule in **PTrG** is applicable in the spirit of the gluing condition for graphs [5, Def. 3.9].

Proposition 21. *Let a rule* $p = (L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R)$ *and a match* $m : L \to G$ *in* **PTrG** *with* $L = (L_S \overset{\sigma_L}{\hookleftarrow} \tilde{L_S} \overset{\iota_{L_S}}{\hookrightarrow} L_C \overset{\iota_{L_T}}{\hookleftarrow} \tilde{L_T} \overset{\tau_L}{\hookrightarrow} L_T)$, $l = (l_S, \tilde{l_S}, l_C, \tilde{l_T}, l_T)$, *and* $m = (m_S, \tilde{m_S}, m_C, \tilde{m_T}, m_T)$ *be given. Then* p *is applicable at match* m *if and only if*

1. $m_S, m_C,$ *and* m_T *satisfy the* gluing condition*, i.e., none of these morphisms identifies an element that is to be deleted with another element and none of these morphisms determines a node to be deleted that has adjacent edges which are not to be deleted as well [5, Def. 3.9] and*
2. *for every referenced element in the source and the target graphs that is deleted the reference is deleted as well, i.e., for every element* $x \in G_S$ *that has a preimage in* L_S *under* m_S*, no preimage under* $m_S \circ l_S$ *in* K_S *but a preimage in* $\tilde{G_S}$ *under* σ_G*, there is an element* $y \in \tilde{L_S}$ *such that* $m_S(\sigma_L(y)) = x =$

$\sigma_G(\tilde{m_S}(y))$. Analogously, for every element $x \in G_T$ that has a preimage in L_T under m_T, no preimage under $m_T \circ l_T$ in K_T but a preimage in $\tilde{G_T}$ under τ_G, there is an element $y \in \tilde{L_T}$ such that $m_T(\tau_L(y)) = x = \tau_G(\tilde{m_T}(y))$.

Starting point for many practical applications of TGGs is the so-called *operationalization* of a rule which is a split of it into a *source* and a *forward rule* (or equivalently: a target and a backward rule). The source rule only performs the action of the original rule on the source part and the forward rule transfers this to the correspondence and the target part. A basic result states that applying a rule to a triple graph is equivalent to applying the source rule followed by an application of the forward rule. In the rest of this section, we present a comparable definition and result for partial triple graphs. However, we generalize the operationalization of rules in two directions: Our rules are rules on partial triple graphs instead of triple graphs and, moreover, they are allowed to be deleting, whereas classically the rules of a TGG are monotonic [27]. To be able to do this, we need to deviate slightly from the original construction. Our source rules perform the action of the original rule on the source side. Moreover, the deletion-action on the domain of the correspondence morphism to the source part is performed. All other actions are performed by the forward rule. In general, the resulting source and forward rules are not rules in **PTrG** any longer but in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$. However, the following theorem shows that the application of a rule in **PTrG** is equivalent to applying first the source and afterwards the forward rule (at a suitable match) in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$.

Definition 22 (Source and forward rule). *Let a rule $p = (L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R)$ in* **PTrG** *be given. Then its* source rule $p_S = (L^S \overset{l^S}{\hookleftarrow} K^S \overset{r^S}{\hookrightarrow} R^S)$ *is defined as depicted in Fig. 11 where $\emptyset$ denotes the empty graph. Its* forward rule $p_F = (L^F \overset{l^F}{\hookleftarrow} K^F \overset{r^F}{\hookrightarrow} R^F)$ *is defined as depicted in Fig. 12.*

$$\begin{array}{ccccccccc}
L^S & = & (L_S & \leftarrow & \tilde{L_S} & \hookrightarrow & L_C & \hookleftarrow & \emptyset & \rightarrow & \emptyset) \\
\uparrow l^S & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
K^S & = & (K_S & \leftarrow & \tilde{K_S} & \hookrightarrow & L_C & \hookleftarrow & \emptyset & \rightarrow & \emptyset) \\
\downarrow r^S & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
R^S & = & (R_S & \leftarrow & \tilde{K_S} & \hookrightarrow & L_C & \hookleftarrow & \emptyset & \rightarrow & \emptyset)
\end{array}$$

Fig. 11. Source rule p_S of a rule p

$$\begin{array}{ccccccccc}
L^F & = & (R_S & \leftarrow & \tilde{K_S} & \hookrightarrow & L_C & \hookleftarrow & \tilde{L_T} & \rightarrow & L_T) \\
\uparrow l^F & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
K^F & = & (R_S & \leftarrow & \tilde{K_S} & \hookrightarrow & K_C & \hookleftarrow & \tilde{K_T} & \rightarrow & K_T) \\
\downarrow r^F & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
R^F & = & (R_S & \leftarrow & \tilde{R_S} & \hookrightarrow & R_C & \hookleftarrow & \tilde{R_T} & \rightarrow & R_T)
\end{array}$$

Fig. 12. Forward rule p_F of a rule p

Theorem 23. *Let a rule $p = (L \overset{l}{\hookleftarrow} K \overset{r}{\hookrightarrow} R)$ in* **PTrG** *with source and forward rules $p_S = (L^S \overset{l^S}{\hookleftarrow} K^S \overset{r^S}{\hookrightarrow} R^S)$ and $p_F = (L^F \overset{l^F}{\hookleftarrow} K^F \overset{r^F}{\hookrightarrow} R^F)$ be given.*

1. *Given a direct transformation $G \Rightarrow_{p,m} H$ in* **PTrG**, *there is a transformation sequence $G \Rightarrow_{p_S,m} G' \Rightarrow_{p_F,n} H$ in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$.*

2. *Given transformation steps* $G \Rightarrow_{p_S,m} G' \Rightarrow_{p_F,n} H$ *in* $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ *where* n *coincides with the comatch of the first transformation step on source and correspondence graph and* G *and* m *are already elements of* **PTrG**, *there is a direct transformation* $G \Rightarrow_{p,m} H$ *in* **PTrG**.

