Parallel Rule Application with Doubling Avoidance

Hans-Jörg Kreowski[1] and Aaron Lye[2(✉)]

[1] Department of Computer Science, University of Bremen, P.O.Box 33 04 40, 28334 Bremen, Germany
kreo@uni-bremen.de

[2] German Aerospace Center (DLR), Institute for the Protection of Maritime Infrastructures, Fischkai 1, 27572 Bremerhaven, Germany
aaron.lye@dlr.de

Abstract. In this paper, the parallelization of parallel independent rule applications within the double-pushout approach over adhesive categories is generalized in such a way that doubling can be avoided. If two rule applications to the same object delete the same part of the commonly accessed part, then independence is restored by removing the deletion part of one of the rule applications. Similarly, one can avoid that certain insertions are doubled. It turns out that parallelization with doubling avoidance is closely related to synchronization by means of amalgamated rules.

1 Introduction

Parallelism and simultaneity are fundamental principles in nature and society. Most biological, chemical and physical processes on one hand and most processes in economy, administration, traffic, and every-day life run in parallel. Often coordination and regulation are needed to avoid collisions, conflicts or undesired effects. As one of the main concerns of Computer Science is the modeling of real-world processes, one encounters quite a spectrum of approaches to the modeling of parallel systems like, e.g., cellular automata, Lindenmayer systems, neural networks, reaction systems, swarm computing, and many more. The same applies, in particular, to the area of graph transformation. See, e.g., Taentzer's parallel and distributed graph transformation [1], Kniemeier's growth grammars [2], Metevier's and Sopena's graph relabeling systems [3], Boehm's, Fonio's and Habel's amalgamation of graph transformation [4], and various others. One of the earliest and simplest approaches to parallel graph transformation was introduced by Ehrig and Kreowski [5] and further studied in [6]. The basic concept is the coproduct of rules used as parallel rule. As this is a rule itself, parallel derivations are just derivations applying parallel rules. Moreover, some very helpful properties of the concept can be shown like sequentialization and parallelization. Given a parallel rule application, the component rules can be applied in arbitrary order yielding the same result as the given rule application.

J. Endrullis and M. Tichy (Eds.): ICGT 2025, LNCS 15720, pp. 44–62, 2025.
https://doi.org/10.1007/978-3-031-94706-3_3

Conversely, rule applications applied to the same graph that are pairwise independent can be applied in parallel. And this holds for two successive sequentially independent rule application, too. A comprehensive survey can be found in Corradini et al. [7], the generalization to adhesive categories in Ehrig et al. [8,9]. A complication is the fact that the search for a matching morphism is NP-complete if rules with arbitrarily large left-hand sides are allowed. Fortunately, a matching morphism of a parallel rule can be constructed from the matching morphisms of the component rules so that the obstacle can be circumvent. In this paper, we generalize this approach to rule-based parallel transformation in such a way that undesired doubling can be avoided. There are two types of doubling with respect to two rule applications to the same object. The first one concerns parallel dependent rule applications that - intuitively seen - delete exactly the same part of the intersection of the matchings. If one modifies one of the rule applications in such a way that the deletion part is removed while the other one is kept, it turns out that they become parallel independent and can be applied in parallel. Moreover, the modified rule application can be reconstructed so that no information gets lost in the process. The second case concerns rule applications that - again intuitively - insert isomorphic parts which may be merged. And again this can be achieved by removing the potentially common part from one of the rule applications while the other one is kept. Both types of doubling may occur together. In Sect. 3, the parallelization with doubling avoidance (dedoubling for short) of two rule applications is introduced and investigated while the concept is generalized to families of rule applications to the same object in Sect. 4. In Sect. 5 we relate our approach to amalgamation as studied in [4].

2 Preliminaries

In this section, the preliminaries are provided as far as needed for this paper. For the well-known categorical notions confer, e.g., Adamek et al. [10]. For the notion and properties of adhesive categories confer, e.g., Lack and Sobocinski [11]. For the notion of graph transformation confer, e.g., Ehrig et al. [8,9].

2.1 Categorical Prerequisites

We assume that the underlying category $\mathbf{C}$ is adhesive. In addition, we require an epi-mono factorization and a strict initial object.

Assumption.

1. $\mathbf{C}$ is adhesive, i.e., $\mathbf{C}$ has all pullbacks, $\mathbf{C}$ has pushouts along monomorphisms (meaning at least one of the two morphisms in the pushout span is a monomorphism), and pushouts along monomorphisms are Van Kampen squares.
2. $\mathbf{C}$ has a strict initial object $\emptyset$, i.e., an initial object $\emptyset$ with the property that every morphism in $\mathbf{C}$ with codomain $\emptyset$ is an isomorphism. This implies that the pushouts of spans of the form $G \leftarrow \emptyset \rightarrow H$ are the coproducts of G and H, denoted by $G + H$.

3. Every morphism in **C** has an epi-mono factorization, i.e., for every morphism f there is a factorization $f = m \circ e$ where e is an epimorphism and m is a monomorphism. The factorization is unique up to isomorphism in adhesive categories.

We use two adhesive categories explicitly: the category Σ-**Graphs** of directed edge-labeled graphs and the category **Rules** of graph transformation rules over an adhesive category **C**. Σ-**Graphs** is a special variant of a diagram category over the category **Sets** of sets. And **Rules** is a diagram category over **C**.

The category **Sets** of sets and mappings is the best-known adhesive category. Every diagram category over an adhesive category is adhesive. This remains true if some target objects are fixed. For instance, consider the diagram $\Sigma \leftarrow \cdot \rightrightarrows \cdot$ over **Sets** for a set Σ (of *labels*). The corresponding adhesive category is the category Σ-**Graphs** of directed edge-labeled graphs. An object is a system $G = (V, E, s, t, l)$ where V is a set of *vertices*, E is a set of *edges*, $s, t \colon E \to V$ and $l \colon E \to \Sigma$ are mappings assigning a *source*, a *target* and a *label* to every edge $e \in E$. An edge e with $s(e) = t(e)$ is called a *loop*. The empty graph is the initial object. Another adhesive category is given by the diagram $\cdot \leftarrow \cdot \rightarrow \cdot$ which remains adhesive if the arrows are restricted to monomorphisms. This provides the category **Rules** of rules over **C**.

2.2 The Double-Pushout Approach

The rewriting formalism which we use throughout this paper is the double-pushout (DPO) approach as introduced by Ehrig, Pfender and Schneider in [12]. It was originally introduced for graphs. However, it is well-defined in adhesive categories.

A *rule* is a span of monomorphisms $p = (L \underset{l}{\leftarrow} K \underset{r}{\rightarrow} R)$. L is called *left-hand side*, R is called *right-hand side*, and K is called *gluing object*.

A rule application to some object G is defined wrt a morphism $g \colon L \to G$ which is called *(left) matching morphism*. G *directly derives* H if the span $L \underset{l}{\leftarrow} K \underset{r}{\rightarrow} R$ and matching morphism g extend to the diagram

$$\begin{tikzcd}
L \arrow[d, "g"'] \arrow[dr, phantom, "(1)"] & K \arrow[l, "l"] \arrow[r, "r"] \arrow[d, "z"'] \arrow[dr, phantom, "(2)"] & R \arrow[d, "h"'] \\
G & Z \arrow[l, "m_{Z\to G}"] \arrow[r, "m_{Z\to H}"'] & H
\end{tikzcd}$$

such that both squares are pushouts. Z is called *intermediate object* and h is called *right matching morphism*. Because rules and rule applications are symmetric, a rule $p = (L \underset{l}{\leftarrow} K \underset{r}{\rightarrow} R)$ induces an inverse rule $p^{-1} = (R \underset{r}{\leftarrow} K \underset{l}{\rightarrow} L)$ such that $G \underset{p}{\Longrightarrow} H$ implies $H \underset{p^{-1}}{\Longrightarrow} G$ and vice versa. The application of a rule p to G wrt g is called *direct derivation* and is denoted by $G \underset{p}{\Longrightarrow} H$ (where g is kept implicit). A *derivation* from G to H is a sequence of direct derivations

$G_0 \underset{p_1}{\Longrightarrow} G_1 \underset{p_2}{\Longrightarrow} \cdots \underset{p_n}{\Longrightarrow} G_n$ with $G_0 = G$, $G_n = H$ and $n \geq 0$. If $p_1, \cdots, p_n \in P$, then the derivation is also denoted by $G \overset{n}{\underset{P}{\Longrightarrow}} H$. If the length of the derivation does not matter, we write $G \overset{*}{\underset{P}{\Longrightarrow}} H$.

