Electronic Communications of the EASST
Volume 26 (2010)

Manipulation of Graphs, Algebras and Pictures

Essays Dedicated to Hans-Jörg Kreowski
on the Occasion of His 60th Birthday

Checking Graph-Transformation Systems for Confluence

Detlef Plump

15 pages

Guest Editors: Frank Drewes, Annegret Habel, Berthold Hoffmann, Detlef Plump
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Checking Graph-Transformation Systems for Confluence

Detlef Plump

The University of York, UK

Abstract: In general, it is undecidable whether a terminating graph-transformation system is confluent or not. We introduce the class of *coverable* hypergraph-transformation systems and show that confluence is decidable for coverable systems that are terminating. Intuitively, a system is coverable if its typing allows to extend each critical pair with a non-deletable context that uniquely identifies the persistent nodes of the pair. The class of coverable systems includes all hypergraph-transformation systems in which hyperedges can connect arbitrary sequences of nodes, and all graph-transformation systems with a sufficient number of unused edge labels.

Keywords: Confluence, graph transformation, coverable systems

1 Introduction

Confluent sets of graph-transformation rules can be executed without backtracking since all terminating derivations produce the same result for a given input graph. Applications of confluence include the efficient recognition of graph classes by graph reduction [ACPS93, BF01, BPR04], the parsing of languages defined by graph grammars [FKZ76, RS97], and the deterministic input/output behaviour of programs in graph-transformation languages such as AGG [Tae04], FUJABA [NNZ00], GrGen [GBG$^+$06] or GP [Plu09].

In the settings of string and term rewriting, confluence is decidable for terminating systems [BO93, BN98, BKV03]: one computes all *critical pairs* $t \leftarrow s \rightarrow u$ of rewrite steps and checks whether t and u are joinable in that they reduce to a common string resp. term. In contrast, confluence is undecidable in general for terminating graph-transformation systems [Plu05]. The problem is, in brief, that the joinability of all critical pairs need not imply confluence of a system. To guarantee confluence, one has to impose extra conditions on the joining derivations, leading to the notion of a strongly joinable critical pair. However, strong joinability of all critical pairs is not a necessary condition for confluence and hence, in general, cannot be used to decide confluence.

In this paper, we introduce *coverable* hypergraph-transformation systems and show that confluence is decidable for coverable systems that are terminating. Intuitively, a system is coverable if its typing allows to extend each critical pair with a non-deletable context—a cover—that uniquely identifies the persistent nodes of the pair. We give a decision procedure for confluence that processes each extended critical pair $\widehat{\Gamma}\colon \widehat{U}_1 \Leftarrow \widehat{S} \Rightarrow \widehat{U}_2$ by reducing $\widehat{U}_1$ and $\widehat{U}_2$ to normal forms X_1 and X_2, and checking whether X_1 and X_2 are isomorphic. If this is the case, then the critical pair underlying $\widehat{\Gamma}$ is strongly joinable; otherwise, a counterexample to confluence has been found.

Roughly speaking, a cover for a critical pair can be constructed if the signature of the hypergraph-transformation system under consideration contains (hyper-)edge labels that do not occur in rules and that can be used to connect the persistent nodes of the critical pair by edges. Such a cover cannot be deleted by rules. Moreover, there must be a unique surjective morphism from the cover to each of its images under a graph morphism. We give different conditions under which covers can be constructed and show, in particular, that the class of coverable systems includes all hypergraph-transformation systems in which hyperedges can connect arbitrary sequences of nodes.

The rest of this paper is organised as follows. The next section recalls some terminology for binary relations and defines hypergraphs and their morphisms. Section 3 reviews the double-pushout approach to (hyper-)graph transformation in a setting where rules are matched injectively and can have non-injective right-hand morphisms. We define confluence of hypergraph-transformation systems and recall the fact that confluence is undecidable for terminating systems. In Section 4 we review the role of critical pairs in establishing confluence. Section 5 introduces covers for critical pairs and coverable systems, discusses our main result and the associated decision procedure for confluence, and presents special cases where confluence is decidable. In Section 6, we conclude and discuss a topic for future work.

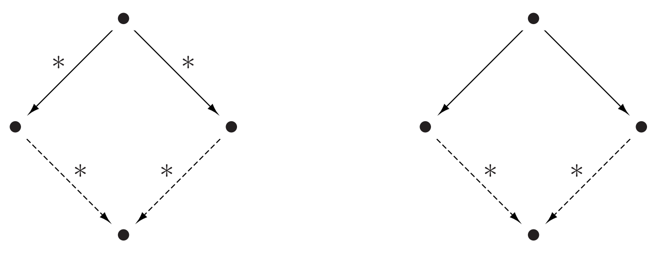
2 Preliminaries

We recall some terminology for binary relations (consistent with [BN98, BKV03]) and define hypergraphs and their morphisms.

2.1 Relations

Let $\rightarrow$ be a binary relation on a set A. The inverse relation of $\rightarrow$ is denoted by $\leftarrow$. We write $\rightarrow^+$ for the transitive closure of $\rightarrow$ and $\rightarrow^*$ for the transitive-reflexive closure of $\rightarrow$. Two elements $a,b \in A$ have a *common reduct* if $a \rightarrow^* c \leftarrow^* b$ for some c. If $a \rightarrow^* c$ and there is no d such that $c \rightarrow d$, then d is a *normal form* of a.

The relation $\rightarrow$ is (1) *terminating* if there is no infinite sequence $a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow \ldots$, (2) *confluent* if for all a, b and c with $b \leftarrow^* a \rightarrow^* c$, elements b and c have a common reduct (see Figure 1(a)), (3) *locally confluent* if for all a, b and c with $b \leftarrow a \rightarrow c$, elements b and c have a common reduct (see Figure 1(b)).



(a) confluence (b) local confluence

Figure 1: Confluence properties

By the following well-known result, local confluence and confluence are equivalent in the presence of termination.

Lemma 1 (Newman's Lemma [New42]) *A terminating relation is confluent if and only if it is locally confluent.*

2.2 Hypergraphs

We deal with directed, labelled hypergraphs and use a simple type system where the label of a hyperedge restricts the number of incident nodes and their labels. A *signature* $\Sigma = \langle \Sigma_\mathrm{V}, \Sigma_\mathrm{E}, \mathrm{Type}\rangle$ consists of a set Σ_V of *node labels*, a set Σ_E of *hyperedge labels* and a mapping Type assigning to each $l \in \Sigma_\mathrm{E}$ a set $\mathrm{Type}(l) \subseteq \Sigma_\mathrm{V}^*$. Unless stated otherwise, we denote by Σ an arbitrary but fixed signature over which all hypergraphs are labelled.

