Efficient Conflict Detection in Graph Transformation Systems by Essential Critical Pairs

Leen Lambers, Hartmut Ehrig

Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin
Germany

Fernando Orejas

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 occurring 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 possesses a unique 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.

Key words: conflict, confluence, critical pair, graph transformation
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 \cite{11} and then for all kinds of transformation systems fitting into the framework of adhesive high-level replacement categories \cite{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 \cite{2}, conflict detection in graph transformation based modeling \cite{8} \cite{1} \cite{8} and model transformation \cite{3} \cite{4}, refactoring \cite{10}, etc. Up to now, however, there is almost no theory which allows an efficient implementation of conflict detection. Therefore our paper \cite{9} and this paper concentrate on exactly this subject.

In \cite{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 \cite{11}, \cite{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 fulfill. 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 appendix 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.

\footnote{Email: leen@cs.tu-berlin.de}
\footnote{Email: ehrig@cs.tu-berlin.de}
\footnote{Email: orejas@lsi.upc.edu}
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 occurring conflict in a graph transformation system where 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 $H_1 \xrightarrow{p_1,m_1} G \xrightarrow{p_2,m_2} H_2$ 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 $p_i : L_i \leftarrow K_i \rightarrow R_i$, with boundary $B_i$ and context $C_i$, defining an initial 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
Given a pair of direct transformations $H_1 \overset{p_1 \cdot m_1}{\triangleleft} G \overset{p_2 \cdot m_2}{\triangleright} H_2$.

- $(S_1, o_1 : S_1 \rightarrow C_1, q_{12} : S_1 \rightarrow L_2)$ the pullback of $(m_1 \circ g_1, m_2)$ satisfies the conflict condition if: $\not\exists s_1 : S_1 \rightarrow B_1 \in M$ such that $c_1 \circ s_1 = o_1$

- $(S_2, q_{21} : S_2 \rightarrow L_1, o_2 : S_2 \rightarrow C_2)$ the pullback of $(m_1, m_2 \circ g_2)$ satisfies the conflict condition if: $\not\exists s_2 : S_2 \rightarrow B_2 \in M$ such that $c_2 \circ s_2 = o_2$

In the example in Fig. 1 $(S_1, o_1 : S_1 \rightarrow C_1, q_{12} : S_1 \rightarrow L_2)$ satisfies, but $(S_2, q_{21} : S_2 \rightarrow L_1, o_2 : S_2 \rightarrow C_2)$ 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 $H_1 \overset{p_1 \cdot m_1}{\triangleleft} G \overset{p_2 \cdot m_2}{\triangleright} H_2$ with $(S_1, o_1 : S_1 \rightarrow C_1, q_{12} : S_1 \rightarrow L_2)$ the pullback of $(m_1 \circ g_1, m_2)$ and $(S_2, q_{21} : S_2 \rightarrow L_1, o_2 : S_2 \rightarrow C_2)$ the pullback of $(m_2, m_1 \circ g_1)$ then the following equivalence holds:

$$H_1 \overset{p_1 \cdot m_1}{\triangleleft} G \overset{p_2 \cdot m_2}{\triangleright} H_2 \text{ are in conflict} \iff (S_1, o_1, q_{12}) \lor (S_2, q_{21}, o_2) \text{ satisfies the conflict condition}$$

Theorem 2.2 (proof see appendix B) teaches us, that a pair of direct transformations $H_1 \overset{p_1 \cdot m_1}{\triangleleft} G \overset{p_2 \cdot m_2}{\triangleright} H_2$ is in conflict, because one of the following three
Fig. 2. symmetrical conflict

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)

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 \((m_1 \circ g_1 \circ o_1 : S_1 \to G_1, m_2 \circ g_2 \circ o_2 : S_2 \to G_2)\) 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 \(m_1 \circ g_1 \circ o_1 \circ a_1 = m_2 \circ g_2 \circ o_2 \circ a_2 = \)
Fig. 3. construction of the conflict reason for symmetrical conflicts

$m_1 \circ q_{21} \circ a_2$. 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 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 $H_1 \xleftarrow{p_1 \circ m_1} G \xrightarrow{p_2 \circ m_2} H_2$ in conflict, the conflict reason span of $H_1 \xleftarrow{p_1 \circ m_1} G \xrightarrow{p_2 \circ m_2} H_2$ 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 $P_1 \xleftarrow{p_1 \circ m_1} G \xrightarrow{p_2 \circ m_2} P_2$...
$K^{p_2,m_2} \Rightarrow P_2$ is an essential critical pair for the pair of rules $(p_1, p_2)$ if the following holds: $P_1 \xleftarrow{p_1,m_1} K^{p_2,m_2} \Rightarrow P_2$ 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 $P_1 \xleftarrow{p_1,m_1} K^{p_2,m_2} \Rightarrow P_2$ according to Definition 2.3.