2.3 Local Church-Rosser Properties

Let $p = (L \underset{l}{\leftarrow} K \underset{r}{\rightarrow} R)$ and $\overline{p} = (\overline{L} \underset{\overline{l}}{\leftarrow} \overline{K} \underset{\overline{r}}{\rightarrow} \overline{R})$ be two rules.

1. Two direct derivations $G \underset{p}{\Longrightarrow} H$ and $G \underset{\overline{p}}{\Longrightarrow} \overline{H}$ with matching morphisms $g \colon L \to G$ and $\overline{g} \colon \overline{L} \to G$ are *parallel independent* if there exist two morphisms $f \colon L \to \overline{Z}$ and $\overline{f} \colon \overline{L} \to Z$ such that $g = m_{\overline{Z} \to G} \circ f$ and $\overline{g} = m_{Z \to G} \circ \overline{f}$. The situation is depicted in the following diagram.

$$\begin{tikzcd}
R \arrow[d, "h"'] & K \arrow[l, "r"'] \arrow[r, "l"] \arrow[d, "z"'] & L \arrow[dr, "g"] \arrow[drrr, bend left, "f"] & & \overline{L} \arrow[dl, "\overline{g}"'] \arrow[dlll, bend right, "\overline{f}"'] & \overline{K} \arrow[l, "\overline{l}"'] \arrow[r, "\overline{r}"] \arrow[d, "\overline{z}"'] & \overline{R} \arrow[d, "\overline{h}"'] \\
H & Z \arrow[l, "m_{Z \to H}"] \arrow[rr, "m_{Z \to G}"'] & & G & & \overline{Z} \arrow[ll, "m_{\overline{Z} \to G}"] \arrow[r, "m_{\overline{Z} \to \overline{H}}"'] & \overline{H}
\end{tikzcd}$$

2. Successive direct derivations $G \underset{\overline{p}}{\Longrightarrow} \overline{H} \underset{p}{\Longrightarrow} X$ with the right matching morphism $\overline{h} \colon \overline{R} \to \overline{H}$ and the (left) matching morphism $g' \colon L \to \overline{H}$ are *sequentially independent* if $\overline{H} \underset{\overline{p}^{-1}}{\Longrightarrow} G$ and $\overline{H} \underset{p}{\Longrightarrow} X$ are parallel independent.

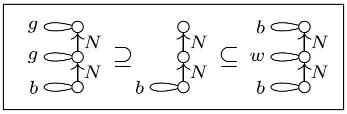
It is well-known that parallel independence induces the direct derivations $H \underset{\overline{p}}{\Longrightarrow} X$ and $\overline{H} \underset{p}{\Longrightarrow} X$ with matching morphisms $m_{Z \to H} \circ \overline{f}$ and $m_{\overline{Z} \to \overline{H}} \circ f$ and that sequential independence induces the derivation $G \underset{p}{\Longrightarrow} H \underset{\overline{p}}{\Longrightarrow} X$. The two constructions in the context of adhesive categories can be found in Lack and Sobocinski [11] (see also [8]).

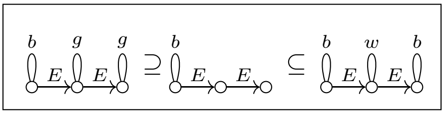
Moreover, the coproduct of $p + \overline{p}$ of two rules p and $\overline{p}$ is called *parallel rule*. It is well-known that an application $G \underset{p+\overline{p}}{\Longrightarrow} X$ yields two sequentially independent sequentializations $G \underset{p}{\Longrightarrow} H \underset{\overline{p}}{\Longrightarrow} X$ and $G \underset{\overline{p}}{\Longrightarrow} \overline{H} \underset{p}{\Longrightarrow} X$ such that the two first steps are parallel independent. Conversely, parallel and sequentially independent rule applications induce corresponding parallel rule applications. The results concerning sequentialization and parallelization are based on the Butterfly Lemma as used in the proof of Theorem 3 in Sect. 4 (cf. [6–8] for more details). Moreover, in [6] is it shown that sequentialization is associative. The proof is done in the category **Graphs**. The same proof holds in adhesive categories.

3 Parallelization with Doubling Avoidance of Two Rule Applications

It is well-known that a set of pairwise parallel independent rule applications to the same object can be parallelized by applying the coproduct of the applied rules as parallel rule (see [6–8]). In this section, parallelization is generalized in such a way that so-called doubling can be avoided. Given two rule applications to the same object, a doubling is a common part of the applied rules on which the matching morphisms coincide. Intuitively, the two rule applications double the common part. Under certain conditions, the two rule applications can be parallelized while the doubling is avoided. This is formally achieved by covering and replacing each of the rule applications by the applications of suitable parallel rules such that their sequentializations and proper re-parallelizations yields the intended effect. This parallelization without doubling can be iterated for sets of rule applications with more than two elements (see Sect. 4). We start with an example that illustrates the idea.

Example 1. Our running example is inspired by pattern generation and tilings. We use two rules

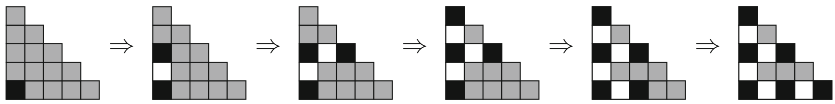
$p_1 =$ 

$p_2 =$ 

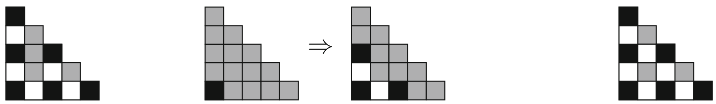
They are applied to finite subgraphs of an underlying infinite background graph defined by $B = (\mathbb{Z} \times \mathbb{Z}, \mathbb{Z} \times \mathbb{Z} \times \{b, g, w, E, N\}, s_B, t_B, l_B)$ where s_B is the projection to $\mathbb{Z} \times \mathbb{Z}$, l_B is the projection to $\{b, g, w, E, N\}$ and t_B is given by, for all $(x, y) \in \mathbb{Z} \times \mathbb{Z}, t_B((x, y), c) = (x, y)$ for $c \in \{b, g, w\}$, $t_B((x, y), E) = (x+1, y)$, $t_B((x, y), N) = (x, y + 1)$. This means that the vertices are the points in the plane with integer coordinates and each vertex is the source of five edges labeled with b, g, w, E and N respectively. The first three are loops colored by b(lack), g(ray), and w(hite). The fourth edge points to the right neighbor, the fifth to the upper neighbor. Each vertex $(x, y) \in \mathbb{Z} \times \mathbb{Z}$ represents a unit square $us(x, y)$ consisting of the four vertices $(x, y), (x + 1, y), (x, y + 1)$ and $(x + 1, y + 1)$ and the four edges $((x, y), E), ((x, y), N), ((x + 1, y), N)$ and $((x, y + 1), E)$. Each set T of vertices with one loop each represents a subgraph $sub(T)$ of B by closing T under the represented unit squares. It can be depicted as a grid of unit squares where each square is colored by the label of the loop at the corresponding vertex in T. The start graph is a rectangular triangle with legs of length 5. The color of the vertex at the bottom left is black; all other vertices have gray loops. A sample derivation applying the rules above to a sample grid is depicted in Fig. 1a. The only alternative result after five steps is depicted in Fig. 1b.

Both rules can be applied to the start pattern. As they are parallel independent, they can be applied in parallel as depicted in Fig. 1c. After the first parallel step, four rule applications are possible. But two of the matches overlap in the middle unit square of the hypotenuse. This is a loop that does not belong to the gluing graph so that the two rule application are parallel dependent. The other pairs are parallel independent so that two maximal parallel derivation steps are defined yielding the two result pattern in Fig. 1a and 1b.

