Initial Conflicts and Dependencies: Critical Pairs Revisited

Leen Lambers[1], Kristopher Born[2], Fernando Orejas[3], Daniel Strüber[4], Gabriele Taentzer[2]

[1] Hasso-Plattner-Institut, Potsdam, Germany, `leen.lambers@hpi.de`
[2] Philipps-Universität Marburg, Germany,
`{born, taentzer}@informatik.uni-marburg.de`
[3] Technical University of Catalunia, Barcelona, Spain,
`orejas@lsi.upc.edu`
[4] Universität Koblenz-Landau, Germany, `strueber@uni-koblenz.de`

Abstract. Considering a graph transformation system, a critical pair represents a pair of conflicting transformations in a minimal context. A conflict between two direct transformations of the same structure occurs if one of the transformations cannot be performed in the same way after the other one has taken place. Critical pairs allow for static conflict and dependency detection since there exists a critical pair for each conflict representing this conflict in a minimal context. Moreover it is sufficient to check each critical pair for strict confluence to conclude that the whole transformation system is locally confluent. Since these results were shown in the general categorical framework of M-adhesive systems, they can be instantiated for a variety of systems transforming e.g. (typed attributed) graphs, hypergraphs, and Petri nets.

In this paper, we take a more declarative view on the minimality of conflicts and dependencies leading to the notions of *initial conflicts* and *initial dependencies*. Initial conflicts have the important new characteristic that for each given conflict a unique initial conflict exists representing it. We introduce initial conflicts for M-adhesive systems and show that the Completeness Theorem and the Local Confluence Theorem still hold. Moreover, we characterize initial conflicts for typed graph transformation systems and show that the set of initial conflicts is indeed smaller than the set of essential critical pairs (a first approach to reduce the set of critical pairs to the important ones). Dual results hold for initial dependencies.

1 Introduction

Graph transformations are often affected by conflicts and dependencies between the included rules. To improve their transformation specifications, users may require a list of all potential conflicts and dependencies occurring between the contained rules. Critical pair analysis (CPA) is a static analysis to enable the automated computation of such a list. The notion of critical pair was coined in the domain of mathematical logic, where it was first introduced for term rewriting
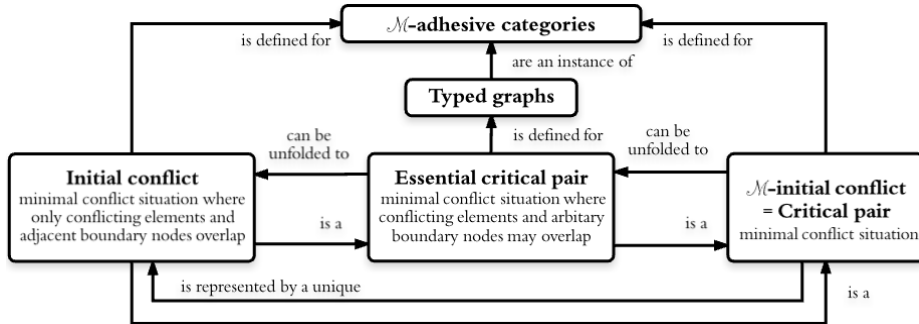
Fig. 1. Overview of critical pair kinds with their formal foundations. Characterizations are given in the category of typed graphs.

systems [1]. More recently, it turned out to be useful for graph transformation systems as well. Plump [2] and Heckel et al. [3] introduced critical pair notions for term graph rewriting and typed attributed graphs, respectively. It was Hartmut Ehrig who, together with his colleagues, came up with a generalized theory of critical pairs for adhesive high-level replacement systems [4]. A remarkable feature that CPA inherits from graph transformations is its versatility. CPA has been used in many scenarios, including conflict detection in functional system requirements [5], detection of product-line feature interactions [6], verification of model transformations [7], and numerous other software engineering scenarios. In these settings, CPA was used to show the correctness of a specification, to improve a rule set by fostering the independent application of its rules, and to support developers during design decisions.

The original critical-pair notion was focused on *delete-use* conflicts, i.e., situations where a rule deletes an element required by the second one, and the dual counterpart of *produce-use* dependencies. To consider *produce-forbid* conflicts as well, the notion of critical pair was extended to rules with negative application conditions in [8]. Each critical pair represents one such conflict situation in a *minimal context*: It comprises a graph specifying an overlap of the two considered rules, together with two jointly surjective match morphisms embedding the rules' left-hand sides into this graph.

Essential critical pairs [9] were introduced to optimize static conflict detection and local confluence analysis. They specify a well-defined subset of the set of critical pairs between a pair of rules to support a more "compact" representation of potential conflicts and dependencies, while providing the same main benefits as regular critical pairs: *completeness*, i.e. each potential conflict or dependency is represented by a critical pair in a minimal context, and *analyzibility of local confluence*, i.e., strict confluence of each critical pair implies local confluence. However, we shall see that essential critical pairs do not provide the *most* compact representation of potential conflicts and dependencies.

In this paper, we consider the following question: *Can the set of essential critical pairs be reduced even further without losing completeness and local con-*

fluence? To answer this question, we introduce the notion of *initial conflicts*. As shown in Fig. 1, initial conflicts further reduce the set of critical pairs, in the sense that the same initial conflict represents multiple essential critical pairs. More precisely, the initial conflict is obtained from these essential critical pairs by "unfolding" them, i.e., reducing the overlap of the conflicting rules. A similar relationship between essential and regular critical pairs was shown in [9]. In contrast to essential critical pairs, initial critical pairs are defined declaratively and generically in a categorical way, rather than constructively and restricted to the category of typed graphs. In sum, we make the following contributions:

– We define the notion of initial conflicts in a purely category-theoretical way, using the framework of $\mathcal{M}$-adhesive categories.
– We provide results to show that the set of initial conflicts still enjoys the completeness property and the local confluence theorem. Moreover, we introduce $\mathcal{M}$-initial conflicts and show that they are equivalent to critical pairs.
– We characterize initial conflicts for typed graph transformation systems and show that the set of initial conflicts is effectively smaller than the set of essential critical pairs.
– Dually to initial conflicts, we introduce initial dependencies.

The rest of this paper is structured as follows. Sect. 2 introduces a running example, whereas Sect. 3 revisits the necessary preliminaries. Sect. 4 introduces the notion of initial conflicts for $\mathcal{M}$-adhesive systems and its relationship with critical pairs. Readers mainly interested in initial conflicts for graph transformation systems may skip this section. Sect. 5 formally characterizes initial conflicts in the category of typed graphs. Sect. 6 outlines how new results can be transfered to dependencies. Sect. 7 discusses related work and concludes our work.

2 Motivating example

Throughout this paper, we illustrate the new concepts with an example, which specifies the operational semantics of finite automata by graph transformation. Finite automata are mainly used to recognize words that conform to regular expressions. A finite automaton consists of a set of states, a set of labeled transitions running between states, a start state, and a set of end states. If the whole word can be read by the automaton such that it finally reaches an end state, the word is in the language of this automaton. In the literature, deterministic automata are distinguished from non-deterministic ones. An automaton is *deterministic* if, for every state and symbol, there is at most one transition starting in that state and being labeled with that symbol. We will see that the specification of non-deterministic automata shows conflicts.

