Efficient Conflict Detection in Graph Transformation Systems by Essential Critical Pairs

Leen Lambers , Hartmut Ehrig [1,2]

Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin
Germany

Fernando Orejas [3]

Dept. L.S.I.
Tech. Univ. Catalonia
Barcelona, Spain

Abstract

The well-known notion of critical pairs already allows a static conflict detection, which is important for all kinds of applications and already implemented in AGG. Unfortunately the standard construction is not very efficient. This paper introduces the new concept of essential critical pairs allowing a more efficient conflict detection. This is based on a new conflict characterization, which determines for each conflict occuring between the rules of the system the exact conflict reason. This new notion of conflict reason leads us to an optimization of conflict detection. Efficiency is obtained because the set of essential critical pairs is a proper subset of all critical pairs of the system and therefore the set of representative conflicts to be computed statically diminishes. It is shown that for each conflict in the system, there exists an essential critical pair representing it. Moreover each essential critical pair is unique with regard to its conflict reason and thus represents each conflict not only in a minimal, but also in a unique way. Main new results presented in this paper are a characterization of conflicts, completeness and uniqueness of essential critical pairs and a local confluence lemma based on essential critical pairs. The theory of essential critical pairs is the basis to develop and implement a more efficient conflict detection algorithm in the near future.

Keywords: conflict, confluence, critical pair, graph transformation

[1] Email: leen@cs.tu-berlin.de
[2] Email: ehrig@cs.tu-berlin.de
[3] Email: orejas@lsi.upc.edu

1   Introduction

Static conflict detection is a well-known important task for all kinds of rewriting systems especially also for graph transformation systems. To enable a *static* conflict detection the notion of critical pairs was developed at first for hypergraph rewriting [12] and then for all kinds of transformation systems fitting into the framework of adhesive high-level replacement categories [6]. Usually a straightforward way (i.e. directly according to the definition) is used to compute the set of all critical pairs of a graph transformation system. This is very important for all kinds of applications like for example graph parsing [2], conflict detection in graph transformation based modeling [8] [1] and model transformation [3] [4], refactoring [11], etc. Up to now, however, there is almost no theory which allows an efficient implementation of conflict detection. Therefore our paper [9] and this paper concentrate on exactly this subject.

In [9] it was already explained which optimizations lead to a more efficient conflict detection in a graph transformation system. Unfortunately this efficiency could only be obtained for conflicts induced by a pair of rules with one of the rules non-deleting. This is quite a strong restriction, since in particular a lot of conflicts are induced by a pair of deleting rules. Therefore this paper formulates a characterization of conflicts, covering also these kind of conflicts. Moreover this conflict characterization leads us to the identification of the conflict reason of each conflict.

The notion of critical pair introduced in [12], [6] expresses each conflict in its minimal context. In some cases though two different critical pairs express the same kind of conflict. Therefore exploiting the uniqueness of each conflict reason mentioned above, it is possible to further reduce the set of critical pairs to a subset of essential critical pairs. This subset expresses each kind of conflict which can occur in a graph transformation system in a minimal context and moreover in a unique way. This uniqueness property and the constructive conflict reason definition facilitates the optimization of detecting all conflicts of a graph transformation system.

The following sections explain how to characterize conflicts and what the conflict reason is, how we come to the definition of essential critical pairs and which properties they fullfill. Main new results presented in this paper are a characterization of conflicts, completeness and uniqueness of essential critical pairs and a local confluence lemma based on essential critical pairs. More details concerning well-known definitions and new proofs are given in the long version of this paper [10] to show the mature status of the theory. The theory of essential critical pairs is the basis to develop and implement a more efficient conflict detection algorithm in the near future.