A closer look at the dependence reveals that both rule applications do exactly the same on the overlap: They replace the g-labeled loop by a b-labeled one. If



(a) A sample derivation



(b) Alternative result (c) Parallel rule application (d) Rule application with overlap

Fig. 1. Sample derivations

it would be allowed to apply this only once instead of twice, then all four rule applications could be parallelized yielding the grid depicted in Fig. 1d.

In order to formally describe this observation, we need the definition of a cover.

Definition 1. *Let* $p = (L \underset{l}{\leftarrow} K \underset{r}{\rightarrow} R), p' = (L' \underset{l'}{\longleftarrow} K' \underset{r'}{\longrightarrow} R')$ *and* $p'' = (L'' \underset{l''}{\longleftarrow} K'' \underset{r''}{\longrightarrow} R'')$ *be rules and* $\langle \lambda'; \lambda'' \rangle \colon L' + L'' \to L$, $\langle \kappa'; \kappa'' \rangle \colon K' + K'' \to K$ *and* $\langle \varrho'; \varrho'' \rangle \colon R' + R'' \to R$ *be epimorphisms such that the diagrams*

$$\begin{tikzcd}
L' + L'' \arrow[d, "\langle \lambda'; \lambda'' \rangle"'] \arrow[dr, phantom, "(3)"] & K' + K'' \arrow[l, "l' + l''"'] \arrow[r, "r' + r''"] \arrow[d, "\langle \kappa'; \kappa'' \rangle"] \arrow[dr, phantom, "(4)"] & R' + R'' \arrow[d, "\langle \varrho'; \varrho'' \rangle"] \\
L & K \arrow[l, "l"] \arrow[r, "r"'] & R
\end{tikzcd}$$

are pushouts. Then the pair (p', p'') *is a* cover *of* p*. The elements* p' *and* p'' *are called* cover components *of* p*. The rule* p'' *is called* complement rule *of* p' *and* p' *complement rule of* p''*.*

Theorem 1. *1. Let* (p', p'') *be a cover of* p *with the left epimorphism* $\langle \lambda'; \lambda'' \rangle \colon L' + L'' \to L$ *and* $G \underset{p}{\Longrightarrow} H$ *for some objects* G *and* H *with the matching morphism* $g \colon L \to G$ *be a rule application. Then* $p' + p''$ *can be applied to* G *with the matching morphism* $g \circ \langle \lambda'; \lambda'' \rangle \colon L' + L'' \to G$ *yielding* $G \underset{p'+p''}{\Longrightarrow} H$*.*

2. Let $G \underset{\overline{p}}{\Longrightarrow} \overline{H}$ *be another rule application that is parallel independent of* $G \underset{p}{\Longrightarrow} H$*. Then* $G \underset{\overline{p}}{\Longrightarrow} \overline{H}$ *is parallel independent of* $G \underset{p'+p''}{\Longrightarrow} H$*, too.*

Proof. 1. Let the diagrams (1) and (2) as given in Sect. 2.2 be the pushouts defining $G \underset{p}{\Longrightarrow} H$. Then the sequential compositions of (1) and (3) and (2) and (4) are pushouts defining $G \underset{p'+p''}{\Longrightarrow} H$.

2. Due to the assumption, there are morphisms $f\colon L \to \overline{Z}$ and $\overline{f}\colon \overline{L} \to Z$ such that $m_{\overline{Z}\to G} \circ f = g$ and $m_{Z\to G} \circ \overline{f} = \overline{g}$ where $\overline{Z}$ is the intermediate object of $G \underset{p}{\Longrightarrow} \overline{H}$ with the monomorphism $m_{\overline{Z}\to G}\colon \overline{Z} \to G$ and $\overline{g}\colon \overline{L} \to G$ is the matching morphism. Composing the first equation with $\langle \lambda'; \lambda'' \rangle$ yields $m_{\overline{Z}\to G} \circ f \circ \langle \lambda'; \lambda'' \rangle = g \circ \langle \lambda'; \lambda'' \rangle$. This proves (together with the second equation) the stated parallel independence. □

If Theorem 1 is applied to two rule applications to the same object in such a way that both covers share one of the component rules, then one gets the following observation by using the known results for sequentialization and parallelization.

Observation 1. Let $d_1 = (G \underset{p_1}{\Longrightarrow} H_1)$ and $d_2 = (G \underset{p_2}{\Longrightarrow} H_2)$ be two rule applications with matchings g_1 and g_2, respectively, and $p_1' + p_1''$ and $p_2' + p_2''$ covers of p_1 and p_2, respectively. Let $G \underset{p_1'+p_1''}{\Longrightarrow} H_1$ and $G \underset{p_2'+p_2''}{\Longrightarrow} H_2$ be the corresponding rule applications according to Theorem 1.1 and $g_1 \circ \lambda_1'' = g_2 \circ \lambda_2''$ where $\lambda_i''\colon L_i'' \to L_i$ for $i = 1, 2$ are the morphisms according to Definition 1. Then the following holds.

1. The sequentializations yield
$$G \underset{p_1'}{\Longrightarrow} H_1' \underset{p_1''}{\Longrightarrow} H_1 \quad G \underset{p_1''}{\Longrightarrow} X_1 \underset{p_1'}{\Longrightarrow} H_1 \quad G \underset{p_2'}{\Longrightarrow} H_2' \underset{p_2''}{\Longrightarrow} H_2 \quad G \underset{p_2''}{\Longrightarrow} X_2 \underset{p_2'}{\Longrightarrow} H_2$$
for some objects H_1', H_2', X_1 and X_2.
2. If $p_0 = p_1'' = p_2''$ and $g_1 \circ \lambda_1'' = g_2 \circ \lambda_2''$ where $\lambda_i''\colon L_i'' \to G$ for $i = 1, 2$ are the morphisms mapping the left hand sides of p_i'' to the left-hand side of p_i according to the Definition 1. Then $X = X_1 = X_2$ and $(G \underset{p_0}{\Longrightarrow} X) = (G \underset{p_1''}{\Longrightarrow} X_1) = (G \underset{p_2''}{\Longrightarrow} X_2)$ such that the rule applications form the following diagram
$$\begin{tikzcd}
 & H_1' \arrow[rr, Rightarrow, "p_0"] & & H_1 \\
G \arrow[ur, Rightarrow, "p_1'"] \arrow[urrr, Rightarrow, "p_0+p_1'" near end] \arrow[rr, Rightarrow, "p_0"] \arrow[dr, Rightarrow, "p_2'"'] \arrow[drrr, Rightarrow, "p_0+p_2'"' near end] & & X \arrow[ur, Rightarrow, "p_1'"'] \arrow[dr, Rightarrow, "p_2'"] & \\
 & H_2' \arrow[rr, Rightarrow, "p_0"'] & & H_2
\end{tikzcd}$$
3. Moreover, $G \underset{p_0}{\Longrightarrow} X$ and $G \underset{p_1'}{\Longrightarrow} H_1'$ as well as $G \underset{p_0}{\Longrightarrow} X$ and $G \underset{p_2'}{\Longrightarrow} H_2'$ are parallel independent.
4. If, in addition, $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_2'}{\Longrightarrow} H_2'$ are parallel independent, then the parallelization with $G \underset{p_0}{\Longrightarrow} X$ yields $G \underset{p_0+p_1'+p_2'}{\Longrightarrow} Y$ for some object Y and one obtains $G \underset{p_0+p_1'}{\Longrightarrow} H_1 \underset{p_2'}{\Longrightarrow} Y$ and $G \underset{p_0+p_2'}{\Longrightarrow} H_2 \underset{p_1'}{\Longrightarrow} Y$ by sequentialization.

The special role of p_0 is reflected in the following definition.

Definition 2. *Assume the situation of the previous observation. Moreover, let p_0 not be a span of isomorphisms.*

1. *Then $G \underset{p_0}{\Longrightarrow} X$ is called a* doubling *of d_1 and d_2.*
2. *If, in addition, $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_2'}{\Longrightarrow} H_2'$ are parallel independent, then the doubling is* useful.

Remark 1. If a rule is a span of isomorphisms, then its application derives an object that is isomorphic to the host object meaning that is has no effect and – in particular – nothing is doubled.