Example 24. Delta-Rule as depicted in Fig. 4 is split by our construction into *Delta-Source-Rule* and *Delta-Forward-Rule* as depicted in Fig. 5. Applying *Delta-Rule* to the triple graph from Fig. 2 such that nodes `p` and `f` are deleted gives the same result as first applying *Delta-Source-Rule* at the according match, which gives the partial triple graph from Fig. 3 (a) as intermediate result, and *Delta-Forward-Rule* subsequently (with consistent match). Namely, both yield the triple graph from Fig. 3 (c).

6 Related Works

In [7] we have already used partial triple graphs to develop an optimized synchronization process for correlated models where the correspondence relationship has been formalized using TGGs. However, we only defined rule application for special cases, only obtained a very restricted version of Theorem 23, and generally left the thorough investigation of that category to future work.

The work that is most closely related to ours with regard to the formal content is the introduction of a new concept of attribution by Kastenberg and Rensink in [17]. They use subgraphs for attribution and prove that the category of reflected monos ($[\bullet \hookrightarrow \bullet, \mathcal{C}]$ in our notation) is adhesive if $\mathcal{C}$ is. Since proving this for an arbitrary S-cartesian subcategory $[\mathcal{X}_S, \mathcal{C}]$ reduces to inspection of a single naturality square at a morphism $m \in S$, our proofs are similar in places. However, they do not consider the case of $\mathcal{C}$ being adhesive HLR and do not relate to the full functor category $[\bullet \rightarrow \bullet, \mathcal{C}]$.

Golas et al. provide a formalization of TGGs in [11] which allows to generalize correspondence relations between source and target graphs as well. They use special typings for the source, target, and correspondence parts of a triple graph and introduce edges between correspondence and source and target nodes instead of using graph morphisms. Hence, they allow for even more flexible correspondence relations and for more flexible deletion and creation of references than possible in our approach. In contrast, we allow for references also between edges and are more in line with the standard formalization of TGGs.

Double pushout rewriting of graph transformation rules by so-called 2-rules has been extensively studied by Machado et al. in [23]. They, too, identify the problem that applying a 2-rule to a rule does not need to result in a rule again since the resulting morphisms are not necessarily injective. Instead of restricting the allowed morphisms, they equip their 2-rules by suitable application conditions. However, their approach is specific to rewriting of graph transformation rules and not directly generalizable to a purely categorical setting. It is not difficult to see that, instantiated for typed graphs, their framework is more general than ours: None of the involved morphisms needs to form a pullback square for a

2-rule to be applicable and result in a rule again. This evokes the research question whether it is possible to increase the classes of morphisms in the categories we presented and still obtain categories that are adhesive HLR.

In contrast to TGGs, where correspondence between elements is defined by total morphisms, partial morphisms have already been used to formalize the correspondence of elements in situations where more than two meta-models are involved [19,29]. As mentioned in Sect. 1, this can also be seen as an instantiation of our general framework. In our practical application to partial triple graphs, we are interested in allowing for partial correspondence morphisms to obtain a more incremental synchronization process. Overall correspondence is still defined via total morphisms.

Partial morphisms have long been a research topic in the area of graph transformation, in particular in connection with the single pushout approach to graph transformation as, e.g., in [18,22,4]. Moreover, there has been research computing limits in categories of partial morphisms [28] or relating properties of pushouts in a category to properties of a pushout in the according category of partial morphisms [15,14]. In this line of research, one enlarges the class of morphisms of a given category by considering also partial morphisms, whereas our framework allows to consider partial morphisms as objects but pushouts and pullbacks are still computed along total morphisms (componentwise).

In [6], Ehrig et al. also consider certain functors as objects of a new category to model distributed objects. They prove (co-)completeness in case the base category is. However, their functors allow for change in the category (or graph) from which the functor starts and the considered morphisms are accordingly quite different from ours.

7 Conclusion

In this paper, we present a new way to construct a category that is adhesive HLR out of a given one, namely as a certain subcategory of a functor category. This construction unifies several categories for which rewriting has been discussed separately so far. As a new application case, we present a category of partial triple graphs. This inspection (as well as comparison to another approach to rewriting rules) shows that, while still interesting in practice, the restriction to a certain kind of morphisms comes with a price. Searching for a (categorical) way to relax this restriction is interesting future work. Moreover, we plan to apply our formal framework to other instances, e.g., to rewriting of constraints.

Acknowledgments We would like to thank the anonymous reviewers for their valuable feedback. This work was partially funded by the German Research Foundation (DFG), project “Triple Graph Grammars (TGG) 2.0”.

References

1. Abou-Saleh, F., Cheney, J., Gibbons, J., McKinna, J., Stevens, P.: Introduction to bidirectional transformations. In: Gibbons, J., Stevens, P. (eds.) Bidirectional