2   Conflict Characterization and Conflict Reason

In this section we formulate a theory which leads us to the identification of the conflict reason for each occuring conflict in a graph transformation system where
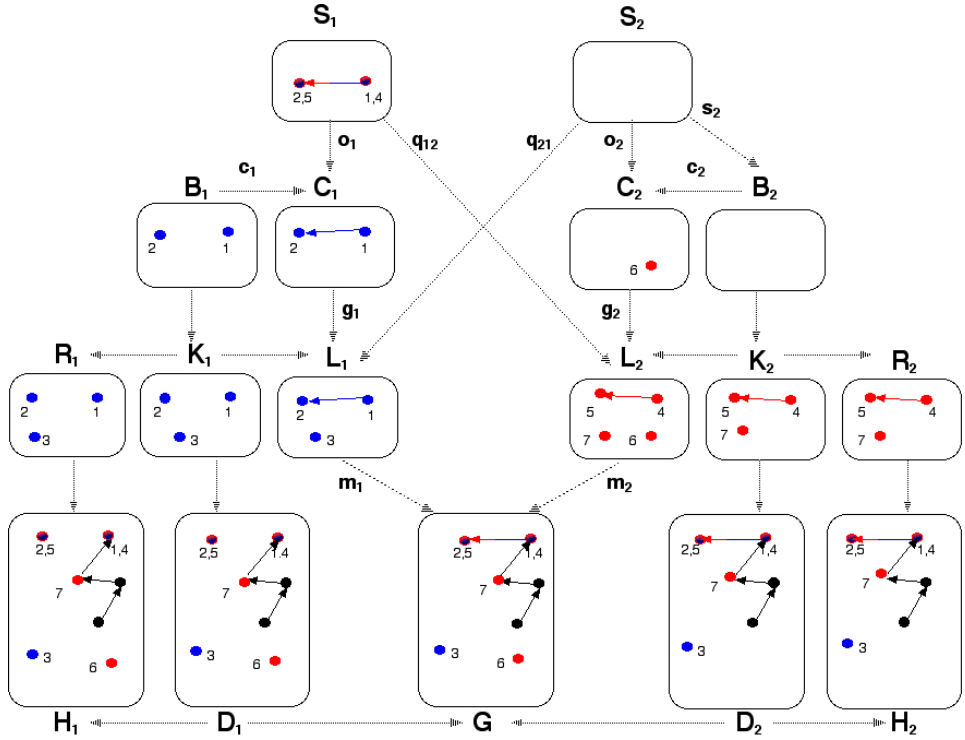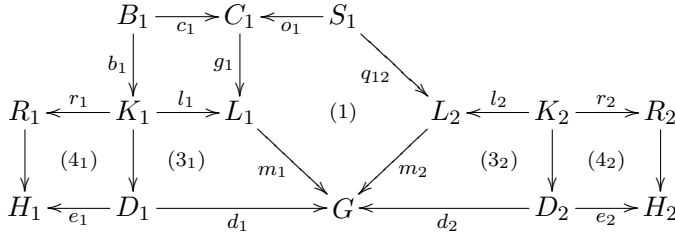
Fig. 1. asymmetrical delete-use-conflict

we only consider injective matches. This new notion of conflict reason will help us consequently in the next sections to detect in a static way all representative conflicts of a graph transformation system. At first, we look at an example of two direct transformations $H_1 \overset{p_1,m_1}{\Longleftarrow} G \overset{p_2,m_2}{\Longrightarrow} H_2$ in conflict in Fig. 1, generated by two deleting rules $p_1 : L_1 \leftarrow K_1 \rightarrow R_1$ and $p_2 : L_2 \leftarrow K_2 \rightarrow R_2$. Looking at both direct transformations we can describe the reason for the conflict between them as follows. The left transformation deletes edge $(1, 4 - 2, 5)$ and that is why rule p_2 can not be applied anymore to the same location on graph H_1. The structure (S_1, o_1, q_{12}), constructed as pullback of $(m_1 \circ g_1, m_2)$, captures exactly the conflict reason for this conflict, because it holds the edge $(1, 4-2, 5)$ to be deleted by the left transformation, but used by the other one. The following definitions and theorem explain how to formalize this new notion of conflict reason. Please note, that for all subsequent definitions and theorems the following pair of rules $p_i : L_i \overset{l_i}{\leftarrow} K_i \overset{r_i}{\rightarrow} R_i$ with boundary B_i and context C_i, defining an inital pushout (1) over l_i (see [6]) and injective graph morphisms b_i, c_i, g_i, l_i are given, i.e. b_i, c_i, g_i, l_i in M $(i = 1, 2)$, where M is the set of all injective graph morphisms.

$$
\begin{array}{ccc}
B_i & \xrightarrow{\;c_i\;} & C_i \\
{\scriptstyle b_i}\big\downarrow & {\scriptstyle (1)} & \big\downarrow{\scriptstyle g_i} \\
K_i & \xrightarrow{\;l_i\;} & L_i
\end{array}
$$

