Initial Conflicts for Transformation Rules with Nested Application Conditions

Leen Lambers[1](✉) and Fernando Orejas[2]

[1] Hasso Plattner Institute, University of Potsdam, Potsdam, Germany
leen.lambers@hpi.de
[2] Universitat Politècnica de Catalunya, Barcelona, Spain
orejas@lsi.upc.edu

Abstract. We extend the theory of initial conflicts in the framework of $\mathcal{M}$-adhesive categories to transformation rules with ACs. We first show that for rules with ACs, conflicts are in general neither inherited from a bigger context any more, nor is it possible to find a finite and complete subset of finite conflicts as illustrated for the category of graphs. We define initial conflicts to be special so-called symbolic transformation pairs, and show that they are minimally complete (and in the case of graphs also finite) in this symbolic way. We show that initial conflicts represent a proper subset of critical pairs again. We moreover demonstrate that (analogous to the case of rules without ACs) for each conflict a unique initial conflict exists representing it. We conclude with presenting a sufficient condition illustrating important special cases for rules with ACs, where we do not only have initial conflicts being complete in a symbolic way, but also find complete (and in the case of graphs also finite) subsets of conflicts in the classical sense.

Keywords: Graph transformation · Critical pairs · Conflicts

1 Introduction

Detecting and analyzing conflicts is an important issue in software analysis and design, which has been addressed successfully using powerful techniques from graph transformation (see, e.g., [12,15,17,24]), most of them based on critical pair analysis. The *power of critical pairs* is a consequence of the fact that: a) they are complete, in the sense that they represent all conflicts; b) there is a finite number of them; and c) they can be computed statically. The main problem is that their computation has exponential complexity in the size of the preconditions of the rules. For this reason, some significantly smaller subsets of critical pairs that are still complete have been defined [1,19,21], clearing the way for a more efficient computation. In particular, recently, in [19], a new approach

F. Orejas has been supported by the Salvador de Madariaga grant PRX18/00308 and by funds from the Spanish Research Agency (AEI) and the European Union (FEDER funds) under grant GRAMM (ref. TIN2017-86727-C2-1-R).

F. Gadducci and T. Kehrer (Eds.): ICGT 2020, LNCS 12150, pp. 109–127, 2020.
https://doi.org/10.1007/978-3-030-51372-6_7

Table 1. Critical pairs versus initial conflicts

	Plain rules	Rules with NACs	Rules with ACs
Critical pairs (CPs)	Subset of conflicts, complete [27]	Subset of conflicts, complete [17,20]	Symbolic, complete [6,9]
Initial conflicts	Subset of conflicts, min. complete, proper subset of CPs [1,19]	Symbolic (Definition 10), min. complete, regular (Theorem 6) & conservative extension of CPs (Theorem 7)	Symbolic (Definition 10), min. complete, proper subset of CPs (Theorem 3)

for conflict detection was introduced based on a different intuition. Instead of considering conflicts in a minimal context, as for critical pairs, we used the notion of initiality to characterize a complete set of minimal conflicts, showing that *initial conflicts* form a strict subset of critical pairs. In particular, we have that every conflict is represented by a unique initial conflict, as opposed to the fact that each conflict may be represented by many critical pairs.

Most of the work on critical pairs only applies to *plain* graph transformation systems, i.e. transformation systems with unconditional rules. Nevertheless, in practice, we often need to limit the application of rules, defining some kind of *application conditions* (ACs). In this sense, in [17,20] we defined critical pairs for rules with negative application conditions (NACs), and in [6,9] for the general case of ACs, where conditions are as expressive as arbitrary first-order formulas on graphs. However, to our knowledge, no work has addressed up to now the problem of finding significantly smaller subsets of critical pairs for this kind of rules. In this paper we generalize the theory of initial conflicts to rules with ACs in the framework of $\mathcal{M}$-adhesive transformation systems. In particular, the main contributions of this paper (as summarized in Table 1) are:

- The definition of the *notion of initial conflict for rules with ACs*, based on a notion of *symbolic transformation pair*, showing that the set of initial conflicts is a *proper subset* of the set of critical pairs and that it is *minimally complete*[1], in the sense that, no smaller set of symbolic transformation pairs exists that is also complete. In particular, the *cardinality* of the set of initial conflicts is, at most, the cardinality of the set of initial conflicts for the plain case, when disregarding the ACs, plus one. Moreover, as in the plain case, every conflict is an instance of a *unique* initial conflict.
- The identification of a class of *regular* initial conflicts that demonstrate a certain kind of regularity in their application conditions. This allows us to unfold them into a *complete (and in the case of graphs also finite) subset of conflicts*. In particular, we show that, in the case of rules with NACs, initial conflicts are regular, implying that our initial conflicts represent a *conservative extension* of the critical pair theory for rules with NACs.

The paper is organized as follows. We describe *related work* in Sect. 2 and, in Sect. 3, we present some *preliminary material*, where we also include some new

[1] Provided that the considered category has initial conflicts for the plain case.

results. More precisely, in Subsect. 3.1 and Subsect. 3.2 we briefly reintroduce the framework of $\mathcal{M}$-adhesive categories and of rules with ACs; in Subsect. 3.3 we reintroduce critical pairs for rules with ACs following [6,9]; in Subsect. 3.4 we reintroduce initial conflicts for plain rules, and in Sect. 4 we introduce initial parallel independent transformation pairs. This result is used in Sect. 4, where we present the main results of the paper about *initial conflicts for rules with ACs*. Then, in Sect. 5 we show our results on *unfolding initial conflicts*. Finally, we conclude in Sect. 6 discussing some future work. Detailed proofs can be found in the full version of the paper [23].

2 Related Work

Most work on checking *confluence* for rule-based rewriting systems is based on the seminal paper from Knuth and Bendix [14], who reduced the problem of checking local confluence to checking the joinability of a finite set of *critical pairs*. This technique has been extensively studied in the area of term rewriting systems (see, e.g., [25]), and it was introduced in the area of *graph transformation* by Plump [27] in the context of term-graph and hypergraph rewriting. Moreover, he also proved that (local) confluence of graph transformation systems is undecidable, even for terminating systems, as opposed to what happens in the area of term rewriting systems. However, recently, in [2] it is shown that confluence of terminating DPO transformation of graphs with interfaces is decidable. The authors explain that the reason is that interfaces play the same role as variables in term rewriting systems, where confluence is undecidable for terminating ground (i.e., without variables) systems, but decidable for non-ground ones.

Computing critical pairs in graph transformation, as introduced by Plump [27], is exponential in the size of the preconditions of the rules. For this reason, different *proper subsets of critical pairs* with a considerably reduced size were studied that are still complete [1,19,21], clearing the way for a more efficient computation. The notion of *essential critical pair* [21] for graph transformation systems already allowed for a significant reduction, and, the notion of *initial conflict* [19], introduced for the more general $\mathcal{M}$-adhesive systems, allowed for an even larger reduction. However, not all $\mathcal{M}$-adhesive categories have initial conflicts. In [19] it is shown that typed graphs do have them and [1] extended that result proving that arbitrary $\mathcal{M}$-adhesive categories satisfying some given conditions also have initial conflicts.

A recent line of work concentrates on the development of *multi-granular conflict detection techniques* [3,18,24]. An extensive literature survey shows [24] that conflict detection is used at different levels of granularity depending on its application field. The overview shows that conflict detection can be used for the analysis and design phase of software systems (e.g. for finding inconsistencies in requirement specifications), for model-driven engineering (e.g. supporting model version management), for testing (e.g. generation of interesting test cases), or for optimizing rule-based computations (e.g. avoiding backtracking). These multi-granular techniques are presented for rules without application conditions (ACs).

Our work builds further foundations for providing multi-granular techniques also in the case of rules with ACs in the future.