Transformations – International Summer School. Lecture Notes in Computer Science, vol. 9715, pp. 1–28. Springer (2018)
2. Awodey, S.: Category Theory, Oxford Logic Guides, vol. 52. Oxford University Press, Inc., New York, NY, USA, 2nd edn. (2010)
3. Czarnecki, K., Foster, J.N., Hu, Z., Lämmel, R., Schürr, A., Terwilliger, J.F.: Bidirectional transformations: A cross-discipline perspective. In: Paige, R.F. (ed.) Theory and Practice of Model Transformations. pp. 260–283. Springer, Berlin (2009)
4. Ehrig, H., Heckel, R., Korff, M., Löwe, M., Ribeiro, L., Wagner, A., Corradini, A.: Algebraic Approaches to Graph Transformation – Part II: Single Pushout Approach and Comparison with Double Pushout Approach. In: Rozenberg, G. (ed.) Handbook of Graph Grammars and Computing by Graph Transformation, chap. 4, pp. 247–312. World Scientific, Singapore (1997)
5. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. Monographs in Theoretical Computer Science, Springer (2006)
6. Ehrig, H., Orejas, F., Prange, U.: Categorical foundations of distributed graph transformation. In: Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G. (eds.) Graph Transformations. pp. 215–229. Springer, Berlin (2006)
7. Fritsche, L., Kosiol, J., Schürr, A., Taentzer, G.: Efficient Model Synchronization by Automatically Constructed Repair Processes. In: van der Aalst, W., Hähnle, R. (eds.) Proceedings of FASE '19. pp. 1–18. No. 11424 in LNCS (2019)
8. Fritsche, L., Kosiol, J., Schürr, A., Taentzer, G.: Short-Cut Rules. Sequential Composition of Rules Avoiding Unnecessary Deletions. In: Mazzara, M., Ober, I., Salaün, G. (eds.) Software Technologies: Applications and Foundations. pp. 415–430. Springer International Publishing, Cham (2018)
9. Giese, H., Hildebrandt, S.: Efficient model synchronization of large-scale models. Tech. Rep. 28, Hasso-Plattner-Institut (2009)
10. Giese, H., Wagner, R.: From model transformation to incremental bidirectional model synchronization. Software & Systems Modeling **8**(1), 21–43 (Feb 2009)
11. Golas, U., Lambers, L., Ehrig, H., Giese, H.: Toward bridging the gap between formal foundations and current practice for triple graph grammars. In: Ehrig, H., Engels, G., Kreowski, H.J., Rozenberg, G. (eds.) Graph Transformations. pp. 141–155. Springer, Berlin (2012)
12. Greenyer, J., Pook, S., Rieke, J.: Preventing information loss in incremental model synchronization by reusing elements. In: France, R.B., Kuester, J.M., Bordbar, B., Paige, R.F. (eds.) Modelling Foundations and Applications. pp. 144–159. Springer, Berlin (2011)
13. Habel, A., Pennemann, K.H.: Correctness of High-Level Transformation Systems Relative to Nested Conditions. Mathematical Structures in Computer Science **19**, 245–296 (2009)
14. Hayman, J., Heindel, T.: On pushouts of partial maps. In: Giese, H., König, B. (eds.) Graph Transformation. pp. 177–191. Springer International Publishing, Cham (2014)
15. Heindel, T.: Hereditary pushouts reconsidered. In: Ehrig, H., Rensink, A., Rozenberg, G., Schürr, A. (eds.) Graph Transformations. pp. 250–265. Springer, Berlin (2010)
16. Hermann, F., Ehrig, H., Orejas, F., Czarnecki, K., Diskin, Z., Xiong, Y., Gottmann, S., Engel, T.: Model synchronization based on triple graph grammars: correctness, completeness and invertibility. Software & Systems Modeling **14**(1), 241–269 (Feb 2015)

17. Kastenberg, H., Rensink, A.: Graph Attribution Through Sub-Graphs. In: Heckel, R., Taentzer, G. (eds.) Graph Transformation, Specifications, and Nets: In Memory of Hartmut Ehrig, pp. 245–265. Springer International Publishing, Cham (2018)
18. Kennaway, R.: Graph rewriting in some categories of partial morphisms. In: Ehrig, H., Kreowski, H.J., Rozenberg, G. (eds.) Graph Grammars and Their Application to Computer Science. pp. 490–504. Springer, Berlin (1991)
19. König, H., Diskin, Z.: Efficient consistency checking of interrelated models. In: Anjorin, A., Espinoza, H. (eds.) Modelling Foundations and Applications. pp. 161–178. Springer International Publishing, Cham (2017)
20. Lack, S., Sobociński, P.: Adhesive and quasiadhesive categories. Theoretical Informatics and Applications **39**(3), 511–545 (2005)
21. Lauder, M., Anjorin, A., Varró, G., Schürr, A.: Efficient model synchronization with precedence triple graph grammars. In: Ehrig, H., Engels, G., Kreowski, H.J., Rozenberg, G. (eds.) Graph Transformations. pp. 401–415. Springer, Berlin (2012)
22. Löwe, M.: Algebraic approach to single-pushout graph transformation. Theoretical Computer Science **109**(1), 181–224 (1993)
23. Machado, R., Ribeiro, L., Heckel, R.: Rule-based transformation of graph rewriting rules: Towards higher-order graph grammars. Theoretical Computer Science **594**, 1–23 (2015)
24. Montanari, U., Ribeiro, L.: Linear Ordered Graph Grammars and Their Algebraic Foundations. In: Corradini, A., Ehrig, H., Kreowski, H.J., Rozenberg, G. (eds.) Graph Transformation. pp. 317–333. Springer, Berlin (2002)
25. Orejas, F., Lambers, L.: Symbolic Attributed Graphs for Attributed Graph Transformation. Electronic Communications of the EASST **30**(International Colloquium on Graph and Model Transformation (GraMoT) 2010) (2010)
26. Robinson, E., Rosolini, G.: Categories of partial maps. Information and Computation **79**(2), 95–130 (1988)
27. Schürr, A.: Specification of graph translators with triple graph grammars. In: Mayr, E.W., Schmidt, G., Tinhofer, G. (eds.) Graph-Theoretic Concepts in Computer Science. Lecture Notes in Computer Science, vol. 903, pp. 151–163. Springer (1995)
28. Shir Ali Nasab, A.R., Hosseini, S.N.: Pullback in partial morphism categories. Applied Categorical Structures **25**(2), 197–225 (2017)
29. Stünkel, P., König, H., Lamo, Y., Rutle, A.: Multimodel correspondence through inter-model constraints. In: Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming. pp. 9–17. ACM, New York (2018)

A Proofs and Additional Preliminaries

A.1 Additional Preliminaries

Amongst others, adhesive and adhesive HLR categories have the following properties. We restrict to those that we will use in our proofs.