**Fact 3.2** Each essential critical pair $P_1 \xleftarrow{p_1,m_1} K^{p_2,m_2} \Rightarrow P_2$ 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 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 $P_1 \xleftarrow{p_1,m_1} K^{p_2,m_2} \Rightarrow P_2$ 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 $P_1' \xleftarrow{p_1,m_1'} K' \Rightarrow P_2'$ (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 [12] 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, occurring in a graph transformation system. Therefore we show, that the set of essential critical pairs fulfills the following three properties. At first, we demonstrate that each conflict, occurring in the system can be expressed by an essential critical pair (completeness). The second property says, that each essential critical pair is induced by a unique conflict reason. 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 $P_1 \xleftarrow{p_1,m_1} K^{p_2,m_2} \Rightarrow P_2$ of $(p_1, p_2)$ there exists a unique essential critical pair $P_1 \xleftarrow{p_1,m_1} K^{p_2,m_2} \Rightarrow P_2$ of $(p_1, p_2)$ with the same conflict reason span and extension diagrams (1) and (2).

$$
\begin{array}{ccc}
P_1 & \xleftarrow{m} & K & \xrightarrow{m} & P_2 \\
1 \downarrow & & \downarrow & & 2 \\
& & P_1' & \xleftarrow{m} & P_2'
\end{array}
$$

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

The proof of this theorem is given in appendix C.
The set of essential critical pairs is unique in the following sense:

**Theorem 4.2 (Uniqueness of Essential Critical Pairs)** Each essential critical pair possesses a unique 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 $P_1^{p_1,m_1} \xleftarrow{K} P_2$ and an essential critical pair $P_1^{p_1,m_1} \xleftarrow{K'} P'_2$) 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 [11] [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. It is similar to the proof of the local confluence lemma in [6], but avoids to assume that $m : K \rightarrow K'$.
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 [12]. 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


APPENDICES

A  DPO Graph Transformation: Basic Definitions, Conflicts and Critical Pairs

The theory of confluence and critical pairs has been worked out for different graph transformation approaches [11]. This paper explains how to apply the theory of confluence and critical pairs, developed for graph transformation in the double pushout approach (DPO) [6]. Therefore we repeat some main definitions.

Definition A.1 [graph transformation system] A graph transformation system consists of a set of rules \( (p : L \leftarrow K \rightarrow R)_{p \in \mathcal{P}} \) with \( \mathcal{P} \) the set of rule names. Given a rule \( p : L \leftarrow K \rightarrow R \) and a graph \( G \), one can try to apply \( p \) to \( G \) if there is an occurrence of \( L \) in \( G \) i.e. an injective graph morphism, called match \( m : L \rightarrow G \). Remark: In general a match doesn’t have to be injective. Here we restrict to injective matches. Given a graph \( G \), a rule \( p : L \leftarrow K \rightarrow R \) and a match \( m : L \rightarrow G \), a direct graph transformation from \( G \) to \( H \) using \( p \) exists if and only if the double pushout (DPO) diagram

![DPO Diagram](image)

can be constructed. In this case we write \( G \xrightarrow{p,m} H \). Since pushouts in Graph always exist, the DPO can be constructed if the pushout complement of \( K \rightarrow L \rightarrow G \) exists. If so, we say that, the match \( m \) satisfies the gluing condition of rule \( p \). Note, that since a match in this paper is injective, the identification condition is always fulfilled. A graph transformation for a graph transformation system \( \mathcal{G} \) is a sequence of direct graph transformations \( G_{i-1} \xrightarrow{p_i} G_i \), with \( p_i \) a rule in \( \mathcal{G} (i = 1, \cdots, n) \), where for \( n = 0 \) we have the identical transformation of \( G_0 \).