Definition 2.1 [conflict condition] Given a pair of direct transformations $H_1 \overset{p_1,m_1}{\Longleftarrow}$

$G \overset{p_2,m_2}{\Rightarrow} H_2$

- $(S_1, o_1 : S_1 \to C_1, q_{12} : S_1 \to L_2)$ the pullback of $(m_1 \circ g_1, m_2)$ satisfies the *conflict condition* if: $\nexists s_1 : S_1 \to B_1 \in M$ such that $c_1 \circ s_1 = o_1$

$$B_1 \xrightarrow{c_1} C_1 \xleftarrow{o_1} S_1$$

$$
\begin{array}{ccccccccc}
B_1 & \xrightarrow{c_1} & C_1 & \xleftarrow{o_1} & S_1 & & & & \\
\downarrow{b_1} & & \downarrow{g_1} & & & \searrow{q_{12}} & & & \\
R_1 \xleftarrow{r_1} K_1 & \xrightarrow{l_1} & L_1 & (1) & & L_2 & \xleftarrow{l_2} K_2 & \xrightarrow{r_2} & R_2 \\
\downarrow (4_1) \downarrow & (3_1) & \searrow{m_1} & & \swarrow{m_2} & & (3_2) & (4_2) & \downarrow \\
H_1 \xleftarrow{e_1} D_1 & \xrightarrow{d_1} & & G & \xleftarrow{d_2} & & D_2 & \xrightarrow{e_2} & H_2
\end{array}
$$

- $(S_2, q_{21} : S_2 \to L_1, o_2 : S_2 \to C_2)$ the pullback of $(m_1, m_2 \circ g_2)$ satisfies the *conflict condition* if: $\nexists s_2 : S_2 \to B_2 \in M$ such that $c_2 \circ s_2 = o_2$

$$
\begin{array}{ccccccccc}
& & & S_2 & \xrightarrow{o_2} & C_2 & \xleftarrow{c_2} & B_2 & \\
& & \swarrow{q_{21}} & & & \downarrow{g_2} & & \downarrow{b_2} & \\
R_1 \xleftarrow{r_1} K_1 & \xrightarrow{l_1} & L_1 & (1) & & L_2 & \xleftarrow{l_2} K_2 & \xrightarrow{r_2} & R_2 \\
\downarrow (4_1) \downarrow & (3_1) & \searrow{m_1} & & \swarrow{m_2} & & (3_2) & (4_2) & \downarrow \\
H_1 \xleftarrow{e_1} D_1 & \xrightarrow{d_1} & & G & \xleftarrow{d_2} & & D_2 & \xrightarrow{e_2} & H_2
\end{array}
$$
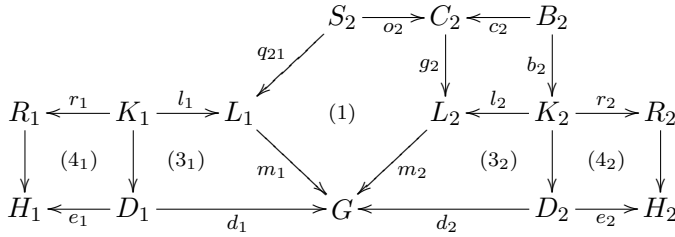
In the example in Fig. 1 $(S_1, o_1 : S_1 \to C_1, q_{12} : S_1 \to L_2)$ satisfies, but $(S_2, q_{21} : S_2 \to L_1, o_2 : S_2 \to C_2)$ doesn't satisfy the conflict condition. The idea behind this conflict condition is that a conflict occurs if graph parts which are deleted are overlapped with parts to be used by the other transformation. This idea is expressed formally by a new characterization of conflicts in the next theorem.

