Relating confluence, innermost-confluence and outermost-confluence properties of term rewriting systems

M.R.K. Krishna Rao*  
Max-Planck-Institut für Informatik, Im Stadtwald, D-66123 Saarbrücken, Germany  
(e-mail: krishna@mpi-sb.mpg.de)

Received October 27, 1994 / May 30, 1995

Abstract. Innermost-confluence is important in giving call-by-value and denotational semantics and outermost-confluence is important in giving call-by-need and lazy semantics of programs. In this paper, we give a few sets of sufficient conditions under which the properties of confluence, innermost-confluence and outermost-confluence coincide.

Confluence and innermost-confluence coincide for weakly innermost normalizing overlay systems and confluence and outermost-confluence coincide for outermost normalizing left-linear overlay systems. In general, every weakly innermost (outermost) normalizing confluent system is innermost (outermost) confluent but the converse is not true.

1 Introduction

In the last few decades, term rewriting systems have played a fundamental role in the analysis and implementation of abstract data type specifications, decidability of word problems, theorem proving, computability theory, design of functional programming languages (e.g. Miranda), integration of functional programming and logic programming paradigms, termination of logic programs, etc. Termination and confluence are two fundamental properties of term rewriting systems. In this paper, we study confluence properties. To be precise, we obtain a number of results relating different notions of confluence. The following quote from Plaisted's recent survey [5] says all about the issues we are dealing with.

Innermost-confluence is interesting since denotational semantics is often defined by a kind of innermost evaluation procedure, which is similar to innermost rewriting. Outermost-confluence is interesting because outermost reduction corresponds to lazy evaluation. If a system is outermost-confluent, that means that the lazy semantics is well-defined, regardless of the order in which outermost reductions are done and regardless of which rules are applied. Also, nonterminating outermost-confluent systems are those for which

* On leave from Tata Institute of Fundamental Research, Bombay
a semantics based on infinite terms seems most natural, since lazy evaluation for nonterminating programs can lead to infinite structures.

Note that confluence of terminating systems implies innermost-confluence and outermost-confluence, but a system can be innermost or outermost-confluent without being confluent. A nonterminating system can be confluent but not innermost (outermost) confluent and vice versa.

The following results are established in the paper.

1. A weakly innermost normalizing overlay system is innermost-confluent if and only if it is confluent.
2. A weakly innermost normalizing system is innermost-confluent if it is confluent. But the converse is not true.
3. A strongly innermost normalizing system is innermost-confluent if $s\theta$ and $t\theta$ are joinable by innermost rewriting for every overlay critical pair $(s, t)$ and a substitution $\theta$ which replaces variables by irreducible terms.
4. Let $R$ be a weakly outermost normalizing left-linear overlay system with the property: if the right-hand side $r_1$ of a rule unifies with a non-variable proper subterm of the left-hand side $l_2$ of a rule then $l_2$ does not unify with the left-hand side of any other rule in $R$. Then $R$ is confluent if and only if it is outermost-confluent.
5. A weakly outermost normalizing system is outermost-confluent if it is confluent. But the converse is not true.
6. Every weakly normalizing almost orthogonal system is outermost-confluent.
7. A strongly outermost normalizing left-linear overlay system is confluent if and only if it is outermost-confluent.

The rest of the paper is organized as follows. The next section gives preliminary definitions and results needed later. In Sect. 3, we give a few examples illustrating the differences between the three notions of confluence. We study innermost-confluence in Sect. 4 and outermost-confluence in Sect. 5. Section 6 concludes the paper with a few open problems.

2 Preliminaries

We assume that the reader is familiar with the basic terminology of term rewriting systems, like contexts, substitutions etc. and properties such as confluence, local confluence and innermost normalization etc. and give definitions only when they are required. The concepts not defined in the paper can be found in Dershowitz and Jouannaud [1], Klop [3] or Plaisted [5].

**Definition 1** (critical pairs). Let $l_1 \sim_r r_1$ and $l_2 \sim_r r_2$ be renamed versions of rules in a rewrite system $\mathcal{R}(\mathcal{F}, R)$ such that they have no variables in common. Suppose $l_1|_p$ is not a variable for some position $p$ and $l_1|_p$ unifies with $l_2$ through a most general unifier $\sigma$. The pair of terms $\langle l_1[r_2]_p\sigma, r_1\sigma \rangle$ is called a critical pair$^1$ of $\mathcal{R}(\mathcal{F}, R)$. If $l_1 \rightarrow r_1$ and $l_2 \rightarrow r_2$ are renamed versions of the same rewrite rule, we do not consider the case $p = \varepsilon$. A critical pair $\langle l_1[r_2]_p\sigma, r_1\sigma \rangle$ with $p = \varepsilon$ is called an overlay, and a critical pair $\langle s, t \rangle$ is trivial if $s \equiv t$.

$^1$ $t|_p$ denotes the subterm of $t$ at position $p$ and $t[u]|_p$ denotes the term obtained from $t$ by replacing the subterm $t|_p$ with $u$. 