A *hypergraph* over Σ is a system $G = \langle \mathrm{V}_G, \mathrm{E}_G, \mathrm{mark}_G, \mathrm{lab}_G, \mathrm{att}_G\rangle$ consisting of two finite sets V_G and E_G of *nodes* (or *vertices*) and *hyperedges*, two labelling functions $\mathrm{mark}_G\colon \mathrm{V}_G \to \Sigma_\mathrm{V}$ and $\mathrm{lab}_G\colon \mathrm{E}_G \to \Sigma_\mathrm{E}$, and an attachment function $\mathrm{att}_G\colon \mathrm{E}_G \to \mathrm{V}_G^*$ such that $\mathrm{mark}_G^*(\mathrm{att}_G(e)) \in \mathrm{Type}(\mathrm{lab}_G(e))$ for each hyperedge e. (The extension $f^*\colon A^* \to B^*$ of a function $f\colon A \to B$ maps the empty string to itself and $a_1 \dots a_n$ to $f(a_1)\dots f(a_n)$.) We write $\mathscr{H}(\Sigma)$ for the set of all hypergraphs over Σ.

In pictures, nodes and hyperedges are drawn as circles and boxes, respectively, with labels inside. Lines represent the attachment of hyperedges to nodes, where numbers specify the left-to-right order in the attachment string. For example, Figure 2 shows a hypergraph with four nodes (all labelled with $\bullet$) and three hyperedges (labelled with B and S).

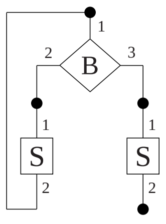


Figure 2: A hypergraph

A hypergraph G is a *graph* if each hyperedge e is an ordinary edge, that is, if $\mathrm{att}_G(e)$ has length two. Ordinary edges may be drawn as arrows with labels written next to them.

Given hypergraphs G and H, a *hypergraph morphism* (or *morphism* for short) $f\colon G \to H$ consists of two functions $f_\mathrm{V}\colon \mathrm{V}_G \to \mathrm{V}_H$ and $f_\mathrm{E}\colon \mathrm{E}_G \to \mathrm{E}_H$ that preserve labels and attachment to nodes, that is, $\mathrm{mark}_H \circ f_\mathrm{V} = \mathrm{mark}_G$, $\mathrm{lab}_H \circ f_\mathrm{E} = \mathrm{lab}_G$ and $\mathrm{att}_H \circ f_\mathrm{E} = f_\mathrm{V}^* \circ \mathrm{att}_G$. A morphism $incl\colon G \to H$ is an *inclusion* if $incl_\mathrm{V}(v) = v$ and $incl_\mathrm{E}(e) = e$ for all $v \in \mathrm{V}_G$ and $e \in \mathrm{E}_G$. In this case G is a *subhypergraph* of H which is denoted by $G \subseteq H$. Every morphism $f\colon G \to H$ induces a subhypergraph of H, denoted by $f(G)$, which has nodes $f_\mathrm{V}(\mathrm{V}_G)$ and hyperedges $f_\mathrm{E}(\mathrm{E}_G)$. Morphism f is *injective* (*surjective*) if f_V and f_E are injective (surjective). If f is surjective, then H is an *image* of G. If f is both injective and surjective, then it is an *isomorphism*. In this case G and H are *isomorphic*, which is denoted by $G \cong H$.

The *composition* of two morphisms $f\colon G \to H$ and $g\colon H \to M$ is the morphism $g \circ f\colon G \to M$ consisting of the composed functions $g_V \circ f_V$ and $g_E \circ f_E$. The composition is also written as $G \to H \to M$ if f and g are clear from the context.

A *partial hypergraph morphism* $f\colon G \to H$ is a hypergraph morphism $S \to H$ such that $S \subseteq G$. Here S is the *domain of definition* of f, denoted by $\mathrm{Dom}(f)$.

3 Graph Transformation

We briefly review the *double-pushout approach* to graph transformation. In our setting, rules are matched injectively and can have non-injective right-hand morphisms. (See [HMP01] for a comparison with other variants of the double-pushout approach.)

3.1 Rules and derivations

A *rule* $r\colon \langle L \leftarrow K \to R\rangle$ consists of two hypergraph morphisms with a common domain, where $K \to L$ is an inclusion. The hypergraphs L and R are the *left-* and *right-hand side* of r, and K is the *interface*. The rule is *injective* if the morphism $K \to R$ is injective.

Let G and H be hypergraphs, $r\colon \langle L \leftarrow K \to R\rangle$ a rule and $f\colon L \to G$ an injective morphism. Then G *directly derives* H by r and f, denoted by $G \Rightarrow_{r,f} H$, if there exist two pushouts as in Figure 3. Given a set of rules $\mathscr{R}$, we write $G \Rightarrow_{\mathscr{R}} H$ to express that there exist $r \in \mathscr{R}$ and a

$$\begin{tikzcd}
L \arrow[d, "f"'] & K \arrow[l] \arrow[r] \arrow[d] & R \arrow[d] \\
G & D \arrow[l] \arrow[r] & H
\end{tikzcd}$$

Figure 3: A double-pushout

morphism f such that $G \Rightarrow_{r,f} H$.

We refer to [Plu05] for the definition and construction of hypergraph pushouts. Intuitively, the left pushout corresponds to the construction of D from G by removing the items in $L - K$, and the right pushout to the construction of H from D by merging items according to $K \to R$ and adding the items in R that are not in the image of K.

A double-pushout as in Figure 3 is called a *direct derivation* from G to H and may be denoted by $G \Rightarrow_{r,f} H$ or just by $G \Rightarrow_r H$ or $G \Rightarrow H$. A *derivation* from G to H is a sequence of direct derivations $G = G_0 \Rightarrow \ldots \Rightarrow G_n = H$, $n \geq 0$, and may be denoted by $G \Rightarrow^* H$.

Given a rule $r\colon \langle L \leftarrow K \to R\rangle$, an injective morphism $f\colon L \to G$ satisfies the *dangling condition* if no hyperedge in $\mathrm{E}_G - f_E(\mathrm{E}_L)$ is incident to a node in $f_V(\mathrm{V}_L - \mathrm{V}_K)$. It can be shown that, given r and f, a direct derivation as in Figure 3 exists if and only if f satisfies the dangling condition [HMP01].