Given a graph \( G \), we may have several rules that can be applied to \( G \). However, this situation is not necessarily a conflictive one. In particular if we have two rules \( p_1 : L_1 \leftarrow K_1 \rightarrow R_1 \) and \( p_2 : L_2 \leftarrow K_2 \rightarrow R_2 \) such that they can both be applied to \( G \) via the matches \( m_1 \) and \( m_2 \), the situation is not a conflict if, after applying any of the rules, we can still apply the other one, i.e. if the transformation defined by the former does not destroy the application of the latter. The following definitions characterize this situation:

Definition A.2 [parallel independence] Two direct transformations \( G \xrightarrow{(p_1,m_1)} H_1 \) and \( G \xrightarrow{(p_2,m_2)} H_2 \) are parallel independent if \( m_1(L_1) \cap m_2(L_2) \subseteq m_1(l_1(K_1)) \cap m_2(l_2(K_2)) \)
This condition can be expressed categorically in the following way:
\[ \exists h_1 : L_1 \rightarrow D_2 : d_2 \circ h_1 = m_1 \wedge \exists h_2 : L_2 \rightarrow D_1 : d_1 \circ h_2 = m_2 \]

\[
\begin{array}{c}
R_1 \xrightarrow{h_1} K_1 \xrightarrow{b_1} L_1 \xrightarrow{h_2} K_2 \xrightarrow{b_2} R_2 \\
H_1 \xleftarrow{d_1} D_1 \xleftarrow{d_2} G \xleftarrow{d_2} D_2 \xleftarrow{d_2} H_2
\end{array}
\]

**Definition A.3** [conflict] Two direct transformations \( G^{(p_1,m_1)} \Rightarrow H_1 \) and \( G^{(p_2,m_2)} \Rightarrow H_2 \) are in conflict if they are not parallel independent. **Remark:** This type of conflict is also called delete-use-conflict. In particular rule \( p_2 \) deletes something, what \( p_1 \) uses if \( m_1(L_1) \cap m_2(L_2) \nsubseteq m_2(l_2(K_2)) \) and/or \( p_2 \) deletes something, what \( p_1 \) uses if \( m_1(L_1) \cap m_2(L_2) \nsubseteq m_1(l_1(K_1)) \).

A conflict situation in a minimal context can be characterized by the notion of critical pair:

**Definition A.4** [critical pair] A critical pair for the pair of rules \((p_1, p_2)\) is a pair of direct transformations \( K^{(p_1,m_1)} \Rightarrow P_1 \) and \( K^{(p_2,m_2)} \Rightarrow P_2 \) in conflict, s.t. \( m_1 \) and \( m_2 \) are jointly surjective morphisms.

\[
\begin{array}{c}
R_1 \xrightarrow{h_1} K_1 \xrightarrow{b_1} L_1 \xrightarrow{h_2} K_2 \xrightarrow{b_2} R_2 \\
P_1 \xleftarrow{d_1} D_1 \xleftarrow{d_2} K \xleftarrow{d_2} D_2 \xleftarrow{d_2} P_2
\end{array}
\]

The context is minimal, because \( m_1 \) and \( m_2 \) are required to be jointly surjective morphisms or so-called overlappings.

Two notions that are important for the rest of the paper are the concepts of boundary and context introduced in [5]:

**Definition A.5** [boundary - context] The boundary \( B \) of an injective graph morphism \( f : A \rightarrow A' \) consists of all nodes \( a \in A \) such that \( f(a) \) is adjacent to an edge in \( A' \setminus f(A) \). The context \( C = A' \setminus f(A) \cup f(b(B)) \) can be glued to \( A \) over the boundary \( B \) obtaining the pushout object \( A' \). This situation is expressed by the following pushout, called boundary pushout with \( b \) and \( g \) graph inclusions.

\[
\begin{array}{c}
B \xrightarrow{b} A \\
C \xrightarrow{g} A'
\end{array}
\]

**Remark:** As described in [5] the boundary pushout is an initial pushout.

**B Proofs of Theorem 2.2: Characterization Conflict**

Given a pair of direct transformations \( H_1^{p_1,m_1} \Rightarrow G^{p_2,m_2} \Rightarrow H_2 \) with \((S_1,o_1 : S_1 \rightarrow C_1, q_1 : S_1 \rightarrow L_2)\) the pullback of \((m_1 \circ g_1, m_2)\) and \((S_2, q_2 : S_2 \rightarrow L_1, o_2 : \]

12
Given \((S_2 \rightarrow C_2)\) the pullback of \((m_2, m_1 \circ g_1)\) then the following equivalence holds:

\[
H_1 \overset{p_1, m_1}{\iff} G \overset{p_2, m_2}{\implies} H_2 \text{ are in conflict}
\]

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

**Remark:** The proof of this Theorem is given in the context of AHLR systems [6], which implies that the conflict characterization proposed here holds not only for graph transformation systems as introduced in this paper, but for AHLR systems with matches that are monomorphisms.