Fact 25 (Properties of adhesive HLR categories). *If $\boldsymbol{C}$ is an adhesive (HLR) category, the following properties hold [20,5]:*

1. *Monomorphisms ($\mathcal{M}$-morphisms) are stable under pushout and pushouts along monomorphisms ($\mathcal{M}$-morphisms) are pullbacks.*
2. *If m in Fig. 6 is a monomorphism ($\mathcal{M}$-morphism), pushout complements for $g \circ m$ are unique (up to isomorphism).*
3. *If $\mathcal{C}$ is adhesive, it is balanced, i.e., each morphism that is a mono- and an epimorphism is already an isomorphism. Analogously, if $\mathcal{C}$ is adhesive HLR, each morphism that is an $\mathcal{M}$- and an epimorphism is an isomorphism.*
4. *If $\mathcal{C}$ is adhesive, every functor category $[\mathcal{X}, \mathcal{C}]$ where $\mathcal{X}$ is small and every slice or co-slice category $\mathcal{C}/C$ or $C/\mathcal{C}$ for an object C of $\mathcal{C}$ is adhesive. Analogously, if $\mathcal{C}$ is adhesive HLR, every functor category $[\mathcal{X}, \mathcal{C}]$ where $\mathcal{X}$ is small and every slice or co-slice category $\mathcal{C}/C$ or $C/\mathcal{C}$ for an object C of $\mathcal{C}$ is adhesive HLR. Here, the new class of $\mathcal{M}$-morphisms is given by natural transformations that consist of $\mathcal{M}$-morphisms componentwise, or the restriction of $\mathcal{M}$ to the morphisms in $\mathcal{C}/C$ or $C/\mathcal{C}$, respectively.*

In [20], the balancedness is proven for adhesive categories but the same proof works for adhesive HLR categories.

A.2 Proofs

Proof (of Proposition 7). Let F be an arbitrary S-functor from $\mathcal{X}$ to $\mathcal{C}$. The identity natural transformation is a morphism in $[\mathcal{X}_S, \mathcal{C}]$ since for every $S \ni m : x \to y$, the square depicted in Fig. 13 is a pullback square.

Given two S-cartesian natural transformations $\sigma : F \to G$ and $\tau : G \to H$, the composition $\tau \circ \sigma$ is S-cartesian since for every $S \ni m : x \to y$, the composition of the two pullbacks $(1a)$ and $(1b)$ depicted in Fig. 14 is a pullback.

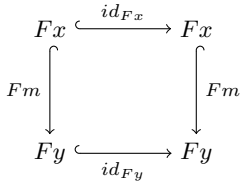


Fig. 13. Identity morphism in $[\mathcal{X}_S, \mathcal{C}]$

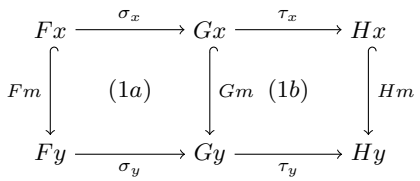


Fig. 14. Composition of morphisms in $[\mathcal{X}_S, \mathcal{C}]$

Moreover, mapping every S-functor from $[\mathcal{X}_S, \mathcal{C}]$ to itself as object in $[\mathcal{X}, \mathcal{C}]$ and every S-cartesian natural transformation from $[\mathcal{X}_S, \mathcal{C}]$ to itself as morphism in $[\mathcal{X}, \mathcal{C}]$ gives a faithful functor that is also injective on objects. □

Proof (of Proposition 8). If a (co)limit of a certain type exists in the category $\mathcal{C}$, a functor category $[\mathcal{X}, \mathcal{C}]$ has (co)limits of this type as well and they are computed componentwise [2]. Hence, if $\mathcal{C}$ has pullbacks along $\mathcal{M}$-morphisms, $[\mathcal{X}, \mathcal{C}]$ has pullbacks along morphisms where every component is an $\mathcal{M}$-morphism and they are computed componentwise. It remains to show, that the inclusion functor I creates these.

Let two S-cartesian natural transformations $\sigma : B \to D$ and $\tau : C \to D$ between S-functors B, C, D be given and let every component of τ be an $\mathcal{M}$-morphism. Since I is an inclusion, we have to show that the componentwise computed pullback A in $[\mathcal{X}, \mathcal{C}]$ with natural transformations $\eta : A \to B$ and $\mu : A \to C$ is a pullback in $[\mathcal{X}_S, \mathcal{C}]$ as well. This means, we have to show (i) that A is an S-functor and that η and μ are S-cartesian and (ii) that given another S-functor Q with S-cartesian natural transformations $\nu_1 : Q \to B$ and $\nu_2 : Q \to C$ such that $\sigma \circ \nu_1 = \tau \circ \nu_2$ the unique mediating natural transformation $\epsilon : Q \to A$, guaranteed to exist in $[\mathcal{X}, \mathcal{C}]$, is S-cartesian as well (i.e., is a morphism in $[\mathcal{X}_S, \mathcal{C}]$).

To show (i), let $S \ni m : x \to y$ be an arbitrary morphism. Computing the pullbacks of the co-spans $Bx \to Dx \leftarrow Cx$ and $By \to Dy \leftarrow Cy$ leads to the left cube depicted in Fig. 15. These pullbacks exist in $\mathcal{C}$ by assumption. The morphism Am is the unique morphism determined by the universal property of the pullback square at the bottom of the cube. Since the front faces are pullbacks (by assumption) and the top and the bottom square are computed as pullbacks, the faces to the back are pullbacks as well. Moreover, the morphism Am is an $\mathcal{M}$-morphism since it arises as pullback of the $\mathcal{M}$-morphism Bm (or, equivalently, Cm). Hence, A is an S-functor and η and μ are S-cartesian.