With every derivation $\Delta\colon G_0 \Rightarrow^* G_n$ a partial hypergraph morphism can be associated that tracks the items of G_0 through the derivation: this morphism is undefined for all items in G_0 that are removed by Δ at some stage, and maps all other items to the corresponding items in G_n.

Definition 1 (Track morphism) Given a direct derivation $G \Rightarrow H$ as in Figure 3, the *track morphism* $\mathrm{tr}_{G\Rightarrow H}\colon G \to H$ is the partial hypergraph morphism defined by

$$\mathrm{tr}_{G\Rightarrow H}(x) = \begin{cases} c'(c^{-1}(x)) & \text{if} \quad x \in c(D), \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Here $c\colon D \to G$ and $c'\colon D \to H$ are the morphisms in the lower row of Figure 3 and $c^{-1}\colon c(D) \to D$ maps each item $c(x)$ to x.

The track morphism of a derivation $\Delta\colon G_0 \Rightarrow^* G_n$ is defined by $\mathrm{tr}_\Delta = \mathrm{id}_{G_0}$ if $n = 0$ and $\mathrm{tr}_\Delta = \mathrm{tr}_{G_1\Rightarrow^* G_n} \circ \mathrm{tr}_{G_0\Rightarrow G_1}$ otherwise, where id_{G_0} is the identity morphism on G_0.

Definition 2 (Hypergraph-transformation system) A *hypergraph-transformation system* $\langle\Sigma, \mathscr{R}\rangle$ consists of a signature Σ and a set $\mathscr{R}$ of rules over Σ. The system is *injective* if all rules in $\mathscr{R}$ are injective. It is a *graph-transformation system* if for each label l in Σ_E, all strings in $\mathrm{Type}(l)$ are of length two.

As graph-transformation systems are special hypergraph-transformation systems, results for the latter also apply to the former. In particular, Theorem 2, Theorem 3 and Corollary 1 below hold for graph-transformation systems, too.

Example 1 Figure 4 shows hypergraph-transformation rules for reducing control-flow graphs (see also [Plu05]). The associated signature contains a single node label • and two hyperedge

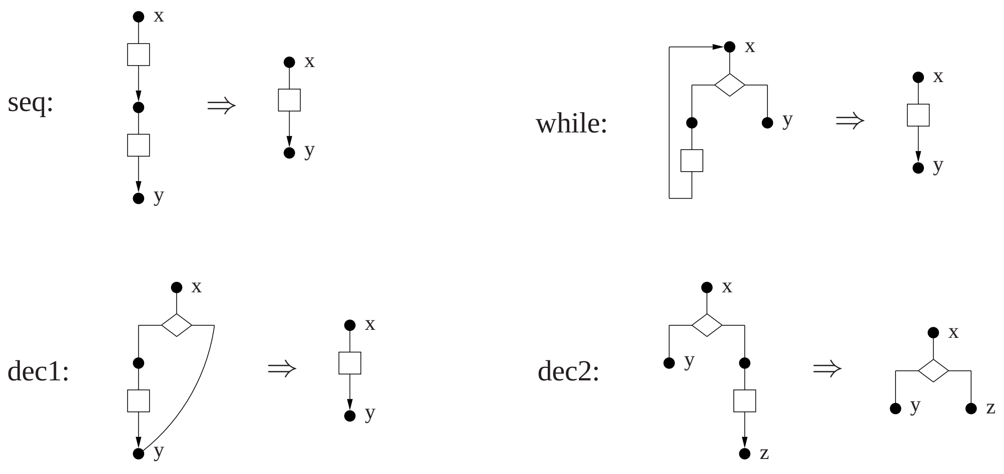


Figure 4: Hypergraph-transformation system for flow-graph reduction

labels which are graphically represented by hyperedges formed as squares and rhombs. Instead of using numbers to represent the attachment function, we use an arrow to point to the second attachment node of a square and define the order among the links of a rhomb to be "top-left-right". The rules are shown in a shorthand notation where only the left- and right-hand sides are depicted, the interface and the morphisms are implicitly given by the node names x,y,z. This example will be continued as Example 2, where it is shown that the system is confluent. □

3.2 Independence and confluence

Two direct derivations $H_1 \Leftarrow_{r_1} G \Rightarrow_{r_2} H_2$ do not interfere with each other if, roughly speaking, the intersection of the left-hand sides of r_1 and r_2 in G consists of common interface items. If one of the rules is not injective, however, an additional injectivity condition is needed. For $i = 1,2$, let r_i denote a rule $\langle L_i \leftarrow K_i \rightarrow R_i \rangle$.

Definition 3 (Independence) Direct derivations $H_1 \Leftarrow_{r_1} G \Rightarrow_{r_2} H_2$ as in Figure 5 are *independent* if there are morphisms $L_1 \rightarrow D_2$ and $L_2 \rightarrow D_1$ such that the following holds:
Commutativity: $L_1 \rightarrow D_2 \rightarrow G = L_1 \rightarrow G$ and $L_2 \rightarrow D_1 \rightarrow G = L_2 \rightarrow G$.
Injectivity: $L_1 \rightarrow D_2 \rightarrow H_2$ and $L_2 \rightarrow D_1 \rightarrow H_1$ are injective.

$$\begin{tikzcd}
R_1 \arrow[d] & K_1 \arrow[l] \arrow[r] \arrow[d] & L_1 \arrow[dr] \arrow[drrr, dashed] & & L_2 \arrow[dl] \arrow[dlll, dashed] & K_2 \arrow[l] \arrow[r] \arrow[d] & R_2 \arrow[d] \\
H_1 & D_1 \arrow[l] \arrow[rr] & & G & & D_2 \arrow[ll] \arrow[r] & H_2
\end{tikzcd}$$

Figure 5: Independent direct derivations

If r_1 and r_2 are injective, the direct derivations of Figure 5 are independent if and only if the intersection of the two left-hand sides coincides with the intersection of the two interfaces.

Lemma 2 (Independence for injective rules) *Let r_1 and r_2 be injective rules. Then direct derivations $H_1 \Leftarrow_{r_1,g_1} G \Rightarrow_{r_2,g_2} H_2$ are independent if and only if $g_1(L_1) \cap g_2(L_2) \subseteq g_1(K_1) \cap g_2(K_2)$.*

To define confluence and local confluence of hypergraph-transformation systems, we slightly relax the properties of Figure 1. Rather than require that converging $\Rightarrow_{\mathscr{R}}$-derivations must end in the *same* graph, we allow them to end in isomorphic graphs.