Theorem 2.2 (Characterization Conflict) *Given a pair of direct transformations* $H_1 \overset{p_1,m_1}{\Leftarrow} G \overset{p_2,m_2}{\Rightarrow} H_2$ *with* $(S_1, o_1 : S_1 \to C_1, q_{12} : S_1 \to L_2)$ *the pullback of* $(m_1 \circ g_1, m_2)$ *and* $(S_2, q_{21} : S_2 \to L_1, o_2 : S_2 \to C_2)$ *the pullback of* $(m_2, m_1 \circ g_1)$ *then the following equivalence holds:*

$$H_1 \overset{p_1,m_1}{\Leftarrow} G \overset{p_2,m_2}{\Rightarrow} H_2 \text{ are in conflict}$$

$$\Leftrightarrow$$

$$(S_1, o_1, q_{12}) \lor (S_2, q_{21}, o_2) \text{ satisfies the conflict condition}$$

Theorem 2.2 (proof see [10]) teaches us, that a pair of direct transformations $H_1 \overset{p_1,m_1}{\Leftarrow} G \overset{p_2,m_2}{\Rightarrow} H_2$ is in conflict, because one of the following three reasons:

(i) (S_1, o_1, q_{12}) satisfies and (S_2, q_{21}, o_2) doesn't satisfy the conflict condition (asymmetrical delete-use-conflict)

(ii) (S_1, o_1, q_{12}) doesn't satisfy and (S_2, q_{21}, o_2) satisfies the conflict condition (asymmetrical use-delete-conflict)

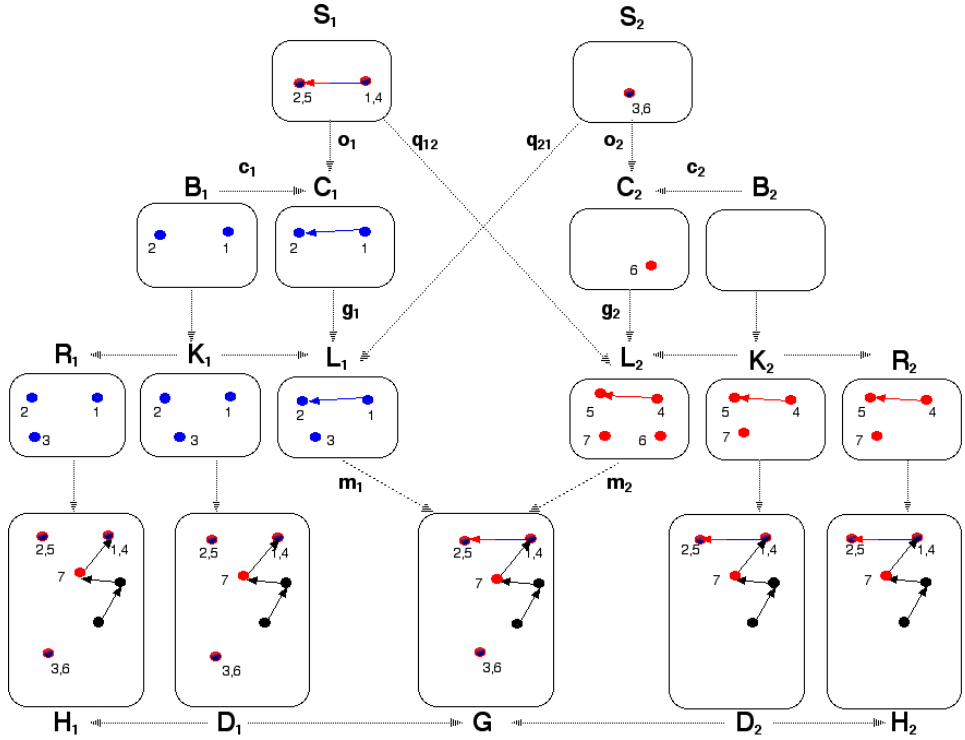(iii) both (S_1, o_1, q_{12}) and (S_2, q_{21}, o_2) satisfy the conflict condition (symmetrical conflict)