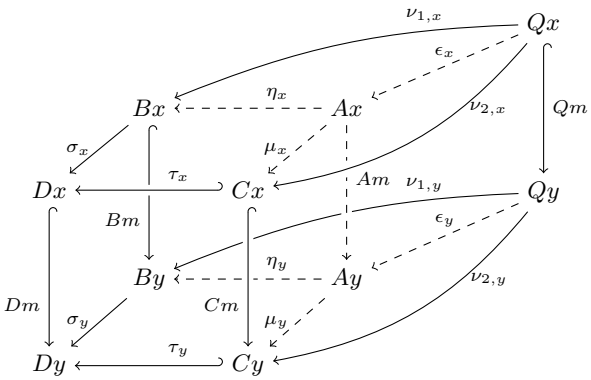


Fig. 15. Pullback of a cartesian naturality

(ii) Since the square $Qx \to Cx \to Cy \leftarrow Qy \leftarrow Qx$ is a pullback by assumption and $Ax \to Cx \to Cy \leftarrow Ay \leftarrow Ax$ is a pullback as well, pullback decomposition implies that the square $Qx \to Ax \to Ay \leftarrow Qy \leftarrow Qx$ is also a pullback. Hence, ϵ is an S-cartesian natural transformation. □

Proof (of Lemma 9). Every morphism in a functor category $[\mathcal{X}, \mathcal{C}]$ that is a monomorphism componentwise is a monomorphism and the converse holds if $\mathcal{C}$ has pullbacks [2]. Since I is a faithful functor from $[\mathcal{X}_S, \mathcal{C}]$ to $[\mathcal{X}, \mathcal{C}]$ and faithful functors reflect monomorphisms, a morphism in $[\mathcal{X}_S, \mathcal{C}]$ that has only monomorphisms as components is a monomorphism. In particular, every morphism in $[\mathcal{X}, \mathcal{C}]$ or in $[\mathcal{X}_S, \mathcal{C}]$ where each component is an $\mathcal{M}$-morphism is a monomorphism (independent of the existence of all pullbacks in $\mathcal{C}$).

Provided pullbacks in $\mathcal{C}$ and given a monomorphism $\mu : A \to B$ in $[\mathcal{X}_S, \mathcal{C}]$, the span $A \xleftarrow{id_A} A \xrightarrow{id_A} A$ is a pullback of μ along itself [2]. By Proposition 8, this means that for every $x \in \mathcal{X}$ the pullback of $Ax \xrightarrow{\mu_x} Bx \xleftarrow{\mu_x} Ax$ in $\mathcal{C}$ is given by $Ax \xleftarrow{id_{Ax}} Ax \xrightarrow{id_{Ax}} Ax$. By the same characterization of monomorphisms as used above, this implies that every component μ_x is a monomorphism. □

Proof (of Proposition 10). Like in the proof of Proposition 8, because $\mathcal{C}$ has pushouts along $\mathcal{M}$-morphisms, $[\mathcal{X}, \mathcal{C}]$ has pushouts along morphisms where every component is an $\mathcal{M}$-morphism and they are computed componentwise. It remains to show, that the inclusion functor I creates these.

Let two S-cartesian natural transformations $\sigma : A \to B$ and $\tau : A \to C$ between S-functors A, B, C be given and let every component of τ be an $\mathcal{M}$-morphism. Since I is an inclusion, we have to show that the componentwise computed pushout D in $[\mathcal{X}, \mathcal{C}]$ with natural transformations $\eta : B \to D$ and $\mu : C \to D$ is a pushout in $[\mathcal{X}_S, \mathcal{C}]$ as well. This means, we have to show (i) that D is an S-functor and that η and μ are S-cartesian and (ii) that given another S-functor Q with S-cartesian natural transformations $\nu_1 : B \to Q$ and $\nu_2 : C \to Q$ such that $\nu_1 \circ \sigma = \nu_2 \circ \tau$ the unique mediating natural transformation $\epsilon : D \to Q$, guaranteed to exist in $[\mathcal{X}, \mathcal{C}]$, is S-cartesian as well (i.e., is a morphism in $[\mathcal{X}_S, \mathcal{C}]$).

To show (i), let $S \ni m : x \to y$ be an arbitrary morphism. Computing the pushouts of the spans $Bx \leftarrow Ax \to Cx$ and $By \leftarrow Ay \to Cy$ leads to the left cube depicted in Fig. 16. These pushouts exist, since $\mathcal{C}$ is adhesive HLR and $\tau_x, \tau_y \in \mathcal{M}$ by assumption. The morphism Dm is the unique morphism determined by the universal property of the pushout square at the top of the cube. That the front faces are pullbacks and Dm is an $\mathcal{M}$-morphism is a direct consequence of the bottom square being a partial van Kampen square. Hence, D is an S-functor and η and μ are S-cartesian.

(ii) It remains to show that $Dx \to Qx \to Qy \leftarrow Dy \leftarrow Dx$ is a pullback square. To see this, first compute the pullback Dx' of $Qx \hookrightarrow Qy \leftarrow Dy$ which exists since $Qm \in \mathcal{M}$. The universal property if this pullback implies the existence of a morphism $Dm' : Dx \to Dx'$ such that $Dm = Dm'' \circ Dm'$ and $\epsilon_x = \epsilon_{x'} \circ Dm'$. We show that Dm' is an isomorphism. For this, by balancedness of $\mathcal{C}$, it is enough to show that $Dm' \in \mathcal{M}$ and that Dm' is an epimorphism.

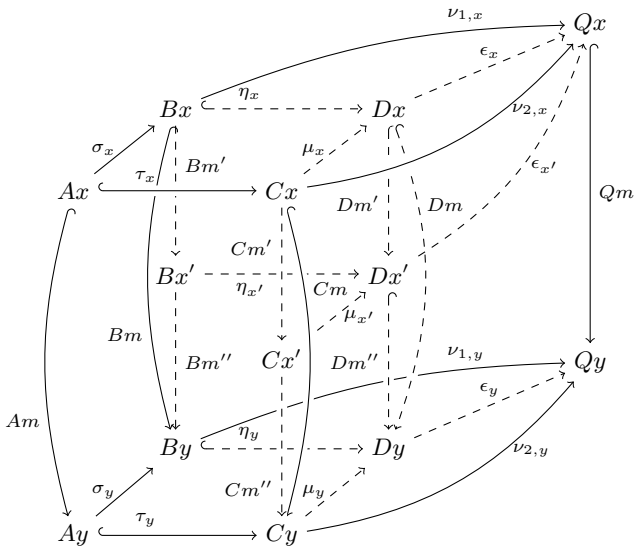


Fig. 16. Pushout of a cartesian naturality square

We already stated that $Dm \in \mathcal{M}$ and, moreover, $Dm'' \in \mathcal{M}$ since it arises by pullback along the $\mathcal{M}$-morphism Qm. By decomposition of $\mathcal{M}$-morphisms, $Dm' \in \mathcal{M}$.