Definition 4 (Confluence of $\langle \Sigma, \mathscr{R} \rangle$) A hypergraph-transformation system $\langle \Sigma, \mathscr{R} \rangle$ is *confluent* (*locally confluent*) if for all $G, G_1, G_2 \in \mathscr{H}(\Sigma)$, $G_1 \Leftarrow^*_{\mathscr{R}} G \Rightarrow^*_{\mathscr{R}} G_2$ ($G_1 \Leftarrow_{\mathscr{R}} G \Rightarrow_{\mathscr{R}} G_2$) implies that there are $H_1, H_2 \in \mathscr{H}(\Sigma)$ such that $G_1 \Rightarrow^*_{\mathscr{R}} H_1 \cong H_2 \Leftarrow^*_{\mathscr{R}} G_2$.

This definition is equivalent to that in Subsection 2.1 as long as the converging derivations $G_1 \Rightarrow^*_{\mathscr{R}} H_1$ and $G_2 \Rightarrow^*_{\mathscr{R}} H_2$ do not both have length 0. This is because, by pushout properties, $A \Rightarrow_{\mathscr{R}} B \cong B'$ always implies $A \Rightarrow_{\mathscr{R}} B'$. If the converging derivations have length 0, however, we may have $G_1 \cong G_2$ without G_1 and G_2 being transformable into a common graph. It is natural to consider this still as confluence, because in (double-pushout) graph transformation the results of rule applications are unique only up to isomorphism.

This view on confluence can be substantiated by considering hypergraph transformation "up to isomorphism", that is, the transformation of isomorphism classes of hypergraphs. Given a hypergraph G, denote by $[G]$ the isomorphism class $\{G' \mid G' \cong G\}$. For a hypergraph-transformation system $\langle \Sigma, \mathscr{R} \rangle$, define the relation $\Rightarrow_{\mathscr{R},\cong}$ on isomorphism classes of hypergraphs over Σ by: $[G] \Rightarrow_{\mathscr{R},\cong} [H]$ if $G \Rightarrow_{\mathscr{R}} H$. (This is well-defined since $G' \cong G \Rightarrow_{\mathscr{R}} H \cong H'$ implies $G' \Rightarrow_{\mathscr{R}} H'$.) Then (local) confluence in the sense of Definition 4 is equivalent to (local) confluence of $\Rightarrow_{\mathscr{R},\cong}$ in the sense of Subsection 2.1, as shown by the next lemma.

Lemma 3 ([Plu05]) *A hypergraph-transformation system $\langle \Sigma, \mathscr{R} \rangle$ is confluent (locally confluent) if and only if the relation $\Rightarrow_{\mathscr{R},\cong}$ is confluent (locally confluent).*

A system $\langle \Sigma, \mathscr{R} \rangle$ is *terminating* if the relation $\Rightarrow_{\mathscr{R}}$ is terminating. The following result follows with Newman's Lemma.

Lemma 4 *A terminating hypergraph-transformation system is confluent if and only if it is locally confluent.*

Proof. The "only if" direction holds trivially, so assume that $\langle \Sigma, \mathscr{R} \rangle$ is terminating and locally confluent. Then $\Rightarrow_{\mathscr{R},\cong}$ is locally confluent by Lemma 3. Moreover, $\Rightarrow_{\mathscr{R},\cong}$ is terminating because $[G] \Rightarrow_{\mathscr{R},\cong} [H]$ if and only if $G \Rightarrow_{\mathscr{R}} H$. Thus, by Lemma 1, $\Rightarrow_{\mathscr{R},\cong}$ is confluent. Using Lemma 3 again shows that $\langle \Sigma, \mathscr{R} \rangle$ is confluent. □

In general, confluence is undecidable even for terminating graph-transformation systems. The precise result is as follows.

Theorem 1 ([Plu05]) *The following problem is undecidable in general:*

Instance: *An injective and terminating graph-transformation system $\langle \Sigma, \mathscr{R} \rangle$ such that Σ_V is a singleton and Σ_E and $\mathscr{R}$ are finite.*
Question: *Is $\langle \Sigma, \mathscr{R} \rangle$ confluent?*

Note that since graph-transformation systems are special hypergraph-transformation systems, the result also applies to the latter.

4 Critical Pairs

Critical pairs consist of direct derivations of minimal size that are not independent. We recall their definition from [Plu93, Plu05].

Definition 5 (Critical pair) Let $r_i \colon \langle L_i \leftarrow K_i \rightarrow R_i \rangle$ be rules, for $i = 1, 2$. A pair of direct derivations $U_1 \Leftarrow_{r_1,g_1} S \Rightarrow_{r_2,g_2} U_2$ is a *critical pair* if

(1) $S = g_1(L_1) \cup g_2(L_2)$ and

(2) the steps are not independent.

Moreover, we require $g_1 \neq g_2$ in case $r_1 = r_2$.

Two critical pairs $U_1 \Leftarrow_{r_1,g_1} S \Rightarrow_{r_2,g_2} U_2$ and $U_1' \Leftarrow_{r_1,g_1'} S' \Rightarrow_{r_2,g_2'} U_2'$ are *isomorphic* if there is an isomorphism $f \colon S \rightarrow S'$ such that for $i = 1, 2$, $g_i' = f \circ g_i$. In the sequel, we equate isomorphic critical pairs so that condition (1) guarantees that a finite set of rules has only a finite number of critical pairs.

Example 2 Figure 6 shows the critical pairs of the hypergraph-transformation system of Figure 4 and demonstrates that all pairs are *strongly joinable* in the sense of the next definition. (Track morphisms are indicated by node names.) □