Parallelization with dedoubling is captured in the following theorem.

Theorem 2. *Let $G \underset{p_0}{\Longrightarrow} X$ be a useful doubling of $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2}{\Longrightarrow} H_2$ with the corresponding cover components $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_2'}{\Longrightarrow} H_2'$, respectively.*

1. *Then $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2'}{\Longrightarrow} H_2'$ as well as $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_2}{\Longrightarrow} H_2$ are parallel independent.*
2. *Therefore, the parallelizations $G \underset{p_1+p_2'}{\Longrightarrow} Y_{12}$ for some object Y_{12} and $G \underset{p_2+p_1'}{\Longrightarrow} Y_{21}$ for some object Y_{21} are defined inducing the sequentialization*

$$G \underset{p_1}{\Longrightarrow} H_1 \underset{p_2'}{\Longrightarrow} Y_{12} \quad G \underset{p_2'}{\Longrightarrow} H_2' \underset{p_1}{\Longrightarrow} Y_{12} \quad G \underset{p_2}{\Longrightarrow} H_2 \underset{p_1'}{\Longrightarrow} Y_{21} \quad G \underset{p_1'}{\Longrightarrow} H_1' \underset{p_2}{\Longrightarrow} Y_{21}.$$

3. *Moreover, $Y_{12} \cong Y_{21}$.*

Proof. 1. According to the reasoning in Observation 1, the rule applications $G \underset{p_0+p_2'}{\Longrightarrow} H_2$ and $G \underset{p_1'}{\Longrightarrow} H_1'$ are parallel independent. Let

$$\begin{tikzcd}
K_0+K_2' \arrow[r, "l_0+l_2'"] \arrow[d, "\langle \kappa_{02};\kappa_2' \rangle"] & L_0+L_2' \arrow[d, "\langle \lambda_{02};\lambda_2' \rangle"] & & L_1' \arrow[d, "\lambda_1'"] & K_1' \arrow[l, "l_1'"'] \arrow[dd, "z_1'"] \\
K_2 \arrow[r, "l_2"] \arrow[d, "z_2"] & L_2 \arrow[dr, "g_2"] & & L_1 \arrow[dl, "g_1"'] & \\
Z_2 \arrow[rr, "m_{Z_2 \to G}"] & & G & & Z_1' \arrow[ll, "m_{Z_1' \to G}"]
\end{tikzcd}$$

be the corresponding left pushouts. Then there are morphisms $f_{12}\colon L_1' \to Z_2$ and $f_{21}\colon L_0 + L_2' \to Z_1'$ such that

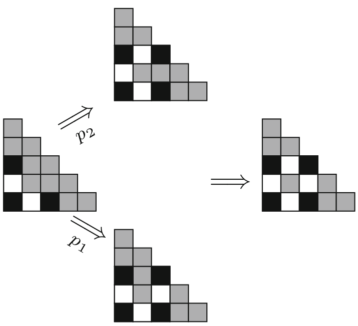
(a) $m_{Z_2 \to G} \circ f_{12} = g_1 \circ \lambda_1'$
(b) $m_{Z_1' \to G} \circ f_{21} = g_2 \circ \langle \lambda_0; \lambda_2' \rangle$.

Let $g_2 = \overline{m} \circ \overline{e}$ for $\overline{e}\colon L_2 \to \overline{L}$ and $\overline{m}\colon \overline{L} \to G$ be the epi-mono factorization of g_2 and $f_{21} = \hat{m} \circ \hat{e}$ with $\hat{e}\colon L_0 + L_2' \to \hat{L}$ and $\hat{m}\colon \hat{L} \to Z_1'$ be the epi-mono factorization of f_{21}. Then one gets the epi-mono factorization of the morphism of (b) in two variants by sequential composition with the monomorphism $m_{Z_1' \to G}$ on one hand and with the epi $\langle \lambda_{02}; \lambda_2' \rangle$ on the other hand, i.e., $m_{Z_1' \to G} \circ \hat{m} \circ \hat{e} = \overline{m} \circ \overline{e} \circ \langle \lambda_{02}; \lambda_2' \rangle$. The uniqueness of epi-mono factorization implies that, without loss of generality, $\overline{L} = \hat{L}$, $m_{Z_1' \to G} \circ \hat{m} = \overline{m}$ and $\hat{e} = \overline{e} \circ \langle \lambda_{02}; \lambda_2' \rangle$. Therefore, one get $m_{Z_1' \to G} \circ \hat{m} \circ \overline{e} = \overline{m} \circ \overline{e} = g_2$. Combined with (a), this proves the parallel independence of $G \underset{p_2}{\Longrightarrow} H_2$ and $G \underset{p_1'}{\Longrightarrow} H_1'$.

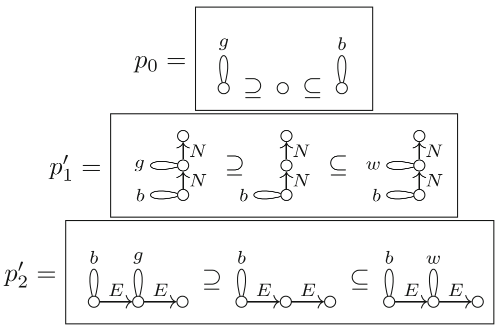
The second case follows by symmetry.

2. The second statement follows from the first one.
3. Using Theorem 1, the first steps of $G \underset{p_1}{\Longrightarrow} H_1 \underset{p_2'}{\Longrightarrow} Y_{12}$ and $G \underset{p_2}{\Longrightarrow} H_2 \underset{p_1'}{\Longrightarrow} Y_{21}$ can be replaced by $G \underset{p_0+p_1'}{\Longrightarrow} H_1$ and $G \underset{p_0+p_2'}{\Longrightarrow} H_2$ respectively yielding two of the sequentializations of $G \underset{p_0+p_1'+p_2'}{\Longrightarrow} Y$ as shown in Observation 1. This means that $Y_{12} \cong Y \cong Y_{21}$. □

Example 2. We continue the discussion in Example 1. After the parallel step in Fig. 1c, the two parallel independent steps are depicted on the left in Fig. 2a. The common component rule p_0 is given at the top in Fig. 2b. It is applied to the middle of the hypotenuse. The complementary component rule p_1' of the upper application of rule p_1 is given in the middle in Fig. 2b, and the complementary component rule p_2' of the lower application of rule p_2 at the bottom in Fig. 2b. They are applied in such a way that all three rule applications overlap in the vertex corresponding to the middle hypotenuse so that they are pairwise parallel dependent. (p_0, p_1') covers p_1, and (p_0, p_2') covers p_2 so that the application of p_0 is a useful doubling, $p_0 + p_i'$ replaces p_i for $i = 1, 2$ and $p_0 + p_1' + p_2', p_1 + p_2'$ and $p_2 + p_1'$ can be applied to the start pattern in Fig. 2a all deriving the pattern on the right.



(a) Parallel rule application with dedoubling



(b) derived rules from dedoubling

Fig. 2. Sample derivations and rules of Example 2

Remark 2. Two rule applications d and d' to the same object are not parallel independent if at least one of the matching morphisms - say the matching morphism of d - cannot be restricted to the intermediate object of the other rule application d'. Intuitively, this means that d matches items that are deleted by d'. There are two cases: (1) d deletes the same items, or (2) d keeps some of them as part of the gluing object. In the second case, one faces an essential conflict of actions that are mutually exclusive. In the first case, deletion cannot be done twice. But if the deletion takes place at all, the purposes of both rule applications are served. With respect to deletion, this is the idea of parallelization with dedoubling. With respect to insertion, the situation is different as one has the choice between doubling or dedoubling.

4 Parallelization with Doubling Avoidance of Families of Rule Applications

In this section, the parallelization with dedoubling of two rule applications is generalized to families of rule applications. The parallelization with dedoubling as introduced in the previous section is used as an operator on families of rule applications so that it can be iterated. The operator chooses two rule applications of the family with a useful doubling non-deterministically and replaces one of the rule applications by its complement rule application. As shown in Theorem 2, the other rule application and the complement rule application are parallel independent. Moreover we show that the operator preserves parallel independence in the sense that a third rule application of the family which is parallel independent of the replaced rule application is also parallel independent of the replacing one. Therefore, the number of pairs of rule applications of the family cannot increase, and it decreases if the processed pair is not parallel independent. If one is interested in getting rid of the dependent pairs and applies the operator only to those, the iteration terminates either with a family of pairwise parallel independent rule applications or one gets stuck because there remain pairs without useful doublings. In the first case, the family can be parallelized. In the second case, parallelization fails.