In the upper left corner of Fig. 2, a simple type graph for finite automata and input words is shown. A *Transition* has a *(s)ource* and a *(t)arget* edge to two *States* and has a *Label*. The *Cursor* points to the *(c)urrent* state. An input word is given by a *Queue* of *Elements* corresponding to labels. The queue points to the *(n)ext* symbol to be recognized.
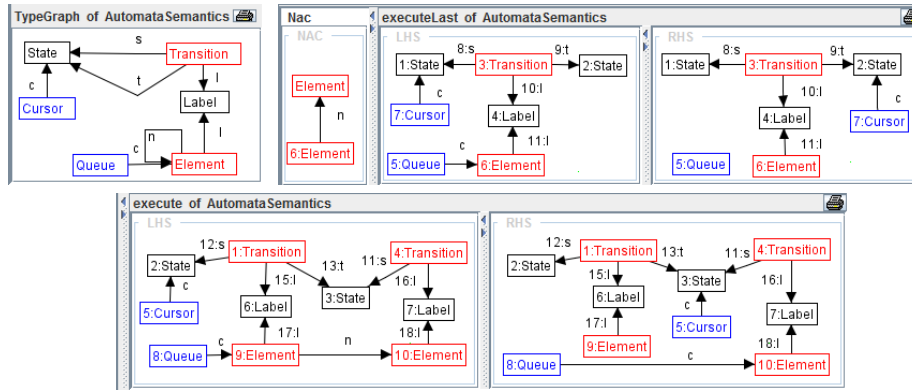
Fig. 2. Type graph and rules for executing transitions in finite automata

Additionally, Fig. 2 depicts two rules specifying the execution of automata. Rule *execute* executes a transition which is not a loop. The cursor is set to the next state and the input queue cursor points to the next element. An analogous rule is needed to execute a loop transition. For the last symbol we use a special rule, called *executeLast* which just consumes the last symbol and sets the cursor to the next state. Finally, all queue elements may be deleted.

Fig. 3 shows an example automaton A in concrete and abstract syntax. This automaton recognizes the language $L(A) = \{ab^n c | n \geq 0\}$. An example input word is *abbc*. The abstract syntax graph in Fig. 3 shows an instance graph conforming to the type graph in Fig. 2. It contains the abstract syntax information for both the example automaton and the input word, glued at all *Label*-nodes. Note that n-typed edges define the order of symbols in the input word.
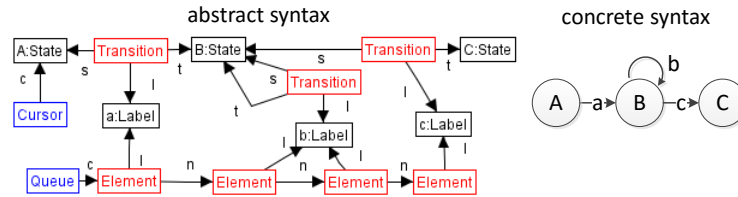

Fig. 3. An example automaton with an example input word

To recognize label a, rule *execute* is applied at its only possible match. As the result, the cursor points to *B:State*, the first n-edge is deleted, and the queue points to the first element containing label b. To recognize the whole word three further rule applications are needed.

The execute-rules cause many potential conflicts; for example, the pair *(execute, execute)* has 49 essential critical pairs. It will turn out that most of them just show variants of basically the same conflicts. Their set of initial conflicts,

however, contains just 7 pairs. (By the way, AGG [10] runs out of memory when computing all critical pairs after checking over 12,000 rule overlaps. Verigraph [11] found 51,602 overlaps with monomorphic matches, where 21,478 of them represent critical pairs.)

3   Preliminaries

In this section, we give a short introduction to $\mathcal{M}$-adhesive categories [4, 12] and $\mathcal{M}$-adhesive systems to define the setting for the categorical results in Section 4. Moreover, we recall the classical notions of conflict and critical pair as well as the corresponding results Completeness Theorem and Local Confluence Theorem within this categorical framework [4, 13].

By considering $\mathcal{M}$-adhesive categories it is possible to avoid similar investigations for different instantiations like e.g. attributed graphs, Petri nets, hypergraphs, and algebraic specifications. An $\mathcal{M}$-*adhesive category* $\langle \mathbf{C}, \mathcal{M} \rangle$ is a category $\mathbf{C}$ with a distinguished class $\mathcal{M}$ of monomorphisms satisfying certain properties. The most important one is the van Kampen (VK) property stating a certain kind of compatibility of pushouts and pullbacks along $\mathcal{M}$-morphisms. In [13] it is proven that the category of typed graphs $\langle \mathbf{Graphs_{TG}}, \mathcal{M} \rangle$ with the class $\mathcal{M}$ of all injective typed graph morphisms is $\mathcal{M}$-adhesive. In Section 5 we will instantiate the idea of initial conflicts to this category.

Within this categorical framework we introduce our notion of rule, direct transformation, and $\mathcal{M}$-adhesive system following the so-called DPO approach [13]. Note that these definitions can be instantiated to the case of typed graph transformation by replacing each object with a typed graph and each morphism with a typed graph morphism. In the category $\mathbf{Graphs}_{TG}$, $\mathcal{M}$-adhesive systems are then called typed graph transformation systems.

Definition 1 (Rule, direct transformation, $\mathcal{M}$-adhesive system). *Given an $\mathcal{M}$-adhesive category $\langle \mathbf{C}, \mathcal{M} \rangle$, then we define the following:*

- *A rule $p : L \xleftarrow{l} K \xrightarrow{r} R$ is a span of morphisms $l, r \in \mathcal{M}$. We call L (resp. R), the left-hand side (LHS) (resp. right-hand side (RHS)) of rule p.*
- *A direct transformation $G \xRightarrow{p,m} H$ from G to H via a rule $p : L \leftarrow K \rightarrow R$ and a morphism $m : L \rightarrow G$, called match, consists of the double pushout (DPO) [14] as depicted in Fig. 4. Since pushouts along $\mathcal{M}$-morphisms in an $\mathcal{M}$-adhesive category always exist, the DPO can be constructed if the pushout complement of $m \circ l$ exists. Then, the match m satisfies the gluing condition of rule p.*
- *A transformation, denoted as $G_0 \xRightarrow{*} G_n$, is a sequence $G_0 \Rightarrow G_1 \Rightarrow \cdots \Rightarrow G_n$ of direct transformations. For $n = 0$, we have the identical transformation $G_0 \Rightarrow G_0$. Moreover, for $n = 0$ we also allow isomorphisms $G_0 \cong G_0'$, because pushouts, and hence also direct transformations, are only unique up to isomorphism.*
- *Given a set of rules $\mathcal{R}$, triple $(\mathcal{C}, \mathcal{M}, \mathcal{R})$ is an $\mathcal{M}$-adhesive system.*

$$L \xleftarrow{\ \ l\ \ } K \xrightarrow{\ \ r\ \ } R \qquad\qquad B \xrightarrow{\ \ b\ \ } C$$
$$\Big\downarrow{\scriptstyle m}\quad (1)\quad \Big\downarrow\quad (2)\quad \Big\downarrow{\scriptstyle m'} \qquad\qquad \Big\downarrow{\scriptstyle k}\quad (IPO)\quad \Big\downarrow{\scriptstyle c}$$
$$G \longleftarrow D \longrightarrow H \qquad\qquad K \xrightarrow{\ \ l\ \ } L$$
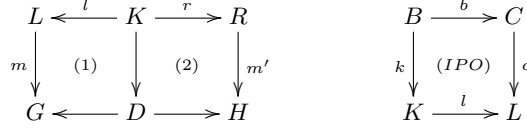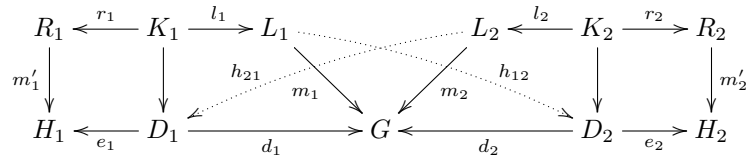
Fig. 4. Direct transformation as DPO, deletion graph constructed by initial pushout

The classical definitions for transformation pairs in conflict and critical pairs are recalled in [13]. The latter represent conflicts in a minimal context materialized by a pair of matches being jointly epimorphic. In particular, for the critical pair definition it is assumed that the $\mathcal{M}-$adhesive category comes with a so-called $\mathcal{E}'$-$\mathcal{M}$ pair factorization, generalizing the classical epi-mono factorization to a pair of morphisms with the same codomain. It is proven in [13] that the category **Graphs$_{TG}$** of typed graphs has a unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization, where $\mathcal{E}'$ is the class of jointly surjective typed graph morphism pairs. Note that we stick to the notation $\mathcal{E}'$ for jointly epimorphic morphisms as in [13], where $\mathcal{E}$ on the other hand is used to denote a class of epimorphisms.