The use of (negative) *application conditions*, to limit the application of graph transformation rules, was introduced in [8,10], while the more general approach, using *nested conditions*, was introduced by Habel and Penneman [11]. Checking confluence for graph transformation systems with application conditions (ACs) has been studied in [17,20] for the case of negative application conditions (NACs), and in [6,9] for the more general case of ACs. However, it is an open issue to find proper subsets of critical pairs of considerably reduced size in the general case.

3 Preliminaries

We start with a brief introduction of $\mathcal{M}$-adhesive categories, *rules with nested application conditions (ACs)* (cf. Subsect. 3.2), and the main parts of *critical pair theory* for this type of rules [6,9] (cf. Subsect. 3.3). Thereafter, we reintroduce the notion of *initial conflicts* [19] for *plain* rules, i.e. rules without ACs (cf. Subsect. 3.4). We also introduce the notion of *initial parallel independent transformation pairs* as a counterpart (cf. Subsect. 3.5), since it will play a particular role when defining initial conflicts for rules with ACs in Subsect. 3.4. We assume that the reader is acquainted with the basic theory of DPO graph transformation and, in particular, the standard definitions of typed graphs and typed graph morphisms (see, e.g., [5]) and its associated category, $\mathbf{Graphs}_{TG}$.

3.1 Graphs and High-Level Structures

Our results do not only apply to a specific class of graph transformation systems, like standard (typed) graph transformation systems, but to systems over any $\mathcal{M}$-adhesive category [5]. The idea behind considering $\mathcal{M}$-adhesive categories is to avoid similar investigations for different instantiations like e.g. Petri nets or hypergraphs. An $\mathcal{M}$-adhesive category is a category $\mathcal{C}$ with a distinguished morphism 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. Moreover, additional properties are needed in our context: initial pushouts, describing the existence of a special "smallest" pushout over a morphism, $\mathcal{E}'$-$\mathcal{M}$ pair factorizations, extending the classical epi-mono factorization to a pair of morphisms with the same codomain. The definitions of these properties can be found in [6,7].

Assumption 1. *We assume that $\langle \mathcal{C}, \mathcal{M} \rangle$ is an $\mathcal{M}$-adhesive category with a unique $\mathcal{E}'$-$\mathcal{M}$ pair factorization (needed for Lemma 1, Definition 5, Theorem 3, Corollary 1) and binary coproducts (needed for Lemma 3, Definition 8, Theorem 1). For the Local Confluence Theorem for initial conflicts of rules with ACs we in addition need initial pushouts (cf. Subsect. 4.4).*

Remark 1. Most categories of structures used for specification are $\mathcal{M}$-adhesive and satisfy these additional properties [5], including the category $\langle \mathrm{Graphs}_{TG}, \mathcal{M} \rangle$ with $\mathcal{M}$ being the class of all injective typed graph morphisms.

3.2 Rules with Application Conditions and Parallel Independence

Nested application conditions [11] (in short, application conditions, or just ACs) generalize the corresponding notions in [4,10,15], where a negative (positive) application condition, short NAC (PAC), over a graph P, denoted $\neg\exists a$ ($\exists a$) is defined in terms of a morphism $a : P \to C$. Informally, a morphism $m : P \to G$ satisfies $\neg\exists a$ ($\exists a$) if there does not exist a morphism $q : C \to G$ extending a to m (if there exists q extending a to m). Then, an AC is either the special condition true or a pair of the form $\exists(a, \mathrm{ac}_C)$ or $\neg\exists(a, \mathrm{ac}_C)$, where ac_C is an additional AC on C. Intuitively, a morphism $m : P \to G$ satisfies $\exists(a, \mathrm{ac}_C)$ if m satisfies a and the corresponding extension q satisfies ac_C. Moreover, ACs may be combined with the usual logical connectors. For a concrete definition of ACs we address the reader to [11] or [6].

ACs are used to restrict the application of rules to a given object. The idea is to equip the precondition (or left hand side) of rules with an application condition. Then we can only apply a given rule to an object G if the corresponding match morphism satisfies the AC of the rule. However, for technical reasons, we also introduce the application of rules *disregarding* the associated ACs.

Definition 1 (rules and transformations). *A* rule $\rho = \langle p, \mathrm{ac}_L\rangle$ *consists of a* plain *rule* $p = \langle L \hookleftarrow I \to R\rangle$ *with* $I \hookrightarrow L$ *and* $I \hookrightarrow R$ *morphisms in* $\mathcal{M}$ *and an application condition* ac_L *over* L.

$$\begin{tikzcd}
\mathrm{ac}_L \arrow[r, phantom, "\blacktriangleright"] \arrow[rd, phantom, "\models"] & L \arrow[d, "m"'] \arrow[rd, phantom, "(1)"] & I \arrow[l, hook'] \arrow[r, hook] \arrow[d] \arrow[rd, phantom, "(2)"] & R \arrow[d, "m^*"] \\
 & G & D \arrow[l, hook'] \arrow[r, hook] & H
\end{tikzcd}$$

A direct transformation $t : G \Rightarrow_{\rho,m,m^*} H$ *consists of two pushouts (1) and (2), called DPO, with match* m *and comatch* m^* *such that* $m \models \mathrm{ac}_L$. $G \hookleftarrow D \hookrightarrow H$ *is called the* derived span *of* t. *An* AC-disregarding direct transformation $G \Rightarrow_{\rho,m,m^*} H$ *consists of DPO (1) and (2), where* m *does not necessarily need to satisfy* ac_L. *Given a set of rules* $\mathcal{R}$ *for* $\langle\mathcal{C}, \mathcal{M}\rangle$, *the triple* $\langle\mathcal{C}, \mathcal{M}, \mathcal{R}\rangle$ *is an* $\mathcal{M}$-adhesive system*.*

Remark 2. In the rest of the paper we assume that each rule (resp. transformation or $\mathcal{M}$-adhesive system) comes with ACs. Otherwise, we state that we have a *plain* rule (resp. transformation or $\mathcal{M}$-adhesive system), which can be seen as a special case, in the sense that the ACs are (equivalent to) *true*.

ACs can be shifted over morphisms and rules as shown in the following lemma (for constructions see [7][2] and [7,11], respectively).

[2] Since this construction entails the enumeration of jointly epimorphic morphism pairs, its computation has exponential complexity in the size of the precondition of the rule and the size of the AC.

Lemma 1 (shift ACs over morphisms [7]). *There is a transformation* Shift *from morphisms and ACs to ACs such that for each AC,* ac_P*, and each morphism* $b\colon P \to P'$*,* Shift *transforms* ac_P *via* b *into an AC* $\mathrm{Shift}(b, \mathrm{ac}_P)$ *over* P' *such that for each morphism* $n\colon P' \to H$ *it holds that* $n \circ b \models \mathrm{ac}_P \Leftrightarrow n \models \mathrm{Shift}(b, \mathrm{ac}_P)$*.*

Lemma 2 (shift ACs over rules [7,11]). *There is a transformation* L *from rules and ACs to ACs such that for every rule* $\rho : L \hookleftarrow I \hookrightarrow R$ *and every AC on* R*,* ac_R*,* L *transforms* ac_R *via* ρ *into the AC* $\mathrm{L}(\rho, \mathrm{ac}_R)$ *on* L*, such that for every direct transformation* $G \Rightarrow_{\rho,m,m^*} H$*,* $m \models \mathrm{L}(\rho, \mathrm{ac}_R) \Leftrightarrow m^* \models \mathrm{ac}_R$*.*

For *parallel independence*, when working with rules with ACs, we need not only that each rule does not delete any element which is part of the match of the other rule, but also that the resulting transformation defined by each rule application still satisfies the ACs of the other rule application.

