15

Hypergraph Rewriting: Critical Pairs and Undecidability of Confluence

Detlef Plump

15.1 INTRODUCTION

In their pioneering paper [KB70], Knuth and Bendix showed that confluence (or, equivalently, the Church-Rosser property) is decidable for terminating term rewriting systems. It suffices to compute all *critical pairs* $t \leftarrow s \rightarrow u$ of rewrite steps in which s is the superposition of the left-hand sides of two rules, and to check whether t and u reduce to a common term. This procedure is justified by the so-called Critical Pair Lemma [Hue80] which states that a term rewriting system is locally confluent if and only if all critical pairs have a common reduct.

For (hyper)graph rewriting systems, however, no such simple characterization of local confluence is possible. The reason is that the embedding of derivations into "context" is more complicated than for tree rewriting. It is shown below that in the graph case, confluence of all critical pairs need not imply general local confluence. This phenomenon refutes a critical pair lemma published by Raoult [Rao84] (personal communication). Okada and Hayashi [OH92] avoid the problem by giving a critical pair lemma under the strong restriction that distinct nodes in a graph must not have the same label.

In this chapter a critical pair lemma for general hypergraph rewriting is presented which provides a sufficient condition for local confluence. It requires that all critical pairs are confluent by derivations that satisfy certain conditions. The second part of

Term Graph Rewriting: Theory and Practice, eds. M.R.Sleep, M.J.Plasmeijer and M.C. van Eekelen

this chapter reveals that a simple characterization of local confluence is indeed impossible: confluence is shown to be undecidable for terminating hypergraph rewriting systems.

15.2 HYPERGRAPH REWRITING

In this section the "Berlin approach" to graph rewriting is briefly reviewed (see [Ehr79] for a comprehensive survey), but all notions are lifted to the hypergraph case which is more flexible in applications. In particular, three theorems of the Berlin approach are recalled concerning the commutation, restriction, and extension of derivations. These results are essential tools in the proof of the Critical Pair Lemma.

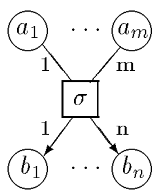
15.2.1 Hypergraphs and hypergraph morphisms

Let $\Sigma = \langle \Sigma_V, \Sigma_E \rangle$ be a *signature*, that is, Σ_V and Σ_E are sets (of node and edge labels), and each $\sigma \in \Sigma_E$ comes with a pair $\mathit{type}(\sigma) = \langle \alpha, \beta \rangle$ of strings $\alpha, \beta \in \Sigma_V^*$.

A *hypergraph* over Σ is a system $G = \langle V_G, E_G, l_G, m_G, s_G, t_G \rangle$, where V_G and E_G are finite sets of *nodes* and *hyperedges* (or *edges* for short), $l_G\colon V_G \to \Sigma_V$ and $m_G\colon E_G \to \Sigma_E$ are *labeling functions*, and $s_G, t_G\colon E_G \to V_G^*$ are functions that assign strings $s_G(e), t_G(e)$ of *source* and *target nodes* to each hyperedge e such that $\mathit{type}(m_G(e)) = \langle l_G^*(s_G(e)), l_G^*(t_G(e)) \rangle$. (The extension $f^*\colon A^* \to B^*$ of a function $f\colon A \to B$ maps the empty string to itself and $a_1 \ldots a_n$ to $f(a_1) \ldots f(a_n)$.)

G is said to be *discrete* if $E_G = \emptyset$.

In pictures of hypergraphs, nodes are drawn as circles and hyperedges as boxes, both with inscribed labels. Lines without arrowheads connect a hyperedge with its source nodes, while arrows point to the target nodes. For example, the graphical structure



represents a hyperedge together with its source and target nodes, where $\mathit{type}(\sigma) = \langle a_1 \ldots a_m, b_1 \ldots b_n \rangle$. "Ordinary" edges with one source and one target node are frequently depicted as arrows, with labels written aside.

Let G,H be hypergraphs. Then G is a *subhypergraph* of H, denoted by $G \subseteq H$, if $V_G \subseteq V_H$, $E_G \subseteq E_H$, and l_G, m_G, s_G, t_G are restrictions of the corresponding functions of H.

A *hypergraph morphism* $f\colon G \to H$ consists of two functions $f_V\colon V_G \to V_H$ and $f_E\colon E_G \to E_H$ that preserve labels and assignments of source and target nodes, that is, $l_H \circ f_V = l_G$, $m_H \circ f_E = m_G$, $s_H \circ f_E = f_V^* \circ s_G$, and $t_H \circ f_E = f_V^* \circ t_G$. f is *injective* (*surjective*) if f_V and f_E are injective (surjective). f is an *isomorphism* if it is injective and surjective; in this case G and H are *isomorphic*, denoted by $G \cong H$.

The subhypergraph of H with node set $f_V(V_G)$ and edge set $f_E(E_G)$ is denoted by fG. If $G \subseteq H$, then $G \hookrightarrow H$ denotes the inclusion morphism.

15.2.2 Rules and derivations

A *rule* $r = (L \supseteq K \to R)$ consists of three hypergraphs L,K,R and a morphism $K \to R$, where $K \subseteq L$.

A *hypergraph rewriting system* $\mathcal{G} = (\Sigma, \mathcal{R})$ consists of a signature Σ and a set $\mathcal{R}$ of rules with hypergraphs over Σ. For the rest of this section and the following section, $\mathcal{G}$ denotes an arbitrary hypergraph rewriting system.