Fig. 2. symmetrical conflict

In the case of asymmetrical conflicts rule p_1 (resp. p_2) deletes something, what is used by rule p_2 (resp. p_1), but *not* the other way round. Let us consider in more detail the case of symmetrical conflicts. In Fig. 2 you can see an example of two direct transformations, having a symmetrical conflict. Then (S_1, o_1, q_{12}) expresses the part which is deleted by p_1 and used by rule p_2 and (S_2, p_1, o_2) expresses the part which is deleted by p_2 and used by rule p_1. In order to summarize both parts into one graph expressing exactly the graph parts of L_1 and L_2 responsible for the conflict, we make the construction depicted in Fig. 3. In this construction (S', a_1, a_2) is the pullback of $(m_1 \circ g_1 \circ o_1 : S_1 \rightarrow G_1, m_2 \circ g_2 \circ o_2 : S_2 \rightarrow G_2)$ and (S, s'_1, s'_2) is the pushout of (S', a_1, a_2). This is, we determine the part S', which is deleted by both rules and glue S_1 and S_2 together over this part leading to S. Note, that in the example in Fig. 2 S' would be the empty graph. Now we have $g_1 \circ o_1 \circ a_1 = q_{21} \circ a_2$ and similar $g_1 \circ o_2 \circ a_2 = q_{12} \circ a_1$ because m_1 is mono and $m_1 \circ g_1 \circ o_1 \circ a_1 = m_2 \circ g_2 \circ o_2 \circ a_2 = m_1 \circ q_{21} \circ a_2$. Together with the pushout property of S this implies, that there exists a unique $s_1 : S \rightarrow L_1$ (resp. $s_2 : S \rightarrow L_2$) s.t. $g_1 \circ o_1 = s_1 \circ s'_1$ and $q_{21} = s_1 \circ s'_2$ (resp. $g_2 \circ o_2 = s_2 \circ s'_2$ and $q_{12} = s_2 \circ s'_1$). Moreover using PO-property of S we can conclude $m_1 \circ s_1 = m_2 \circ s_2$. Please note, that in Fig. 3 we left out q_{21} and q_{12}. Thus in the end (S, s_1, s_2) summarizes which parts of L_1 and L_2 are responsible for the symmetrical conflict. **Remark:** $S = S_1 = S_2$ if and only if all elements deleted by p_1 are also deleted by p_2 and the other way round (pure delete-delete-conflict). $S' = \emptyset$ if and only if all elements deleted by p_1 are not deleted, but used by p_2 and the other way round (pure delete-use-conflict
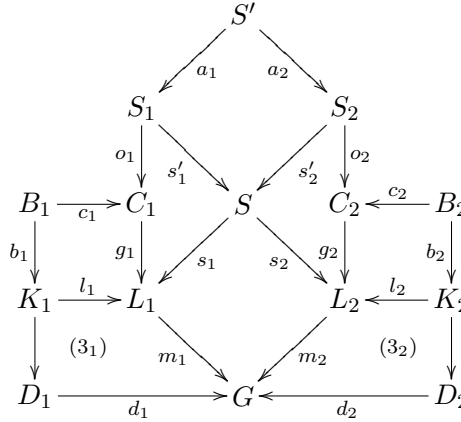
Fig. 3. construction of the conflict reason for symmetrical conflicts

as in the example in Fig. 3).

We can resume these observations into the following definition.

Definition 2.3 [conflict reason span] Given a pair of direct transformations $H_1 \overset{p_1,m_1}{\Leftarrow} G \overset{p_2,m_2}{\Rightarrow} H_2$ in conflict, the *conflict reason span* of $H_1 \overset{p_1,m_1}{\Leftarrow} G \overset{p_2,m_2}{\Rightarrow} H_2$ is one of the following spans using the notation of Def.2.1:

- $(S_1, g_1 \circ o_1, q_{12})$ if (S_1, o_1, q_{12}) satisfies and (S_2, q_{21}, o_2) doesn't satisfy the conflict condition

- $(S_2, q_{21}, g_2 \circ o_2)$ if (S_1, o_1, q_{12}) doesn't satisfy and (S_2, q_{21}, o_2) satisfies the conflict condition

- (S, s_1, s_2) if (S_1, o_1, q_{12}) and (S_2, q_{21}, o_2) both satisfy the conflict condition and (S, s_1, s_2) is constructed as above

3   Definition of Essential Critical Pairs

By means of the new notion of conflict reason it is possible to define the new notion of essential critical pairs. The idea behind this notion is that for each conflict reason we have an essential critical pair, expressing the conflict caused by exactly this conflict reason in a minimal context.

Definition 3.1 [essential critical pair] A pair of direct transformations $P_1 \overset{p_1,m_1}{\Leftarrow} K \overset{p_2,m_2}{\Rightarrow} P_2$ is an *essential critical pair* for the pair of rules (p_1, p_2) if the following holds: $P_1 \overset{p_1,m_1}{\Leftarrow} K \overset{p_2,m_2}{\Rightarrow} P_2$ are in conflict and (K, m_1, m_2) is a pushout of the conflict reason span $(S_1, g_1 \circ o_1, q_{12}),(S_2, q_{21}, g_2 \circ o_2)$ or (S, s_1, s_2) of $P_1 \overset{p_1,m_1}{\Leftarrow} K \overset{p_2,m_2}{\Rightarrow} P_2$ according to Definition 2.3.