Definition 2 (transformation pairs and parallel independence). *A* transformation pair $H_1 \Leftarrow_{\rho_1,o_1} G \Rightarrow_{\rho_2,o_2} H_2$ *is* parallel independent *if there exists a morphism* $d_{12}\colon L_1 \to D_2$ *such that* $k_2 \circ d_{12} = o_1$ *and* $c_2 \circ d_{12} \models \mathrm{ac}_{L_1}$ *and there exists a morphism* $d_{21}\colon L_2 \to D_1$ *such that* $k_1 \circ d_{21} = o_2$ *and* $c_1 \circ d_{21} \models \mathrm{ac}_{L_2}$*.*

$$\begin{tikzcd}
R_1 \arrow[d] & I_1 \arrow[l, hook'] \arrow[r, hook] \arrow[d] & L_1 \overset{\mathrm{ac}_{L_1}}{\triangleleft} \arrow[dr, "o_1"'] \arrow[drrr, dotted, "d_{12}"] & & \overset{\mathrm{ac}_{L_2}}{\triangleright} L_2 \arrow[dl, "o_2"] \arrow[dlll, dotted, "d_{21}"'] & I_2 \arrow[l, hook'] \arrow[r, hook] \arrow[d] & R_2 \arrow[d] \\
H_1 & D_1 \arrow[l, hook', "c_1"] \arrow[rr, hook, "k_1"'] & & G & & D_2 \arrow[ll, hook', "k_2"] \arrow[r, hook, "c_2"'] & H_2
\end{tikzcd}$$

A transformation pair is *in conflict* or *conflicting* if it is parallel dependent. We distinguish different conflict types, generalizing straightforwardly the conflict characterization introduced for rules with NACs [20]. The transformation pair $H_1 \Leftarrow_{\rho_1,o_1} G \Rightarrow_{\rho_2,o_2} H_2$ is a *use-delete* (resp. *delete-use*) conflict if in Definition 2 the commuting morphism d_{12} (resp. d_{21}) does not exist, i.e. the second (resp. first) rule deletes something used by the first (resp. second) one. Moreover, it is an *AC-produce* (resp. *produce-AC*) conflict if in Definition 2 the commuting morphism d_{12} (resp. d_{21}) exists, but an extended match is produced by the second (resp. first) rule that does not satisfy the rule AC of the first (resp. second) rule. If a transformation pair is an *AC-produce* or *produce-AC* conflict, then we also say that it is an *AC conflict* or *AC conflicting.*

Remark 3 (use-delete XOR AC-produce). A use-delete (resp. delete-use) conflict cannot occur simultaneously to an AC-produce (resp. produce-AC) conflict, since the AC of the first (resp. second) rule can only be violated iff there exists an extended match for the first (resp. second) rule. However, a use-delete (resp. delete-use) conflict may occur simultaneously to a produce-AC (resp. AC-produce) conflict.

For grasping the notion of completeness of transformation pairs w.r.t. a property like parallel (in-)dependence, it is first important to understand how a given transformation can be extended to another transformation. In particular, an *extension diagram* describes how a transformation $t\colon G_0 \Rightarrow^* G_n$ can be extended to a transformation $t'\colon G'_0 \Rightarrow^* G'_n$ via the same rules and an *extension morphism* $k_0\colon G_0 \rightarrow G'_0$ that maps G_0 to G'_0 as shown in the following diagram on the left. For each rule application and transformation step, we have two double pushout diagrams as shown on the right, where the rule ρ_{i+1} is applied to G_i and G'_i.

$$\begin{tikzcd}
G_0 \arrow[r, Rightarrow, "*"] \arrow[d, "k_0"'] \arrow[dr, phantom, "(1)"] & G_n \arrow[d, "k_n"] \\
G'_0 \arrow[r, Rightarrow, "*"] & G'_n
\end{tikzcd}
\qquad
\begin{tikzcd}
L_{i+1} \arrow[d] & I_{i+1} \arrow[l] \arrow[r] \arrow[d] & R_{i+1} \arrow[d] \\
G_i \arrow[d] & D_i \arrow[l] \arrow[r] \arrow[d] & G_{i+1} \arrow[d] \\
G'_i & D'_i \arrow[l] \arrow[r] & G'_{i+1}
\end{tikzcd}$$

We introduce two notions of completeness, distinguishing $\mathcal{M}$-completeness from regular completeness, depending on the membership of the extension morphism in $\mathcal{M}$. It is known that critical pairs (resp. initial conflicts) for *plain rules* are $\mathcal{M}$-complete (resp. complete) w.r.t. parallel dependence [5,19]. In Subsect. 3.3, we reintroduce the fact that critical pairs for rules with ACs are $\mathcal{M}$-complete w.r.t. parallel dependence, but as symbolic transformation pairs. We learn in Sect. 4 that initial conflicts for rules with ACs are also complete in this symbolic way.

Definition 3 (($\mathcal{M}$-)completeness of transformation pairs). *A set of* transformation pairs $\mathcal{S}$ *for a pair of rules* $\langle \rho_1, \rho_2 \rangle$ *is* complete *(resp.* $\mathcal{M}$-complete*) w.r.t. parallel (in-)dependence if there is a pair* $P_1 \Leftarrow_{\rho_1,o_1} K \Rightarrow_{\rho_2,o_2} P_2$ *from* $\mathcal{S}$ *and an extension diagram via extension morphism* m *(resp.* $m \in \mathcal{M}$*) for each parallel (in-)dependent direct transformation pair* $H_1 \Leftarrow_{\rho_1,m_1} G \Rightarrow_{\rho_2,m_2} H_2$.

$$\begin{tikzcd}
P_1 \arrow[d] & K \arrow[l, Rightarrow, "{\rho_1, o_1}"'] \arrow[r, Rightarrow, "{\rho_2, o_2}"] \arrow[d, "m"] & P_2 \arrow[d] \\
H_1 & G \arrow[l, Rightarrow, "{\rho_1, m_1}"] \arrow[r, Rightarrow, "{\rho_2, m_2}"'] & H_2
\end{tikzcd}$$

Fig. 1. ($\mathcal{M}$-)completeness of transformation pairs

3.3 Critical Pairs

Critical pairs for plain rules are just transformation pairs, where morphisms o_1 and o_2 are in $\mathcal{E}'$ (i.e., roughly, K is an overlapping of L_1 and L_2). In the category of **Graphs** they lead to finite and complete subsets of finite conflicts [4] (assumed that the rule graphs are also finite). However, when rules include ACs, we cannot use the same notion of critical pair since, as we show in Theorem 2, in general, for any two rules with ACs, there is no complete set of transformation pairs that is finite. To avoid this problem, our critical pairs for rules with ACs also include ACs, as in [6,9], where they are proved to be $\mathcal{M}$-*complete*, and they are also finite in the category of **Graphs** (assumed again that the rules are finite).

In particular, critical pairs are based on the notion of *symbolic transformation pairs*, which are pairs of *AC-disregarding transformations* on some object K with two special ACs on K. These two ACs, ac_K (*extension AC*) and ac_K^* (*conflict-inducing AC*), are used to characterize which embeddings of this pair, via some morphism $m : K \to G$, give rise to a transformation pair that is parallel dependent. If $m \models \mathrm{ac}_K$, then $m \circ o_1 : L_1 \to G$ and $m \circ o_2 : L_2 \to G$ are two morphisms, satisfying the associated ACs of ρ_1 and ρ_2, respectively. Moreover, if $m \models \mathrm{ac}_K^*$, then the two transformations $H_1 \Leftarrow_{\rho_1, m\circ o_1} G \Rightarrow_{\rho_2, m\circ o_2} H_2$ are parallel dependent. Symbolic transformation pairs allow us to present critical pairs as well as initial conflicts (cf. Subsect. 3.4) in a compact and unified way, since they both are instances of symbolic transformation pairs. Finally, note that each symbolic transformation pair $stp_K : \langle tp_K, ac_K, ac_K^* \rangle$ is by definition uniquely determined (up to isomorphism and equivalence of the extension AC and conflict-inducing AC) by its underlying AC-disregarding transformation pair.