Let G, H be hypergraphs. Given a rule $r = (L \supseteq K \to R)$ from $\mathcal{G}$ and a morphism $g\colon L \to G$, *G directly derives H through r and g*, denoted by $G \Rightarrow_{r,g} H$, if there are two hypergraph pushouts of the following form:

$$\begin{array}{ccccc} L & \hookleftarrow & K & \to & R \\ {\scriptstyle g}\downarrow & & \downarrow & & \downarrow \\ G & \hookleftarrow & D & \stackrel{c}{\to} & H \end{array}$$

(See [Ehr79] for the definition and construction of graph pushouts; the extension to hypergraphs is straightforward.) Intuitively, D is obtained from G by removing the nodes and edges in $gL - gK$, and H is constructed from D by identifying items in gK as specified by $K \to R$ and by adding the items in $R - K$.

The relations $\Rightarrow_r$ and $\Rightarrow$ are defined in the obvious way. $G \Rightarrow^\lambda H$ means $G \Rightarrow H$ or $G \cong H$. *G derives H*, denoted by $G \Rightarrow^* H$, if $G \cong H$ or there are hypergraphs $G_0,\ldots,G_n$ $(n \geq 1)$ such that $G = G_0 \Rightarrow G_1 \Rightarrow \ldots \Rightarrow G_n = H$.

Proposition **15.2.1** *Let G be a hypergraph, $r = (L \supseteq K \to R)$ be a rule, and $g\colon L \to G$ be a morphism. Then there exists a direct derivation $G \Rightarrow_{r,g} H$ if and only if the following two conditions are satisfied:*
Contact Condition. *No edge in $G - gL$ is incident to any node in $gL - gK$.*
Identification Condition. *For all items x, y in L, $g(x) = g(y)$ implies $x = y$ or $x, y \in K$.*

The following *track function* allows to "follow nodes through derivations". For a direct derivation $G \Rightarrow H$, $track_{G\Rightarrow H}\colon V_G \to V_H$ is the partial function defined by

$$track_{G\Rightarrow H}(v) = \begin{cases} c_V(v) & \text{if } v \in D, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

For a derivation $G \Rightarrow^* H$, $track_{G\Rightarrow^* H} = i_V$ if $G \Rightarrow^* H$ by an isomorphism $i\colon G \to H$, and $track_{G\Rightarrow^* H} = track_{G_{n-1}\Rightarrow G_n} \circ \ldots \circ track_{G_0 \Rightarrow G_1}$ if $G \Rightarrow^* H$ by a sequence $G = G_0 \Rightarrow G_1 \Rightarrow \ldots \Rightarrow G_n = H$.

$\mathcal{G}$ is *confluent* if for all hypergraphs G, H_1, H_2 with $H_1 \;{}^*\!\Leftarrow G \Rightarrow^* H_2$ there is a hypergraph M such that $H_1 \Rightarrow^* M \;{}^*\!\Leftarrow H_2$. $\mathcal{G}$ is *locally confluent* if for all direct derivations of the form $H_1 \Leftarrow G \Rightarrow H_2$ there is an M such that $H_1 \Rightarrow^* M \;{}^*\!\Leftarrow H_2$. Finally, $\mathcal{G}$ is *terminating* if it does not admit an infinite sequence $G_1 \Rightarrow G_2 \Rightarrow G_3 \Rightarrow \ldots$ of direct derivations.

15.2.3 Commutation, restriction, and extension of derivations

The following three theorems were originally formulated for graphs rather than for hypergraphs. But inspecting their proofs shows that they can be extended to the hypergraph case without further ado.

THEOREM **15.2.2 (Commutation theorem [EK76])** *Let* $H_1 \;{}_{r_1,g_1}\!\Leftarrow G \Rightarrow_{r_2,g_2} H_2$ *be direct derivations through rules* $r_i = (L_i \supseteq K_i \to R_i)$, *for* $i = 1, 2$. *If* $g_1L_1 \cap g_2L_2 = g_1K_1 \cap g_2K_2$, *then there is a hypergraph* M *such that* $H_1 \Rightarrow_{r_2} M \;{}_{r_1}\!\Leftarrow H_2$.

The following variant of the so-called Clip Theorem applies only to direct derivations, which suffices for the purposes of the present chapter.

THEOREM **15.2.3 ([Kre77])** *Let* $G \Rightarrow_{r,g} H$ *be a direct derivation through a rule* $r = (L \supseteq K \to R)$. *If* S *is a subhypergraph of* G *such that* $gL \subseteq S$, *then* $S \Rightarrow_{r,g'} U$ *where* g' *is the restriction of* g *to* S *and* $U \subseteq H$. *Moreover,* $track_{S \Rightarrow U}$ *is the restriction of* $track_{G \Rightarrow H}$.

The next theorem allows a derivation to extend by arbitrary context, provided that context edges are not attached to nodes that are removed by the derivation. The present form of the theorem is tailored to the proof of the Critical Pair Lemma.