**Proof.**

- Given \((S_2, q_{21}, o_2) \lor (S_1, o_1, q_{12})\) satisfying the conflict condition. It remains to show that \(H_1 \overset{p_1, m_1}{\iff} G \overset{p_2, m_2}{\implies} H_2\) are in conflict. Assume parallel independence, than we have \(m'_1 : L_1 \rightarrow D_2\) with \(d_2 \circ m'_1 = m_1\) and \(m'_2 : L_2 \rightarrow D_1\) with \(d_1 \circ m'_2 = m_2\). It suffices to construct \(s_2 : S_2 \rightarrow B_2 \in M\) (resp. \(s_1 : S_1 \rightarrow B_1 \in M\)) with \(c_2 \circ s_2 = o_2\) (resp. \(c_1 \circ s_1 = o_1\)) which violates the conflict condition for \((S_2, q_{21}, o_2)\) (resp. \((S_1, o_1, q_{12})\)).

- Given \((S_2, q_{21}, o_2)\) satisfying the conflict condition. It remains to show parallel independence of \(H_1 \overset{p_1, m_1}{\iff} G \overset{p_2, m_2}{\implies} H_2\). It suffices to construct \(s_2 : S_2 \rightarrow B_2 \in M\) (resp. \(s_1 : S_1 \rightarrow B_1 \in M\)) with \(c_2 \circ s_2 = o_2\) and \(k_2 \circ b_2 \circ s_2 = m'_1 \circ q_{21}\). Now \(o_2, c_2 \in M\) implies \(s_2 \in M\), because \(M\) is closed under decomposition.

Analogously we can construct \(s_1 : S_1 \rightarrow B_1 \in M\) s.t. \(c_1 \circ s_1 = o_1\). This is a contradiction and proves, that \(H_1 \overset{p_1, m_1}{\iff} G \overset{p_2, m_2}{\implies} H_2\) are in conflict.

- Given \(P_1 \overset{p_1, m_1}{\iff} K \overset{p_2, m_2}{\implies} P_2\) in conflict and neither \((S_1, o_1, q_{12})\) nor \((S_2, q_{21}, o_2)\) satisfy the conflict condition. Then there exists a morphism \(s_1 : S_1 \rightarrow B_1 \in M\) with \(c_1 \circ s_1 = o_1\) and there exists a morphism \(s_2 : S_2 \rightarrow B_2 \in M\) with \(c_2 \circ s_2 = o_2\). It suffices to show parallel independence of \(H_1 \overset{p_1, m_1}{\iff} G \overset{p_2, m_2}{\implies} H_2\). This is equivalent to constructing \(m'_1 : L_1 \rightarrow D_2\) with \(d_2 \circ m'_1 = m_1\) and
$m'_2 : L_2 \to D_1$ with $d_1 \circ m'_2$. We begin with the construction of $m'_1$. Therefore consider the following picture:

Let $(P, x_1, x_2)$ be the pullback of $(G, m_1, d_2)$ in the front square. The left square is a pullback by construction. The back square is a pullback because $c_2$ is a monomorphism. The front pullback leads to a unique morphism $x_3 : S_2 \to P$ s.t. $q_{21} = x_1 \circ x_3$ and $x_2 \circ x_3 = k_2 \circ b_2 \circ s_2$. The top square is a pullback because $x_1$ is a monomorphism. The bottom square is a pushout by construction and hence also pullback. This implies by pullback composition and decomposition that also the right square is a pullback. Now the Van Kampen property with bottom pushout and $c_2 \in M$ implies that the top is a pushout as well. This implies $x_1$ is an isomorphism. Now let $m'_1 = x_2 \circ (x_1)^{-1} : L_1 \to D_2$, then $d_2 \circ m'_1 = d_2 \circ x_2 \circ (x_1)^{-1} = m_1 \circ x_1 \circ (x_1)^{-1} = m_1$. Analogously we can construct $m'_2$. This is a contradiction since $H_1 \xleftarrow{p_1,m_1} G \xrightarrow{p_2,m_2} H_2$ are direct transformations in conflict.

\begin{proof}
For each critical pair $P' \xleftarrow{p_1,m'_1} K' \xrightarrow{p_2,m'_2} P'$ there exists an essential critical pair $P_1 \xleftarrow{p_1,m_1} K \xrightarrow{p_2,m_2} P_2$ with extension diagrams (1) and (2).

\[
\begin{array}{c}
\xymatrix{ P_1 \ar[r] & K \ar[r] & P_2 \\
(1) \ar[d] & m \ar[d] & (2) \\
P'_1 \ar[r] & K' \ar[r] & P'_2 }
\end{array}
\]

\textbf{Remark:} $m : K \to K'$ is an epimorphism, but not necessarily a monomorphism. The proof of this theorem is not given in the context of AHLR systems \cite{6}, thus it is not known yet if the completeness property of essential critical pairs can be generalized from graph transformation to AHLR systems. This is subject of ongoing work.