Definition 4 (symbolic transformation pair). *Given rules* $\rho_1 = \langle p_1, \mathrm{ac}_{L_1} \rangle$ *and* $\rho_2 = \langle p_2, \mathrm{ac}_{L_2} \rangle$*, a* symbolic transformation pair $stp_K : \langle tp_K, ac_K, ac_K^* \rangle$ *for* $\langle \rho_1, \rho_2 \rangle$ *consists of a pair* $tp_K : P_1 \Leftarrow_{\rho_1, o_1} K \Rightarrow_{\rho_2, o_2} P_2$ *of AC-disregarding transformations together with ACs* ac_K *and* ac_K^* *on* K *given by:*

$$\mathrm{ac}_K = \mathrm{Shift}(o_1, \mathrm{ac}_{L_1}) \wedge \mathrm{Shift}(o_2, \mathrm{ac}_{L_2}),\ \textit{called}\ \text{extension AC},\ \textit{and}$$
$$\mathrm{ac}_K^* = \neg(\mathrm{ac}_{K,d_{12}}^* \wedge \mathrm{ac}_{K,d_{21}}^*),\ \textit{called}\ \text{conflict-inducing AC}$$

with $\mathrm{ac}_{K,d_{12}}^*$ *and* $\mathrm{ac}_{K,d_{21}}^*$ *given as follows:*

$$\begin{aligned}
\textit{if}\,(\exists\ d_{12}\ \textit{with}\ k_2 \circ d_{12} = o_1)\ \textit{then}\ \mathrm{ac}_{K,d_{12}}^* &= \mathrm{L}(p_2^*, \mathrm{Shift}(c_2 \circ d_{12}, \mathrm{ac}_{L_1}))\\
\textit{else}\ \mathrm{ac}_{K,d_{12}}^* &= \text{false}\\
\textit{if}\,(\exists\ d_{21}\ \textit{with}\ k_1 \circ d_{21} = o_2)\ \textit{then}\ \mathrm{ac}_{K,d_{21}}^* &= \mathrm{L}(p_1^*, \mathrm{Shift}(c_1 \circ d_{21}, \mathrm{ac}_{L_2}))\\
\textit{else}\ \mathrm{ac}_{K,d_{21}}^* &= \text{false}
\end{aligned}$$

where $p_1^* = \langle K \overset{k_1}{\hookleftarrow} D_1 \overset{c_1}{\hookrightarrow} P_1 \rangle$ *and* $p_2^* = \langle K \overset{k_2}{\hookleftarrow} D_2 \overset{c_2}{\hookrightarrow} P_2 \rangle$ *are defined by the corresponding double pushouts.*

$$\begin{tikzcd}
 & & & \mathrm{ac}_{L_1} & & \mathrm{ac}_{L_2} & & & \\
p_1 : & R_1 \arrow[d] & I_1 \arrow[l, hook'] \arrow[r, hook] \arrow[d] & L_1 \triangleleft \arrow[dr, "o_1"] \arrow[dddrrr, phantom] \arrow[drrr, dotted, "d_{12}"] & & \triangleright L_2 \arrow[dl, "o_2"'] \arrow[dlll, dotted, "d_{21}"'] & I_2 \arrow[l, hook'] \arrow[r, hook] \arrow[d] & R_2 \arrow[d] & : p_2\\
p_1^* : & P_1 & D_1 \arrow[l, hook', "c_1"] \arrow[rr, hook, "k_1"'] & & K & & D_2 \arrow[ll, hook', "k_2"] \arrow[r, hook, "c_2"'] & P_2 & : p_2^*
\end{tikzcd}$$

A *critical pair* is now a symbolic transformation pair in a minimal context such that there exists at least one extension to a pair of transformations being parallel dependent.

Definition 5 (critical pair). *Given rules* $\rho_1 = \langle p_1, \mathrm{ac}_{L_1}\rangle$ *and* $\rho_2 = \langle p_2, \mathrm{ac}_{L_2}\rangle$*, a* critical pair *for* $\langle\rho_1, \rho_2\rangle$ *is a symbolic transformation pair* $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K\rangle$*, where the match pair* (o_1, o_2) *of* tp_K *is in* $\mathcal{E}'$*, and there exists a morphism* $m\colon K \to G \in \mathcal{M}$ *such that* $m \models \mathrm{ac}_K \wedge \mathrm{ac}^*_K$ *and* $m_i = m \circ o_i$*, for* $i = 1, 2$*, satisfy the gluing conditions, i.e.* m_i *has a pushout complement w.r.t.* p_i*.*

Definition 6 (($\mathcal{M}$-)completeness of symbolic transformation pairs). *A set of symbolic transformation pairs* $\mathcal{S}$ *for a pair of rules* $\langle\rho_1, \rho_2\rangle$ *is* complete *(resp.* $\mathcal{M}$-complete*) w.r.t. parallel dependence if there is a symbolic transformation pair* $stp_K : \langle tp_K : P_1 \Leftarrow_{\rho_1,o_1} K \Rightarrow_{\rho_2,o_2} P_2, \mathrm{ac}_K, \mathrm{ac}^*_K\rangle$ *from* $\mathcal{S}$ *and an extension diagram as depicted in Fig.* 1 *with* $m : K \to G$ *(resp.* $m : K \to G \in \mathcal{M}$*) and* $m \models \mathrm{ac}_K \wedge \mathrm{ac}^*_K$ *for each parallel dependent direct transformation* $H_1 \Leftarrow_{\rho_1,m_1} G \Rightarrow_{\rho_2,m_2} H_2$*.*

In [6,9] it is shown that the set of *critical pairs* for a pair of rules is $\mathcal{M}$-complete w.r.t. parallel dependence.

3.4 Initial Conflicts for Plain Rules

Initial conflicts for plain rules follow an alternative approach to the original idea of critical pairs. Instead of considering all conflicting transformations in a minimal context (materialized by a pair of jointly epimorphic matches), initial conflicts use the notion of *initiality of transformation pairs* to obtain a more declarative view on the minimal context of critical pairs. Each initial conflict is a critical pair but not the other way round. Moreover, all initial conflicts for plain rules are complete w.r.t. parallel dependence and they still satisfy the Local Confluence Theorem for plain rules. Consequently, initial conflicts for plain rules represent an important, proper subset of critical pairs for performing static conflict detection as well as local confluence analysis.

Definition 7 (initial transformation pair and initial conflict). *Given a pair of plain direct transformations* $tp : H_1 \Leftarrow_{p_1,m_1} G \Rightarrow_{p_2,m_2} H_2$*, then* $tp^I : H^I_1 \Leftarrow_{p_1,m^I_1} G^I \Rightarrow_{p_2,m^I_2} H^I_2$ *is an* initial transformation pair *for tp if it can be embedded into tp via extension diagrams (1) and (2) and extension morphism* f^I*, as in the left diagram below, such that for each transformation pair* $tp' : H'_1 \Leftarrow_{p_1,m'_1} G' \Rightarrow_{p_2,m'_2} H'_2$ *that can be embedded into tp via extension diagrams (3) and (4) and extension morphism* f*, as in the left diagram below, it holds that* tp^I *can be embedded into* tp' *via unique extension diagrams (5) and (6) and unique vertical morphism* f'^I *s.t.* $f \circ f'^I = f^I$*.*