THEOREM **15.2.4 ([Ehr77, Kre77])** *Let* $S \Rightarrow_{r,g} T \Rightarrow^* U$ *be a derivation and* G *be a hypergraph with* $S \subseteq G$. *Let Boundary be the discrete subhypergraph of* S *that consists of all nodes that are touched by any edge in* $G - S$. *If* $track_{S \Rightarrow T \Rightarrow^* U}$ *is defined for all nodes in Boundary, then there is a derivation* $G \Rightarrow_{r,\overline{g}} H \Rightarrow^* M$ *such that* $T \subseteq H$ *and* $\overline{g}$ *is the extension of* g *to* G. *Moreover,* M *is defined by the pushout*

$$\begin{array}{ccc} Boundary & \xrightarrow{tr} & U \\ \downarrow & & \downarrow \\ Context & \longrightarrow & M \end{array}$$

where $Context = (G - S) \cup Boundary$ *is a subhypergraph of* G, $Boundary \to Context$ *is the inclusion of Boundary in Context, and* tr *is the restriction of* $track_{S \Rightarrow T \Rightarrow^* U}$ *to Boundary (considered as a morphism).*

15.3 THE CRITICAL PAIR LEMMA

The quest for a critical pair lemma is motivated by the problem of testing hypergraph rewriting systems for (local) confluence. The idea is to infer the confluence of arbitrary divergent steps $H_1 \;{}_{r_1}\!\Leftarrow G \Rightarrow_{r_2} H_2$ from the confluence of those steps where G represents a "critical overlap" of the left-hand sides of r_1 and r_2. By the Commutation Theorem 15.2.2, such an overlap is critical only if it comprises nodes or edges that are removed by r_1 or r_2. This suggests the following definition of a critical pair.

DEFINITION **15.3.1 (Critical pair)** *Let* $r_i = (L_i \supseteq K_i \to R_i)$ *be rules, for* $i = 1, 2$. *A pair of direct derivations of the form* $T \;{}_{r_1,g_1}\!\Leftarrow S \Rightarrow_{r_2,g_2} U$ *is a* critical pair *if* $S = g_1L_1 \cup g_2L_2$ *and* $g_1L_1 \cap g_2L_2 \neq g_1K_1 \cap g_2K_2$. *Moreover,* $g_1 \neq g_2$ *is required for the case* $r_1 = r_2$.

In the sequel, two critical pairs are not distinguished if they differ only by renaming of nodes and edges. The critical pairs arising from r_1 and r_2 can be computed by constructing all pairs of direct derivations $T \;{}_{r_1}\!\Leftarrow (L_1+L_2)_{/\approx} \Rightarrow_{r_2} U$ where $(L_1+L_2)_{/\approx}$ is a quotient of the disjoint union $L_1 + L_2$ that identifies at least one item in $L_1 - K_1$ (resp. $L_2 - K_2$) with some item in L_2 (L_1).

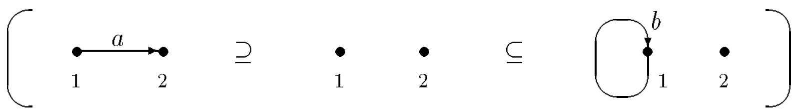
By the Commutation Theorem 15.2.2 a strong confluence property can be established for the case that $\mathcal{G}$ has no critical pairs at all. This is substantially different from term rewriting where only local confluence holds (see for example [Hue80]).

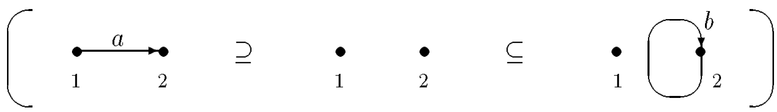
THEOREM 15.3.2 *Hypergraph rewriting systems without critical pairs are* strongly confluent, *that is, whenever* $H_1 \Leftarrow G \Rightarrow H_2$, *then there is a hypergraph* X *such that* $H_1 \Rightarrow^{\lambda} X \;{}^{\lambda}\!\Leftarrow H_2$.

PROOF. Let $H_1 \;{}_{r_1,g_1}\!\Leftarrow G \Rightarrow_{r_2,g_2} H_2$. If $g_1L_1 \cap g_2L_2 = g_1K_1 \cap g_2K_2$, then there are direct derivations $H_1 \Rightarrow_{r_2} M \;{}_{r_1}\!\Leftarrow H_2$ by Theorem 15.2.2. Assume therefore the contrary. The Clip Theorem 15.2.3 yields direct derivations $T \;{}_{r_1,g_1'}\!\Leftarrow (g_1L_1 \cup g_2L_2) \Rightarrow_{r_2,g_2'} U$ with $g_1'L_1 \cap g_2'L_2 \neq g_1'K_1 \cap g_2'K_2$. Because there are no critical pairs, $r_1 = r_2$ and $g_1 = g_2$ must hold. Then $H_1 \cong H_2$ since the result of a direct derivation is determined uniquely up to isomorphism. □

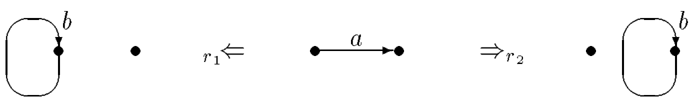
DEFINITION 15.3.3 *A critical pair* $T \Leftarrow S \Rightarrow U$ *is* joinable *if there is a hypergraph* X *such that* $T \Rightarrow^* X \;{}^*\!\Leftarrow U$.