\end{proof}
Proof.

\[
\begin{array}{c}
R_1 \xrightarrow{r_1} K_1 \xrightarrow{l_1} L_1 \\
\downarrow k_1 \downarrow m_1' \downarrow m_1 \\
P'_1 \xrightarrow{e_1'} D_1' \xrightarrow{d_1'} K' \xrightarrow{d_2'} D'_2 \xrightarrow{e_2'} P'_2 \\
L_2 \xleftarrow{l_2} K_2 \xleftarrow{r_2} R_2 \\
\end{array}
\]

Since a critical pair is in particular a pair of direct transformations in conflict according to Theorem 2.2 one of the following cases occurs:

(i) \((S_1, o_1, q_{12})\) satisfies and \((S_2, q_{21}, o_2)\) doesn’t satisfy the conflict condition (asymmetrical conflict)

(ii) \((S_1, o_1, q_{12})\) doesn’t satisfy and \((S_2, q_{21}, o_2)\) satisfies the conflict condition (asymmetrical conflict)

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

with \((S_1, o_1 : S_1 \to C_2, q_{12} : S_1 \to L_2)\) the pullback of \((m_1 \circ g_1, m_2)\), \((S_2, q_{21} : S_2 \to L_1, o_2 : S_2 \to C_2)\) the pullback of \((m_1, m_2 \circ g_1)\) and \((S_1, s_1, s_2)\) constructed out of \((S_1, o_1, q_{12})\) and \((S_2, q_{21}, o_2)\).

(i) Analog to the following case.

(ii) Construct the pushout \((9) (K, m_1 : L_1 \to K, m_2 : L_2 \to K)\) of the conflict reason span \((S_2, q_{21}, g_2 \circ o_2)\).

Since \((9)\) is a pushout and \(m'_1 \circ q_{21} = m'_2 \circ g_2 \circ o_2\) a unique morphism \(m : K \to K'\) exists such that \(m'_1 = m \circ m_1\) and \(m'_2 = m \circ m_2\). Now we can construct the pullback \((4) (D_1, d_1 : D_1 \to K, a'_1 \circ D_1 \to D_1')\) of \((d'_1, m)\) and the pullback \((7) (D_2, d_2 : D_1 \to K, a'_2 \circ D_2 \to D_2')\) of \((d'_2, m)\). Since \((4)\) (resp. \((7)\)) is a pullback and \(d'_1 \circ k_1 = m \circ m_1 \circ l_1\) (resp. \(d'_2 \circ k_2 = m \circ m_2 \circ l_2\)) a morphism \(k_1 : K_1 \to D_1\) resp. \(k_2 : K_2 \to D_2\) exists s.t. \(a'_1 \circ k_1 = k'_1\) (resp. \(a'_2 \circ k_2 = k'_2\)) and \(d_1 \circ k_1 = m_1 \circ l_1\) (resp. \(d_2 \circ k_2 = m_2 \circ l_2\)). Now we prove, that \((d_1, m_1)\) and \((d_2, m_2)\) are jointly surjective.

(a) We start with proving that \((d_1, m_1)\) is jointly surjective. Since \((m_1, m_2)\) are jointly surjective \(K\) can also be written as \(K = K_{L_2 \setminus L_1} \cup K_{L_1}\) with \(K_{L_2 \setminus L_1} = m_2(L_2) \setminus m_1(L_1)\) and \(K_{L_1} = m_1(L_1)\). For all \(x \in K_{L_2 \setminus L_1}\) we have to prove that they have a preimage in \(D_1\). So we assume,
that $\exists y_2 \in L_2 : m_2(y_2) = x$ and $\exists y_1 \in L_1 : m_1(y_1) = x$. Since $(m'_1, m'_2)$ are also jointly surjective $K' = K'_{L_2 \setminus L_1} \cup K'_{L_1}$ with $K'_{L_2 \setminus L_1} = m'_2(L_2) \setminus m'_1(L_1)$ and $K'_{L_1} = m'_1(L_1)$.

- $m(x) \in K'_{L_2 \setminus L_1}$ implies that $m(x)$ doesn’t have a preimage in $L_1$.
  But since $(m'_1, d'_1)$ jointly surjective, $\exists x_1 \in D'_1 : d'_1(x_1) = m(x)$ and since (4) is a pullback we have $(x_1, x) \in D_1$ with $d_1(x_1, x) = x$.