Definition 2 (conflict, critical pair). *A pair of direct transformations* $t_1 : G \overset{p_1,m_1}{\Rightarrow} H_1$ *and* $t_2 : G \overset{p_2,m_2}{\Rightarrow} H_2$ *is in* conflict *if* $\nexists h_{12} : L_1 \to D_2 : d_2 \circ h_{12} = m_1$ *or* $\nexists h_{21} : L_2 \to D_1 : d_1 \circ h_{21} = m_2$.

$$R_1 \xleftarrow{\ r_1\ } K_1 \xrightarrow{\ l_1\ } L_1 \qquad\qquad L_2 \xleftarrow{\ l_2\ } K_2 \xrightarrow{\ r_2\ } R_2$$
$$\Big\downarrow{\scriptstyle m_1'}\qquad \Big\downarrow\qquad\qquad\qquad\qquad\qquad\qquad \Big\downarrow\qquad \Big\downarrow{\scriptstyle m_2'}$$
$$H_1 \xleftarrow{\ e_1\ } D_1 \xrightarrow[\ \ \ d_1\ \ \ ]{} G \xleftarrow[\ \ \ d_2\ \ \ ]{} D_2 \xrightarrow{\ e_2\ } H_2$$

Given an $\mathcal{E}'$-$\mathcal{M}$ *pair factorization, a* critical pair *is a pair of direct transformations* $K \overset{p_1,m_1}{\Rightarrow} P_1$ *and* $K \overset{p_2,m_2}{\Rightarrow} P_2$ *in conflict with* (m_1, m_2) *in* $\mathcal{E}'$.

Now, we recall the Completeness Theorem for critical pairs, where we need the notion of extension morphism and extension diagram as presented in [4, 13].

Definition 3 (Extension diagram). *An* extension diagram *is a diagram (1) as shown on the left of Fig. 5 where* $f : G' \to G$ *is a morphism, called* extension morphism, *and* $t : G \overset{p}{\Longrightarrow} H$ *as well as* $t' : G' \overset{p}{\Longrightarrow} H'$ *are two direct transformations via the same rule* p *with matches* m' *and* $f \circ m'$ *respectively, defined by the four pushouts in the middle of Fig. 5.*

Transformations are actually extended by extending their context D' to D. Morphisms $f : G' \to G$ and $f' : H' \to H$ are the resulting pushout morphisms. In the category **Graphs**$_{TG}$, this means that the context graph D' may be embedded into a larger one and/or elements of it may be glued together. Corresponding actions are reflected in f and f' but no additional actions may happen.

$$L \xleftarrow{} K \xrightarrow{} R$$

$$
\begin{array}{ccc}
G' \xRightarrow{p} H' & \quad G' \xleftarrow{} D' \xrightarrow{} H' & \quad P_1 \xLeftarrow{p_1} K \xRightarrow{p_2} P_2 \\
f \downarrow \;\; (1) \;\; \downarrow f' & \quad f\downarrow \qquad \downarrow \qquad \downarrow f' & \quad \downarrow \;\; (1) \;\; m \;\; (2) \;\; \downarrow \\
G \xRightarrow{p} H & \quad G \xleftarrow{} D \xrightarrow{} H & \quad H_1 \xLeftarrow{p_1} G \xRightarrow{p_2} H_2
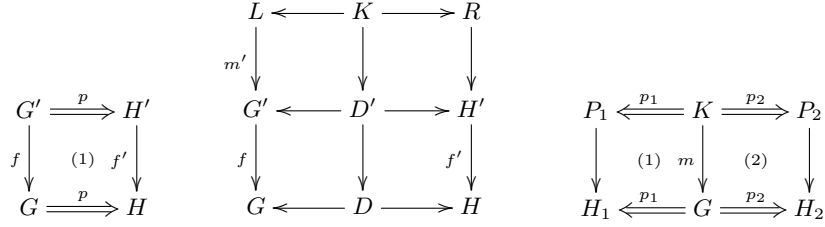\end{array}
$$

Fig. 5. Extension diagram (overview and more detailed), extension diagram for transformation pair

The Completeness Theorem [4, 13] for critical pairs states that each potential conflict can be represented in a minimal context by some critical pair. For conciseness reasons in the following we sometimes write that the $\mathcal{M}$-adhesive system comes with an $\mathcal{E}'$-$\mathcal{M}$ pair factorization (or some other additional requirement) if the corresponding $\mathcal{M}$ adhesive category does.

Theorem 4 (Completeness Theorem). *Let an $\mathcal{M}$-adhesive system with an $\mathcal{E}'$-$\mathcal{M}$ pair factorization be given. For each pair of direct transformations $H_1 \xLeftarrow{p_1} G \xRightarrow{p_2} H_2$ in conflict, there is a critical pair $P_1 \xLeftarrow{p_1} K \xRightarrow{p_2} P_2$ with extension diagrams (1) and (2) and $m \in \mathcal{M}$ as depicted on the right of Fig. 5.*

The Local Confluence Theorem [4, 13] states that, by checking each critical pair for *strict confluence*, one can conclude local confluence of the overall transformation system. Strict confluence ensures that the largest subobject of K preserved by both t_1 and t_2 is preserved by the transformations establishing local confluence. Note that for this result the $\mathcal{M}$-adhesive category needs to fulfill an additional requirement: The category needs so-called initial pushouts describing the existence of a "smallest" pushout over a morphism [13]. It is proven in [13] that the category **Graphs$_{\mathbf{TG}}$** of typed graphs has initial pushouts.

Theorem 5 (Local Confluence Theorem). *Given an $\mathcal{M}$-adhesive system with an $\mathcal{E}'$-$\mathcal{M}$ pair factorization and initial pushouts over $\mathcal{M}$-morphisms, it is locally confluent if all its critical pairs are strictly confluent.*

For a closer look at conflicts we have to identify the following two rule parts: the deletion object comprising the part to be deleted and its boundary specifying how the deletion object is connected to the preserved rule part.

