Confluence of Graph Transformation Revisited

Detlef Plump

Department of Computer Science, The University of York,
York YO10 5DD, United Kingdom
det@cs.york.ac.uk

Abstract. It is shown that it is undecidable in general whether a terminating graph rewriting system is confluent or not—in contrast to the situation for term and string rewriting systems. Critical pairs are introduced to hypergraph rewriting, a generalisation of graph rewriting, where it turns out that the mere existence of common reducts for all critical pairs of a graph rewriting system does not imply local confluence. A Critical Pair Lemma for hypergraph rewriting is then established which guarantees local confluence if each critical pair of a system has joining derivations that are compatible in that they map certain nodes to the same nodes in the common reduct.

1 Introduction

To compute efficiently with graph transformation rules requires some way of cutting down the nondeterminism in the derivation spaces of graphs. A common solution to this problem in rule-based formalisms is to rely on *confluent* sets of rules so that all terminating derivations from an initial state will yield the same result, making backtracking unnecessary. For example, confluence properties are important for efficiently recognizing graph classes and executing graph algorithms by graph reduction [2, 5, 7], for parsing graph languages by using the inverse rules of graph grammars [16, 31], and for verifying the deterministic behaviour of programs in graph rewriting languages such as PROGRES [34], AGG [15] and GP [29].

In the setting of term rewriting systems, Knuth and Bendix [22] showed that confluence is decidable for terminating sets of rules. 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 Critical Pair Lemma [20]—stating that a term rewriting system is locally confluent if and only if all its critical pairs have common reducts—and by Newman's Lemma which asserts the equivalence of confluence and local confluence in the presence of termination.

In contrast to the situation for term and string rewriting, Theorem 5 below will show that confluence is undecidable for terminating graph rewriting systems. Roughly, the reason is that the embedding of derivations into context is more complicated for graphs than for terms and strings, meaning that the existence of common reducts for all critical pairs need not imply local confluence of the system.

A. Middeldorp et al. (Eds.): Processes... (Klop Festschrift), LNCS 3838, pp. 280–308, 2005.

The second major result of this paper is a Critical Pair Lemma for hypergraph rewriting which provides a sufficient condition for local confluence and hence for confluence of terminating systems (Theorem 7). The result requires each critical pair $T \Leftarrow S \Rightarrow U$ to be joinable by two derivations $T \Rightarrow^* W_1 \cong W_2 \Leftarrow^* U$ such that an isomorphism $W_1 \to W_2$ is compatible with the joining derivations, in that each node in S that is preserved by both $S \Rightarrow T$ and $S \Rightarrow U$ is mapped to the same node in W_2 by $S \Rightarrow^* W_2$ and $S \Rightarrow^* W_1$ followed by $W_1 \to W_2$.

The next section recalls some properties of binary relations and defines labelled and directed hypergraphs. Section 3 reviews the double-pushout approach to graph transformation, adapted to the setting of hypergraphs. Subsections 3.2 and 3.3 provide results about the restriction, extension and independence of derivations which will be needed to prove the Critical Pair Lemma. Section 4 starts by arguing, in Subsection 4.1, that confluence modulo isomorphism rather than confluence is the right notion to consider in graph transformation. Subsection 4.2 then presents a reduction of the Post Correspondence Problem showing that confluence is undecidable for terminating graph rewriting systems. Subsection 4.3 introduces critical pairs to hypergraph rewriting and proves the Critical Pair Lemma. Section 5 concludes by mentioning related work and topics for future work. Finally, the Appendix summarises some properties of hypergraph pushouts that are needed in proofs.

The confluence results of this paper were established in [27], but the undecidability of confluence was only shown for hypergraph rewriting; the current result also covers graph rewriting. In addition, the undecidability proof has been simplified so that the number of rule schemata in the reduction of the Post Correspondence Problem decreased from 21 to 12. Moreover, this paper is rigorous with respect to the role of confluence modulo isomorphism.

2 Preliminaries

This section fixes some terminology for binary relations (see also [3, 4]) and introduces hypergraphs and their morphisms.

2.1 Relations

Let $\to$ be a binary relation on a set A. The inverse relation of $\to$ is denoted by $\leftarrow$. The identity on A is the relation $\to^0 = \{\langle a, a\rangle \mid a \in A\}$. The reflexive closure of $\to$ is $\to^= = \to \cup \to^0$. The compositon of two binary relations $\to_1$ and $\to_2$ on A is $\to_1 \circ \to_2 = \{\langle a, c\rangle \mid a \to_1 b \text{ and } b \to_2 c \text{ for some } b\}$. For every $n > 0$, the n-fold composition of $\to$ is $\to^n = \to \circ \to^{n-1}$. The transitive closure of $\to$ is $\to^+ = \cup_{n>0} \to^n$, and the transitive-reflexive closure of $\to$ is $\to^* = \to^+ \cup \to^0$. Two elements a and b have a *common reduct* if $a \to^* c \leftarrow^* b$ for some c. If $a \to^* c$ and there is no d such that $c \to d$, then d is a *normal form* of a.

Definition 1 (Termination and confluence). The relation $\to$ is

(1) *terminating* if there is no infinite sequence of the form $a_1 \to a_2 \to a_3 \to \ldots$,
(2) *confluent* if for all a, b and c with $b \leftarrow^* a \to^* c$, elements b and c have a common reduct (see Figure 1(a)),