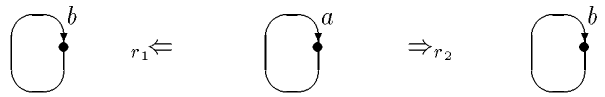
It turns out that the joinability of all critical pairs of $\mathcal{G}$ does not guarantee local confluence. This problem may occur if $S \Rightarrow T \Rightarrow^* X$ and $S \Rightarrow U \Rightarrow^* X$ send some node in S to different nodes in X. As an example, let $\mathcal{G}$ contain the following two rules (the node indices 1,2 indicate the inclusion morphisms):

$r_1 =$ 

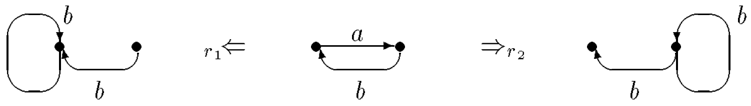
$r_2 =$ 

There are only two critical pairs, both being joinable:





However, $\mathcal{G}$ is not locally confluent:



The outer hypergraphs are non-isomorphic and irreducible, hence they have no common reduct.

Here the embedding of the first critical pair into context destroys the isomorphism between the outer hypergraphs. This is possible because the two direct derivations of the critical pair—although resulting in the same hypergraph—have different track functions. In order to overcome this problem one can introduce the rules

$r_3 =$ 

$r_4 =$ 

which allow the outer hypergraphs of both critical pairs to the empty hypergraph to be reduced. r_3 and r_4 do not create new critical pairs, so all critical pairs have "confluent derivations with identical track functions". Still, this is not sufficient for local confluence: r_3 and r_4 cannot be applied to the outer hypergraphs of the last derivation pair because of the contact condition for direct derivations. In other words, the confluent derivations cannot be embedded into context since r_3 and r_4 remove nodes.

This example suggests that the confluent derivations of critical pairs should preserve certain nodes and send these to the same nodes in the common reduct.

Definition **15.3.4** *Let $T \Leftarrow S \Rightarrow U$ be a critical pair, and let Protect(S) be the discrete subhypergraph of S that consists of all nodes v such that $track_{S \Rightarrow T}(v)$ and $track_{S \Rightarrow U}(v)$ are defined. Then $T \Leftarrow S \Rightarrow U$ is* strongly joinable *if there are derivations $T \Rightarrow^* X \ {}^*\!\Leftarrow U$ such that for all nodes v in Protect(S), $track_{S \Rightarrow T \Rightarrow^* X}(v)$ and $track_{S \Rightarrow U \Rightarrow^* X}(v)$ are defined and equal.*

LEMMA **15.3.5 (Critical Pair Lemma)** *A hypergraph rewriting system is locally confluent if all its critical pairs are strongly joinable.*

PROOF. Assume that all critical pairs of $\mathcal{G}$ are strongly joinable. Consider two direct derivations $H_1 \ {}_{r_1,g_1}\!\Leftarrow G \Rightarrow_{r_2,g_2} H_2$ through rules $r_i = (L_i \supseteq K_i \to R_i)$, $i = 1, 2$. If $g_1L_1 \cap g_2L_2 = g_1K_1 \cap g_2K_2$, then there is a hypergraph M such that $H_1 \Rightarrow_{r_2} M \ {}_{r_1}\!\Leftarrow H_2$ by the Commutation Theorem 15.2.2 . Assume therefore $g_1L_1 \cap g_2L_2 \neq g_1K_1 \cap g_2K_2$. Assume further that $r_1 \neq r_2$ or $g_1 \neq g_2$, as otherwise $H_1 \cong H_2$. Let $S = g_1L_1 \cup g_2L_2$. By Theorem 15.2.3 there are restricted derivation steps $U_1 \ {}_{r_1,g_1'}\!\Leftarrow S \Rightarrow_{r_2,g_2'} U_2$ where g_i' is the restriction of g_i to S and $U_i \subseteq H_i$, for $i = 1, 2$. Clearly these two steps constitute a critical pair. Hence, by assumption, there are derivations $U_1 \Rightarrow^* X \ {}^*\!\Leftarrow U_2$ such that $track_{S \Rightarrow U_1 \Rightarrow^* X}(v)$ and $track_{S \Rightarrow U_2 \Rightarrow^* X}(v)$ are defined and equal for each $v \in Protect(S)_V$.

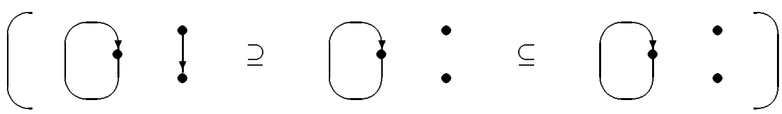
Let *Boundary* be the discrete subhypergraph of S that consists of all nodes that are touched by any edge in $G - S$. Both $track_{G \Rightarrow H_1}$ and $track_{G \Rightarrow H_2}$ are defined for all nodes in *Boundary*, because $G \Rightarrow H_1$ and $G \Rightarrow H_2$ satisfy the contact condition. Then, in particular, $track_{S \Rightarrow U_1}$ and $track_{S \Rightarrow U_2}$ are defined on *Boundary*, that is, $Boundary \subseteq Protect(S)$. Hence, for $i = 1, 2$, $track_{S \Rightarrow U_i \Rightarrow^* X}$ is defined on *Boundary*. Therefore, by Theorem 15.2.4, there are derivations $G \Rightarrow_{r_i,\overline{g_i}} \overline{H_i} \Rightarrow^* M_i$ with $U_i \subseteq \overline{H_i}$, for $i = 1, 2$. $\overline{g_i}$ is the extension of g_i' to G, so $\overline{g_i} = g_i$ and consequently $\overline{H_i} \cong H_i$, for $i = 1, 2$. Moreover, Theorem 15.2.4 states that, for $i = 1, 2$, M_i is defined by the pushout

$$\begin{array}{ccc} Boundary & \stackrel{tr_i}{\rightarrow} & X \\ \downarrow & & \downarrow \\ Context & \rightarrow & M_i \end{array}$$