Given a plain $\mathcal{M}$*-adhesive system with initial transformation pairs for conflicts, an* initial conflict *is a conflict tp isomorphic to* tp^I*.*

$$\begin{tikzcd}
H_1^I \arrow[d, "g_1^I"'] \arrow[dr, phantom, "(1)"] & G^I \arrow[l, Leftarrow, "{p_1,m_1^I}"'] \arrow[r, Rightarrow, "{p_2,m_2^I}"] \arrow[d, "f^I"] \arrow[dr, phantom, "(2)"] & H_2^I \arrow[d, "g_2^I"] \\
H_1 & G \arrow[l, Leftarrow, "{p_1,m_1}"'] \arrow[r, Rightarrow, "{p_2,m_1}"] & H_2
\end{tikzcd}
\qquad
\begin{tikzcd}
H_1^I \arrow[d, "g_1'^I"'] \arrow[dr, phantom, "(5)"] & G^I \arrow[l, Leftarrow, "{p_1,m_1^I}"'] \arrow[r, Rightarrow, "{p_2,m_2^I}"] \arrow[d, "f'^I"] \arrow[dr, phantom, "(6)"] & H_2^I \arrow[d, "g_2'^I"] \\
H_1' \arrow[d, "g_1"'] \arrow[dr, phantom, "(3)"] & G' \arrow[l, Leftarrow, "{p_1,m'_1}"'] \arrow[r, Rightarrow, "{p_2,m'_2}"] \arrow[d, "f"] \arrow[dr, phantom, "(4)"] & H_2' \arrow[d, "g_2"] \\
H_1 & G \arrow[l, Leftarrow, "{p_1,m_1}"'] \arrow[r, Rightarrow, "{p_2,m_2}"] & H_2
\end{tikzcd}$$

The idea of representing all conflicts by a (finite) set of initial conflicts is based on the requirement of the *existence of initial transformation pairs* for parallel dependent or *conflicting* plain transformation pairs. This requirement holds for the category of typed graphs [19] and for any arbitrary $\mathcal{M}$-adhesive category fulfilling some extra conditions [1].

For plain $\mathcal{M}$-adhesive systems, initial conflicts are critical pairs [19]. Moreover, they are complete and minimal as transformation pairs w.r.t. parallel dependence, whereas critical pairs for plain rules are $\mathcal{M}$-complete [4].

3.5 Initial Parallel Independent Transformation Pairs for Plain Rules

In this section, we show the existence of initial transformation pairs for *parallel independent transformation pairs* (Fig. 2), allowing us to define a *complete* subset also w.r.t. parallel independence. The proof requires that binary coproducts exist.

Lemma 3 (existence of initial transformation pair for parallel independent transformation pair). *Given a pair of parallel independent plain direct transformations* $tp : H_1 \Leftarrow_{p_1,m_1} G \Rightarrow_{p_2,m_2} H_2$*, then* $tp_{L_1+L_2} : R_1 + L_2 \Leftarrow_{p_1,i_1} L_1 + L_2 \Rightarrow_{p_2,i_2} L_1 + R_2$*, where* $i_1 : L_1 \to L_1 + L_2$ *and* $i_2 : L_2 \to L_1 + L_2$ *are the coproduct morphisms, is initial for tp.*

$$\begin{tikzcd}
p_1 : & R_1 \arrow[d] & I_1 \arrow[l, hook'] \arrow[r, hook] \arrow[d] & L_1 \overset{\mathrm{ac}_{L_1}}{\triangleleft} \arrow[dr, "i_1"'] \arrow[drrr, dotted, "d_{12}"] & & \overset{\mathrm{ac}_{L_2}}{\triangleright} L_2 \arrow[dl, "i_2"] \arrow[dlll, dotted, "d_{21}"'] & I_2 \arrow[l, hook'] \arrow[r, hook] \arrow[d] & R_2 \arrow[d] & : p_2 \\
p_1^* : & R_1 + L_2 \arrow[d] & I_1 + L_2 \arrow[l, hook', "c_1"] \arrow[rr, hook, "k_1"'] \arrow[d] & & L_1 + L_2 \arrow[d, "m"'] & & L_1 + I_2 \arrow[ll, hook', "k_2"] \arrow[r, hook, "c_2"'] \arrow[d] & L_1 + R_2 \arrow[d] & : p_2^* \\
& H_1 & D_1 \arrow[l, hook'] \arrow[rr, hook] & & G & & D_2 \arrow[ll, hook'] \arrow[r, hook] & H_2 &
\end{tikzcd}$$

Fig. 2. Initial parallel independent transformation pair $tp_{L_1+L_2}$

Uniqueness of initial transformation pairs up to isomorphism implies that for each pair of *plain rules* $\langle p_1, p_2 \rangle$ there is a *unique initial parallel independent transformation pair* $tp_{L_1+L_2} : R_1 + L_2 \Leftarrow_{p_1,i_1} L_1 + L_2 \Rightarrow_{p_2,i_2} L_1 + R_2$.

Definition 8 (initial parallel independent transformation pair). *A pair of parallel independent plain transformations* $tp : H_1 \Leftarrow_{p_1,m_1} G \Rightarrow_{p_2,m_2} H_2$ *is an* initial parallel independent transformation pair *if it is isomorphic to the transformation pair* $tp_{L_1+L_2} : R_1 + L_2 \Leftarrow_{p_1,i_1} L_1 + L_2 \Rightarrow_{p_2,i_2} L_1 + R_2$.

The one-element set consisting of the initial parallel independent transformation pair for a given pair of rules is *complete w.r.t. parallel independence.*

Theorem 1 (completeness of initial parallel independent transformation pairs). *The set consisting of the initial parallel independent transformation pair* $tp_{L_1+L_2} : R_1 + L_2 \Leftarrow_{p_1,i_1} L_1 + L_2 \Rightarrow_{p_2,i_2} L_1 + R_2$ *for a pair of plain rules* $\langle p_1, p_2 \rangle$ *is* complete *w.r.t. parallel independence.*

4 Initial Conflicts

We start with showing why it is not possible to straightforwardly generalize the idea of initial conflicts from plain rules to rules with ACs. On the one hand, *conflict inheritance*, which was the basis for showing completeness of initial conflicts for plain rules, does not hold any more. Moreover, it is *impossible* in general to find a *finite and complete subset of finite conflicts* for rules with ACs (cf. Subsect. 4.2). This motivates again the need for having *symbolic transformation pairs* as introduced in Definition 4, allowing us to define *initial conflicts* (cf. Subsect. 4.3) as a set of specific symbolic transformation pairs, being complete w.r.t. parallel dependence indeed (as shown in Subsect. 4.4).

4.1 Conflict Inheritance

Conflicts are in general not inherited (as opposed to the case of plain rules [19]), i.e., not each (initial) transformation pair that can be embedded into a conflicting one will be conflicting again. This may happen in particular for AC conflicts. Use-delete (resp. delete-use) conflicts for rules with ACs are still inherited.

Lemma 4 (Use-delete (delete-use) conflict inheritance). *Given a pair of direct transformations tp in use-delete (resp. delete-use) conflict and another pair of direct transformations tp′ that can be embedded into tp via extension morphism f and corresponding extension diagrams, then tp′ is also in use-delete (resp. delete-use) conflict.*

Example 1 (No inheritance for AC conflicts). Consider rules $p_1 : \bigcirc \leftarrow \bigcirc \rightarrow \bigcirc\!\!\rightarrow\!\!\bigcirc$ (with AC *true*), producing an outgoing edge with a node, and $p_2 : \bigcirc \leftarrow \bigcirc \rightarrow \bigcirc\,\bigcirc$ with NAC $\neg\exists n : \bigcirc \rightarrow \bigcirc\,\bigcirc\,\bigcirc$, producing a node only if two other nodes do not exist already. Consider graph $G = \bigcirc\,\bigcirc$, holding two nodes. Applying both rules to G (with the matches sharing one node in G) we obtain a produce-AC conflict since the first rule creates a third node, forbidden by the second rule. Now both rules can be applied similarly to the shared node in the subgraph $G' = \bigcirc$ of G obtaining parallel independent transformations, illustrating that AC-conflicts are not inherited.

4.2 Complete Subset of Conflicts

We show that in $\mathcal{M}$-adhesive categories, in particular in the category of graphs, it is in general impossible to find a finite and complete subset of conflicts for finite rules with ACs. If it would always exist, we could derive that each first-order formula is equivalent to a finite disjunction of atomic formulas.