Fact 3.2 *Each essential critical pair $P_1 \overset{p_1,m_1}{\Leftarrow} K \overset{p_2,m_2}{\Rightarrow} P_2$ of (p_1, p_2) is a critical pair of (p_1, p_2).*

Proof. Each essential critical pair is a pair of direct transformations in conflict. The overlappings (m_1, m_2) of an essential critical pair are jointly surjective, because they
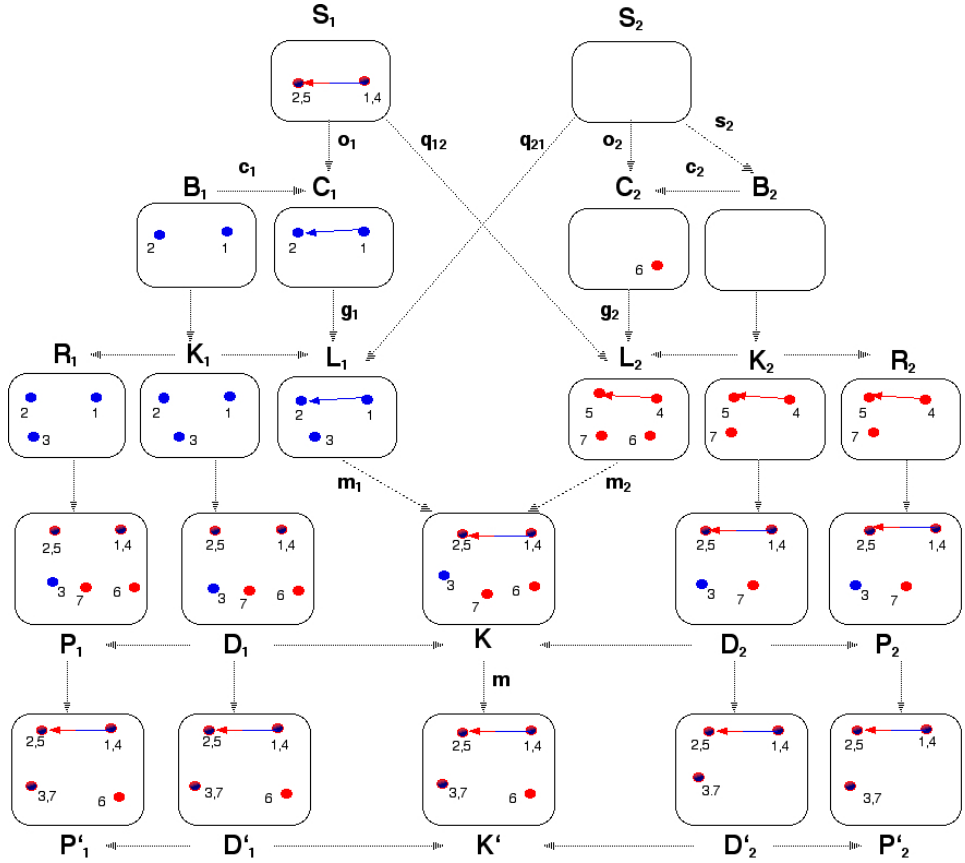
Fig. 4. essential crit. pair $P_1 \overset{p_1,m_1}{\Longleftarrow} K \overset{p_2,m_2}{\Longrightarrow} P_2$ into crit. pair $P'_1 \overset{p_1,m'_1}{\Longleftarrow} K' \overset{p_2,m'_2}{\Longrightarrow} P'_2$

are constructed via a pushout. □

Remark: The main idea shown in the next section is that it is sufficient to consider essential critical pairs and not every critical pair is an essential critical pair. This is shown in the example in Fig. 4. The essential critical pair $P_1 \overset{p_1,m_1}{\Longleftarrow} K \overset{p_2,m_2}{\Longrightarrow} P_2$ of (p_1, p_2) only overlaps the edge $(1 - 2)$ with $(4 - 5)$, since this is exactly the reason for the delete-use-conflict. However the matches (m'_1, m'_2) of the critical pair $P'_1 \overset{p_1,m'_1}{\Longleftarrow} K' \overset{p_2,m'_2}{\Longrightarrow} P'_2$ (with $m'_1 = m_1 \circ m$ and $m'_2 = m_2 \circ m$) overlap in addition node 7 with node 3, which are not responsible for the conflict at all. The pair of rules, used in the example in Fig. 1,2 and 4 induces, according to the critical pair detection in [13] AGG 14 critical pairs, but only 3 of them are essential critical pairs.