Definition 6 (Boundary and deletion objects). *Let an $\mathcal{M}$-adhesive system with initial POs [13] over $\mathcal{M}$ and a rule $p : L \xleftarrow{l} K \xrightarrow{r} R$ as well as an initial pushout (IPO) (see Fig. 4) over morphism l be given. Then we say that B is the* boundary object *for rule p and the context object C in this IPO is the* deletion object *for rule p.*

4 Initial Conflicts

The original idea of critical pairs consists of considering all possible conflicting transformations in a minimal context. In the classical critical pair definition this minimal context is materialized by a pair of jointly epimorphic matches from a special set $\mathcal{E}'$ arising from the $\mathcal{E}'$-$\mathcal{M}$ pair factorization as additional requirement to the $\mathcal{M}$-adhesive category. We propose here a more *declarative* view on a pair of direct transformations in conflict to be minimal resulting in the subsequent definition of *initial conflicts*. In categorical terms, one can use actually the notion of *initiality of transformation pairs* to obtain this new view on critical pairs. Interestingly, it will turn out that each initial conflict is a critical pair but not the other way round. We will show however at the end of this section that all initial conflicts still satisfy the Completeness Theorem as well as the Local Confluence Theorem. Consequently, we have found an important subset within the set of classical critical pairs for performing static conflict detection as well as local confluence analysis for $\mathcal{M}$-adhesive systems. Finally, we will see also that the notion of $\mathcal{M}$-initiality allowing merely $\mathcal{M}$-morphisms as extension morphisms leads to the notion of $\mathcal{M}$-*initial conflicts*, representing an equivalent characterization of critical pairs provided that the $\mathcal{E}'$-$\mathcal{M}$ pair factorization for building them is unique. We will see that by definition ($\mathcal{M}$-)initial conflicts have the important new characteristic that for each given pair of conflicting transformations there exists a *unique* ($\mathcal{M}$-)initial conflict representing it.

Definition 7 (($\mathcal{M}$-)Initial transformation pair). *Given a pair of direct transformations* $(t_1, t_2) : H_1 \overset{p_1, m_1}{\Longleftarrow} G \overset{p_2, m_2}{\Longrightarrow} H_2$, *then* $(t_1^I, t_2^I) : H_1^I \overset{p_1, m_1^I}{\Longleftarrow} G^I \overset{p_2, m_2^I}{\Longrightarrow} H_2^I$ *is an* initial transformation pair *(resp. $\mathcal{M}$-initial transformation pair) for* (t_1, t_2) *if it can be embedded into* (t_1, t_2) *via extension diagrams (1) and (2) and extension morphism f^I (resp. $f^I \in \mathcal{M}$) as in Fig. 6 such that for each transformation pair* $(t_1', t_2') : H_1' \overset{p_1, m_1'}{\Longleftarrow} G' \overset{p_2, m_2'}{\Longrightarrow} H_2'$ *that can be embedded into* (t_1, t_2) *via extension diagrams (3) and (4) and extension morphism f (resp. $f \in \mathcal{M}$) as in Fig. 6 it holds that* (t_1^I, t_2^I) *can be embedded into* (t_1', t_2') *via unique extension diagrams (5) and (6) and unique vertical morphism f'^I (resp. $f'^I \in \mathcal{M}$) s.t. $f \circ f'^I = f^I$.*

Lemma 8 (Uniqueness of ($\mathcal{M}$-)initial transformation pair). *Given a pair of direct transformations* $(t_1, t_2) : H_1 \overset{p_1, m_1}{\Longleftarrow} G \overset{p_2, m_2}{\Longrightarrow} H_2$ *then, if* $(t_1^I, t_2^I) : H_1^I \overset{p_1, m_1^I}{\Longleftarrow} G^I \overset{p_2, m_2^I}{\Longrightarrow} H_2^I$ *is an initial pair of transformations (resp. $\mathcal{M}$-initial pair of transformations) for* (t_1, t_2), *any other initial transformation pair (resp. $\mathcal{M}$-initial transformation pair) for* (t_1, t_2) *is isomorphic to* (t_1^I, t_2^I).

Proof. Consider some other initial pair $(t_1'^I, t_2'^I) : H_1'^I \overset{p_1, m_1^I}{\Longleftarrow} G'^I \overset{p_2, m_2^I}{\Longrightarrow} H_2'^I$ for (t_1, t_2). Then the extension diagrams in Fig. 7 can be built by definition of ($\mathcal{M}$)-initial pairs. Now consider for (t_1^I, t_2^I) trivial extension diagrams via the identity extension morphism $id : G^I \to G^I$. The extension morphism of the extension diagrams (7)+(5) and (8)+(6) w.r.t. (t_1, t_2) needs to be equal to the identity extension morphism by definition. Analogously, one can argue for (5)+(7) and (6)+(8). Therefore both initial pairs are isomorphic.

$$
\begin{array}{ccc}
H_1^I \xLeftarrow{p_1,m_1^I} G^I \xRightarrow{p_2,m_2^I} H_2^I \qquad & \qquad H_1^I \xLeftarrow{p_1,m_1^I} G^I \xRightarrow{p_2,m_2^I} H_2^I \\
\Big\downarrow g_1^I \quad (1) \quad f^I \Big\downarrow \quad (2) \quad \Big\downarrow g_2^I \qquad & \qquad g_1'^I \Big\downarrow \quad (5) \quad f'^I \Big\downarrow \quad (6) \quad \Big\downarrow g_2'^I \\
H_1 \xLeftarrow{p_1,m_1} G \xRightarrow{p_2,m_1} H_2 \qquad & \qquad H_1' \xLeftarrow{p_1,m'_1} G' \xRightarrow{p_2,m'_2} H_2' \\
 & \qquad \Big\downarrow g_1 \quad (3) \quad f \Big\downarrow \quad (4) \quad \Big\downarrow g_2 \\
 & \qquad H_1 \xLeftarrow{p_1,m_1} G \xRightarrow{p_2,m_2} H_2
\end{array}
$$

Fig. 6. ($\mathcal{M}$-)initial transformation pair $H_1^I \xLeftarrow{p_1,m_1^I} G^I \xRightarrow{p_2,m_2^I} H_2^I$ for $H_1 \xLeftarrow{p_1,m_1} G \xRightarrow{p_2,m_2} H_2$

$$
\begin{array}{ccc}
H_1'^I \xLeftarrow{p_1,m'^I_1} G'^I \xRightarrow{p_2,m'^I_2} H_2'^I \qquad & \qquad H_1^I \xLeftarrow{p_1,m_1^I} G^I \xRightarrow{p_2,m_2^I} H_2^I \\
\Big\downarrow \quad (5) \quad \Big\downarrow \quad (6) \quad \Big\downarrow \qquad & \qquad \Big\downarrow \quad (7) \quad \Big\downarrow \quad (8) \quad \Big\downarrow \\
H_1^I \xLeftarrow{p_1,m_1^I} G^I \xRightarrow{p_2,m_2^I} H_2^I \qquad & \qquad H_1'^I \xLeftarrow{p_1,m'^I_1} G'^I \xRightarrow{p_2,m'^I_2} H_2'^I \\
\Big\downarrow g_1^I \quad (1) \quad f^I \Big\downarrow \quad (2) \quad \Big\downarrow g_2^I \qquad & \qquad g_1'^I \Big\downarrow \quad (3) \quad f'^I \Big\downarrow \quad (4) \quad \Big\downarrow g_2'^I \\
H_1 \xLeftarrow{p_1,m_1} G \xRightarrow{p_2,m_2} H_2 \qquad & \qquad H_1 \xLeftarrow{p_1,m_1} G \xRightarrow{p_2,m_2} H_2
\end{array}
$$

Fig. 7. Uniqueness of ($\mathcal{M}$-)initial transformation pair

Our key notion of initial conflicts is based on the *existence of initial transformation pairs* for *conflicting* transformation pairs. It describes the "smallest" conflict that can be embedded into a given conflict. It is an open issue to come up with a constructive categorical characterization in the context of $\mathcal{M}$-adhesive systems, which is the reason for having it as an additional requirement (formulated in Def. 9) for now. It is possible, however, to constructively characterize $\mathcal{M}$-initial transformation pairs for conflicts provided that a unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization is given (see Lemma 10). The key difference between initiality and $\mathcal{M}$-initiality is that the extension morphism used to embed the "smallest" conflict into a given conflict is general or needs to be in $\mathcal{M}$, respectively.