Theorem 2. *Given finite rules $\rho_1 = \langle p_1, \mathrm{ac}_{L_1} \rangle$ and $\rho_2 = \langle p_2, \mathrm{ac}_{L_2} \rangle$ for the $\mathcal{M}$-adhesive category* **Graphs***, in general, there is no finite set of finite transformation pairs $\mathcal{S}$ for ρ_1 and ρ_2 that is complete w.r.t. parallel dependence.*

4.3 Initial Conflicts

We generalize the notion of *initial conflicts* for plain rules to rules with ACs. In particular, we introduce them as special symbolic transformation pairs. They are *conflict-inducing* meaning that there needs to exist an unfolding of the symbolic transformation pair into a concrete conflicting transformation pair. Moreover, their AC-disregarding transformation pair needs to be an initial conflict or initial parallel independent transformation pair. We also show the *relationship between initial conflicts and critical pairs* as reintroduced in Subsect. 3.3, demonstrating that initial conflicts represent a proper subset of critical pairs.

Definition 9 (unfolding of symbolic transformation pair). *Given a symbolic transformation pair $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$ for rule pair $\langle \rho_1, \rho_2 \rangle$, then its unfolding $\mathcal{U}(stp_K)$ consists of all transformation pairs $H_1 \Leftarrow_{\rho_1, m_1} G \Rightarrow_{\rho_2, m_2} H_2$ representing the lower row of the extension diagrams via some extension morphism $m : K \to G$ as shown in Fig. 1 (with AC-disregarding transformation pair tp_K in the upper row). Moreover, we say that stp_K is* conflict-inducing *if its unfolding includes a conflicting transformation pair.*

Definition 10 (initial conflict). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules. An* initial conflict *for rules $\rho_1 = \langle p_1, \mathrm{ac}_{L_1} \rangle$ and $\rho_2 = \langle p_2, \mathrm{ac}_{L_2} \rangle$ is a conflict-inducing symbolic transformation pair $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$ with the AC-disregarding transformation pair tp_K being initial, i.e. either tp_K is an initial conflict for rules p_1 and p_2 (in this case stp_K is called a* use-delete/delete-use initial conflict*) or it is the initial parallel independent transformation pair $tp_{L_1+L_2}$ for rules p_1 and p_2 (in this case $stp_K = stp_{L_1+L_2} = \langle tp_{L_1+L_2}, \mathrm{ac}_{L_1+L_2}, \mathrm{ac}^*_{L_1+L_2} \rangle$ is called the* AC initial conflict*).*

Note that the unfolding of a conflict-inducing symbolic transformation pair (and in particular of an AC initial conflict) may also include parallel independent transformation pairs. All conflicts in the unfolding of an AC initial conflict are AC conflicts, and never use-delete/delete-use conflicts (because otherwise we would get a contradiction using Lemma 4).

Example 2 (initial conflict). Consider again the rules from Example 1. Applying both rules to $L_1 + L_2 = \bigcirc\ \bigcirc$ (with disjoint matches) we obtain the AC initial conflict $stp_K = stp_{L_1+L_2} = \langle tp_{L_1+L_2}, \mathrm{ac}_{L_1+L_2}, \mathrm{ac}^*_{L_1+L_2}\rangle$. Thereby $\mathrm{ac}_{L_1+L_2}$ is equivalent to $\neg\exists(\bigcirc_1\ \bigcirc_2 \to \bigcirc_1\ \bigcirc_2\ \bigcirc) \wedge \neg\exists(\bigcirc_1\ \bigcirc_2 \to \bigcirc_{1,2}\ \bigcirc\ \bigcirc)$, expressing that when during extension both nodes are merged, no two additional nodes, otherwise not one additional node should be given. Moreover, $\mathrm{ac}^*_{L_1+L_2}$ is equivalent to $\exists(\bigcirc_1\ \bigcirc_2 \to \bigcirc_{1,2}\ \bigcirc) \vee \exists(\bigcirc_1\ \bigcirc_2 \to \bigcirc_1\ \bigcirc_2)$, expressing that either both nodes are not merged during extension, otherwise one additional node should be present for a conflict to arise. Both transformation pairs (the conflicting one from $G = \bigcirc\ \bigcirc$ as well as the parallel independent one from its subgraph $G' = \bigcirc$, sharing the merged node in their matches) described in Example 1 belong to its unfolding.

Each initial conflict is in particular also a critical pair.

Theorem 3 (initial conflict is critical pair). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules. Each initial conflict $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K\rangle$ is a critical pair.*

The reverse direction of Theorem 3 does not hold. In general, critical pairs $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K\rangle$ where tp_K represents a use-delete/delete-use conflict (but is not initial yet) are represented by the initial conflict $stp_I : \langle tp_I, \mathrm{ac}_I, \mathrm{ac}^*_I\rangle$ with tp_I the unique initial conflict for tp_K as plain transformation pair. Moreover, critical pairs $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K\rangle$ where tp_K is parallel independent as plain transformation pair are represented by one initial conflict $stp_{L_1+L_2} : \langle tp_{L_1+L_2}, \mathrm{ac}_{L_1+L_2}, \mathrm{ac}^*_{L_1+L_2}\rangle$ with $tp_{L_1+L_2}$ the initial parallel independent transformation pair.

Example 3 (initial conflicts: proper subset of critical pairs). Consider again the rules from Example 1 and their application to $G' = \bigcirc$. The symbolic transformation pair $stp_{G'} : \langle tp_{G'}, \mathrm{ac}_{G'}, \mathrm{ac}^*_{G'}\rangle$ is a critical pair, but not an initial conflict.

4.4 Completeness

We show that initial conflicts are complete (not $\mathcal{M}$-complete as in the case of critical pairs) w.r.t. parallel dependence as symbolic transformation pairs.

Theorem 4 (completeness of initial conflicts). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules. The set of* initial conflicts *for a pair of rules $\langle \rho_1, \rho_2\rangle$ is* complete *w.r.t. parallel dependence.*

Remark 4 (uniqueness of initial conflicts). For each conflict a *unique* (up-to-isomorphism) initial conflict exists representing it, since this property is inherited from the one for plain rules [19] and the fact that the initial parallel independent pair of transformations is unique w.r.t. a given rule pair.

Initial conflicts are also minimally complete, i.e. we are able to generalize the corresponding result for plain rules to rules with ACs.

5 Unfoldings of Initial Conflicts

We show a *sufficient condition* for being able to unfold initial conflicts into a *complete set of conflicts* that is *finite* if the set of initial conflicts is finite (cf. Subsect. 5.1). We demonstrate moreover that this sufficient condition is fulfilled for the special case of having merely *NACs* as rule application conditions (cf. Subsect. 5.2). Finally, we show that in this case we obtain in particular specific critical pairs for rules with negative application conditions (NACs) as introduced in [20] again. In this sense we show explicitly that initial conflicts as introduced in this paper represent a conservative extension of the critical pair theory for rules with NACs.

5.1 Finite and Complete Unfolding

We introduce *regular initial conflicts* leading to $\mathcal{M}$-complete subsets of conflicts by unfolding them in some particular way (cf. *disjunctive unfolding* in Definition 11). The idea is that the extension and conflict-inducing AC (ac_K and ac^*_K, respectively) of such a conflict $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$ have a specific form that is amenable to finding $\mathcal{M}$-complete unfoldings. We expect the condition $\mathrm{ac}_K \wedge \mathrm{ac}^*_K$ to consist of a *disjunction of positive literals* (conditions of the form $\exists(a_i : K \to C_i, c_i)$) with a *negative remainder* (i.e. a condition $c_i = \wedge_{j \in J} \neg\exists(b_j : C_i \to C_j, d_j)$). Intuitively, this means that there is a finite number of possibilities to unfold the symbolic conflict into a concrete conflict by adding some specific positive context (expressed by the morphism a_i). The negative remainder c_i ensures that by adding this positive context to the context K of the symbolic transformation pair within the initial conflict, we indeed find a concrete conflict when not extending further at all. Moreover, it expresses under which condition the corresponding concrete representative conflict leads to further conflicts by extension. Finally, the subsets of $\mathcal{M}$-complete conflicts built using the disjunctive unfolding are *finite* if the set of initial conflicts it is derived from is finite.