- $m(x) \in K'_{L_1}$ implies that there exists an $y_1 \in L_1$ s.t. $m'_1(y_1) = m(x)$.
  Now again we distinguish two cases:
  - Let $y_1 \in L_1 \setminus C_1$. Then $\exists x_1 \in K_1 : l_1(x_1) = y_1$ and $m(x) = m'_2(l_1(x)) = m(m_1(l_1(x))) = d'_1(a'_1(k_1(x_1)))$. This implies that $(a'_1(k_1(x_1)), x) \in D_1$ with $d_1(a'_1(k_1(x_1)), x) = x$, since (4) is a pullback.
  - On the other hand, if $y_1 \in C_1$ then $y_1 = g_1(y_1)$.

Moreover since $\exists y_2 \in L_2 : m_2(y_2) = x$ it follows that $m'_2(y_2) = m(m_2(y_2)) = m(x)$. Then $m'_1(g_1(y_1)) = m'_1(y_1) = m(x) = m'_2(y_2)$ and since $(S_1, o_1, q_1)$ is a pullback of $(m'_1 \circ g_1, m'_2)$ it follows that $(y_1, y_2) \in S_1$ with $o_1(y_1, y_2) = y_1$. Since $(S_1, o_1, q_1)$ doesn’t satisfy the conflict condition there exists an $s_1 : S_1 \to B_1$ s.t. $o_1 = c_1 \circ s_1$. This implies $y_1 = o_1(y_1, y_2) = c_1(s_1(y_1, y_2))$ with $s_1(y_1, y_2) \in B_1$.

Because of the initial pushout over $l_1 : K_1 \to L_1$ we know that $y_1 = g_1(y_1) = g_1(c_1(s_1(y_1, y_2))) = g_1(c_1(y_1)) = l_1(b_1(y_1)) = l_1(y_1)$ since $b_1$ is an inclusion. Thus $l_1(y_1) = y_1$ with $y_1 \in K_1$. Since $m(x) = m'_2(y_2) = m'_1(y_1) = m(m_1(l_1(y_1))) = d'_1(a'_1(k_1(y_1)))$ we know that $(a'_1(k_1(y_1)), x) \in D_1$ with $d_1(a'_1(k_1(y_1)), x) = x$ since (4) is a pullback.

(b) Now we prove, that $(d_2, m_2)$ are jointly surjective. Since $(m_1, m_2)$ are jointly surjective $K = K_{L_1 \setminus L_2} \cup K_{L_2}$. It suffices, to show that for each $x \in K_{L_1 \setminus L_2}$ there exists $y_2 \in D_2 : d_2(y_2) = x$. So we assume, that $\exists y_1 \in L_1 : m_1(y_1) = x$ and $\exists y_2 \in L_2 : m_2(y_2) = x$. Since also $(m'_1, m'_2)$ are jointly surjective $K' = K'_{L_1 \setminus L_2} \cup K'_{L_2}$ we distinguish the following two cases:

- $m(x) \in K'_{L_1 \setminus L_2}$ implies, that $m(x)$ doesn’t have a preimage in $L_2$.
  Since $(m'_2, d'_2)$ are jointly surjective then there exists $x_2 \in D'_2 : d'_2(x_2) = m(x)$. Because (7) is a pullback $(x_2, x) \in D_2$ with $d_2(x_2, x) = x$.

- $m(x) \in K'_{L_2}$ implies, that there exists $y_2 \in L_2 : m'_2(y_2) = m(x)$.

16
Now again we distinguish two cases:

- Let \( y_2 \in L_2 \setminus C_2 \). Then \( \exists x_2 \in K_2 : l_2(x_2) = y_2 \) and \( m(x) = m_2'(y_2) = m_2'(l_2(x_2)) = m(m_2(l_2(x_2))) = d'_2(a'_2(k_2(x_2))) \). This implies that \( (a'_2(k_2(x_2)), x) \in D_2 \) with \( d_2(a'_2(k_2(x_2), x) = x, \) since (7) is a pullback.

- On the other hand, if \( y_2 \in C_2 \) then \( g_2(y_2) = y_2 \) and \( m_2'(g_2(y_2)) = m_2'(y_2) = m(x) = m(m_1(y_1)) = m_1'(y_1) \) and since \((S_2, q_{21}, o_2)\) is a pullback of \( (m'_1, m_2' \circ g_2) \) it follows that \( (y_1, y_2) \in S_2 \). But since (9) is a pushout this implies \( m_1(y_1) = m_1(q_{21}(y_1, y_2)) = m_2(g_2(o_2(y_1, y_2))) = m_2(y_2) \). This is a contradiction since now \( m_1(y_1) = x = m_2(y_2) \), but \( \not\exists y_2 \in L_2 : m_2(y_2) = x \).