Definition 9 (Existence of initial transformation pair for conflict). *An $\mathcal{M}$-adhesive system has* initial transformation pairs for conflicts *if, for each transformation pair in conflict (t_1, t_2), the initial transformation pair (t_1^I, t_2^I) exists.*

Lemma 10 (Existence of $\mathcal{M}$-initial transformation pair for conflict). *In an $\mathcal{M}$-adhesive system with unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization, for each pair of transformations (t_1, t_2) in conflict, there exists an $\mathcal{M}$-initial transformation pair (t_1^I, t_2^I). In particular, it corresponds to the classical critical pair as constructed in Theorem 4.*

Proof. Consider the critical pair (t_1^I, t_2^I) as given by Theorem 4. We show that this is indeed an $\mathcal{M}$-initial transformation pair for (t_1, t_2). Given matches (m_1, m_2) of transformation pair (t_1, t_2) and matches (m_1^I, m_2^I) for the pair (t_1^I, t_2^I) built via the pair factorization (as on the left of Fig. 6). Then $(m_1^I, m_2^I) \in \mathcal{E}'$ and the extension morphism f^I from (t_1^I, t_2^I) to (t_1, t_2) is in $\mathcal{M}$ and $f^I \circ m_1^I = m_1$ and $f^I \circ m_2^I = m_2$. Consider some other pair (t_1', t_2') that can be embedded via some extension morphism $f : G' \to G \in \mathcal{M}$ into (t_1, t_2) (as on the right of Fig. 6). According to Theorem 4 we again have a critical pair with matches in $\mathcal{E}'$ that can be embedded into (t_1', t_2') via some extension morphism f'^I in $\mathcal{M}$. Since the $\mathcal{E}'$-$\mathcal{M}$ pair factorization is unique and $\mathcal{M}$-morphisms are closed under composition, this will actually be indeed the same critical pair (t_1^I, t_2^I) as for (t_1, t_2).

Now we are ready to introduce our notion of ($\mathcal{M}$-)initial conflicts representing the set of all possible "smallest" conflicts. Like for classical critical pairs they are defined for a given $\mathcal{M}$-adhesive system allowing for static conflict detection.

Definition 11 (($\mathcal{M}$-)Initial conflict). *Given an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts, a pair of direct transformations in conflict* $(t_1, t_2) : H_1 \overset{p_1}{\Longleftarrow} G \overset{p_2}{\Longrightarrow} H_2$ *is an* initial conflict *if it is isomorphic to the initial transformation pair for* (t_1, t_2).

Given an $\mathcal{M}$-adhesive system with unique $\mathcal{E}'$-$\mathcal{M}$-pair factorization, a pair of direct transformations in conflict $(t_1, t_2) : H_1 \overset{p_1}{\Longleftarrow} G \overset{p_2}{\Longrightarrow} H_2$ *is an $\mathcal{M}$-initial* conflict *if it is isomorphic to the $\mathcal{M}$-initial transformation pair for* (t_1, t_2).

It follows quite straightforwardly that the set of $\mathcal{M}$-initial conflicts corresponds to the classical set of critical pairs for an $\mathcal{M}$-adhesive system with unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization.[1] Moreover, it follows that each initial conflict is an $\mathcal{M}$-initial conflict (or critical pair), in particular. A counterexample for the reverse direction will be given in the next section.

Theorem 12 ($\mathcal{M}$-Initial conflict = critical pair). *In an $\mathcal{M}$-adhesive system with unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization, each $\mathcal{M}$-initial conflict is a critical pair and vice versa.*

Proof. Given some $\mathcal{M}$-initial conflict $(t_1^I, t_2^I) : H_1^I \overset{p_1}{\Longleftarrow} G^I \overset{p_2}{\Longrightarrow} H_2^I$. Then it follows directly from Def. 2, Def. 11 and Lemma 10 that (t_1^I, t_2^I) is a critical pair because it is in conflict and its matches are in $\mathcal{E}'$.

Given some critical pair $(t_1^I, t_2^I) : H_1^I \overset{p_1}{\Longleftarrow} G^I \overset{p_2}{\Longrightarrow} H_2^I$, then we need to show that it is an $\mathcal{M}$-initial conflict. When constructing the initial transformation pair for (t_1^I, t_2^I) according to Lemma 10, a pair of isomorphic transformations w.r.t. (t_1^I, t_2^I) would be constructed because of the $\mathcal{E}'$-$\mathcal{M}$ pair factorization being unique and the fact that one could choose alternatively as extension morphism the identity morphism on G^I (being in $\mathcal{M}$), since the matches are already in $\mathcal{E}'$.

---

[1] Classical critical pairs are slightly more general since they do not require uniqueness of the $\mathcal{E}'$-$\mathcal{M}$ pair factorization.

Theorem 13 (Initial conflict is $\mathcal{M}$-Initial conflict). *In an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts and a unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization, each initial conflict is an $\mathcal{M}$-initial conflict.*

Proof. Given some initial conflict $(t_1^I, t_2^I) : H_1^I \overset{p_1}{\Longleftarrow} G^I \overset{p_2}{\Longrightarrow} H_2^I$, then because of Lemma 10 we can construct an $\mathcal{M}$-initial transformation pair for it. By definition, each $\mathcal{M}$-initial transformation pair is also an initial transformation pair since each morphism in $\mathcal{M}$ is a regular morphism. Because of Lemma 8, such an initial pair is unique and, for an initial conflict, isomorphic to (t_1^I, t_2^I) in particular, such that the initial transformation pair is indeed an $\mathcal{M}$-initial pair.

To decide if initial conflicts can replace critical pairs for detecting conflicts and analyzing local confluence statically, we investigate now if the Completeness Theorem and Local Confluence Theorem hold. The Completeness Theorem for initial conflicts can indeed be formulated in a slightly modified way w.r.t. Thm 4. This is because the extension morphism is not necessarily in $\mathcal{M}$ anymore. Informally speaking, we are able to represent several critical pairs by one initial conflict by unfolding elements that were overlapped unnecessarily (i.e. without having importance for the described conflict). Note also that, instead of requiring an $\mathcal{E}'$-$\mathcal{M}$ pair factorization as in the classical Completeness Theorem for critical pairs, we assume the existence of initial transformation pairs for conflicts.

Lemma 14 (Conflict inheritance). *Given a pair of direct transformations $(t_1, t_2) : H_1 \overset{p_1}{\Longleftarrow} G \overset{p_2}{\Longrightarrow} H_2$ in conflict and another pair of direct transformations $(t_1', t_2') : H_1' \overset{p_1}{\Longleftarrow} G' \overset{p_2}{\Longrightarrow} H_2'$ that can be embedded into (t_1, t_2) via extension diagrams (1) and (2) and extension morphism f in Fig. 8, then (t_1', t_2') is also in conflict.*
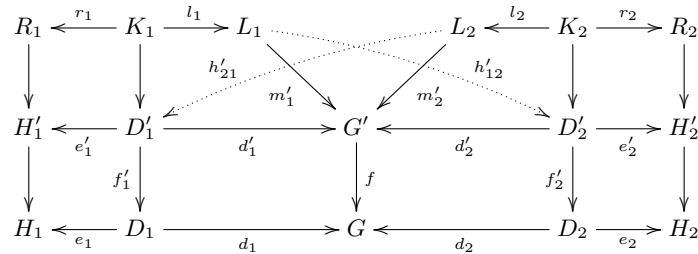

Fig. 8. Conflict inheritance

Proof. Assume that $(t_1', t_2') : H_1' \overset{p_1, m_1'}{\Longleftarrow} G' \overset{p_2, m_2'}{\Longrightarrow} H_2'$ are parallel independent. This means that some morphism h_{12}' (and h_{21}') exists such that $d_1' \circ h_{12}' = m_2'$ (and $d_2' \circ h_{21}' = m_1'$). Then $(t_1, t_2) : H_1 \overset{p_1, m_1}{\Longleftarrow} G \overset{p_2, m_2}{\Longrightarrow} H_2$ with $f \circ m_1' = m_1$ and $f \circ m_2' = m_2$ would be parallel independent as well, which is a contradiction. This

is because a morphism $h_{12} = f'_1 \circ h'_{12}$ would exist such that $d_1 \circ h_{12} = d_1 \circ f'_1 \circ h'_{12} = f \circ d'_1 \circ h'_{12} = f \circ m'_2 = m_2$ and similarly, a morphism $h_{21} = f'_2 \circ h'_{21}$ would exist such that $d_2 \circ h_{21} = m_1$.