Definition 3. *1. Let $\mathcal{F}$ be the class of all finite families of rule applications of the form $F = \{G \underset{p_i}{\Longrightarrow} H_i\}_{i \in I}$ for some common start object G and a finite index set I. Then the non-deterministic* dedoubling *operator $DD \in \mathcal{F} \times \mathcal{F}$ is defined as follows: Let $F = \{G \underset{p_i}{\Longrightarrow} H_i\}_{i \in I}, \overline{F} = \{G \underset{\overline{p}_i}{\Longrightarrow} \overline{H}_i\}_{i \in I} \in \mathcal{F}$. Then $(F, \overline{F}) \in DD$ if there are $j, k \in I$ and a useful doubling $G \underset{p_0}{\Longrightarrow} X$ of $G \underset{p_j}{\Longrightarrow} H_j$ and $G \underset{p_k}{\Longrightarrow} H_k$ with the complement component $G \underset{p'_j}{\Longrightarrow} H'_j$ of $G \underset{p_j}{\Longrightarrow} H_j$ such that $\overline{F}$ is obtained from F by replacing $G \underset{p_j}{\Longrightarrow} H_j$ by $G \underset{p'_j}{\Longrightarrow} H'_j$. $(F, \overline{F}) \in DD$ is denoted by $F \rightsquigarrow \overline{F}$.*

2. As a binary relation of $\mathcal{F}$, DD can be iterated $F_0 \rightsquigarrow F_1 \rightsquigarrow \cdots \rightsquigarrow F_n$ for some $n \in \mathbb{N}$. This is denoted by $F \overset{n}{\rightsquigarrow} \overline{F}$ with $F = F_0$ and $\overline{F} = F_n$ or by $F \overset{*}{\rightsquigarrow} \overline{F}$ if the length of the dedoubling sequence is not needed explicitly.

An interesting question concerning iterated processes is about termination. In many cases, a sufficient condition that guarantees termination is a natural variable which decreases in every process step. In the case of dedoubling, this applies if one wants to get rid of parallel dependence because the processed pair of rule applications results in a parallel independent pair (cf. Theorem 2). Moreover, it can be shown that dedoubling preserves parallel independence.

Theorem 3. *Let $G \underset{p_i}{\Longrightarrow} H_i$ for $i = 1, 2, 3$ be rule applications. Let $G \underset{p_0}{\Longrightarrow} X$ be a useful dedoubling of $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2}{\Longrightarrow} H_2$ with the corresponding cover component $G \underset{p_1'}{\Longrightarrow} H_1'$ of $G \underset{p_1}{\Longrightarrow} H_1$. Let $G_1 \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_3}{\Longrightarrow} H_3$ be parallel independent. Then $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_3}{\Longrightarrow} H_3$ are parallel independent, too.*

Proof. Consider the following diagram

$$\begin{tikzcd}
 & K_1' \arrow[r, "l_1'"] \arrow[d, "\kappa_1"'] \arrow[ddl, bend right, "z_1'"'] \arrow[dr, phantom, "(3_1')"] & L_1' \arrow[d, "\lambda_1"'] \arrow[ddr, bend left, "g_1'"] & & & \\
 & K_1 \arrow[r, "l_1"'] \arrow[d, "z_1"'] & L_1 \arrow[dr, "g_1"] \arrow[drrr, bend left, "f_{13}"] \arrow[d, phantom, "(1_1)"] & & L_3 \arrow[dl, "g_3"'] \arrow[dlll, bend right, "f_{31}"'] \arrow[d, phantom, "(1_3)"] & K_3 \arrow[l, "l_3"'] \arrow[d, "z_3"] \\
Z_1' \arrow[rrr, bend right, "m_{Z_1' \to G}"'] & Z_1 \arrow[l, dashed, "m"'] \arrow[rr, "m_{Z_1 \to G}"'] & & G & & Z_3 \arrow[ll, "m_{Z_3 \to G}"]
\end{tikzcd}$$

$G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_3}{\Longrightarrow} H_3$ provide the pushouts (1_1) and (1_3). As they are parallel independent, there are morphisms $f_{13}\colon L_1 \to Z_3$ and $f_{31}\colon L_3 \to Z_1$ such that (a) $m_{Z_3 \to G} \circ f_{13} = g_1$ and (b) $m_{Z_1 \to G} \circ f_{31} = g_3$. $G \underset{p_1'}{\Longrightarrow} H_1'$ provides the pushout depicted on the left outer square above. Choosing (c) $f_{13}' = f_{13} \circ \lambda_1$ yields

$$\text{(d) } m_{Z_3 \to G} \circ f_{13}' \underset{(c)}{=} m_{Z_3 \to G} \circ f_{13} \circ \lambda_1 \underset{(a)}{=} g_1 \circ \lambda_1 = g_1'.$$

If there is a morphism $m\colon Z_1 \to Z_1'$ with (e) $m_{Z_1' \to G} \circ m = m_{Z_1 \to G}$, then choosing (f) $f_{31}' = m \circ f_{31}$ yields

$$\text{(g) } m_{Z_1' \to G} \circ f_{31}' \underset{(f)}{=} m_{Z_1' \to G} \circ m \circ f_{31} \underset{(e)}{=} m_{Z_1 \to G} \circ f_{31} \underset{(b)}{=} g_3.$$

(d) and (g) show that $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_3}{\Longrightarrow} H_3$ are parallel independent.

The morphism m is obtained by applying the Butterfly Lemma to the left pushout of $G \underset{p_1'+p_0}{\Longrightarrow} H_1$:

$$
\begin{tikzcd}
L_1' + L_0 \arrow[d, "\langle \lambda_1'; \lambda_0 \rangle"'] & K_1' + K_0 \arrow[l, "l_1'+l_0"'] \arrow[d, "\langle \kappa_1'; \kappa_0 \rangle"] \arrow[dl, phantom, "(3_1)"] \\
L_1 \arrow[d, "g_1"] & K_1 \arrow[l, "l_1"] \arrow[d, "z_1"] \arrow[dl, phantom, "(1_1)"] \\
G & Z_1 \arrow[l, "m_{Z_1 \to G}"]
\end{tikzcd}
$$

implies

$$
\begin{tikzcd}
 & K_1' \arrow[dl, "l_1'"'] \arrow[dr, "z_1 \circ \kappa_1'"] \arrow[d, phantom, "(A)"] & & K_0 \arrow[dl, "z_1 \circ \kappa_0"'] \arrow[dr, "l_0"] \arrow[d, phantom, "(B)"] & \\
L_1' \arrow[dr, "f_{10}'"] \arrow[ddrr, bend right, "g_1 \circ \lambda_1 = g_1'"'] & & Z_1 \arrow[dl, "m'"'] \arrow[dr, "m"] \arrow[dd, "m_{Z_1 \to G}"] \arrow[d, phantom, "(C)\quad(D)"] & & L_0 \arrow[dl, "f_{01}'"] \arrow[ddll, bend left, "g_1 \circ \lambda_0 = g_1''"] \\
 & Z_0 \arrow[dr, "m_{Z_0 \to G}"] \arrow[d, phantom, "(E)"] & & Z_1' \arrow[dl, "m_{Z_1' \to G}"'] \arrow[d, phantom, "(F)"] & \\
 & & G & &
\end{tikzcd}
$$

where the diagram $(C)+(D)$ is a pushout. As l_1' is a monomorphism, the pushout complement of $g_1' \circ l_1'$ is unique (up to isomorphism). Moreover, (D) is commutative so that $m\colon Z_1 \to Z_1'$ is the morphism needed above. This completes the proof. $\square$

Definition 4. *Let $F = \{G \underset{p_i}{\Longrightarrow} H_i\}_{i \in I} \in \mathcal{F}$. Then $di(F)$ is the number of pairs of F that are parallel dependent, called* dependence index *of F.*

Corollary 1. *1. Let $F \rightsquigarrow \overline{F}$ for $F, \overline{F} \in \mathcal{F}$ and the processed pair of rule applications be parallel dependent. Then $di(F) > di(\overline{F})$.*

2. Let $F_0 \overset{n}{\rightsquigarrow} F_n$ be a sequence of dedoublings where all processed pairs are parallel dependent. Then $di(F_0) \geq n$.