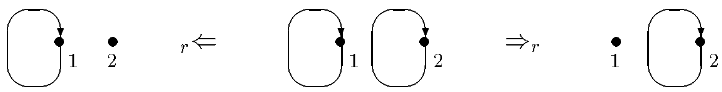
where $Context = (G - S) \cup Boundary$, $Boundary \to Context$ is the inclusion of *Boundary* in *Context*, and tr_i is the restriction of $track_{S \Rightarrow U_i \Rightarrow^* X}$ to *Boundary* (considered as a morphism). Now $tr_1 = tr_2$ implies $M_1 \cong M_2$ since pushout objects are unique up to isomorphism. Thus $H_1 \Rightarrow^* M_1 \ {}^*\!\Leftarrow H_2$. □

In contrast to term and string rewriting, the Critical Pair Lemma cannot provide a characterization of local confluence: the following example shows that even confluent and terminating systems may possess critical pairs that are not strongly joinable.

Let the label sets Σ_V and Σ_E be singletons, and let $\mathcal{G}$ contain only the following rule:

$r =$ 

$\mathcal{G}$ is terminating because every rule application decreases the number of edges by one. To see that $\mathcal{G}$ is confluent, consider two derivations $H_1 \ {}^*\!\Leftarrow G \Rightarrow^* H_2$. Then either G contains no loop and $H_1 \cong G \cong H_2$, or G, H_1, H_2 contain at least one loop and have the same number of nodes. In the latter case holds $H_1 \Rightarrow^* M \ {}^*\!\Leftarrow H_2$ for the hypergraph M with $|V_G|$ nodes, one loop, and no other edges. So $\mathcal{G}$ is confluent. But the following critical pair is not strongly joinable (the nodes are numbered to indicate the track functions):



15.4 UNDECIDABILITY OF CONFLUENCE

The above example demonstrates that terminating and confluent hypergraph rewriting systems need not have strongly joinable critical pairs. So the well-known decision procedure for the confluence of terminating term rewriting systems—which reduces the terms of a critical pair to normal form and checks equality—cannot be adapted to the hypergraph case (by checking strong joinability of critical pairs). This leads to the question whether confluence is decidable at all for terminating systems. By the following result, the answer is negative.

THEOREM **15.4.1** *It is undecidable in general whether a finite, terminating hypergraph rewriting system is confluent.*

Here a hypergraph rewriting system $\mathcal{G} = \langle \Sigma, \mathcal{R} \rangle$ is said to be finite if Σ_V, Σ_E and $\mathcal{R}$ are finite sets.

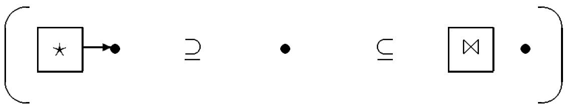
The rest of this section is devoted to the proof of Theorem 15.4.1. The proof idea is inspired by the proof of Kapur, Narendran, and Otto [KNO90] that ground-confluence is undecidable for terminating term rewriting systems. In the following the Post Correspondence Problem (PCP) is reduced to the problem of deciding confluence for terminating hypergraph rewriting systems. Recall that the PCP is the following decision problem: Given two nonempty lists $A = \langle u_1, \ldots, u_n \rangle$ and $B = \langle v_1, \ldots, v_n \rangle$ of nonempty words over some alphabet Γ, decide whether there is a sequence $i_1, \ldots, i_k$ of indices such that $u_{i_1} \ldots u_{i_k} = v_{i_1} \ldots v_{i_k}$.

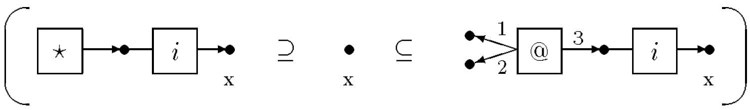
The pair $\langle A, B \rangle$ is called an *instance* of the PCP, and a sequence $i_1, \ldots, i_k$ as above is a *solution* of this instance. It is well-known that it is undecidable whether an arbitrary instance of the PCP has a solution (see for example [HU79]).

Let now $\langle A, B \rangle$ be an arbitrary instance of the PCP with $A = \langle u_1, \ldots, u_n \rangle$, $B = \langle v_1, \ldots, v_n \rangle$. The plan is to construct a finite, terminating hypergraph rewriting system $\mathcal{G}(A, B)$ that is confluent if and only if $\langle A, B \rangle$ has no solution.