Theorem 15 (Completeness theorem for initial conflicts). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts. For each pair of direct transformations $(t_1, t_2) : H_1 \overset{p_1}{\Longleftarrow} G \overset{p_2}{\Longrightarrow} H_2$ in conflict, there is an initial conflict $(t_1^I, t_2^I) : P_1 \overset{p_1}{\Longleftarrow} K \overset{p_2}{\Longrightarrow} P_2$ with extension diagrams (1) and (2).*

Proof. We can assume the existence of the initial transformation pair (t_1^I, t_2^I) for the given pair (t_1, t_2) in conflict. It remains to show that the initial transformation pair (t_1^I, t_2^I) for (t_1, t_2) is indeed an initial conflict according to Def. 11. Firstly, the transformation pair (t_1^I, t_2^I) is in conflict according to Lemma 14. Secondly, each initial conflict for (t_1^I, t_2^I) needs to be isomorphic to (t_1^I, t_2^I) since we would have found a non-isomorphic initial transformation pair for (t_1, t_2) by composition of extension diagrams otherwise. This would contradict Lemma 8.

The Local Confluence Theorem can be formulated for initial conflicts similarly to the one for classical critical pairs because its proof actually does not need the requirement that extension morphisms should be in $\mathcal{M}$.

Theorem 16 (Local confluence theorem for initial conflicts). *Given an $\mathcal{M}$-adhesive system with initial pushouts and initial transformation pairs for conflicts, an $\mathcal{M}$-adhesive system is locally confluent if all its initial conflicts are strictly confluent.*

Proof. The proof runs completely analogously to the proof of the regular Local Confluence Theorem (Theorem 5 in [13]). The only difference is that for this proof, we need initial pushouts over general morphisms whereas in Theorem 5 initial pushouts over $\mathcal{M}$-morphisms are sufficient. The proof requires initial pushouts over the extension morphism m embedding a critical pair (or initial conflict) into a pair of conflicting transformations. This extension morphism belongs to the special subset $\mathcal{M}$ of monomorphisms for classical critical pairs, but it is a general morphism in the case of initial conflicts.

In summary, given an $\mathcal{M}$-adhesive system, we obtain the Completeness and Local Confluence Theorem in slightly different flavors. For Completeness of $\mathcal{M}$-initial conflicts (or classical critical pairs) we assume to have a unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization and for Local Confluence we in addition require initial POs over $\mathcal{M}$. For Completeness of initial conflicts we assume the existence of initial transformation pairs for conflicts (*) and for Local Confluence we in addition require initial POs. For the category of typed graphs it is shown in [13] that all these requirements hold apart from requirement (*) proven in the next section.

5 Initial Conflicts for Typed Graph Transformation

In this section, we discuss how initial conflicts look like in graph transformation systems, i.e., in the category $\mathbf{Graphs}_{TG}$. Moreover, we clarify how they are related to essential critical pairs which were introduced in [9] as a first optimization

of critical pairs in graph transformation systems. Essential critical pairs form a subset of critical pairs for which the Completeness Theorem as well as the Local Confluence Lemma still hold. Therefore, an obvious question is the following: Does the set of initial conflicts correspond to the set of essential critical pairs in the case of typed graph transformation systems? It turns out that, in general, the set of initial conflicts is a proper subset of the set of essential critical pairs here. First, we show an initial conflict occurring in our running example.
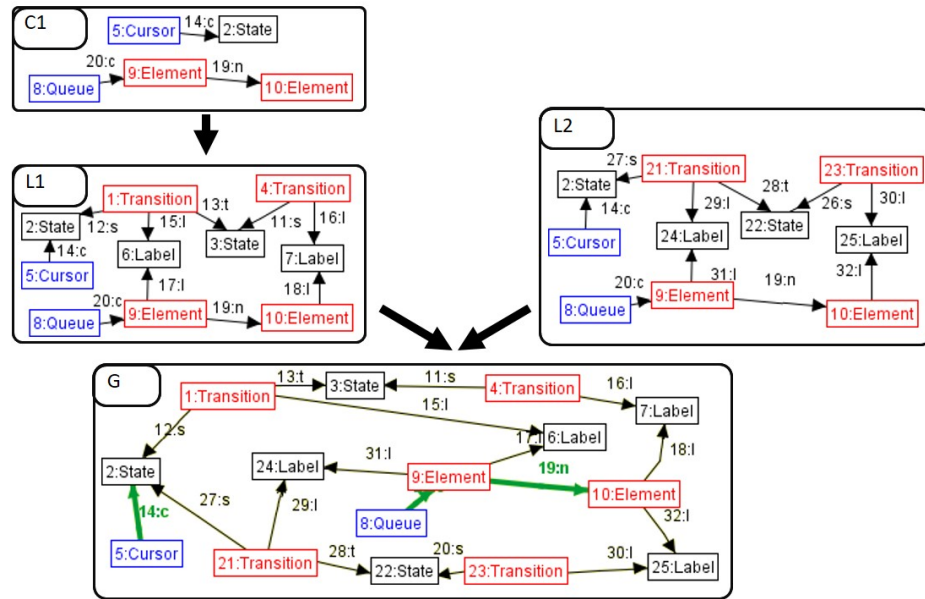
Fig. 9. Example for an initial conflict

Example 17 (Initial conflict). In a non-deterministic automaton there may be a state with two subsequent transitions, both triggered by the same label. This situation is described symbolically by the (excerpt of the) initial conflict in Fig. 9. [2] Both transitions can be executed, i.e., the rule *execute* is applicable with different results at two different matches. These matches lead to transformations in conflict since they are both triggered and therefore change the current queue pointer as well as the current cursor position. The corresponding edges are highlighted in the overlap graph (green) at the bottom of the figure. Together with their adjacent nodes they form the actual overlap of both matches. Note that applying

---

[2] Note that this situation is somewhat unrealistic, since it allows a symbol to be connected to two different labels in principle. However, graph G is supposed to be embedded into realistic situations to check if a pair of transformations is conflicting. It is part of future work to integrate the notion of constraints into our theory of initial conflicts, leading – if possible – to realistic situations already in initial conflicts.

rule *execute* at these matches leads to an initial conflict since the overlap is in deleted elements and their adjacent boundary nodes only. (See also Lemma 21 below.) If the overlap is such small, no other transformation pair is embeddable since unfoldings can occur in preserved elements only.