To see that Dm' is an epimorphism, compute the pullbacks Cx' of $Dx' \hookrightarrow Dy \leftarrow Cy$ and Bx' of $Dx' \hookrightarrow Dy \leftarrow By$ (which both exist since $Dm'' \in \mathcal{M}$). Since $Cx \to Qx \to Qy \leftarrow Cy \leftarrow Cx$ is a pullback, its universal property implies the existence of a unique morphism $C\bar{m}' : Cx' \to Cx$ with $\nu_{2,x} \circ C\bar{m}' = \epsilon_{x'} \circ \mu_{x'}$ and $Cm \circ C\bar{m}' = Cm''$. One uses this to first calculate $Cm'' \circ Cm' \circ C\bar{m}' = Cm \circ C\bar{m}' = Cm''$. Moreover,

$$\begin{aligned} Dm'' \circ \mu_{x'} \circ Cm' \circ C\bar{m}' &= \mu_y \circ Cm'' \circ Cm' \circ C\bar{m}' \\ &= \mu_y \circ Cm \circ C\bar{m}' \\ &= \mu_y \circ Cm'' \\ &= Dm'' \circ \mu_{x'} \end{aligned}$$

Since Dm'' is a monomorphism, this implies $\mu_{x'} \circ Cm' \circ C\bar{m}' = \mu_{x'}$. But $id_{C_{x'}}$ is the unique morphism with these two properties and hence $id_{C_{x'}} = \circ Cm' \circ C\bar{m}'$. In particular, Cm' is an epimorphism. Analogously, there exists a morphism $B\bar{m}' : Bx' \to Bx$ that can be used to show that Bm' is an epimorphism. Moreover, since the bottom square is a van Kampen square, the morphisms $\mu_{x'}$ and $\eta_{x'}$ are jointly epimorphic (as coprojections of a pushout). Since this is a composition with epimorphisms, $\eta_{x'} \circ Bm'$ and $\mu_{x'} \circ Cm'$ are jointly epimorpic as well.

To finally show that Dm' is an epimorphism, let $f, g : Dx' \to Z$ be two morphisms (in $\mathcal{C}$) with $f \circ Dm' = g \circ Dm'$. Then

$$\begin{aligned} f \circ \mu_{x'} \circ Cm' &= f \circ Dm' \circ \mu_x \\ &= g \circ Dm' \circ \mu_x \\ &= g \circ \mu_{x'} \circ Cm' \end{aligned}$$

and analogously $f \circ \eta_{x'} \circ Bm' = g \circ \eta_{x'} \circ Bm'$. Hence, $f = g$ and Dm' is an epimorphism. □

Proof (of Proposition 12). That a rule application in $[\mathcal{X}_S, \mathcal{C}]$ is a rule application in $[\mathcal{X}, \mathcal{C}]$ is a corollary to Proposition 10 and Theorem 11.

Let the rule p be applicable to G at match μ in $[\mathcal{X}, \mathcal{C}]$. Let D be the pushout complement computed in the first step of the rule application and $\kappa : K \to D$ and $\phi : D \to G$ the resulting morphisms. It suffices to show that κ and ϕ are S-cartesian and that D is an S-functor. Then, the pushout complement is a pushout complement in $[\mathcal{X}_S, \mathcal{C}]$ as well. That computing the second pushout of the rule application is the same in $[\mathcal{X}_S, \mathcal{C}]$ or in $[\mathcal{X}, \mathcal{C}]$ is, again, stated by Proposition 10.

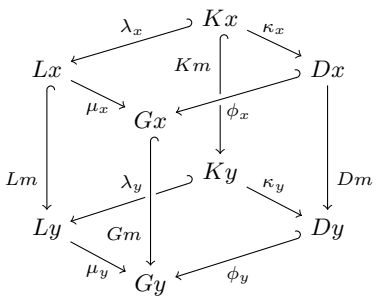


Fig. 17. Pushout complement for one component

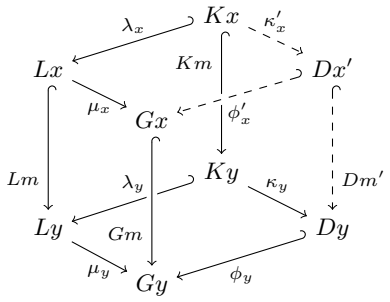


Fig. 18. Pullback resulting in pushout complement

Let $\mathcal{M} \ni m : x \to y$ be arbitrary. Figure 17 depicts the pushout complement at the components x and y with morphism $Dm : Dx \to Dy$ which exist by assumption. Since λ and μ are S-cartesian, the two vertical squares to the left are pullbacks and hence, $Kx \to Gx \to Gy \leftarrow Ky \leftarrow Kx$ is a pullback. Hence, if we show that $Dx \to Gx \to Gy \leftarrow Dy \leftarrow Dx$ is a pullback square, $Kx \to Ky \to Dy \leftarrow Dx \leftarrow Kx$ is a pullback square as well by pullback decomposition and Dm is an $\mathcal{M}$-morphism since it arises as pullback along the $\mathcal{M}$-morphism Gm. Computing the pullback of the co-span $Gx \to Gy \leftarrow Dy$ (which exists in $\mathcal{C}$ since Gm is an $\mathcal{M}$-morphism) results in an object Dx' with $\mathcal{M}$-morphism $Dm' : Dx' \hookrightarrow Dy$. Then, the pullback of κ_y along Dm' exists and results (up to

isomorphism) in Kx again since it completes the pullback of the co-span $Gx \hookrightarrow Gy \leftarrow Ky$. Since the bottom square is a van Kampen square in $\mathcal{C}$ and all vertical faces are pullbacks, the top square is a pushout. Since pushout complements are unique in categories that are adhesive HLR, there is an isomorphism between Dx' and Dx that is compatible with ϕ_x and ϕ'_x. Hence, the square at the right front is a pullback. □