Definition 11 (regular initial conflict, disjunctive unfolding). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules. Given an initial conflict $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$ for rules $\langle \rho_1, \rho_2 \rangle$, then we say that it is* regular *if $\mathrm{ac}_K \wedge \mathrm{ac}^*_K$ is equivalent to a condition $\vee_{i \in I} \exists(a_i : K \to C_i, c_i)$ with $c_i = \wedge_{j \in J} \neg\exists(b_j : C_i \to C_j, d_j)$ a condition on C_i, b_j*

non-isomorphic and I some non-empty index set. Given a regular initial conflict $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$, then $\mathcal{U}^{\mathcal{D}}(stp_K) = \cup_{i \in I}\{tp_{C_i} : D_{1,i} \Leftarrow_{\rho_1, a_i \circ o_1} C_i \Rightarrow_{\rho_2, a_i \circ o_2} D_{2,i}\}$ is the disjunctive unfolding *of stp_K.*

Remark 5 (disjunctive unfolding). The disjunctive unfolding of a regular conflict is non-empty, but might consist of less elements than literals in the disjunction $\vee_{i \in I} \exists(a_i : K \to C_i, c_i)$: if a morphism a_i does not satisfy the gluing condition of the derived spans, then also every extension morphism starting from there will not satisfy the gluing condition, and we can safely ignore that case from the disjunctive unfolding.

Theorem 5 (finite and complete unfolding). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules. Given a rule pair $\langle \rho_1, \rho_2 \rangle$ with set $\mathcal{S}$ of initial conflicts such that each initial conflict stp in S is regular, then $\cup_{stp \in \mathcal{S}} \mathcal{U}^{\mathcal{D}}(stp)$ is $\mathcal{M}$-complete w.r.t. parallel dependence. Moreover, $\cup_{stp \in \mathcal{S}}\ \mathcal{U}^{\mathcal{D}}(stp)$ is finite if $\mathcal{S}$ is finite.*

It is possible to automatically check if some initial conflict is regular by using dedicated automated reasoning [22] as well as symbolic model generation for ACs [28] as follows. The reasoning mechanism [22] is shown to be refutationally complete ensuring that if the condition $\mathrm{ac}_K \wedge \mathrm{ac}^*_K$ of some initial conflict is unsatisfiable, this will be detected eventually. Moreover, the related symbolic model generation mechanism [28] is able to automatically transform each condition $\mathrm{ac}_K \wedge \mathrm{ac}^*_K$ into some disjunction $\vee_{i \in I} \exists(a_i : K \to C_i, c_i)$ with c_i a negative remainder if such an equivalence holds.

5.2 Unfolding for Rules with NACs

We show that for rules with NACs initial conflicts are regular. This means that in this special case there exists a complete subset of conflicts that is finite (in the case of graphs and assuming finite rules). This conforms to the findings in [17,20], where an $\mathcal{M}$-complete set of critical pairs – as specific subset of conflicts – for graph transformation rules with NACs was introduced [20] (and generalized to $\mathcal{M}$-adhesive transformation systems [17]).

Theorem 6 (regular initial conflicts for rules with NACs). *Consider an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules. Given some initial conflict $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$ for a pair of rules $\langle \rho_1, \rho_2 \rangle$ with $\mathrm{ac}_{L_i} = \wedge_{j \in J} \neg \exists n_j : L_i \to N_j$ for $i = 1, 2$ and J some finite index set, then it is* regular*. In particular, $\mathrm{ac}_K \wedge \mathrm{ac}^*_K$ is equivalent to a condition $\vee_{i \in I} \exists(a_i : K \to C_i, c_i)$ with $c_i = \wedge_{q \in Q} \neg \exists n_q$ a condition on C_i and I some non-empty index set.*

The negative remainder c_i of each literal in $\vee_{i \in I} \exists(a_i : K \to C_i, c_i)$ of a regular initial conflict for rules with NACs thus consists of a set of NACs. Intuitively this means that we obtain for each initial conflict an $\mathcal{M}$-complete subset of concrete conflicts by adding the context described by a_i. As long as no NAC from c_i is violated we can extend such a concrete conflict to further ones.

Corollary 1 (complete unfolding: rules with NACs). *Consider an $\mathcal{M}$-adhesive system, with initial transformation pairs for conflicts along plain rules. Given a rule pair $\langle \rho_1, \rho_2 \rangle$ with $\mathrm{ac}_{L_i} = \wedge_{j \in J} \neg \exists n_j : L_i \to N_j$ for $i = 1, 2$, then $\cup_{stp \in \mathcal{S}} \mathcal{U}^{\mathcal{D}}(stp)$ is $\mathcal{M}$-complete w.r.t. parallel dependence.*

We show moreover that the initial conflict definition is a *conservative extension* of the critical pair definition for rules with NACs as given in [17,20], i.e., we show that each conflict in the disjunctive unfolding of an initial conflict as chosen in the proof of Theorem 6 is a critical pair for rules with NACs.

Theorem 7 (conservative unfolding). *In an $\mathcal{M}$-adhesive system with initial transformation pairs for conflicts along plain rules, if $stp_K : \langle tp_K, \mathrm{ac}_K, \mathrm{ac}^*_K \rangle$ is an initial conflict for rules $\langle \rho_1, \rho_2 \rangle$ with $\mathrm{ac}_{L_i} = \wedge_{j \in J} \neg \exists n_j : L_i \to N_j$ for $i = 1, 2$ and J some finite index set, then each conflict as chosen in the proof of Theorem 6 in $\mathcal{U}^{\mathcal{D}}(stp)$ is in particular a critical pair for $\langle \rho_1, \rho_2 \rangle$ as given in [17,20].*

Example 4 (conservative unfolding). Consider again the rules from Example 1 (having only NACs as ACs) and their application to the graph $G =$ ○ ○. The corresponding transformation pair tp_G is a critical pair for rules with NACs as given in [17,20]. This is because it is in particular a conflicting pair of transformations, and the morphism violating the NAC (since finding the three nodes) and therefore causing the conflict after applying the first rule to $G =$ ○ ○ obtaining some graph $H_1 =$ ○→○ ○ is jointly surjective together with the corresponding co-match. As argued already in Example 2 this critical pair for rules with NACs belongs to the unfolding (and in particular to the disjunctive unfolding) of the unique AC initial conflict $stp_{L_1+L_2} : \langle tp_{L_1+L_2}, \mathrm{ac}_{L_1+L_2}, \mathrm{ac}^*_{L_1+L_2} \rangle$.

6 Conclusion and Outlook

In this paper we have *generalized the theory of initial conflicts* (from plain rules, i.e. rules without application conditions) to *rules with application conditions* (ACs) in the framework of $\mathcal{M}$-adhesive transformation systems. We build on the notion of symbolic transformation pairs, since it turns out that it is not possible to find a complete subset of concrete conflicting transformation pairs in the case of rules with ACs. We have shown that initial conflicts are (minimally) complete w.r.t. parallel dependence as symbolic transformation pairs. Moreover, initial conflicts represent (analogous to the case of plain rules) proper subsets of critical pairs in the sense that for each critical pair (or also for each conflict), there exists a unique initial conflict representing it. We concluded the paper by showing sufficient conditions for finding unfoldings of initial conflicts that lead to (finite and) complete subsets of conflicts (as in the case of rules with NACs). Thereby we have shown that initial conflicts for rules with ACs represent a conservative extension of the critical pair theory for rules with NACs.