In the category $\mathbf{Graphs}_{TG}$, a critical pair is essential if two injective matches overlap in deleted elements and boundary nodes only [9]. The following example illustrates that indeed not each essential critical pair is an initial conflict.
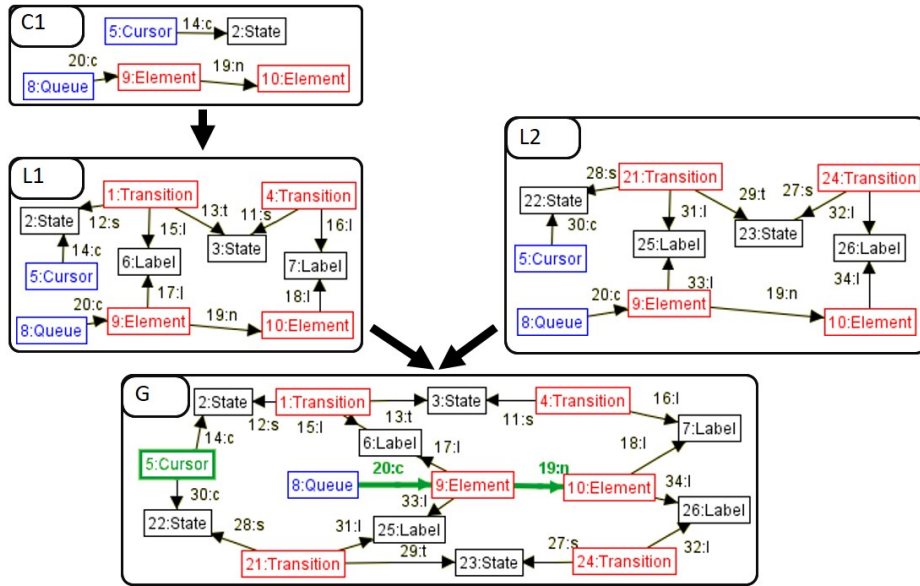
Fig. 10. Example for an essential critical pair not being an initial conflict

Example 18 (Essential critical pair not being an initial conflict). A *parallel* automaton may have several transitions that can be executed in the current states. Such a situation is described by the (excerpt of the) critical pair in Fig. 10. There exist two different current states with outgoing transitions both recognizing the same label. In this case, both transitions could be executed, i.e., the rule *execute* can be applied at two different matches. Since the matches overlap in deleted elements and isolated boundary nodes (as highlighted in green) only, the critical pair shown in Fig. 10 is essential. In particular, the isolated boundary node *5:Cursor* occurs in the overlap (and not the adjacent edges that are to be deleted). The same conflict would be specified if the cursor nodes of both LHSs were not overlapped. Hence, this essential critical pair is not an initial conflict. A similar critical pair with two cursors would be embeddable into the depicted one. Since that cannot be further unfolded, it would represent the corresponding initial conflict.

The following lemma shows that the category $\mathbf{Graphs}_{TG}$ has initial transformation pairs for conflicts and hence, initial conflicts. As a preparatory work, we define matches that do not overlap in isolated boundary nodes. If they would, then it would be possible to unfold the matches at these isolated boundary nodes.

Definition 19 (No isolated boundary node). *Given two rules p_1 and p_2 with LHSs L_1 and L_2, boundary graphs B_1 and B_2 as well as deletion graphs C_1 and C_2 as in Def. 6. Morphisms $m_1 : L_1 \to G$ and $m_2 : L_2 \to G$ do not overlap in isolated boundary nodes if $\forall x \in m_1(c_1(b_1(B_1))) \cap m_2(L_2) :$*
$\exists e \in m_1(c_1(C_1)) \cap m_2(L_2) : x = src(e) \vee x = tgt(e)$ *and*
$\forall x \in m_2(c_2(b_2(B_2))) \cap m_1(L_1) :$
$\exists e \in m_2(c_2(C_2)) \cap m_1(L_1) : x = src(e) \vee x = tgt(e)$

Lemma 20 (Existence of initial transformation pairs in $\mathbf{Graphs}_{TG}$). *Given a pair of direct transformations (t_1, t_2) in conflict, there is an initial transformation pair for (t_1, t_2), in the category $\mathbf{Graphs}_{TG}$.*

Proof. Due to the Completeness Theorem for critical pairs [13] there is a critical pair $cp : H_1 \overset{p_1,m_1}{\Longleftarrow} G \overset{p_2,m_2}{\Longrightarrow} H_2$ for (t_1, t_2). By the critical pair definition the matches m_1 and m_2 are jointly surjective. If $O = m_1(L_1) \cap m_2(L_2)$ contained some graph elements preserved by both rules, cp is tried to be unfolded at these nodes and edges, i.e., a critical pair cp' is searched which does not map these elements to the same one in O. This is always possible for edges. It is also possible for nodes if they do not have incident edges to be deleted, also being in O. The dangling edge condition for unfolded nodes cannot be violated after unfolding if it was not violated before since the same amount or fewer incident edges per unfolded node arise. The identification condition is also fulfilled after unfolding if it was fulfilled before since fewer elements are identified afterwards. Unfolding a critical pair as much as possible in this way yields the transformation pair $itp : H_1^I \overset{p_1,m_1^I}{\Longleftarrow} G^I \overset{p_2,m_2^I}{\Longrightarrow} H_2^I$ where the only preserved elements in $m_1^I(L_1) \cap m_2^I(L_2)$ are boundary nodes with incident edges to be deleted. A further unfolding is not possible since we would not find a corresponding extension diagram. Remember that an extension morphism can only unfold elements that are commonly preserved by both transformations. Preserved nodes with at least one incident edge to be deleted being overlapped as well cannot be unfolded since this edge would have to be unfolded as well.

We have to show now that itp is an initial transformation pair for (t_1, t_2). It is obvious that itp can be embedded into cp, which can be embedded into (t_1, t_2) via extension diagrams and extension morphisms. Given any other transformation pair tp that can be embedded into (t_1, t_2), tp may differ from (t_1, t_2) just by having fewer commonly preserved elements or by unfolding of preserved elements. itp can be embedded into tp since it contains the minimal number of preserved elements and the minimal overlap of preserved elements. The uniqueness of the corresponding extension diagrams and morphism follows from the construction of itp, i.e., the construction of critical pairs uses a unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization and the unfolding is canonical.

As Lemma 20 suggests an initial conflict is a transformation pair in conflict with minimal context and maximal unfolding of preserved elements.

Theorem 21 (Initial conflict in Graphs$_{TG}$). *In the category* **Graphs**$_{TG}$*, a transformation pair* $ic : H_1 \stackrel{p_1,m_1}{\Longleftarrow} G \stackrel{p_2,m_2}{\Longrightarrow} H_2$ *is an initial conflict iff ic has the following properties:*

1. *Minimal context: m_1 and m_2 are jointly surjective.*
2. *At least one element in delete-use conflict:*
   $m_1(L_1) \cap m_2(L_2) \nsubseteq m_1(l_1(K_1)) \cap m_2(l_2(K_2))$.
3. *Overlap in deletion graphs only:*
   $m_1(L_1) \cap m_2(L_2) \subseteq (m_1(c_1(C_1) \cap m_2(L_2)) \cup (m_1(L_1) \cap m_2(c_2(C_2)))$ *with*
   $c_1 : C_1 \to L_1$ *and* $c_2 : C_2 \to L_2$ *being defined as in Def. 6.*
4. *No isolated boundary node in overlap graph: m_1, m_2 as given in Def. 19.*

Proof. Given the initial conflict ic, we show that it fulfills items 1. to 4.: According to Def. 11, ic is isomorphic to the initial transformation pair for ic. This transformation pair can be constructed as in Lemma 20 and it is unique due to Lemma 8. Hence, we follow this construction and deduce the properties ic has to satisfy. The first step yields a critical pair which fulfills items 1. and 2. as shown in e.g. [13]. After the maximal unfolding of this critical pair, items 1. and 2. are still fulfilled since unfolding does not add context (item 1.) and does not unfold elements to be deleted (item 2.). In addition, items 3. and 4. are fulfilled.