Proof. 1. According to Theorem 3, $di(F)$ does not increase after dedoubling. According to Theorem 2, it decreases at least by 1 wrt the processed pair.

2. According to Point 1, the length of a dedoubling is bounded by $di(F_0)$ if only dependent pairs are processed. $\square$

The situation is more difficult if the processed pairs of dedoublings are parallel independent because no general termination criterion is known. But whenever a family is reached by a sequence of dedoublings such that each pair of members is parallel independent, one may stop the process. The result provides a parallelization with doubling avoidance of the initial family.

Another interesting question about iterated dedoubling concerns the semantic relation between a given family of rule applications and one obtained by a sequence of dedoublings. A family member remains unchanged or is repeatedly replaced by a component rule application that is complementary to the removed component rule application. A component is only removed if it is a doubling so that it remains present in another member of the family and no information gets lost. This is reflected in the following theorem stating that each member of the start family is not changed or can be reconstructed as parallel rule application of the resulting member and all the removed component rule applications that are involved in the processing of all members with the same index as the initial member.

Theorem 4. *Let* $F = \{d_i = (G \underset{p_i}{\Longrightarrow} H_i)\}_{i\in I}, \overline{F} = \{\overline{d}_i = (G \underset{\overline{p}_i}{\Longrightarrow} \overline{H}_i)\}_{i\in I} \in \mathcal{F}$ *with* $dd = (F \overset{*}{\rightsquigarrow} \overline{F})$*. Then for each* $i \in I$*, either* $d_i = \overline{d}_i$ *or there is a parallelization* $G \underset{(\sum_{l=1}^{m(i)} p_{0l})+\overline{p}_i}{\Longrightarrow} H_i$ *of* $G \underset{p_i}{\Longrightarrow} \overline{H}_i$ *and the dedoublings* $G \underset{p_{0l}}{\Longrightarrow} H_{0l}$ *for* $l = 1, \ldots, m(i)$ *which are used to diminish the ith component of* dd*.*

Proof. Induction on the length of dd.

Base for $F \overset{0}{\rightsquigarrow} \overline{F}$: Then $F = \overline{F}$ and $d_i = \overline{d}_i$ for all $i \in I$.

Step for $(F \overset{n+1}{\rightsquigarrow} \overline{F}) = (F \overset{n}{\rightsquigarrow} \hat{F} \overset{1}{\rightsquigarrow} \overline{F})$ for some $\hat{F}$: Then the last step is given by a dedoubling of $\hat{d}_j = (G \underset{\hat{p}_j}{\Longrightarrow} \hat{H}_j)$ and $\hat{d}_k = (G \underset{\hat{p}_k}{\Longrightarrow} \hat{H}_k)$ with a useful dedoubling $\hat{d}_0 = (G \underset{\hat{p}_0}{\Longrightarrow} \hat{H}_0)$ corresponding cover components $\hat{d}'_j = (G \underset{\hat{p}'_j}{\Longrightarrow} \hat{H}'_j)$ and $\hat{d}'_k = (G \underset{\hat{p}'_k}{\Longrightarrow} \hat{H}'_k)$ as well as $\overline{d}_j = \hat{d}'_j$ and $\overline{d}_k = \hat{d}'_k$. This means for $i \neq j$ that $\overline{d}_i = \hat{d}_i$ so hat the statement holds by induction hypothesis. For $j \in I$, one has (a) $d_j = \hat{d}_j$ and (b) $D_j = (G \underset{(\sum_{l=1}^{m(j)} p_{0l})+\hat{p}_j}{\Longrightarrow} \hat{H}_j)$ being the parallelization of $\hat{d}_j = (G \underset{\hat{p}_j}{\Longrightarrow} \hat{H}_j)$ and the dedoubling $d_{0l} = (G \underset{p_{0l}}{\Longrightarrow} H_{0l})$ for $l = 1, \ldots, m(j)$ by induction hypothesis. By construction, one has $\hat{d}_j = (G \underset{\hat{p}_0+\hat{p}'_j}{\Longrightarrow} \hat{H}_j)$. The first step of the sequentialization of $\hat{D}_j$ are $\hat{d}'_j = (G \underset{\hat{p}'_j}{\Longrightarrow} \hat{H}'_j)$ and $\hat{d}_i = (G \underset{\hat{p}}{\Longrightarrow} \hat{H}_j)$. In the case (a), this proves the statement. In the case (b), $\hat{d}_j$ and all d_{0l} are pairwise parallel independent as components of a parallelization. $\hat{D}_j$ covers $\hat{d}_j$ so that $\hat{D}_j$ is parallel independent of all d_{0l} according to Theorem 2. Therefore, the parallelization $(G \underset{(\sum_{l=1}^{m(j)} p_{0l})+\hat{p}_0+\hat{p}_j}{\Longrightarrow} H_j)$ is defined proving the statement for the case (b) and completing the proof. □

5 Relating Parallelization with Doubling Avoidance to Amalgamation

Boehm, Fonio and Habel [4] introduced and studied amalgamation of graph transformation as a generalization of parallelization of parallel independent rule applications. Their approach is generalized to adhesive categories in Ehrig et al. [9]. They consider the same basic situation as in our approach, i.e., two rules p_1 and p_2 sharing the same component rule p_0 together with the embedding morphisms $e_{01}\colon p_0 \to p_1$ and $e_{02}\colon p_0 \to p_2$ both accompanied with complementary component rules p_1' and p_2' resp., which are called remainders. Then the amalgamated rule p_{12} is defined as the pushout of the span given by e_{01} and e_{02} (provided that it exists). The Amalgamation Theorem states the following: Let $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2}{\Longrightarrow} H_2$ be two rule applications the matching morphisms of which coincide on the embeddings of the left-hand side of p_0. Moreover, let $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_2'}{\Longrightarrow} H_2'$ be parallel independent rule applications defined by the matching morphisms that are composed of the given matching morphisms and the embeddings of left-hand sides of p_1' and p_2' into the left-hand sides of p_1 and p_2 resp. Then there are derivations $G \underset{p_{12}}{\Longrightarrow} Y, G \underset{p_1}{\Longrightarrow} H_1 \underset{p_2'}{\Longrightarrow} Y$ and $G \underset{p_2}{\Longrightarrow} H_2 \underset{p_1'}{\Longrightarrow} Y$ for some object Y. We show that the amalgamated rule p_{12} is covered by our parallel rule $p_0 + p_1' + p_2'$ (Theorem 5) such that the Amalgamation Theorem turns out to be a kind of corollary of Theorem 2. The proof is based on the relation among the component rules of a useful doubling as summarized in Observation 2.