As future work we aim at finding *further interesting classes* allowing finite and (minimally) complete unfoldings into *subsets of conflicts*. This will serve as a guideline to be able to *develop and implement efficient conflict detection* techniques for rules with (specific) ACs, which has been an open challenge until today.

We are moreover planning to develop (semi-)automated detection of unfoldings of initial conflicts of rules with arbitrary ACs using dedicated automated reasoning and model finding for graph conditions [22,26,28]. It would be interesting to investigate in which *use cases* initial conflicts (or critical pairs) are useful already as symbolic transformation pairs, and in which use cases we rather need to consider unfoldings indeed. This is in line with the research on multi-granular conflict detection [3,18,24] investigating different levels of granularity that can be interesting from the point of view of applying conflict detection to different use cases. Finally, we plan to investigate conflict detection in the light of initial conflict theory for *attributed graph transformation* [5,13,16], and in particular the case of rules with so-called attribute conditions more specifically. It would also be interesting to further investigate initial conflicts for transformation rules (with ACs) not following the DPO approach.

Acknowledgement. We thank Jens Kosiol for pointing out that the set of initial conflicts for plain rules is not only complete, but also minimally complete. We were able to transfer this result to rules with ACs in this paper. Many thanks also to the reviewers for their detailed and constructive comments helping to finalize the paper.

References

1. Azzi, G.G., Corradini, A., Ribeiro, L.: On the essence and initiality of conflicts in m-adhesive transformation systems. J. Log. Algebr. Methods Program. **109**, 100482 (2019)
2. Bonchi, F., Gadducci, F., Kissinger, A., Sobociński, P., Zanasi, F.: Confluence of graph rewriting with interfaces. In: Yang, H. (ed.) ESOP 2017. LNCS, vol. 10201, pp. 141–169. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54434-1_6
3. Born, K., Lambers, L., Strüber, D., Taentzer, G.: Granularity of conflicts and dependencies in graph transformation systems. In: de Lara, J., Plump, D. (eds.) ICGT 2017. LNCS, vol. 10373, pp. 125–141. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-61470-0_8
4. Ehrig, H., Ehrig, K., Habel, A., Pennemann, K.: Theory of constraints and application conditions: from graphs to high-level structures. Fundam. Inform. **74**(1), 135–166 (2006)
5. 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
6. 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), 35–63 (2012). https://doi.org/10.3233/FI-2012-705
7. Ehrig, H., Golas, U., Habel, A., Lambers, L., Orejas, F.: $\mathcal{M}$-adhesive transformation systems with nested application conditions. Part 1: parallelism, concurrency and amalgamation. Math. Struct. Comput. Sci. **24**(4) (2014). https://doi.org/10.1017/S0960129512000357
8. Ehrig, H., Habel, A.: Graph grammars with application conditions. In: Rozenberg, G., Salomaa, A. (eds.) The Book of L, pp. 87–100. Springer, Heidelberg (1986). https://doi.org/10.1007/978-3-642-95486-3_7

9. Ehrig, H., Habel, A., Lambers, L., Orejas, F., Golas, U.: Local confluence for rules with nested application conditions. In: Ehrig, H., Rensink, A., Rozenberg, G., Schürr, A. (eds.) ICGT 2010. LNCS, vol. 6372, pp. 330–345. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15928-2_22
10. Habel, A., Heckel, R., Taentzer, G.: Graph grammars with negative application conditions. Fundam. Inform. **26**(3/4), 287–313 (1996). https://doi.org/10.3233/FI-1996-263404
11. Habel, A., Pennemann, K.: Correctness of high-level transformation systems relative to nested conditions. Math. Struct. Comput. Sci. **19**(2), 245–296 (2009). https://doi.org/10.1017/S0960129508007202
12. 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: Tracz, W., Young, M., Magee, J. (eds.) Proceedings of the 24th International Conference on Software Engineering, ICSE 2002, pp. 105–115. ACM (2002). https://doi.org/10.1145/581339.581355
13. Hristakiev, I., Plump, D.: Attributed graph transformation via rule schemata: Church-Rosser theorem. In: Milazzo, P., Varró, D., Wimmer, M. (eds.) STAF 2016. LNCS, vol. 9946, pp. 145–160. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50230-4_11
14. Knuth, D., Bendix, P.: Simple word problems in universal algebras. In: Leech, J. (ed.) Computational Problems in Abstract Algebra, pp. 263–297. Pergamon Press, Oxford (1970)
15. Koch, M., Mancini, L.V., Parisi-Presicce, F.: Graph-based specification of access control policies. J. Comput. Syst. Sci. **71**(1), 1–33 (2005). https://doi.org/10.1016/j.jcss.2004.11.002
16. Kulcsár, G., Deckwerth, F., Lochau, M., Varró, G., Schürr, A.: Improved conflict detection for graph transformation with attributes. In: Rensink, A., Zambon, E. (eds.) Proceedings GaM@ETAPS 2015. EPTCS, vol. 181, pp. 97–112 (2015). https://doi.org/10.4204/EPTCS.181.7
17. Lambers, L.: Certifying rule-based models using graph transformation. Ph.D. thesis, Berlin Institute of Technology (2009). http://opus.kobv.de/tuberlin/volltexte/2010/2522/
18. Lambers, L., Born, K., Kosiol, J., Strüber, D., Taentzer, G.: Granularity of conflicts and dependencies in graph transformation systems: a two-dimensional approach. J. Log. Algebr. Methods Program. **103**, 105–129 (2019). https://doi.org/10.1016/j.jlamp.2018.11.004
19. Lambers, L., Born, K., Orejas, F., Strüber, D., Taentzer, G.: Initial conflicts and dependencies: critical pairs revisited. In: Heckel, R., Taentzer, G. (eds.) Graph Transformation, Specifications, and Nets. LNCS, vol. 10800, pp. 105–123. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75396-6_6
20. Lambers, L., Ehrig, H., Orejas, F.: Conflict detection for graph transformation with negative application conditions. In: Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G. (eds.) ICGT 2006. LNCS, vol. 4178, pp. 61–76. Springer, Heidelberg (2006). https://doi.org/10.1007/11841883_6
21. Lambers, L., Ehrig, H., Orejas, F.: Efficient conflict detection in graph transformation systems by essential critical pairs. Electr. Notes Theor. Comput. Sci. **211**, 17–26 (2008)
22. Lambers, L., Orejas, F.: Tableau-based reasoning for graph properties. In: Giese, H., König, B. (eds.) ICGT 2014. LNCS, vol. 8571, pp. 17–32. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09108-2_2

23. Lambers, L., Orejas, F.: Initial conflicts for transformation rules with nested application conditions (2020). arXiv:2005.05901 [cs.LO]
24. Lambers, L., Strüber, D., Taentzer, G., Born, K., Huebert, J.: Multi-granular conflict and dependency analysis in software engineering based on graph transformation. In: Chaudron, M., Crnkovic, I., Chechik, M., Harman, M. (eds.) Proceedings of the 40th International Conference on Software Engineering, ICSE 2018, pp. 716–727. ACM (2018). https://doi.org/10.1145/3180155.3180258
25. Ohlebusch, E.: Advanced Topics in Term Rewriting. Springer, Heidelberg (2002). https://doi.org/10.1007/978-1-4757-3661-8
26. Pennemann, K.: Development of correct graph transformation systems. Ph.D. thesis, University of Oldenburg, Germany (2009). http://oops.uni-oldenburg.de/volltexte/2009/948/
27. Plump, D.: Hypergraph rewriting: Critical pairs and undecidability of confluence. In: Sleep, R., Plasmeijer, R., van Eekelen, M. (eds.) Term Graph Rewriting: Theory and Practice, pp. 201–213. Wiley, Chichester (1993)
28. Schneider, S., Lambers, L., Orejas, F.: Symbolic model generation for graph properties. In: Huisman, M., Rubin, J. (eds.) FASE 2017. LNCS, vol. 10202, pp. 226–243. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54494-5_13