(iii) If both \((S_1, o_1, q_{12})\) and \((S_2, q_{21}, o_2)\) satisfy the conflict condition at first we construct the conflict reason span \((S, s_1, s_2)\).

Remember that for this construction \( g_1 \circ o_1 = s_1 \circ s_1', q_{21} = s_1 \circ s_2, q_{12} = s_2 \circ s_1' \) and \( g_2 \circ o_2 = s_2 \circ s_2' \). Then we construct the pushout \((K, m_1, m_2)\) of the conflict reason span \((S, s_1, s_2)\). Since \( m'_1 \circ s_1 \circ s_1' = m'_1 \circ o_1 = m'_2 \circ q_{12} = m'_2 \circ s_2 \circ s_1' \) and \( m'_1 \circ o_1 \circ s_2 = m'_1 \circ q_{21} = m'_2 \circ q_2 \circ o_2 = m'_2 \circ s_2 \circ s_2' \) and \((s_1', s_2')\) jointly surjective we can conclude that \( m'_1 \circ s_1 = m'_2 \circ s_2 \). Because of the pushout property of (9) then \( \exists m : K \to K' \) s.t. \( m \circ m_1 = m'_1 \) and \( m \circ m_2 = m'_2 \). In the same way as in case (i) and (ii) we build pushouts (4) and (7) and get morphisms \( k_1 : K_1 \to D_1 \) and \( k_2 : K_2 \to D_2 \). Now we prove, that \((d_1, m_1)\) and \((d_2, m_2)\) are jointly surjective.

(a) We start proving, that \((d_1, m_1)\) are jointly surjective. Since \((m_1, m_2)\) are jointly surjective we can assume that \( K = K_{L_1} \cup K_{L_2 \setminus L_1} \). It suffices to show, that for each \( x \in K_{L_2 \setminus L_1} : \exists y_1 : d_1(y_1) = x \). Thus we assume that \( \exists y_2 \in L_2 : m_2(y_2) = x \) and \( \not\exists y_1 \in L_1 : m_1(y_1) = x \). Since \((m'_1, m'_2)\) are also jointly surjective \( K' = K'_{L_1} \cup K'_{L_2 \setminus L_1} \). Now we distinguish two cases:

- Let \( m(x) \in K'_{L_2 \setminus L_1} \), then \( \not\exists y_1 \in L_1 : m'_1(y_1) = m(x) \). But since
(m'_1, d'_1) are jointly surjective then \( \exists x_1 \in D'_1 : m(x) = d'_1(x_1) \). Since (4) is a pullback \((x_1, x) \in D_1 \) with \( d_1(x_1, x) = x \).

- Let \( m(x) \in K'_{L_1} \). Then \( \exists y_1 \in L_1 : m'_1(y_1) = m(x) \). Now we distinguish two cases:
  - Let \( y_1 \in L_1 \setminus L_1 \) then \( \exists x_1 \in K_1 \) with \( l_1(x_1) = y_1 \). Since \( m(x) = m'_1(y_1) = m'_1(l_1(x_1)) = d'_1(a'_1(k_1(x_1))) \) we have found an \( (a'_1(k_1(x_1)), x) \in D_1 \) with \( d_1(a'_1(k_1(x_1)), x) = x \).
  - Let \( y_1 \in C_1 \), then \( y_1 = g_1(y_1) \) since \( g_1 \) is an inclusion. Then \( m'_1(g_1(y_1)) = m'_1(y_1) = m(x) = m(m_2(y_2)) = m'_2(y_2) \) and because of \((S_1, o_1, q_{12})\) pullback of \((m'_1 \circ g_1, m'_2)\), \((y_1, y_2) \in S_1 \Rightarrow (y_1, y_2) \in S \). This implies since \((9)\) is a pushout that \( m_1(y_1) = m_2(y_2) = x \) which is a contradiction since \( \overline{\exists} y_1 \in L_1 : m_1(y_1) = x \).

(b) Analogously we can prove, that \((d_2, m_2)\) is jointly surjective.