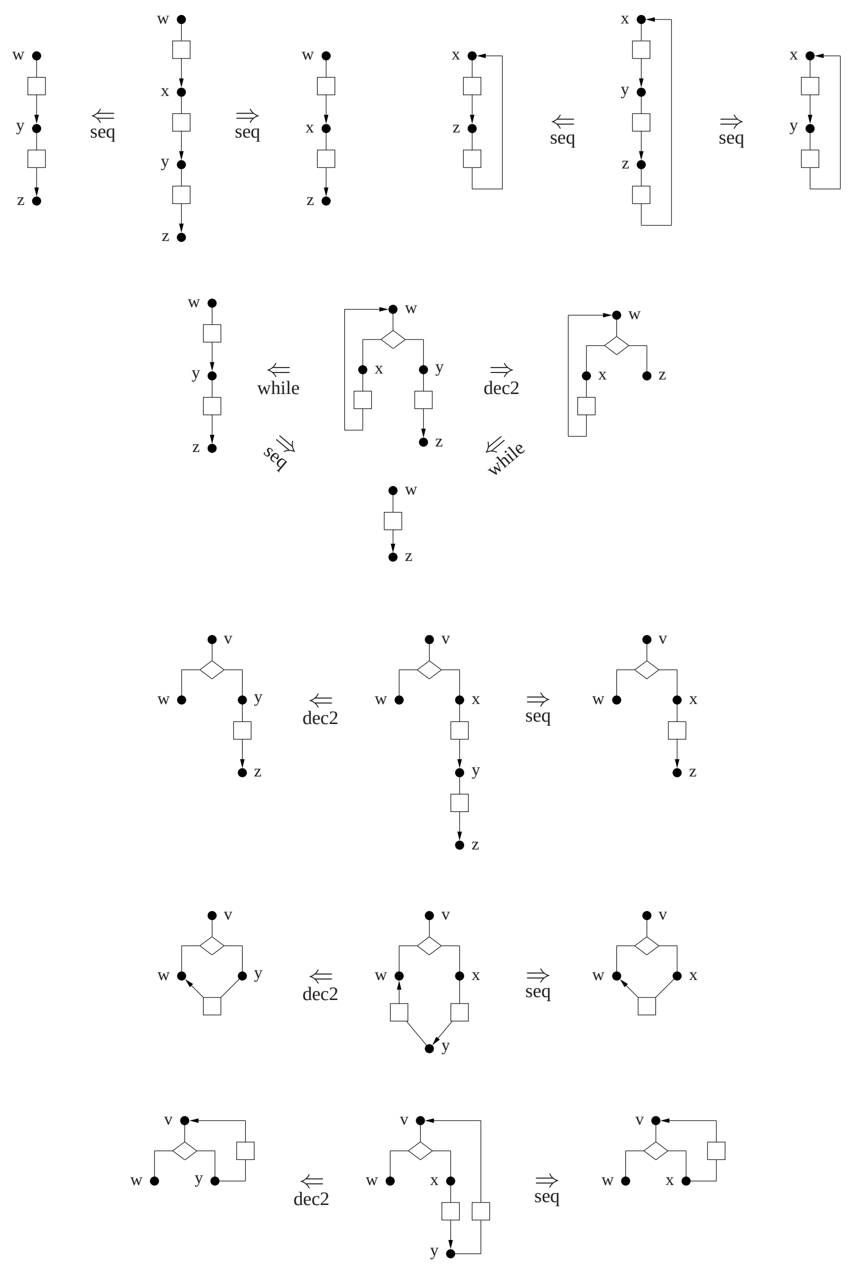


Figure 6: The critical pairs of the system of Figure 4

Given a critical pair $\Gamma\colon U_1 \Leftarrow S \Rightarrow U_2$, let $\mathrm{Persist}_\Gamma$ be the subhypergraph of S consisting of all nodes v such that both $\mathrm{tr}_{S\Rightarrow U_1}(v)$ and $\mathrm{tr}_{S\Rightarrow U_2}(v)$ are defined.

Definition 6 (Joinability) Let $\langle\Sigma,\mathscr{R}\rangle$ be a hypergraph-transformation system. A critical pair $\Gamma\colon U_1 \Leftarrow S \Rightarrow U_2$ is *joinable* if there are derivations $U_i \Rightarrow^*_{\mathscr{R}} X_i$, for $i = 1,2$, and an isomorphism $f\colon X_1 \to X_2$. Moreover, Γ is *strongly joinable* if, in addition, for each node v in $\mathrm{Persist}_\Gamma$,

(1) $\mathrm{tr}_{S\Rightarrow U_1\Rightarrow^* X_1}(v)$ and $\mathrm{tr}_{S\Rightarrow U_2\Rightarrow^* X_2}(v)$ are defined and

(2) $f_{\mathrm{V}}(\mathrm{tr}_{S\Rightarrow U_1\Rightarrow^* X_1}(v)) = \mathrm{tr}_{S\Rightarrow U_2\Rightarrow^* X_2}(v)$.

In [Plu05] it is shown that a hypergraph-transformation system is locally confluent if all its critical pairs are strongly joinable. Combining this result with Newman's Lemma yields a sufficient condition for the confluence of terminating systems.

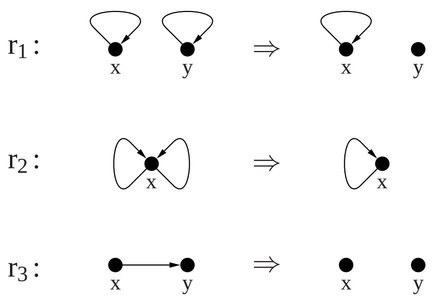
Theorem 2 ([Plu05]) *A terminating hypergraph-transformation system is confluent if all its critical pairs are strongly joinable.*

For example, the system of Figure 4 is terminating since each of the rules reduces the size of a hypergraph it is applied to. Thus, by Theorem 2, the system is confluent.

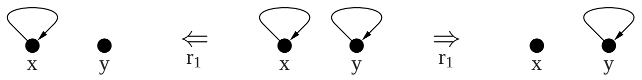
5 Coverable Systems

In general, by Theorem 1, confluence of a terminating hypergraph-transformation system $\langle\Sigma,\mathscr{R}\rangle$ cannot be decided by checking whether all critical pairs are strongly joinable. For, suppose we encounter a critical pair $U_1 \Leftarrow S \Rightarrow U_2$ that is joinable but not strongly joinable, that is, there are hypergraphs X_1 and X_2 such that $U_1 \Rightarrow^*_{\mathscr{R}} X_1 \cong X_2 \Leftarrow^*_{\mathscr{R}} U_2$ but no isomorphism $X_1 \to X_2$ is compatible with the track morphisms $\mathrm{tr}_{S\Rightarrow U_i\Rightarrow^* X_i}$. Then, assuming that all other critical pairs are joinable, $\langle\Sigma,\mathscr{R}\rangle$ may or may not be confluent. This is demonstrated by the following example.

Example 3 Consider the graph-transformation system $\langle\Sigma,\mathscr{R}\rangle$ consisting of singletons Σ_{V} and Σ_{E}, and the following rules:

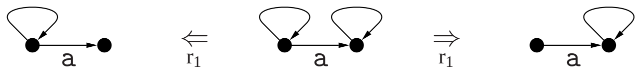


This system is terminating as every rule application reduces the number of edges. It is also confluent since whenever $H_1 \Leftarrow^*_{\mathscr{R}} G \Rightarrow^*_{\mathscr{R}} H_2$, there are derivations $H_1 \Rightarrow^*_{\mathscr{R}} H_1' \cong H_2' \Leftarrow^*_{\mathscr{R}} H_2$ where H_1' and H_2' consist of $|V_G|$ nodes and either no edges (if G is loop-free) or one loop and no other edges. However, despite confluence, the critical pair



is not strongly joinable because the outer graphs are normal forms[1] and the isomorphism between them is not compatible with the track morphisms as required by condition (2) of Definition 6.