Given the transformation pair ic fulfilling items 1. to 4., we show that ic is an initial conflict. When constructing the initial transformation pair for ic according to Lemma 20, a pair of isomorphic transformations to ic would be constructed since items 1. and 2. lead to an isomorphic critical pair and items 3. and 4. ensure that no more unfoldings can be made.

The theorem above shows in particular that each initial conflict is an essential critical pair satisfying properties 1. to 3. Example 18 shows, however, that not each essential critical pair is an initial conflict.

6   Initial Dependencies

To reason about initial dependencies for a rule pair (p_1, p_2), we consider the dual concepts and results that we get when inverting the left transformation of a conflicting pair. This means that we check if $G \stackrel{p_1^{-1},m'_1}{\Longleftarrow} H_1 \stackrel{p_2,m_2}{\Longrightarrow} H_2$ is parallel dependent, which is equivalent to the sequence $G \stackrel{p_1,m_1}{\Longrightarrow} H_1 \stackrel{p_2,m_2}{\Longrightarrow} H_2$ being sequentially dependent. Rule p^{-1} is the inverse of rule p obtained by exchanging morphisms l and r (Def. 1). This exchange is possible since a transformation is symmetrically defined by two pushouts. They ensure in particular that morphisms $m : L \to G$ as well as $m' : R \to H$ fulfill the gluing condition.

Initial transformation sequences and dependencies can then be defined analogously to Defs. 7 and 11. Initial dependencies show dependencies in such a way

that there is no other dependency that can be extended to it. In the category **Graphs**$_{TG}$ this means that each initial dependency is characterized by a jointly surjective pair of morphisms, consisting of the co-match of p_1 and match of p_2, which overlap in at least one graph element produced by p_1 and used by p_2, the overlap consists of produced elements and boundary nodes only, and none of these boundary nodes is isolated. Results presented for conflicts above can be formulated and proven for dependencies in an analogous way.

7 Related Work and Conclusion

The critical pair analysis (CPA) has developed into the standard technique for detecting potential conflicts and dependencies in graph transformation systems [3] and more generally, of $\mathcal{M}$-adhesive systems [4, 12]. We introduced the notions of initial conflict and dependency as a new yardstick to present potential conflicts and dependencies in graph transformation systems in a minimal way. These notions are defined in a purely category-theoretical way within the framework of $\mathcal{M}$-adhesive systems. While each initial conflict is a critical pair, it turns out that this is not true vice versa. Actually, our running example shows that, given a rule pair, the set of initial conflicts can be considerably smaller than the set of critical pairs and even than the set of essential critical pairs. We characterized initial conflicts in graph transformation systems as transformation pairs with minimal context and maximal unfolding of preserved graph elements.

The CPA is offered by the graph transformation tools AGG [10] and Verigraph [11] and the graph-based model transformation tool Henshin [15]. All of them provide the user with a set of (essential) critical pairs for each pair of rules as analysis result at design time. Since initial conflicts turned out to be a real subset of essential critical pairs, we intend to optimize the conflict and dependency analysis (CDA) in AGG and Henshin by prioritizing the initial ones. We also intend to investigate how far we can speed up this analysis by our new results. It would be interesting to come up with some results on the amount of reduction of critical pairs, maybe w.r.t. a particular characterization of the rules.

Novel conflict and dependency concepts at several granularity levels are presented in [16]. It is up to future work to investigate the relation of this work with initial conflicts and dependencies. The CPA is not only available for plain rules but also for rules with application conditions (ACs) [17]. Due to their definition in a purely category-theoretical form, we are quite confident that the theory for initial conflicts and dependencies can be extended to rules with ACs.

References

1. Huet, G.: Confluent reductions: Abstract properties and applications to term rewriting systems: Abstract properties and applications to term rewriting systems. Journal of the ACM (JACM) **27**(4) (1980) 797–821
2. Plump, D.: Critical Pairs in Term Graph Rewriting. In: Mathematical Foundations of Computer Science. Volume 841 of LNCS. (1994) 556–566
3. Heckel, R., Küster, J.M., Taentzer, G.: Confluence of Typed Attributed Graph Transformation Systems. In: First Int. Conf. on Graph Transformation (ICGT). Volume 2505 of LNCS., Springer (2002) 161–176
4. Ehrig, H., Habel, A., Padberg, J., Prange, U.: Adhesive high-level replacement categories and systems. In: Int. Conf. on Graph Transformation (ICGT), Springer (2004) 144–160
5. Hausmann, J.H., Heckel, R., Taentzer, G.: Detection of Conflicting Functional Requirements in a Use Case-Driven Approach: A Static Analysis Technique Based on Graph Transformation. In: 22rd Int. Conf. on Software Engineering (ICSE), ACM (2002) 105–115
6. Jayaraman, P., Whittle, J., Elkhodary, A.M., Gomaa, H.: Model composition in product lines and feature interaction detection using critical pair analysis. In: Int. Conf. on Model Driven Engineering Languages and Systems, Springer (2007) 151–165
7. Baresi, L., Ehrig, K., Heckel, R.: Verification of model transformations: A case study with bpel. In: International Symposium on Trustworthy Global Computing, Springer (2006) 183–199
8. Lambers, L.: Certifying rule-based models using graph transformation. PhD thesis, Berlin Institute of Technology (2010)
9. Lambers, L., Ehrig, H., Orejas, F.: Efficient conflict detection in graph transformation systems by essential critical pairs. Electr. Notes Theor. Comput. Sci. **211** (2008) 17–26
10. Taentzer, G.: AGG: A graph transformation environment for modeling and validation of software. In: Int. Workshop on Applications of Graph Transformations with Industrial Relevance, Springer (2003) 446–453
11. Verigraph: Verigraph. https://github.com/Verites/verigraph
12. Ehrig, H., Golas, U., Hermann, F.: Categorical frameworks for graph transformation and HLR systems based on the DPO approach. Bulletin of the EATCS **102** (2010) 111–121
13. Ehrig, H., Ehrig, K., Prange, U., Taentzer, G.: Fundamentals of Algebraic Graph Transformation. Monographs in Theoretical Computer Science. Springer (2006)
14. Corradini, A., Montanari, U., Rossi, F., Ehrig, H., Heckel, R., Löwe, M.: Algebraic Approaches to Graph Transformation I: Basic Concepts and Double Pushout Approach. In Rozenberg, G., ed.: Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1: Foundations. World Scientific (1997) 163–245
15. Arendt, T., Biermann, E., Jurack, S., Krause, C., Taentzer, G.: Henshin: Advanced Concepts and Tools for In-Place EMF Model Transformations. In: Model Driven Engineering Languages and Systems. Volume 6394 of LNCS. (2010) 121–135 http://www.eclipse .org/henshin/.
16. Born, K., Lambers, L., Strüber, D., Taentzer, G.: Granularity of conflicts and dependencies in graph transformation systems. In Plump, D., de Lara, J., eds.: Int. Conf. on Graph Transformation (ICGT), Springer (2017)
17. Ehrig, H., Golas, U., Habel, A., Lambers, L., Orejas, F.: $\mathcal{M}$-adhesive transformation systems with nested application conditions. part 2: Embedding, critical pairs and local confluence. Fundam. Inform. **118**(1-2) (2012) 35–63