Now we know that if we build \((K, m_1, m_2)\) as a pushout of the conflict reason span of \( P'_1 \overset{p_1\ast m'_1}{\hookleftarrow} K' \overset{p_2\ast m'_2}{\twoheadrightarrow} P'_2 \), \((d_1, m_1)\) and \((d_2, m_2)\) are jointly surjective. Now we can conclude that (2) (resp. (5)) is a pullback since \((d_1, m_1)\) (resp.\((d_2, m_2)\)) are jointly surjective, (4) (resp. (7)) is a pullback and (2)+(4) (resp. (5)+(7)) is a pushout and also a pullback. Since \( l_1 \) (resp. \( l_2 \)) is injective, (2)+(4) (resp. (5)+(7)) is a pushout and (2), (4) (resp. (5), (7)) are pullbacks, this implies that (2), (4) (resp. (5), (7)) are also pushouts. Than we can construct \( P_1 \) and \( P_2 \) as pushouts of \((r_1, k_1)\) resp. \((r_2, k_2)\) and because of the pushout property the two lacking morphisms \( f_1 \) and \( f_2 \) for the essential critical pair are constructed. Because of pushout-pushout decomposition now we can deduce that (1), (3), (6) and (8) are pushouts. Now we know that \( P_1 \overset{p_1\ast m_1}{\hookleftarrow} K \overset{p_2\ast m_2}{\twoheadrightarrow} P_2 \) is a pair of direct transformations.

Since \((K, m_1, m_2)\) was constructed as a pushout over the conflict reason span of \( P'_1 \overset{p_1\ast m'_1}{\hookleftarrow} K' \overset{p_2\ast m'_2}{\twoheadrightarrow} P'_2 \) we still have to prove that the conflict reason span stays the same for \( P_1 \overset{p_1\ast m_1}{\hookleftarrow} K \overset{p_2\ast m_2}{\twoheadrightarrow} P_2 \). Therefore at first, we have to show that if \((S_2, q_{21}, o_2)\) is a pullback of \((m'_1, m'_2 \circ g_2)\) then it is also a pullback of \((m_1, m_2 \circ g_2)\).
Since (9) is a pushout. Moreover if we take another graph $X$ and morphisms $x_1 : X \to L_1$ and $x_2 : X \to L_2$ such that $m_1 \circ x_1 = m_2 \circ g_2 \circ x_2$, this implies $m_1' \circ x_1 = m \circ m_1 \circ x_1 = m \circ m_2 \circ g_2 \circ x_2 = m_2' \circ g_2 \circ x_2$ and because of the pullback property of the outer pullback a unique morphism $x : X \to S_2$ exists s.t. $q_2 \circ x = x_1$ and $o_2 \circ x = x_2$. Thus $(S_2, q_2, o_2)$ is also a pullback of $(m_1, m_2 \circ g_2)$. Analogously we can prove, that $(S_1, q_1, o_1)$ is also a pullback of $(m_1 \circ g_1, m_2)$. Since the conflict reason span $(S, s_1, s_2)$ is constructed from $(S_2, q_2, o_2)$ and $(S_1, q_1, o_1)$ and since further on the satisfaction of the conflict condition only depends on the structure of the rules we know that the conflict reason span $(S_1, g_1 \circ o_1, q_1)$, $(S_2, q_2, g_2 \circ o_2)$ or $(S, s_1, s_2)$ stays the same for $P_1 \xleftarrow{p_1 \cdot m_1} K \xrightarrow{p_2 \cdot m_2} P_2$ which implies by Theorem 2.2 that also $P_1 \xleftarrow{p_1 \cdot m_1} K \xrightarrow{p_2 \cdot m_2} P_2$ is in conflict. Now we have constructed an essential critical pair $P_1 \xleftarrow{p_1 \cdot m_1} K \xrightarrow{p_2 \cdot m_2} P_2$ with extension morphism $m : K \to K'$. The essential critical pair $P_1 \xleftarrow{p_1 \cdot m_1} K \xrightarrow{p_2 \cdot m_2} P_2$ is unique up to isomorphism, since $(K, m_1, m_2)$ is the pushout of conflict reason span $(S_1, g_1 \circ o_1, q_1)$ for case (i), $(S_2, q_2, g_2 \circ o_2)$ for case (ii) and $(S, s_1, s_2)$ for case (iii) and a pushout construction is unique up to isomorphism.

D Proof of Theorem 4.3 (Essential Critical Pair Lemma)

If all essential critical pairs of a graph transformation system are strictly confluent, then this graph transformation system is locally confluent.

Proof. In the proof of the critical pair lemma in [6] for AHLR systems it is demanded that the extension morphism $m$ belongs to a special subset of monomorphisms $M$. For the proof of this lemma though it is sufficient to demand the existence of an initial pushout over the extension morphism $m$. In the case of essential critical pairs $m$ is not necessarily an injective morphism, as shown in Theorem 4.1, but an initial pushout over a non-injective morphism $m$ in the AHLR-category Graph always exists. Therefore the proof of the critical pair lemma can be repeated restricting the set of critical pairs to the set of essential critical pairs.

\[ \square \]