Thus, we cannot report non-confluence if we encounter a joinable critical pair that is not strongly joinable. On the other hand, joinability of all critical pairs does not guarantee confluence. Suppose, for instance, that we add an edge label `a` to Σ_E. Then all critical pairs are still joinable but confluence breaks down, as witnessed by the following counterexample:



This example also shows that signature extensions need not preserve confluence. In particular, hyperedge labels that do not occur in rules turn out to be crucial for ensuring that local confluence implies strong joinability of all critical pairs. $\square$

Given a hyperedge e in a hypergraph G, the pair $\langle \mathrm{lab}_G(e), \mathrm{mark}^*_G(\mathrm{att}_G(e))\rangle$ is the *profile* of e. If $\mathscr{R}$ is a set of hypergraph-transformation rules, we write $\mathrm{Prof}(\mathscr{R})$ for the set of all hyperedge profiles occurring in $\mathscr{R}$ and $\mathrm{Mark}(\mathscr{R})$ for the set of all node labels occurring in $\mathscr{R}$.

Definition 7 ($G^{\mathscr{R}}$ and $G^{\ominus}$) Let $\langle \Sigma, \mathscr{R}\rangle$ be a hypergraph-transformation system and $G \in \mathscr{H}(\Sigma)$. We define subhypergraphs $G^{\mathscr{R}}$ and $G^{\ominus}$ as follows:

(1) $G^{\mathscr{R}}$ consists of all hyperedges with profile in $\mathrm{Prof}(\mathscr{R})$ and all nodes with label in $\mathrm{Mark}(\mathscr{R})$.

(2) $G^{\ominus}$ consists of all hyperedges in $\mathrm{E}_G - \mathrm{E}_{G^{\mathscr{R}}}$, all attachment nodes of these hyperedges, and all nodes in $\mathrm{V}_G - \mathrm{V}_{G^{\mathscr{R}}}$.

It follows that $G = G^{\mathscr{R}} \cup G^{\ominus}$, where $G^{\mathscr{R}}$ and $G^{\ominus}$ may share some attachment nodes of edges in $G^{\ominus}$. These shared nodes cannot be removed by any rule in $\mathscr{R}$, by the dangling condition for direct derivations.

Definition 8 (Cover) Given a critical pair Γ of a hypergraph-transformation system $\langle \Sigma, \mathscr{R}\rangle$, a *cover* for Γ is a hypergraph $C \in \mathscr{H}(\Sigma)$ such that

(1) $\mathrm{Persist}_\Gamma \subseteq C$,

(2) $C^{\ominus} = C$, and

(3) for every image $\tilde{C}$ of C, there is a unique surjective morphism $C \to \tilde{C}$.

Remarks

1. By condition (2), the profiles of the hyperedges in C are distinct from those in $\mathrm{Prof}(\mathscr{R})$. Also, since all node labels in $\mathrm{Persist}_\Gamma$ belong to $\mathrm{Mark}(\mathscr{R})$, (1) and (2) imply that each node in $\mathrm{Persist}_\Gamma$ is incident to some hyperedge in C.

[1] A graph G is a *normal form* with respect to a system $\langle \Sigma, \mathscr{R}\rangle$ if there is no graph H such that $G \Rightarrow_{\mathscr{R}} H$.

2. Intuitively, C uniquely identifies the nodes in $\mathrm{Persist}_\Gamma$ in that for every image $\tilde{C}$ of C, each node in $\mathrm{Persist}_\Gamma$ corresponds to a unique node in $\tilde{C}$. Moreover, the rules in $\mathscr{R}$ can affect C at most by merging some nodes in $\mathrm{Persist}_\Gamma$.

3. By condition (3), C does not possess nontrivial automorphisms. That is, the identity $\mathrm{id}_C \colon C \to C$ is the only isomorphism on C.

Example 4 Consider a critical pair $\Gamma\colon U_1 \Leftarrow S \Rightarrow U_2$.

1. If $\mathrm{Persist}_\Gamma = \emptyset$, then the empty hypergraph is a cover for Γ.

2. Let $\mathrm{Persist}_\Gamma$ consist of a single node v with label m. If there is some $l \in \Sigma_\mathrm{E}$ such that $m \in \mathrm{Type}(l)$ and $\langle l, m\rangle \notin \mathrm{Prof}(\mathscr{R})$, then the hypergraph C consisting of v and an hyperedge e with $\mathrm{lab}_C(e) = l$ and $\mathrm{att}_C(e) = v$ is a cover for Γ. Alternatively, if $mm \in \mathrm{Type}(l)$ and $\langle l, mm\rangle \notin \mathrm{Prof}(\mathscr{R})$, then the graph C consisting of v and an edge e with $\mathrm{lab}_C(e) = l$ and $\mathrm{att}_C(e) = vv$ is a cover for Γ.

3. Let $\mathrm{Persist}_\Gamma$ consist of nodes $v_1,\dots,v_n$ with $n \geq 2$ and $\mathrm{mark}_S(v_i) = m_i$, for $i = 1,\dots,n$. If there is $l \in \Sigma_\mathrm{E}$ such that $m_1 \dots m_n \in \mathrm{Type}(l)$ and $\langle l, m_1 \dots m_n\rangle \notin \mathrm{Prof}(\mathscr{R})$, then C consisting of $v_1,\dots,v_n$ and an hyperedge e with $\mathrm{lab}_C(e) = l$ and $\mathrm{att}_C(e) = v_1 \dots v_n$ is a cover for Γ. Alternatively, suppose that there are distinct labels $l_1,\dots,l_{n-1} \in \Sigma_\mathrm{E}$ such that for $i = 1,\dots,n-1$, $m_i m_{i+1} \in \mathrm{Type}(l_i)$ and $\langle l_i, m_i m_{i+1}\rangle \notin \mathrm{Prof}(\mathscr{R})$. Then a graph cover C for Γ is given by $v_1,\dots,v_n$ and edges $e_1,\dots,e_{n-1}$ where for $i = 1,\dots,n-1$, $\mathrm{lab}_C(e_i) = l_i$ and $\mathrm{att}_C(e_i) = v_i v_{i+1}$. (For instance, the critical pair discussed in Example 3 can be covered in this way after the edge label $\mathtt{a}$ with $\mathrm{Type}(\mathtt{a}) = \{\bullet\bullet\}$ has been added to Σ_E.) $\square$

Figure 7 shows the alternative covers of Example 4.3 for a critical pair with n persistent nodes. Note that $l_1,\dots,l_{n-1}$ need to be distinct as otherwise condition (3) of Definition 8 may be violated.