Observation 2. The following diagram displays the relations among the rules that are involved in a useful doubling where the given epimorphisms are denoted by e_1 and e_2, the injections into coproducts are indicated by i with proper indices and the restrictions of e_1 and e_2 to the coproduct components by e with proper indices. Moreover, the pushout p_{12} of the span given by the common component rule p_0 of p_1 and p_2 with the embeddings $e_{0k}\colon p_0 \to p_k$ for $k = 1, 2$ is added. The pushout morphisms are denoted by $f_k\colon p_k \to p_{12}$ for $k = 1, 2$. And $f_0\colon p_0 \to p_{12}$ is defined as $f_1 \circ e_{01} = f_2 \circ e_{02}$. Altogether all triangles are commutative.

$$
\begin{tikzcd}
 & & & & p_0 + p_1' + p_2' \arrow[dl, phantom, "(G)"] \arrow[dr, phantom, "(H)"] & & & & \\
 & & p_0 + p_1' \arrow[urr, "i_1"] \arrow[dd, "e_1"] & & & & p_0 + p_2' \arrow[ull, "i_2"'] \arrow[dd, "e_2"'] & & \\
p_1' \arrow[urr, "i_{11}"] \arrow[drr, "e_{11}"'] \arrow[r, phantom, "(C)"] & & & & p_0 \arrow[ull, "i_{01}"'] \arrow[uu, "i_0"] \arrow[urr, "i_{02}"] \arrow[dll, "e_{01}"] \arrow[drr, "e_{02}"'] \arrow[dd, "f_0"] \arrow[ll, phantom, "(D)"] \arrow[rr, phantom, "(E)"] & & & & p_2' \arrow[ull, "i_{22}"'] \arrow[dll, "e_{22}"] \arrow[l, phantom, "(F)"] \\
 & & p_1 \arrow[drr, "f_1"'] & & & & p_2 \arrow[dll, "f_2"] & & \\
 & & & & p_{12} \arrow[ul, phantom, "(A)"] \arrow[ur, phantom, "(B)"] & & & &
\end{tikzcd}
$$

Using the coproduct property of $p_0 + p_1' + p_2'$ the morphisms $f_1 \circ e_{11}, f_0 = f_1 \circ e_{01} = f_2 \circ e_{02}$ and $f_2 \circ e_{22}$ induce $e\colon p_0 + p_1' + p_2' \to p_{12}$ with

$$\begin{aligned}
&(\mathrm{I}) \quad e \circ i_k \circ i_{kk} = f_k \circ e_{kk}, \quad k = 1,2 \\
&(\mathrm{J}) \quad e \circ i_2 \circ i_{02} = e \circ i_1 \circ i_{01} = f_1 \circ e_{01} = f_2 \circ e_{02} \\
&(\mathrm{K}) \quad e \circ i_0 = f_0.
\end{aligned}$$

Theorem 5. *Let $G \underset{p_0}{\Longrightarrow} X$ be a useful doubling of $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2}{\Longrightarrow} H_2$ for some rule p with the embeddings $e_{0i}\colon p_0 \to p_i$ and the corresponding complementary rule applications $G \underset{p_i'}{\Longrightarrow} H_i'$ for $k = 1,2$. Let p_{12} with the rule morphisms $f_i\colon p_i \to p_{12}$ for $k = 1,2$ be the pushout of the span $p_1 \underset{e_{01}}{\longleftarrow} p_0 \underset{e_{02}}{\longrightarrow} p_2$ and $e\colon p_0 + p_1' + p_2' \to p_{12}$ be the rule morphism constructed in the observation above. Then*

1. *e is an epimorphism and*
2. *the diagram*

$$\begin{tikzcd}
L_0 + L_1' + L_2' \arrow[d, "e_L"'] & K_0 + K_1' + K_2' \arrow[l, "l_0+l_1'+l_2'"'] \arrow[r, "r_0+r_1'+r_2'"] \arrow[d, "e_K"] & R_0 + R_1' + R_2' \arrow[d, "e_R"] \\
L_{12} & K_{12} \arrow[l, "l_{12}"] \arrow[r, "r_{12}"'] & R_{12}
\end{tikzcd}$$

is a double pushout.

Proof. 1. Let $f, f'\colon p_{12} \to p$ with $(X) \quad f \circ e = f' \circ e$. Then using equations C, F, I and X and D, E, B, A, K and X respectively, one gets $f \circ f_k \circ e_k \circ i_{kk} = f' \circ f_k \circ e_k \circ i_{kk}$ and $f \circ f_k \circ e_k \circ i_{0k} = f' \circ f_k \circ e_k \circ i_{0k}$ for $k = 1,2$. Using the coproduct property of $p_0 + p_k'$ one gets $f \circ f_k \circ e_k = f' \circ f_k \circ e_k$. As e_k is an epimorphism this implies

$$(Y_k) \quad f \circ f_k = f' \circ f_k.$$

Summarizing, one gets

$$\begin{array}{ccccc}
f \circ f_1 \circ e_{01} & \underset{A}{=} & f \circ f_0 & \underset{B}{=} & f \circ f_2 \circ e_{02} \\
{\scriptstyle Y_1}\Vert & & & & {\scriptstyle Y_2}\Vert \\
f' \circ f_1 \circ e_{01} & \underset{A}{=} & f' \circ f_0 & \underset{B}{=} & f' \circ f_2 \circ e_{02}
\end{array} .$$

Because $(A) + (B)$ is a pushout, this implies $f = f'$ such that e is an epimorphism.

2. Consider the following diagram for $k = 1,2$:

$$\begin{tikzcd}
 & L_0+L'_1+L'_2 \arrow[dddl, bend right, "q"'] \arrow[ddd, "e(L)"] & & & K_0+K'_1+K'_2 \arrow[lll, "l_0+l'_1+l'_2"] \arrow[ddd, "e(K)"] \\
 & & L_0+L'_k \arrow[ul, "i_k(L)"] \arrow[d, "e_k(L)"] & K_0+K'_k \arrow[l, "l_0+l'_k"] \arrow[ur, "i_k(K)"] \arrow[d, "e_k(K)"] & \\
 & & L_k \arrow[dl, "f_k(L)"] \arrow[dll, dashed, bend right, "\overline{q}_k"'] & K_k \arrow[l, "l_k"] \arrow[dr, "f_k(K)"] & \\
X & L_{12} \arrow[l, "\overline{q}"'] & & & K_{12} \arrow[lll, "l_{12}"] \arrow[llll, bend left, "q'"]
\end{tikzcd}$$

where the square in the middle is a pushout and the squares surrounding the pushout commute. Let q, q' with $q \circ (l_0 + l'_1 + l'_2) = q' \circ e(K)$ for some object X. Then this implies

$$q \circ i_k(L) \circ (l_0 + l'_k) = q \circ (l_0 + l'_1 + l'_2) \circ i_k(K) = q' \circ e(K) \circ i_k(K) = q' \circ f_k(K) \circ e_k(K).$$

The pushout property of L_k yields that there is a morphism $\overline{q}_k \colon L_k \to X$ with $q \circ i_k(L) = \overline{q}_k \circ e_k(L)$ and $q' \circ f_k(K) = \overline{q}_k \circ l_k$. This implies

$$\begin{aligned}
& \overline{q}_1 \circ e_{01}(L) = \overline{q}_1 \circ e_1(L) \circ i_{01}(L) = q \circ i_1(L) \circ i_{01}(L) = q \circ i_0(L) \\
= \; & q \circ i_2(L) \circ i_{02}(L) = \overline{q}_2 \circ e_2(L) \circ i_{02}(L) = \overline{q}_2 \circ e_{02}(L).
\end{aligned}$$

The pushout property of L_{12} (as component of p_{12}) yields that there is a morphism $\overline{q} \colon L_{12} \to X$ with $\overline{q} \circ f_k(L) = \overline{q}_k$. It remains to prove $\overline{q} \circ e(L) = q$ and $\overline{q} \circ l_{12} = q'$. Using the available commutativities, one gets, for $k = 1, 2$,

$$\overline{q} \circ e(L) \circ i_k(L) = \overline{q} \circ f_k(L) \circ e_k(L) = \overline{q}_k \circ e_k(L) = q \circ i_k(L)$$

and

$$\overline{q} \circ e(L) \circ i_k(L) = \overline{q} \circ f_0 = \overline{q} \circ f_k(L) \circ e_{0k}(L) = \overline{q}_k \circ e_{0k} = q \circ i_k(L).$$

As $L_0 + L'_1 + L'_2$ is a coproduct, this implies $\overline{q} \circ e(L) = q$. Moreover, one gets

$$\overline{q} \circ l_{12} \circ f_k(K) = \overline{q} \circ f_k(L) \circ l_k = \overline{q}_k \circ l_k = q' \circ f_k(K)$$

such that the pushout property of K_{12} implies $\overline{q} \circ l_{12} = q'$. Both together prove the stated equations. Because of the symmetry of double pushouts, the same reasoning works for the right-hand side completing the proof. $\square$

Related Work 1.

1. As the coproduct is associative and commutative, the theorem shows that the pairs $(p_0, p'_1 + p'_2)$, $(p_0 + p'_1, p'_2)$ and $(p_0 + p'_2, p'_1)$ are covers of p_{12}. Therefore, Theorem 2 can be used showing that $p_0 + p'_1 + p'_2$ plays the same role as the amalgamated rule p_{12} in the Amalgamation Theorem (see [4,9]).