4 Properties of Essential Critical Pairs

In this section we will prove that it is enough to compute all essential critical pairs to detect all conflicts, occuring in a graph transformation system. Therefore we show, that the set of essential critical pairs fullfills the following three properties.

At first, we demonstrate that each conflict, occuring in the system can be expressed by an essential critical pair (*completeness*). The second property says, that each essential critical pair is *unique* with regard to its conflict reason span. Finally we will prove a *local confluence lemma* based on essential critical pairs.

Theorem 4.1 (Completeness and Uniqueness of Essential Critical Pairs) *For each critical pair $P_1' \overset{p_1,m_1'}{\Leftarrow} K' \overset{p_2,m_2'}{\Rightarrow} P_2'$ of (p_1, p_2) there exists a unique essential critical pair $P_1 \overset{p_1,m_1}{\Leftarrow} K \overset{p_2,m_2}{\Rightarrow} P_2$ of (p_1, p_2) with the same conflict reason span and extension diagrams (1) and (2).*

$$
\begin{array}{ccccc}
P_1 & \Longleftarrow & K & \Longrightarrow & P_2 \\
\downarrow & (1) \;\; m & \downarrow & (2) & \downarrow \\
P_1' & \Longleftarrow & K' & \Longrightarrow & P_2'
\end{array}
$$

Remark: $m : K \to K'$ *is an epimorphism, but not necessarily a monomorphism.*

The proof of this theorem is given in appendix C in [10].

The set of essential critical pairs is unique in the following sense:

Theorem 4.2 (Uniqueness of Essential Critical Pairs) *Each essential critical pair is unique with regard to its conflict reason span.*

Proof. This follows directly from Theorem 4.1 and Fact 3.2. □

Note, that the set of critical pairs doesn't possess this uniqueness property. The example in Fig. 4 shows two different critical pairs (a normal critical pair $P_1' \overset{p_1,m_1'}{\Leftarrow} K' \overset{p_2,m_2'}{\Rightarrow} P_2'$ and an essential critical pair $P_1' \overset{p_1,m_1'}{\Leftarrow} K' \overset{p_2,m_2'}{\Rightarrow} P_2'$) possessing the same conflict reason span.

The following theorem states that it is enough to check each essential critical pair for strict confluence as defined in [12][6] to obtain local confluence of a graph transformation system.

Theorem 4.3 (Local Confluence Lemma based on Essential Critical Pairs) *If all essential critical pairs of a graph transformation system are strictly confluent, then this graph transformation system is locally confluent.*

The proof of this theorem is given in appendix D in [10]. It is similar to the proof of the local confluence lemma in [6], but avoids to assume that $m : K \to K'$ is a monomorphism. Note, that the theory of essential critical pairs not only simplifies static conflict detection, but in addition confluence analysis of the conflicts in the system. This is because the number of conflicts to be analyzed for strictly confluence diminishes, since the essential critical pairs are a subset of the critical pairs.

5 Summary and Outlook

In this paper we have introduced the new notion of essential critical pairs and corresponding results which are the basis of a more efficient conflict detection and

local confluence analysis than the standard techniques based on usual critical pairs. In a forthcoming paper we will give on this basis an efficient correct construction of all essential critical pairs for each pair of rules and a corresponding algorithm which will improve the current critical pair algorithm of AGG [13]. In addition we assume and will verify that an extension of this theory to graph transformation with non-injective matches is possible, provided that the conflict condition is slightly generalized. Moreover the following question in the context of conflict detection for graph transformation systems is subject of future work. What kind of new conflicts occur and which new critical pair notion is necessary to describe the conflicts in graph transformation systems with application conditions and constraints [5] and what about the more general case of typed, attributed graph transformation systems [7]?