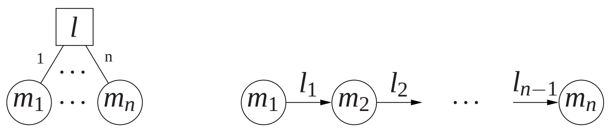


Figure 7: Alternative covers for a critical pair with n persistent nodes

Definition 9 (Coverable system) A hypergraph-transformation system is *coverable* if for each of its critical pairs there exists a cover.

Our main result is that for coverable systems, local confluence is equivalent to the strong joinability of all critical pairs.

Theorem 3 *A coverable hypergraph-transformation system is locally confluent if and only if all its critical pairs are strongly joinable.*

Sketch of proof. Theorem 2 establishes the “if” direction. We outline the proof for the converse, which is based on extending critical pairs with their covers. Consider a critical pair $\Gamma\colon U_1 \Leftarrow S \Rightarrow U_2$ and a cover C for Γ such that $S \cap C = \mathrm{Persist}_\Gamma$. Then there are extended direct derivations $\widehat{U}_1 \Leftarrow \widehat{S} \Rightarrow \widehat{U}_2$, where $\widehat{S} = S \cup C$. By local confluence, there are hypergraphs X_1 and X_2 such that $\widehat{U}_1 \Rightarrow^* X_1 \cong X_2 \Leftarrow^* \widehat{U}_2$. The derivations $\widehat{S} \Rightarrow \widehat{U}_i \Rightarrow^* X_i$, $i = 1,2$, preserve the nodes in $\mathrm{Persist}_\Gamma$ because the latter are incident to edges in C . Hence, after taking the cover C off, one obtains restricted derivations $S \Rightarrow U_i \Rightarrow^* \overline{X}_i$, $i = 1,2$, that satisfy condition (1) of Definition 6. Moreover, one can show that $\overline{X}_1 = X_1^{\mathscr{R}} \cong X_2^{\mathscr{R}} = \overline{X}_2$. Restricting the morphisms $\mathrm{tr}_{\widehat{S}\Rightarrow\widehat{U}_i\Rightarrow^* X_i}$, $i = 1,2$, to $\widehat{S}^{\ominus}$ and $X_i^{\ominus}$ yields surjective morphisms $t_i\colon \widehat{S}^{\ominus} \to X_i^{\ominus}$. Also, given an isomorphism $f\colon X_1 \to X_2$, its restriction $f^{\ominus}\colon X_1^{\ominus} \to X_2^{\ominus}$ is an isomorphism. Hence both $f^{\ominus} \circ t_1\colon \widehat{S}^{\ominus} \to X_2^{\ominus}$ and $t_2\colon \widehat{S}^{\ominus} \to X_2^{\ominus}$ are surjective morphisms. Since $\widehat{S}^{\ominus} = C$, condition (3) of Definition 8 implies $f \circ t_1 = t_2$. It then follows that condition (2) of Definition 6 is satisfied. Thus Γ is strongly joinable. □

Assumption For the rest of this section, we consider hypergraph-transformation systems $\langle \Sigma, \mathscr{R} \rangle$ in which Σ_V, Σ_E and $\mathscr{R}$ are finite.

As a consequence of Theorem 3, confluence of terminating coverable systems is equivalent to the strong joinability of all critical pairs. This allows to decide confluence by testing for the latter property.

Corollary 1 *Confluence is decidable for coverable hypergraph-transformation systems that are terminating.*

Given a terminating and coverable system, Algorithm 1 checks whether all critical pairs are strongly joinable by extending critical pairs with covers and then testing for simple joinability of all “covered pairs”. By the (full) proof of Theorem 3, joinability of a covered pair implies strong joinability of the underlying critical pair. Given a covered pair $\widehat{\Gamma}\colon \widehat{U}_1 \Leftarrow \widehat{S} \Rightarrow \widehat{U}_2$, one nondeterministically computes a normal form X_i of $\widehat{U}_i$, for $i = 1,2$, and checks whether X_1 and X_2 are isomorphic. If they are, then the critical pair Γ underlying $\widehat{\Gamma}$ is strongly joinable, otherwise a counterexample to confluence has been found.

Example 5 Consider again the hypergraph-transformation system of Example 1. Suppose that its typing allows a rhomb hyperedge to have two attachment nodes and a square hyperedge to have three attachment nodes, besides the versions of rhombs and squares occurring in the rules. Then each critical pair of this system can be covered and Algorithm 1 determines that the system is confluent. For instance, Figure 8 shows the extended version of a critical pair of Figure 6 and its joining derivations.

The graph-transformation system of Example 3, on the other hand, is not coverable. It becomes coverable after the edge label a has been added to the signature, when Algorithm 1 determines that the resulting system is non-confluent. □

Algorithm 1 Decision procedure for confluence

```
Input: a terminating and coverable hypergraph-transformation system $\langle\Sigma,\mathscr{R}\rangle$ and its set of critical pairs CP
  for all $\Gamma$: $U_1 \Leftarrow_{r_1,g_1} S \Rightarrow_{r_2,g_2} U_2$ in CP do
    {let $C$ be a cover for $\Gamma$ such that $S \cap C = \mathrm{Persist}_\Gamma$}
    $\widehat{S} := S \cup C$
    {for $i = 1,2$, let $\widehat{g}_i$ be the extension of $g_i$ to $\widehat{S}$}
    for $i = 1$ to $2$ do
      construct a derivation $\widehat{S} \Rightarrow_{r_i,\widehat{g}_i} \widehat{U}_i \Rightarrow^*_{\mathscr{R}} X_i$ such that $X_i$ is a normal form
    end for
    if $X_1 \ncong X_2$ then
      return "non-confluent"
    end if
  end for
  return "confluent"
```