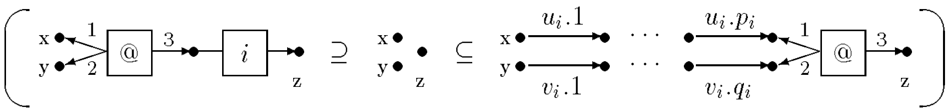
Let $\Sigma_V = \{\bullet\}$ and $\Sigma_E = \Gamma \cup \{1, \ldots, n\} \cup \{\star, \bowtie, @, ?, \exists\}$; the types of the edge labels can be seen from the rules below. The rule set of $\mathcal{G}(A, B)$ is partitioned into subsets $\mathcal{R}_0, \mathcal{R}_1, \mathcal{R}_2$, and $\mathcal{R}_3$. $\mathcal{R}_0$ gives rise to a critical pair which stands for the choice to create an edge labeled by $\bowtie$ or to check a possible solution of $\langle A, B \rangle$. $\mathcal{R}_1$ tests whether a sequence of indices is a solution of $\langle A, B \rangle$, $\mathcal{R}_2$ detects ill-formed hypergraphs, and $\mathcal{R}_3$ performs "garbage collection".

$\mathcal{R}_0$ contains the following rules:

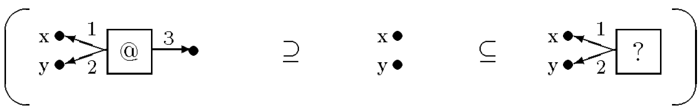


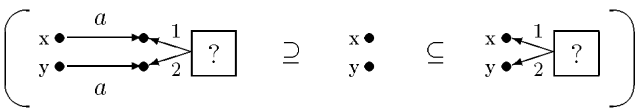
 for $i = 1, \ldots, n$,

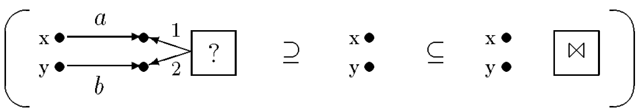
$\mathcal{R}_1$ contains the following rules:

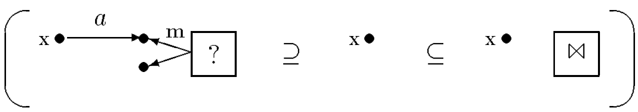


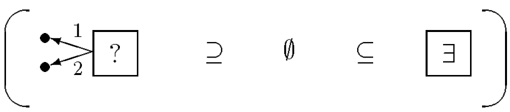
for $i = 1, \ldots, n$, where $u_i = u_i.1 \ldots u_i.p_i$ and $v_i = v_i.1 \ldots v_i.q_i$, with $u_i.j, v_i.j \in \Gamma$,



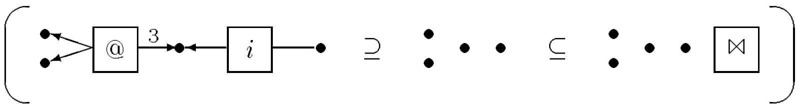
 for all $a \in \Gamma$,

 for all $a, b \in \Gamma$ with $a \neq b$,

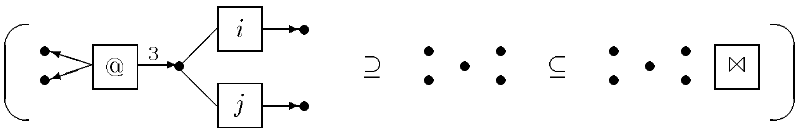
 for $m = 1, 2$ and all $a \in \Gamma$,



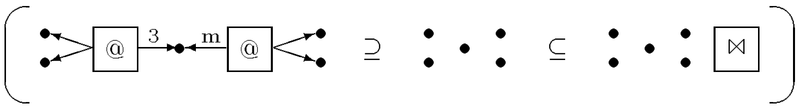
$\mathcal{R}_2$ contains the following rules:



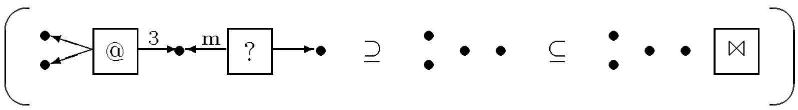
for $i = 1, \ldots, n$,



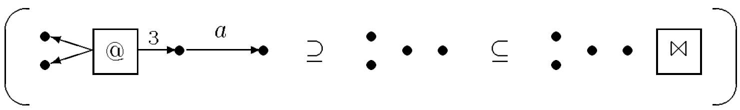
for all $i, j \in \{1, \ldots, n\}$,



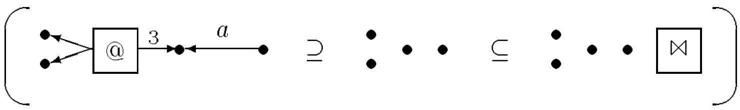
for $m = 1, 2, 3$,



for $m = 1, 2$,

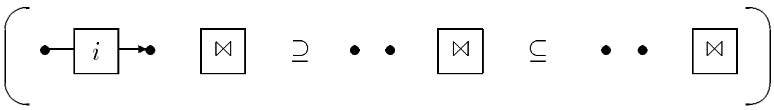


for all $a \in \Gamma$,

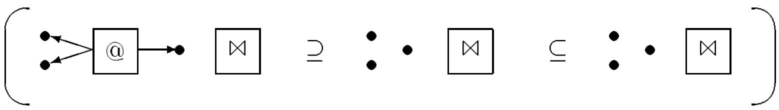


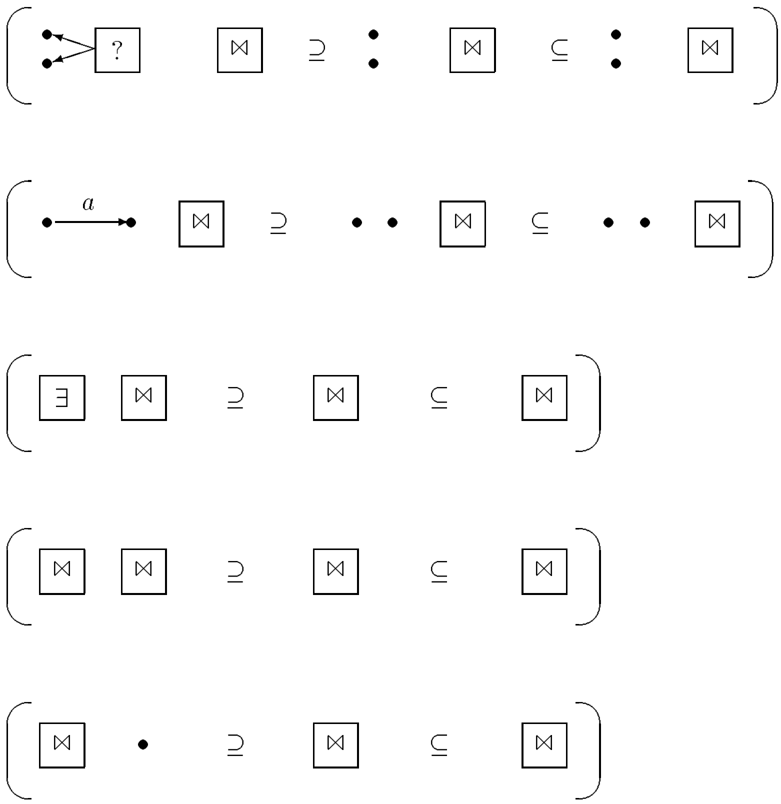
for all $a \in \Gamma$.

$\mathcal{R}_3$ contains the following "garbage collecting" rules:



for $i = 1, \ldots, n$,





for all $a \in \Gamma$,

In the following it is shown that $\mathcal{G}(A, B)$ is terminating (Lemma 15.4.2), and that $\mathcal{G}(A, B)$ is confluent if and only if $\langle A, B \rangle$ has no solution (Lemmas 15.4.4 and 15.4.6). This concludes the proof of Theorem 15.4.1 since $\mathcal{G}(A, B)$ is effectively constructible.

LEMMA **15.4.2** *$\mathcal{G}(A, B)$ is terminating.*

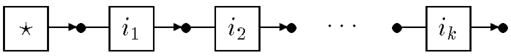
PROOF. Suppose that $\mathcal{G}(A, B)$ admits an infinite sequence $G_1 \Rightarrow G_2 \Rightarrow \ldots$ of direct derivations. No application of any rule in $\mathcal{G}(A, B)$ increases the number of edges with label in $\{\star\} \cup \{1, \ldots, n\}$, so there is some $l \geq 1$ such that the number of these edges is the same in all G_j with $j \geq l$. Consequently $G_l \Rightarrow G_{l+1} \Rightarrow \ldots$ contains no applications of the first three rule schemata. But all other rules in $\mathcal{G}(A, B)$ decrease the sum of the numbers of nodes and edges, and hence $G_l \Rightarrow G_{l+1} \Rightarrow \ldots$ cannot be infinite. □

LEMMA **15.4.3** *Every hypergraph containing an edge labeled by $\bowtie$ reduces to $\boxed{\bowtie}$.*

PROOF. Apply the rules in $\mathcal{R}_3$ and the first rule for $\mathcal{R}_0$ as long as possible. □

LEMMA **15.4.4** *If $\langle A, B \rangle$ has a solution, then $\mathcal{G}(A, B)$ is not confluent.*

PROOF. Let $i_1, \ldots, i_k$ be a solution of $\langle A, B \rangle$. Then



reduces to $\boxed{\bowtie}$ and $\boxed{\exists}$, both being irreducible. □

LEMMA **15.4.5** *If $\langle A, B\rangle$ has no solution and $G \Rightarrow H$ is a direct derivation through the second rule schema for $\mathcal{R}_0$, then $H \Rightarrow^* \boxed{\bowtie}$.*

PROOF. Call a sequence $e_1, \ldots, e_k$ of edges in H an *index chain* if (1) $m_H(e_j) \in \{1, \ldots, n\}$ for $j = 1, \ldots, k$, (2) $t_H(e_j) = s_H(e_{j+1})$ for $j = 1, \ldots, k-1$, and (3) $indegree(s_H(e_j)) = outdegree(s_H(e_j)) = 1$ for $j = 1, \ldots, k$. Let now $e_1, \ldots, e_k$ be the longest index chain in H such that e_1 is created by $G \Rightarrow H$. Then there is a derivation $H \Rightarrow^* H'$ through k successive applications of the first rule schema for $\mathcal{R}_1$, such that the j^{th} step replaces e_j by two sequences of edges representing u_{i_j} and v_{i_j}, with $i_j = m_H(e_j)$. Let v be the third target node of the @-edge e created in the k^{th} step. If v is a source or target node of any other edge, then $H' \Rightarrow H''$ through a rule in $\mathcal{R}_2$ and hence $H'' \Rightarrow^* \boxed{\bowtie}$ by Lemma 15.4.3. On the other hand, if e is the only edge incident to v, then e can be replaced by a ?-edge through the second $\mathcal{R}_1$-rule. The generated strings $u_{i_1} \ldots u_{i_k}$ and $v_{i_1} \ldots v_{i_k}$ cannot be equal as otherwise $i_1, \ldots, i_k$ were a solution of $\langle A, B\rangle$. Therefore an exhaustive application of the $\mathcal{R}_1$-rules for ?-edges results in a hypergraph containing a $\bowtie$-edge. Finally, this hypergraph reduces to $\boxed{\bowtie}$ by Lemma 15.4.3. □