References

[1] Baresi, L., R. Heckel, S. Thöne and V. D., *Modeling and analysis of architectural styles based on graph transformation*, in: I. Crnkovic, H. Schmidt, J. Stafford and K. Wallnau, editors, *Proc. of the 6th ICSE Workshop on Component-Based Software Engineering: Automated Reasoning and Prediction Portland* (2003), pp. 67–72.

[2] Bottoni, P., A. Schürr and G. Taentzer, *Efficient Parsing of Visual Languages based on Critical Pair Analysis and Contextual Layered Graph Transformation*, in: *Proc. IEEE Symposium on Visual Languages*, 2000, long version available as technical report SI-2000-06, University of Rom.

[3] de Lara, J. and G. Taentzer, *Automated Model Transformation and its Validation using AToM³ and AGG*, in: A. Blackwell, K. Marriott and A. Shimojima, editors, *Diagrammatic Representation and Inference* (2004).

[4] Ehrig, H. and K. Ehrig, *Overview of Formal Concepts for Model Transformations based on Typed Attributed Graph Transformation*, in: *Proc. International Workshop on Graph and Model Transformation (GraMoT'05)*, Electronic Notes in Theoretical Computer Science (2005).
URL http://tfs.cs.tu-berlin.de/publikationen/Papers05/EE05.pdf

[5] Ehrig, H., K. Ehrig, A. Habel and K.-H. Pennemann, *Constraints and application conditions: From graphs to high-level structures*, in: F. Parisi-Presicce, P. Bottoni and G. Engels, editors, *Proc. 2nd Int. Conference on Graph Transformation (ICGT'04)*, LNCS 3256 (2004), pp. 287–303.
URL http://www.cs.tu-berlin.de/~ehrig/publications/ICGT04paper3.pdf

[6] Ehrig, H., A. Habel, J. Padberg and U. Prange, *Adhesive high-level replacement categories and systems*, in: F. Parisi-Presicce, P. Bottoni and G. Engels, editors, *Proc. 2nd Int. Conference on Graph Transformation (ICGT'04)*, LNCS 3256 (2004), pp. 144–160.
URL http://www.cs.tu-berlin.de/~ehrig/publications/ICGT04paper1.pdf

[7] Ehrig, H., U. Prange and G. Taentzer, *Fundamental theory for typed attributed graph transformation*, in: F. Parisi-Presicce, P. Bottoni and G. Engels, editors, *Proc. 2nd Int. Conference on Graph Transformation (ICGT'04), Rome, Italy*, LNCS 3256, Springer, 2004 pp. 161–177.
URL http://www.cs.tu-berlin.de/~ehrig/publications/ICGT04paper2.pdf

[8] Hausmann, J., R. Heckel and G. Taentzer, *Detection of Conflicting Functional Requirements in a Use Case-Driven Approach*, in: *Proc. of Int. Conference on Software Engineering 2002*, Orlando, USA, 2002, to appear.

[9] Lambers, L., H. Ehrig and F. Orejas, *Efficient detection of conflicts in graph-based model transformation*, in: *Proc. International Workshop on Graph and Model Transformation (GraMoT'05)*, Electronic Notes in Theoretical Computer Science (2005).
URL http://tfs.cs.tu-berlin.de/publikationen/Papers05/LEO05.pdf

[10] Lambers, L., H. Ehrig and F. Orejas, *Efficient conflict detection in graph transformation systems by essential critical pairs*, Technical report, Technische Universität Berlin (2006).

[11] Mens, T., G. Taentzer and O. Runge, *Detecting Structural Refactoring Conflicts using Critical Pair Analysis*, in: R. Heckel and T. Mens, editors, *Proc. Workshop on Software Evolution through Transformations: Model-based vs. Implementation-level Solutions (SETra'04), Satellite Event of ICGT'04)* (2004).
URL http://www.cs.tu-berlin.de/~gabi/gMTR04.pdf

[12] Plump, D., *Hypergraph Rewriting: Critical Pairs and Undecidability of Confluence*, in: M. Sleep, M. Plasmeijer and M. C. van Eekelen, editors, *Term Graph Rewriting*, Wiley, 1993 pp. 201–214.

[13] Taentzer, G., *AGG: A Graph Transformation Environment for Modeling and Validation of Software*, in: J. Pfaltz, M. Nagl and B. Boehlen, editors, *Application of Graph Transformations with Industrial Relevance (AGTIVE'03)*, LNCS 3062, Springer, 2004 pp. 446 – 456.