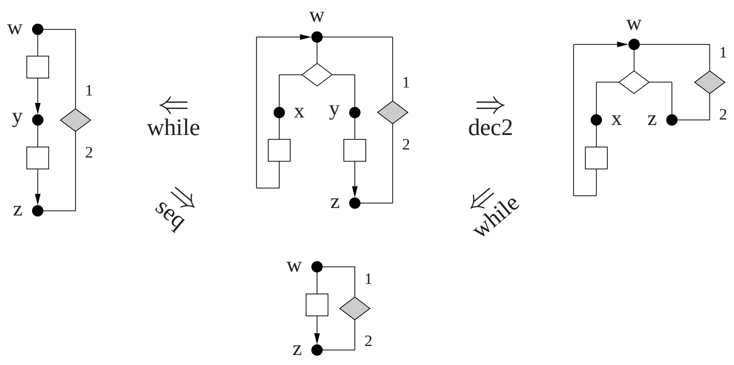


Figure 8: An extended critical pair of the system of Figure 4

Particular classes of hypergraph- and graph-transformation systems for which confluence is decidable can be obtained by specialising Corollary 1 with the conditions given in Example 4.3 or with similar conditions. For instance, in the case of graph transformation, another sufficient condition for terminating systems is that for each critical pair Γ with persistent nodes $v_1,\dots,v_n$, there are distinct labels $l_1,\dots,l_n \in \Sigma_{\mathrm{E}}$ such that for $i = 1,\dots,n$, $\mathrm{mark}_S(v_i)\mathrm{mark}_S(v_i) \in \mathrm{Type}(l_i)$ and $\langle l_i, \mathrm{mark}_S(v_i)\mathrm{mark}_S(v_i)\rangle \notin \mathrm{Prof}(\mathscr{R})$. In this case a cover can be constructed by attaching to $v_1,\dots,v_n$ loops labelled with $l_1,\dots,l_n$.

In the case of hypergraph transformation, a sufficient condition for the decidability of confluence (of terminating systems) can be given purely in terms of the signature Σ. We call a signature Σ *universal* if for each $l \in \Sigma_{\mathrm{E}}$, $\mathrm{Type}(l) = \Sigma_{\mathrm{V}}^*$.

Corollary 2 *Confluence is decidable for terminating hypergraph-transformation systems with universal signatures.*

For, if hyperedges can have arbitrary sequences of attachment nodes, we can cover critical pairs with hyperedges that have longer attachment sequences than any hyperedges in rules by using repeated nodes in the attachment.

6 Conclusion

Confluence is an undecidable property of terminating graph- and hypergraph-transformation systems. We have identified coverable systems as a subclass that comes with a decision procedure for confluence. The class is nontrivial and properly includes all hypergraph-transformation systems with universal signatures.

A topic for future work is to extend Algorithm 1 such that it decides confluence for certain non-coverable systems. The idea is to add to the signature of an input system a hyperedge label whose typing allows to cover all critical pairs. One then runs the algorithm as before: if all extended pairs are joinable, one can conclude that the underlying critical pairs are strongly joinable and hence that the system is confluent. However, if a non-joinable extended pair is encountered whose underlying critical pair is joinable, then the procedure has to give up because the input system may or may not be confluent.

Bibliography

[ACPS93] S. Arnborg, B. Courcelle, A. Proskurowski, D. Seese. An Algebraic Theory of Graph Reduction. *Journal of the ACM* 40(5):1134–1164, 1993.

[BF01] H. L. Bodlaender, B. van Antwerpen-de Fluiter. Reduction Algorithms for Graphs of Small Treewidth. *Information and Computation* 167(2):86–119, 2001.

[BKV03] M. Bezem, J. W. Klop, R. de Vrijer (eds.). *Term Rewriting Systems*. Cambridge University Press, 2003.

[BN98] F. Baader, T. Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998.

[BO93] R. V. Book, F. Otto. *String-Rewriting Systems*. Texts and Monographs in Computer Science. Springer-Verlag, 1993.

[BPR04] A. Bakewell, D. Plump, C. Runciman. Specifying Pointer Structures by Graph Reduction. In *Applications of Graph Transformations With Industrial Relevance (AGTIVE 2003), Revised Selected and Invited Papers*. Lecture Notes in Computer Science 3062, pp. 30–44. Springer-Verlag, 2004.

[FKZ76] R. Farrow, K. Kennedy, L. Zucconi. Graph Grammars and Global Program Data Flow Analysis. In *Proc. 17th Annual Symposium on Foundations of Computer Science*. Pp. 42–56. 1976.

[GBG$^+$06] R. Geiß, G. V. Batz, D. Grund, S. Hack, A. M. Szalkowski. GrGen: A Fast SPO-Based Graph Rewriting Tool. In *Proc. International Conference on Graph Transformation (ICGT 2006)*. Lecture Notes in Computer Science 4178, pp. 383–397. Springer-Verlag, 2006.

[HMP01] A. Habel, J. Müller, D. Plump. Double-Pushout Graph Transformation Revisited. *Mathematical Structures in Computer Science* 11(5):637–688, 2001.

[New42] M. Newman. On Theories with a Combinatorial Definition of "Equivalence". *Annals of Mathematics* 43(2):223–243, 1942.

[NNZ00] U. Nickel, J. Niere, A. Zündorf. The FUJABA Environment. In *Proc. International Conference on Software Engineering (ICSE 2000)*. Pp. 742–745. ACM Press, 2000.

[Plu93] D. Plump. Hypergraph Rewriting: Critical Pairs and Undecidability of Confluence. In Sleep et al. (eds.), *Term Graph Rewriting: Theory and Practice*. Chapter 15, pp. 201–213. John Wiley, 1993.

[Plu05] D. Plump. Confluence of Graph Transformation Revisited. In Middeldorp et al. (eds.), *Processes, Terms and Cycles: Steps on the Road to Infinity: Essays Dedicated to Jan Willem Klop on the Occasion of His 60th Birthday*. Lecture Notes in Computer Science 3838, pp. 280–308. Springer-Verlag, 2005.

[Plu09] D. Plump. The Graph Programming Language GP. In *Proc. Algebraic Informatics (CAI 2009)*. Lecture Notes in Computer Science 5725, pp. 99–122. Springer-Verlag, 2009.

[RS97] J. Rekers, A. Schürr. Defining and Parsing Visual Languages with Layered Graph Grammars. *Journal of Visual Languages and Computing* 8(1):27–55, 1997.

[Tae04] G. Taentzer. AGG: A Graph Transformation Environment for Modeling and Validation of Software. In *Applications of Graph Transformations With Industrial Relevance (AGTIVE 2003), Revised Selected and Invited Papers*. Lecture Notes in Computer Science 3062, pp. 446–453. Springer-Verlag, 2004.