LEMMA **15.4.6** *If $\langle A, B\rangle$ has no solution, then $\mathcal{G}(A, B)$ is confluent.*

PROOF. By Newman's Lemma (see e.g. [Hue80]) it suffices to show local confluence, since $\mathcal{G}(A, B)$ is terminating. Consider two direct derivations $H_1 \;{}_{r_1}\!\Leftarrow G \Rightarrow_{r_2} H_2$ through rules r_1,r_2 from $\mathcal{G}(A, B)$. Assume that the two steps are not independent in the sense of the Commutation Theorem 15.2.2 and that $H_1 \not\cong H_2$, as otherwise the existence of a common reduct is clear.

Case 1: $r_1, r_2 \in \mathcal{R}_2 \cup \mathcal{R}_3$. Then both H_1 and H_2 contain an edge labeled with $\bowtie$, hence they reduce to $\boxed{\bowtie}$ by Lemma 15.4.3.

Case 2: $r_1 \in \mathcal{R}_0 \cup \mathcal{R}_1$, $r_2 \in \mathcal{R}_2 \cup \mathcal{R}_3$. Then H_2 reduces to $\boxed{\bowtie}$. If H_1 contains a $\bowtie$-edge, then H_1 reduces also to $\boxed{\bowtie}$. Otherwise there is a direct derivation $H_1 \Rightarrow H_3$ through a rule in $\mathcal{R}_2$, so H_3 reduces to $\boxed{\bowtie}$.

Case 3: $r_1 \in \mathcal{R}_2 \cup \mathcal{R}_3$, $r_2 \in \mathcal{R}_0 \cup \mathcal{R}_1$. Analogously to case 2.

Case 4: $r_1, r_2 \in \mathcal{R}_0 \cup \mathcal{R}_1$. Then one of the rules, say r_1, is the first rule for $\mathcal{R}_0$ while r_2 is an instance of the second rule schema. By Lemmas 15.4.3 and 15.4.5, H_1 and H_2 reduce to $\boxed{\bowtie}$. □

15.5 CONCLUDING REMARKS

A task for further research is to find a sufficiently large subclass of terminating hypergraph rewriting systems for which confluence is equivalent to strong joinability of critical pairs. For the finite systems in such a class, confluence is decidable since strong joinability becomes decidable under termination.

A possible application of the Critical Pair Lemma not considered in this paper is the completion of non-confluent systems. One could set up a procedure which adds rules to a system until all critical pairs are strongly joinable, where the new rules should preserve the equivalence $\overset{*}{\Leftrightarrow}$ generated by $\Rightarrow$. The hypergraph rewriting systems submitted to such a procedure would have to be terminating, to ensure that strong joinability can be checked. This poses the question of how to test for termination of (hyper)graph rewriting systems, a topic to which apparently very little attention has been paid yet.

ACKNOWLEDGEMENT. I wish to thank Frank Drewes for pointing out an error in a previous version of the proof of Theorem 15.4.1.

REFERENCES

[Ehr77] Hartmut Ehrig. Embedding theorems in the algebraic theory of graph-grammars. In *Proc. Fundamentals of Computation Theory*, pages 245–255. Springer Lecture Notes in Computer Science 56, 1977.

[Ehr79] Hartmut Ehrig. Introduction to the algebraic theory of graph grammars. In *Proc. Graph-Grammars and Their Application to Computer Science and Biology*, pages 1–69. Springer Lecture Notes in Computer Science 73, 1979.

[EK76] Hartmut Ehrig and Hans-Jörg Kreowski. Parallelism of manipulations in multidimensional information structures. In *Proc. Mathematical Foundations of Computer Science*, pages 284–293. Springer Lecture Notes in Computer Science 45, 1976.

[HU79] John E. Hopcroft and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 1979.

[Hue80] Gérard Huet. Confluent reductions: Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4):797–821, 1980.

[KB70] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebras*, pages 263–297. Pergamon Press, 1970.

[KNO90] Deepak Kapur, Paliath Narendran, and Friedrich Otto. On ground-confluence of term rewriting systems. *Information and Computation*, 86:14–31, 1990.

[Kre77] Hans-Jörg Kreowski. Manipulationen von Graphmanipulationen. Dissertation, Technische Universität Berlin, 1977.

[OH92] Yasuyoshi Okada and Masahiro Hayashi. Graph rewriting systems and their application to network reliability analysis. In *Proc. Graph-Theoretic Concepts in Computer Science*. Springer Lecture Notes in Computer Science 570, 1992.

[Rao84] Jean-Claude Raoult. On graph rewritings. *Theoretical Computer Science*, 32:1–24, 1984.