Proof (of Proposition 16). Let a triple graph $G = (G_S \xleftarrow{\sigma_G} G_C \xrightarrow{\tau_G} G_T)$ be given. Define the functor J to map G to the partial triple graph $G = (G_S \xleftarrow{\sigma_G} G_C \xhookrightarrow{id_{G_C}} G_C \xhookleftarrow{id_{G_C}} G_C \xrightarrow{\tau_G} G_T)$. This is clearly injective on objects and id_{G_C} is a monomorphism. The components (f_S, f_C, f_T) of a morphism between triple graphs $G = (G_S \xleftarrow{\sigma_G} G_C \xrightarrow{\tau_G} G_T)$ and $H = (H_S \xleftarrow{\sigma_H} H_C \xrightarrow{\tau_H} H_T)$ are mapped by J to the quintuple $(f_S, f_C, f_C, f_C, f_T)$. As depicted in Fig. 19, f_C makes squares (2) and (3) (since (f_C, f_T) is a morphism between τ_G and τ_H) commute and (2) into a pullback square; the situation on the source part is analogous. Moreover, this mapping is injective. And every morphism $(f_S, f_C, f_C, f_C, f_T)$ between partial triple graphs $G = (G_S \xleftarrow{\sigma_G} G_C \xhookrightarrow{id_{G_C}} G_C \xhookleftarrow{id_{G_C}} G_C \xrightarrow{\tau_G} G_T)$ and $H = (H_S \xleftarrow{\sigma_H} H_C \xhookrightarrow{id_{H_C}} H_C \xhookleftarrow{id_{H_C}} H_C \xrightarrow{\tau_H} H_T)$ where the three central objects are identical (and connected by the identity morphism) has (f_S, f_C, f_T) as a preimage under J. Hence the mapping is full.

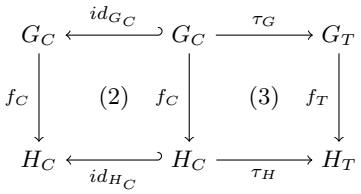


Fig. 19. Mapping a morphism between triple graphs to one between partial triple graphs

That rule application is functorial, is a consequence of Proposition 10: Pushout complements (if they exist) and pushouts along monomorphisms are computed componentwise. Hence, the pushout complement for $J(m) \circ J(l)$, with m, l being ordinary triple graph morphisms, can be chosen as $D_S \xleftarrow{\sigma_D} D_C \xhookrightarrow{id_{D_C}} D_C \xhookleftarrow{id_{D_C}} D_C \xrightarrow{\tau_G} D_T$ where $D_S \xleftarrow{\sigma_D} D_C \xrightarrow{\tau_G} D_T$ is the according pullback complement for $m \circ l$ and given a pullback complement $D_S \xleftarrow{\sigma_D} D_C \xrightarrow{\tau_G} D_T$ for $m \circ l$, then $D_S \xleftarrow{\sigma_D} D_C \xhookrightarrow{id_{D_C}} D_C \xhookleftarrow{id_{D_C}} D_C \xrightarrow{\tau_G} D_T$ is a pullback complement for $J(m) \circ J(l)$. The same holds for pushouts. □

Proof (of Proposition 19). Since partial triple graphs are a category and, hence, identity morphisms exist and composition of morphisms is possible, typed par-

tial triple graphs are a subcategory of the slice category $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]/TG$.

This category is adhesive since in a slice category, pushouts and pullbacks are just computed along the morphisms and the typing is induced by the universal property of the pushout or by composition of morphisms, respectively [5]. This means, e.g., the pushout of morphisms $f : t_G \to t_H$ and $g : t_G \to t_K$ in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]/TG$ is given by t_D where D arises as pushout object of $H \xleftarrow{f} G \xrightarrow{g} K$ in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ and the typing morphism $t_D : D \to TG$ is induced by D's universal property. In this case, the morphisms stem from **PTrG** and by Propositions 8 and 10, computing pullbacks and pushouts along morphims from **PTrG** in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ is the same as computing them directly in **PTrG**. Hence, adhesivity is inherited from **PTrG**.

The rest of the proof is equivalent to the one of Proposition 16. □

Proof (of Proposition 20). By assumption, all squares in Fig. 10 are commuting. Moreover, the hooked morphisms are injective. Hence, all the statements are just translations of the requirement that the squares (4) – (9) are pullback squares into its set-theoretic meaning. □

Proof (of Proposition 21). First, let the rule p be applicable at match m. Since pushouts are computed componentwise in **PTrG**, so are pushout complements. In particular, pushout complements exist for $m_S \circ l_S, m_C \circ l_C$, and $m_T \circ l_T$. Since these are graph morphisms, the existence of the pushout complement implies that the gluing condition is fulfilled [5]. Moreover, there exist morphisms $\sigma_K : \tilde{K_S} \to K_S$ and $\sigma_D : \tilde{D_S} \to D_S$ making the left cube in Fig. 20 commute. Let $x \in G_S$ be such that there is a preimage z_1 for x in L_S and z_2 for x in $\tilde{G_S}$. Since the central vertical square is a pushout square as well, z_2 either has a preimage in $\tilde{L_S}$ or in $\tilde{D_S}$. Assume it to only have in preimage z_3 in $\tilde{D_S}$. Then $d_S(\sigma_D(z_3)) = x$, thus x has a preimage in L_S and in D_S and since the left vertical square is a pushout, there is a preimage for $\sigma_D(z_3)$ in K_S which contradicts the assumption that x has no preimage under $m_S \circ l_S$.