2. Let $F = \{G \underset{p_i}{\Longrightarrow} H_i\}_{i \in I} \in \mathcal{F}$ be a family of rule applications and $g_i \colon L_i \to G$ be the left matching morphisms for $i \in I$ with the particular property that each two members share a useful doubling, meaning that there is a common component rule p_0 and complement rules p'_i for $i \in I$ such that (p_0, p'_i) covers p_i. Then, obviously, $n-1$ dedoublings are possible if I has n elements. Moreover $G \underset{p_0+\sum_{i \in I} p'_i}{\Longrightarrow} Y$ for some object Y is a parallelization with dedoubling of the initial family. This very situation is considered and investigated by Ehrig et al. [9,13] as multi-amalgamation.
3. It turns out that the effect of amalgamation and multi-amalgamation can be obtained by means of particular sequentializations and parallelizations leading to parallelization with dedoubling. In both approaches, one needs to find special component rules (a common one and complements). One may argue that sequentialization and parallelization are slightly simpler and more basic constructions than amalgamation. Moreover, the general amalgamated rule does not always exist in adhesive categories as arbitrary pushouts do not exist in general. A further difference between both approaches concerns the scaling from two rule applications to an arbitrary number of them. Multi-amalgamation in [9,13] is only considered for the special case that all rule applications delete the same part of the host object.

6 Conclusion

In this paper, we have studied the parallelization of rule applications that are constructed by double-pushouts in adhesive categories. The well-known parallelization of parallel independent rule applications by means of coproducts of rules used as parallel rules is generalized in such a way that doubling can be avoided. If the dependence of two rule applications to the same object is caused by the fact that - intuitively - both delete that same part of the commonly accessed part, then independence can be restored by removing the deletion part of one of the rule applications provided that there are proper component rules complementing the deletion parts. Similarly, one can avoid that certain insertions are doubled. We have given sufficient conditions to guarantee that doubling avoidance - called dedoubling for short - works. The construction combines the sequentialization of parallel rule applications and certain re-parallelizations of the resulting rule applications. As a main result, we have shown that dedoubling within a family of rule applications preserves parallel independence. Consequently, a sequence of dedoubling steps yields a family of pairwise parallel independent rule applications that can be applied in parallel, or it gets stuck because none of the pairs of rule applications allow a further dedoubling.

To obtain a better insight into the parallelization with doubling avoidance, the following aspects may be considered:

1. A dedoubling of two rule applications requires to cover them by parallel rules with a common component rule. To find such covers is guesswork so far. Therefore, it would be helpful to come up with a systematic construction of

covers. According to the definition, the component rules are embedded into the given rules by arbitrary rule morphisms. If one assumes monomorphic embeddings and finitary categories, then there is only a finite number of component rules so that all covers can be enumerated.

2. If a family of rule applications has two members $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2}{\Longrightarrow} H_2$, then dedoubling may not be possible or dedoubling yields either $G \underset{p_1'}{\Longrightarrow} H_1'$ and $G \underset{p_2}{\Longrightarrow} H_2$ or $G \underset{p_1}{\Longrightarrow} H_1$ and $G \underset{p_2'}{\Longrightarrow} H_2'$. In Sect. 3, it is shown that $G \underset{p_1'+p_2}{\Longrightarrow} Y, G \underset{p_1+p_2'}{\Longrightarrow} Y$, and $G \underset{p_0+p_1'+p_2'}{\Longrightarrow} Y$ for some object Y. This means that both dedoublings are closely related from a semantic point of view. We expect that a result like this holds for arbitrary families and arbitrary sequences of dedoublings that does not require to store the doublings and is stronger than Theorem 4 in this way.
3. A dedoubling applied to two parallel dependent rule applications yields parallel independence so that all dedoublings have the same effect with respect to the deletion part. Dedoubling with respect to the insertion part behaves quite different as one has various options between a common component rule that does not insert anything and some insertion part that identifies the maximal mergable parts of the two involved right-hand sides. The question is whether there are more interesting standard cases between the two extremes.
4. It would be interesting to study applications of the proposed concepts. As our example in Sects. 3 and 4 indicates, the area of pattern generation is a potential candidate as many patterns, tilings, and fractals can be generated in a parallel mode of construction. Rule-based collaborative text editing (e.g. refactoring source code by multiple people) fits into this view. The authors of [13–15] have pointed out that amalgamation and multi-amalgamation are very helpful concepts for the specification of model transformations. Therefore, it may be interesting to see whether parallelization with dedoubling can play a similar role.

Acknowledgements. We are grateful to the anonymous reviewers for their valuable comments that led to various improvements.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Taentzer, G.: Parallel and distributed graph transformation: formal description and application to communication-based systems. Ph.D. thesis, TU Berlin, Shaker Verlag (1996)
2. Kniemeyer, O.: Design and implementation of a graph grammar based language for functional-structural plant modelling. Ph.D. thesis, Brandenburg University of Technology, Cottbus - Senftenberg, Germany (2008)
3. Métivier, Y., Sopena, E.: Graph relabelling systems: a general overview. Comput. Artif. Intell. **16**(2), 167–185 (1997)

4. Boehm, P., Fonio, H.-R., Habel, A.: Amalgamation of graph transformations with applications to synchronization. In: Ehrig, H., Floyd, C., Nivat, M., Thatcher, J. (eds.) CAAP 1985. LNCS, vol. 185, pp. 267–283. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-15198-2_17
5. Ehrig, H., Kreowski, H.-J.: Parallelism of manipulations in multidimensional information structures. In: Mazurkiewicz, A. (ed.) MFCS 1976. LNCS, vol. 45, pp. 284–293. Springer, Heidelberg (1976). https://doi.org/10.1007/3-540-07854-1_188
6. Kreowski, H.-J.: Manipulationen von Graphmanipulationen. Ph.D. thesis, Technische Universität Berlin (1978)
7. Corradini, A., Montanari, U., Rossi, F., Ehrig, H., Heckel, R., Löwe, M.: Algebraic approaches to graph transformation part I: basic concepts and double pushout approach. In: Rozenberg, G. (ed.) Handbook of Graph Grammars and Computing by Graph Transformation, Vol. 1: Foundations, pp. 163–245. World Scientific, Singapore (1997)
8. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. Monographs in Theoretical Computer Science. An EATCS Series. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-31188-2
9. Ehrig, H., Ermel, C., Golas, U., Hermann, F.: Graph and Model Transformation - General Framework and Applications. Monographs in Theoretical Computer Science. An EATCS Series. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-47980-3
10. Adámek, J., Herrlich, H., Strecker, G.E.: Abstract and Concrete Categories - The Joy of Cats. Dover Publications (2009)
11. Lack, S., Sobociński, P.: Adhesive and quasiadhesive categories. RAIRO Theor. Inform. Appl. **39**(3), 511–545 (2005)
12. Ehrig, H., Pfender, M., Schneider, H.J.: Graph-grammars: an algebraic approach. In: Proceedings of the 14th Annual Symposium on Switching and Automata Theory, Iowa City, Iowa, USA, 15–17 October 1973, pp. 167–180. IEEE Computer Society (1973)
13. Biermann, E., Ehrig, H., Ermel, C., Golas, U., Taentzer, G.: Parallel independence of amalgamated graph transformations applied to model transformation. In: Engels, G., Lewerentz, C., Schäfer, W., Schürr, A., Westfechtel, B. (eds.) Graph Transformations and Model-Driven Engineering. LNCS, vol. 5765, pp. 121–140. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17322-6_7
14. Born, K., Taentzer, G.: An algorithm for the critical pair analysis of amalgamated graph transformations. In: Echahed, R., Minas, M. (eds.) ICGT 2016. LNCS, vol. 9761, pp. 118–134. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40530-8_8
15. Taentzer, G., Golas, U.: Towards local confluence analysis for amalgamated graph transformation. In: Parisi-Presicce, F., Westfechtel, B. (eds.) ICGT 2015. LNCS, vol. 9151, pp. 69–86. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21145-9_5