Secondly, let the two conditions be fulfilled. In particular, the pushout complement object D_C exists. Compute the pullback of the co-span $\tilde{G_S} \hookrightarrow G_C \hookleftarrow D_C$. A morphism $\tilde{k_S} : \tilde{K_S} \to \tilde{D_S}$ is obtained by the universal property of this pullback. Moreover, since the vertical square to the right is a pushout and hence a van Kampen square, the arising vertical square in the center is a pushout square as well (all side faces are pullbacks). It remains to show that there exists a morphism $\sigma_D : \tilde{D_S} \to D_S$ making the left cube commute. Since d_S is injective, the only possible choice to define σ_D such that the bottom square commutes is

$$\sigma_D(x) := d_S^{-1}(\sigma_G(\tilde{d_S}(x))) \ .$$

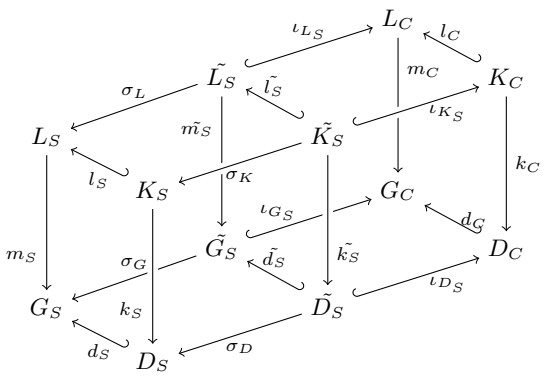


Fig. 20. Characterizing applicability of a rule at a match

Given this definition one calculates

$$\begin{aligned} d_S \circ k_S \circ \sigma_K &= m_S \circ l_S \circ \sigma_K \\ &= m_S \circ \sigma_L \circ \tilde{l_S} \\ &= \sigma_G \circ \tilde{m_S} \circ \tilde{l_S} \\ &= \sigma_G \circ \tilde{d_S} \circ \tilde{k_S} \\ &= d_S \circ \sigma_D \circ \tilde{k_S} \end{aligned}$$

which implies $k_S \circ \sigma_K = \sigma_D \circ \tilde{k_S}$ and hence commutativity of the whole cube, provided that such a σ_D exists. Assume an $x \in G_S$ to exist, that has a preimage z_1 under $\sigma_G \circ \tilde{d_S}$ but no preimage in D_S. Since the left vertical square is a pushout, then x has a preimage z_2 in L_S but z_2 does not have in preimage in K_S. Since $\tilde{d_S}(z_1)$ is a preimage for x in $\tilde{G_S}$, by assumption there is a preimage z_3 for z_2 and $\tilde{d_S}(z_1)$ in $\tilde{L_S}$. Since the vertical square in the center is a pushout, there exists $z_4 \in \tilde{K_S}$ with $\tilde{l_S}(z_4) = z_3$ and $\tilde{k_S}(z_4) = z_1$. By commutativity of the involved squares

$$\begin{aligned} d_S(k_S(\sigma_K(z_4))) &= m_S(l_S(\sigma_K(z_4))) \\ &= m_S(\sigma_L(\tilde{l_S}(z_4))) \\ &= m_S(\sigma_L(z_3)) \\ &= m_S(z_2) \\ &= x \end{aligned}$$

which is a contradiction to x having no preimage under d_S. □

Proof (of Theorem 23). Let p, p_S, and p_F be given. The rules p_S and p_F are rules in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ which is an adhesive HLR category as well. Hence, if we show that p is a concurrent rule for p_S and p_F, i.e., $p = p_S *_E p_F$

for a suitable dependency relation E, the Concurrency Theorem [5, Thm. 5.23] is applicable and provides the desired result in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$.

To see this, first observe that R^S and L^F map jointly epimorphic to L^F. Hence, $R^S = L^F$ with the obvious morphisms is a natural choice for a dependency object. Moreover, the pushout complements for $K^S \overset{r^S}{\hookrightarrow} R^S \hookrightarrow L^F$ and $K^F \overset{l^F}{\hookrightarrow} L^F \overset{id}{\hookrightarrow} L^F$ exist and are given by KL and K^F again where $KL := (K_S \leftarrow \tilde{K_S} \hookrightarrow L_C \hookleftarrow \tilde{L_T} \rightarrow L_T)$ (see (1a) and (1b) in Fig. 21). Computing the pushouts $(2a)$ and $(2b)$ results in L and R, respectively. One easily checks that computing the pullback (3) results in K and that $l^S \circ k_1 = l$ and $r^F \circ k_2 = r$. In summary, $p = p_S *_{R^S} p_F$.

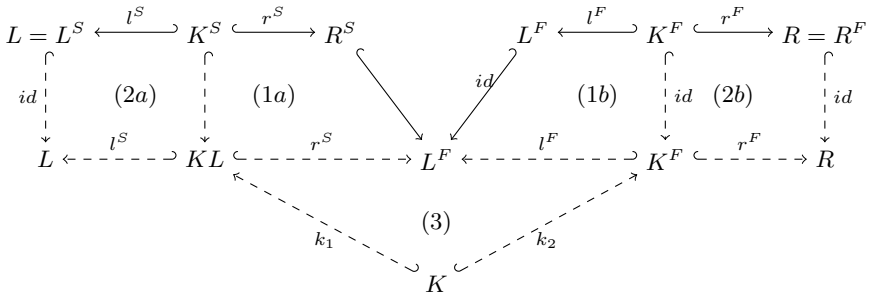


Fig. 21. Original rule p as concurrent rule of p^S and p^F

Applying the Concurrency Theorem gives the desired results at least it $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ and Proposition 16 allows to lift the results to **PTrG**:

1. Given a direct transformation $G \Rightarrow_{p,m} H$ in **PTrG**, by Proposition 16 this is a transformation in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$. Then the Concurrency Theorem guarantees existence of a transformation sequence $G \Rightarrow_{p_S,m} G' \Rightarrow_{p_F,n} H$ in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ where n is the comatch of the first transformation step.
2. Given transformation steps $G \Rightarrow_{p_S,m} G' \Rightarrow_{p_F,n} H$ in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$ where n is the comatch of the first transformation step, the Concurrency Theorem guarantees existence of a direct transformation $G \Rightarrow_{p,m} H$ in $[\bullet \leftarrow \bullet \rightarrow \bullet \leftarrow \bullet \rightarrow \bullet, \mathbf{Graphs}]$. Since G and m are already elements of **PTrG**, by Proposition 16 this is a transformation already in **PTrG**. □